Reconnaissance automatique des gestes de la langue française parlée complétée

Thomas Burger

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque
|__|__|__|__|__|__|__|__|__|__|

T H E S E

pour obtenir le grade de

DOCTEUR DE L'INP Grenoble

Spécialité : *Signal, Image, Parole, télécoms*

préparée au laboratoire GIPSA-Lab / DIS

dans le cadre de **l'Ecole Doctorale** *Electronique, Electrotechnique, Automatique, Traitement du Signal*

présentée et soutenue publiquement

par

Thomas BURGER

le 26/10/2007

RECONNAISSANCE AUTOMATIQUE DES GESTES DE LA LANGUE FRANÇAISE PARLEE COMPLETEE

Sous la direction de Mme. **Alice CAPLIER**
Sous la co-direction de M. **Pascal PERRET**

JURY

M. James CROWLEY	Président de jury,
M. Olivier COLOT	Rapporteur,
M. Michel DHOME	Rapporteur,
M. Jean-Emmanuel VIALLET	Examinateur,
Mme. Alice CAPLIER	Directrice de thèse,
M. Pascal PERRET	Co-directeur de thèse.

A François, mon grand-père,
et à Nohé, un futur lecteur j'espère.

PREFACE & REMERCIEMENTS

La rédaction de ce rapport clôt une tranche de vie qui a commencé durant l'automne 2003 au Jardin de Ville de Grenoble, avec la rencontre de Denis Beautemps. A cette époque, je compensais un désœuvrement immature et un désintérêt total de mes études à l'INPG par une pratique outrancière de la montagne et de la voile. Le monde de la voile était restreint dans le bassin grenoblois, et de formations parallèles en rencontres fortuites, je me suis retrouvé à enseigner le secourisme dans l'association que dirige Denis à l'échelle départemental.

Durant cette journée "Porte ouverte" au Jardin de Ville, nous profitions de l'inintérêt des passants à l'égard de notre stand pour faire connaissance. Il me parla de son travail au CNRS, et des balbutiements du projet TELMA, qu'il dirige maintenant. Je lui ai parlé de mes études : j'avais "remplacé" mon cursus d'ingénierie des télécommunications par les mathématiques appliquées et la recherche opérationnelle, matière très intéressante, mais qui me laissait perplexe quand à ses débouchées. En revanche, le projet de Denis me semblait à la fois passionnant et d'un réel intérêt pratique, malgré le peu de chose que j'y comprenais : il s'agissait d'un domaine que je ne connaissais pas. C'est donc très surpris qu'en fin de journée, j'ai accepté son offre de faire mon stage de DEA sous sa direction, à partir du printemps suivant. Si depuis 4 ans je prends quotidiennement plaisir à travailler, c'est parce que Denis m'a fait confiance ce jour-là au Jardin de Ville, et m'a proposé de participer à ce projet. A ce titre, et malgré nos permanentes divergences scientifiques, je lui suis infiniment reconnaissant de la chance qu'il m'a donnée.

Par la suite, de nombreuses entraves administratives m'ont empêché de prolonger ce travail avec lui, et c'est Alice Caplier qui m'a accepté sous son aile. Ce que j'ai pris au départ comme un aléas, s'est cependant révélé être une seconde intervention de la providence : Alice, de par son pragmatisme, sa patience, son caractère bien trempé, sa spontanéité et son ouverture d'esprit, est la seule personne que j'ai rencontrée à pouvoir aussi facilement gérer dans un cadre professionnel, mon caractère de cochon, ma "grande gueule", ma libre interprétation de l'orthographe et de la grammaire française, et ma tendance pathologique à disperser tous mes efforts. Je me demande combien de fois elle a dû rentrer le soir avec les nerfs en pelote... Surtout durant la première année.

L'intégralité de ces trois années fut réellement enrichissante, et ce document n'ayant aucune vocation autobiographique, je ne vais pas les détailler plus, même si j'aimerais passer plus de temps à décrire un bon nombre de personnes que j'ai rencontrées dans les laboratoires universitaires, à France Telecom R&D, en Turquie, en cours, en conférence, et dans ma vie privée. Je ne les nomme pas de peur de manquer d'exhaustivité, mais le cœur y est. Je sais que je continuerai à en voir certaines, et pour beaucoup d'autres, je ne peux formuler qu'une espérance.

Parmi ces personnes, il y a en particulier eu de nombreux malentendants et leur entourage (orthophonistes, familles, amis, associations). En proportion du temps trop faible que j'ai passé en leur compagnie, ils ont participé de manière prépondérante à la richesse de ces années, en faisant partager leurs expériences et leurs motivations avec une générosité constante et naturelle.

Il y a aussi mon grand-père, parti en décembre 2005 à presque 91 ans et Nohé, rencontré 4 mois plus tard, alors âgé de 4 ans et demi. L'un ne sera jamais mon lecteur. Pour l'autre, il faudra attendre une petite vingtaine d'année. Le premier, émerveillé par le concept d'Internet, aurait trouvé mes travaux magiques. Le second les trouvera, au mieux, paléolitiquement communs.

Je suis le seul à qui ce document a apporté un diplôme. Malgré tout, il s'agit d'un travail collectif dont voici les coauteurs : il y a tout d'abord Alice, bien sûr, dont la présence et l'aide permanente sont allées au-delà de ce qui incombe à une directrice de thèse. Il y a aussi Stéphane Mancini et Pascal Perret, qui ont participé à mon encadrement à divers titres. Il y a Oya Aran et Lale Akarun de l'université d'Istanbul. Il y a Alexandre Benoit, Sébastien Roux, Joël Gardes et Franck Mamalet, dont ce travail est empreint des leurs. Il y a Alexandra Urankar et Pierre Lemaire, qui ont réalisé leur projet de fin d'étude sous mon tutorat, et qui, en plus de m'avoir fait gagner beaucoup de temps par leur excellent travail, m'ont appris qu'il était aussi difficile qu'enrichissant de jouer le rôle de superviseur. Il y a les "reviewers" de mes différentes publications, qui m'ont forcé à me justifier, qui m'ont poussé dans mes retranchements et m'ont obligé à me remettre en cause; ainsi, ils ont participé directement à la maturation encore inachevée de mes convictions scientifiques. Parmi eux, les rapporteurs du jury de soutenance, Olivier Colot et Michel Dhome, mais aussi ses autres membres, James Crowley et Jean-Emmanuel Viallet, ont un rôle des plus importants. Enfin, de manière plus prosaïque, il y a le duo "Mathilde Fort et son stylo rouge", sans qui je serai coupable d'indécence orthographique. Si l'on me joint à ces personnes, le "nous" narrateur du reste du rapport est au grand complet. Je laisserai la parole à ce "nous" dès la page suivante.

J'espère que vous prendrez autant de plaisir à lire ce rapport que j'en ai eu à l'écrire. J'espère aussi que cette lecture vous apprendra au moins un petit quelque chose. Enfin, j'espère que vous serez dorénavant plus sensible à la cause des malentendants.

Bonne lecture,
Tom

SOMMAIRE

LISTE DES FIGURES

LISTE DES TABLEAUX

CHAPITRE I
INTRODUCTION & CONTEXTE GENERAL

I.1 La langue Française parlée complétée (LPC)

En France, entre 8 et 10% de la population souffre d'**hypoacousie**, (diminution pathologique de l'audition pour laquelle la perte moyenne est supérieure à 20**dB**), le degré du trouble variant de la déficience auditive légère (le **malentendant léger** a une perte moyenne de 20 à 40dB) à l'**anacousie** (le **sourd profond** a une perte moyenne supérieure à 90dB). Son étiologie est très variée : traumatismes, intoxications, maladies, origine génétique, ou congénitale non syndromique. On distingue trois types de surdité, en fonction de la zone atteinte : surdité de perception, surdité de transmission et surdité centrale (lésions des aires cérébrales auditives). Enfin on considère l'âge auquel est survenue l'atteinte : anténatale, néonatale, pré-linguale, post-linguale, à l'âge adulte, au troisième âge (**presbyacousie**). Cette dernière classification permet une compréhension plus cognitive ou sociologique que clinique de ce trouble, mais est indispensable à toute mise en place d'une politique d'accessibilité.

Ainsi, cette population est en réalité beaucoup plus complexe à appréhender que ne le croit l'imaginaire collectif des bien-entendants, qui se réfèrent bien souvent à une dichotomie abusive entre "les normaux" qui parlent et entendent, et "les sourds-muets" qui "gesticulent" une langue des signes à propos de laquelle toutes sortes de préjugés circulent.

Par le terme de "langue des signes", il est bien souvent fait référence à la Langue des Signes Française (LSF). Comme son nom l'indique, la LSF est entièrement basée sur des gestes, afin d'être visuellement accessible. Mais son nom indique aussi que c'est une langue à part entière :

– sa grammaire et sa syntaxe lui sont propres, et sont clairement distinctes de celles du français. A titre d'exemple, (1) il n'y a pas de conjugaison mais le temps est représenté par une ligne partant de derrière le signeur et allant vers l'avant, et (2) les pronoms sont remplacés par une représentation du positionnement spatial des substantés.

– son vocabulaire gestuel est potentiellement infini, puisqu'il se compose de concepts.

En d'autres termes, les personnes dont cette langue est la langue maternelle, ne sont pas "natives francophones", mais "natives d'une langue signée". Contrairement à une idée très répandue, la langue des signes n'est en aucun cas universelle. Chaque pays possède sa propre langue (ainsi, pour les francophones, il existe la LSF, LS Québécoise, LS de Belgique Francophone, etc.), et il existe même des "patois" entre différentes régions ou villes. Fort heureusement, la parenté de certains gestes des langues signées avec les pantomimes permet à des personnes ayant pour langue maternelle des langues signées radicalement différentes de communiquer de manière élémentaire avec des signes très iconisés. De même, afin de faire le lien avec les langues orales et écrite, l'alphabet dactylologique permet d'épeler les noms ou mots n'existant pas dans la LSF en signant l'alphabet latin (Figure I-1).

Le fait que la LSF soit une langue à part entière a plusieurs implications :

– La LSF n'a pas d'équivalent écrit. Un natif LSF ne peut donc pas lire dans sa langue maternelle. D'une manière générale, la communication entre bien-entendants et malentendants est très compliquée, puisqu'il s'agit de communiquer en choisissant une langue parmi les deux possibles, alors qu'elles sont radicalement différentes, et qu'elles n'ont même pas la même modalité d'expression.

– Une personne dont l'hypoacousie est acquise longtemps après l'apprentissage de la langue ne sera pas natif LSF, autrement dit, sa langue maternelle ne sera pas la LSF, et cette dernière sera maîtrisée comme telle.

– Un enfant sévèrement sourd de naissance dont les parents sont bien-entendants ne sera pas élevé de la même manière que si les parents sont eux-mêmes sourds : si les parents ne maîtrisent pas complètement la LSF (ce qui est inévitable, même s'ils décident de l'apprendre de manière intensive dès que la surdité est diagnostiquée), celle-ci ne pourra pas être enseignée correctement comme une langue maternelle.

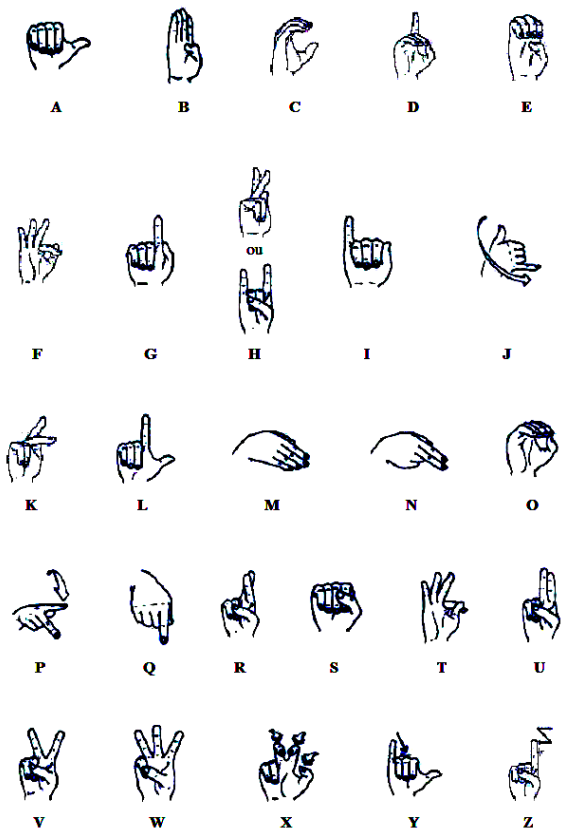


Figure I-1 : dactylologie de la LSF [167].

– Une personne dont la surdité n'est que partielle a énormément de mal à faire coexister les restes auditifs avec une langue signée, et devra éventuellement faire un choix entre les deux modes de communication qu'elle ne peut

appréhender que partiellement (la langue orale entravée par l'hypoacousie d'une part, et la langue signée, maîtrisée comme une langue "étrangère" d'autre part).

La mise en valeur de ces faits permet déjà d'apercevoir l'ébauche de deux courants qui ne sont opposables que dans une description simplifiée du paysage de l'hypoacousie : les **oralisés** (dont la communication s'appuie sur les divers aspects de la langue française telle qu'elle est utilisée par les bien-entendants) et les **non-oralisés** (dont le mode de communication reste indépendant de la communication orale). Comme nous venons de le voir, il est difficile, voire impossible d'appartenir pleinement aux deux communautés. De cette exclusion mutuelle découle une opposition concurrente qu'il est important de comprendre. Pour tout membre d'une de ces deux communautés, la prépondérance éventuelle de l'autre communauté est un fait tragique, puisqu'il marque le risque d'une seconde discrimination : après avoir vécu la discrimination liée au handicap vis-à-vis des bien-entendants, il y a le risque de la discrimination liée à la pratique d'un mode de communication minoritaire au sein des malentendants. Or, le sentiment d'exclusion vécu par tout malentendant ne peut être combattu que par l'appartenance à un groupe faisant unité face à cette exclusion, et au sein duquel cette exclusion n'existe pas. Pour chacune des deux communautés, la peur de voir sa langue maternelle tomber en désuétude au profit d'une autre est vécue comme l'annonce de devoir vivre une nouvelle adaptation en tant que minoritaire dans une nouvelle communauté. C'est pourquoi, toute comparaison d'une communauté par rapport à une autre, même si cela est fait sans aucune intention de nuire, ni sans aucune arrière-pensée politique, sans mise en avant effective d'un système de communication par rapport à l'autre, doit être faite dans le respect de ces craintes. Tout ceci peut sembler un peu "politique", futile et contre-productif à l'égard des démarches d'accessibilité du point de vue extérieur et factuel des bien-entendants, mais cela est malgré tout nécessaire pour tenir compte de la réalité affective du problème.

La prépondérance numéraire de la communication oralisée est due à la prépondérance des personnes ayant soit un fort reste auditif, soit ayant eu une complète accession au langage avant l'hypoacousie (due à l'âge, à une maladie, un barotraumatisme, etc.). Cependant, même si l'écrasante majorité de la population souffrant d'hypoacousie est oralisée, le noyau dur, le plus soudé, le plus autonome et communautaire en terme de langage, le plus porté sur le prosélytisme et celui dont l'existence se remarque le plus du point de vue des bien-entendants est celui de la population non oralisée. Cela est dû à :

– L'utilisation de la LSF qui est plus visible et remarquable que celle des modes de communication oralisée.

– L'utilisation de la LSF par les sourds profonds, et donc par les personnes les plus handicapées par leur déficience auditive.

– L'apprentissage de la LSF de manière naturelle par les enfants sourds de parents sourds.

– La condamnation de l'exclusion des malentendants qui n'est encore que trop récente à l'échelle des générations de malentendants qui se succèdent.

– L'association trompeuse entre les sourds profonds et la communauté malentendante : il ne viendrait à l'idée de personne de demander à ses parents ou grands-parents d'apprendre la langue des signes lorsque leur audition baisse.

Un phénomène récent vient cependant modifier cet équilibre : tout le monde considère désormais que la priorité de l'insertion des personnes malentendantes dans le reste de la société est une évidence. A ce titre, la résistance des non oralisés est en train de faiblir, mais il est nécessaire de volontairement freiner cet affaiblissement, afin de ne pas créer une autre situation d'exclusion. Ainsi, le **Français Signé** a pour objectif un tel rapprochement en douceur, en proposant d'utiliser le vocabulaire gestuel de la LSF, tout en respectant la syntaxe du français oral. Cependant, cela reste encore trop éloigné du français oral, et à plus fort titre, de la représentation partielle du français que permet la lecture labiale.

La **lecture labiale** désigne l'action qui consiste à reconnaître les sons émis par un locuteur en fonction du mouvement de ses lèvres. Cet exercice difficile nécessite d'avoir une pleine vue sur les lèvres du locuteur, mais aussi de connaître le contexte de la discussion. En effet, un grand nombre de phonèmes sont des **sosies labiaux**, c'est-à-dire qu'ils sont visuellement équivalents (ils correspondent à un même **visème** : les différents phonèmes d'un même visème ne sont pas différenciés par le mouvement des lèvres mais par celui d'autres articulateurs, tels que les cordes vocales, qui sont invisibles). Ainsi le /p/ et le /m/ sont des sosies labiaux, de même que le /y/ et le /u/ (cf. Figure I-2 pour l'alphabet phonétique). Il est donc nécessaire aux malentendants de se référer sans cesse à un dictionnaire de mots connus afin de déterminer le mot réel le plus probablement articulé par le locuteur. Dans certains cas, des mots ou des expressions entières sont des synonymes labiaux (par exemple les phrases "Papa sort" et "Maman dort") et seul le contexte permet la discrimination. On considère généralement que cette suppléance mentale permet de retrouver 65-70% du message à partir de la lecture labiale. Il en résulte que :

– La lecture labiale seule peut entraîner d'importantes incompréhensions, qui sont autant d'entraves à une bonne communication.

– Pour le sourd profond de naissance, le français écrit et la projection labiale du français oral sont trop éloignés l'un de l'autre pour permettre une maîtrise correcte de la langue maternelle dans un cadre scolaire, et plus tard universitaire.

– Pour le sourd profond de naissance non exposé à une langue complète (la LSF) mais seulement à une sous-spécification labiale du français, il n'est pas possible d'accéder à une spécification langagière complète (par exemple, il est difficile de comprendre la différence entre le /s/ et le /d/ pour des personnes non oralisées).

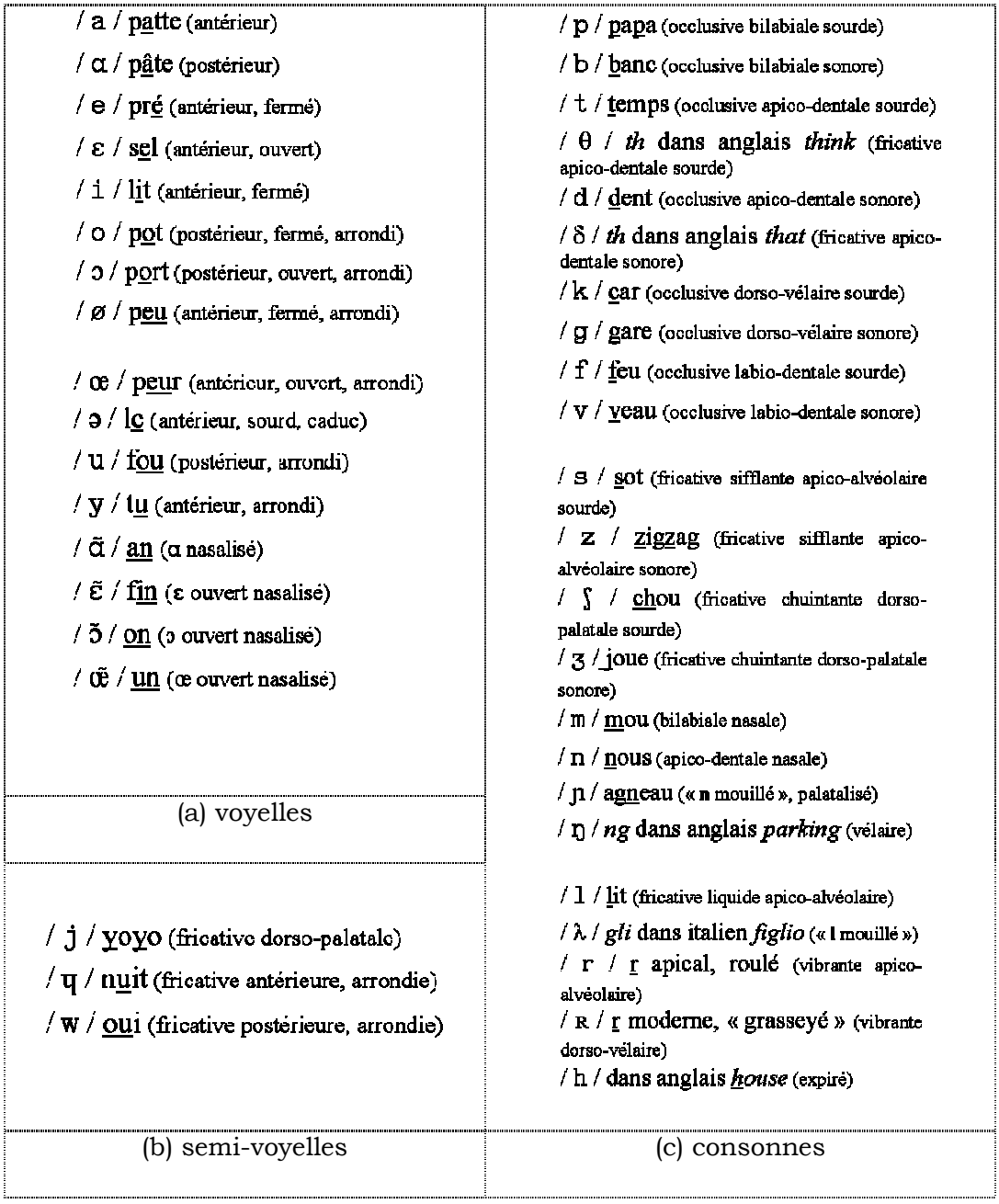


Figure I-2 : alphabet phonétique [168].

Toutes ces considérations ont poussé le Dr. Cornett à mettre au point en 1967 un système de complétion gestuelle à la lecture labiale, dans le but de rendre équivalent l'anglais américain visuel et l'anglais américain oral : le **Cued Speech** [1]. Il s'agit de remplacer l'information fournie par les articulateurs invisibles par des informations équivalentes fournies par des gestes. Le terme "informations équivalentes" est à prendre au sens de la théorie de l'information, c'est-à-dire que celles-ci ne doivent pas être plus riches, plus complètes, moins ambigües, ou issues d'un espace d'expression plus grand. Dans la langue orale, chaque son (et en particulier chaque phonème) est différenciable d'un autre son, soit parce que la sortie du conduit vocal (les lèvres, qui sont visibles) soit parce que

d'autres articulateurs invisibles (les cordes vocales ou la langue par exemple) effectuent un mouvement différent. Il doit en être de même avec les "phonèmes visuels" du Cued Speech : deux phonèmes différents, mais visuellement identiques aux lèvres (appartenant donc au même visème) doivent "s'articuler visuellement" avec des gestes manuels différents, afin de lever l'ambiguïté. De plus, ce geste doit être accessible en parallèle du mouvement labial, et il doit être réalisable de telle sorte qu'il ne ralentisse pas de manière excessive la **prosodie** (le rythme de la parole). Pour toutes ces raisons, Cornett propose d'effectuer un geste manuel de la manière suivante :

– il doit y avoir un geste pour chaque syllabe de type Consonne-Voyelle (telle que "/b/-/a/" par exemple). Des consonnes et des voyelles "muettes" ou "invisibles" sont éventuellement ajoutées pour maintenir cette alternance Consonne-Voyelle (ou CV). Ainsi, avec la convention qu'un phonème muet se note /_/, la phrase "tu as pris le train", se code "/t/-/y/ , /_/-/a/ , /p/-/_/ , /r/-/i/ , /L/-/Ø/ , /t/-/_/ , /r/-/õe/".

– le geste est réalisé avec une seule main, la paume de la main vers le locuteur et son dos vers l'interlocuteur, près du visage et sans cacher les lèvres. Ainsi, l'ensemble des informations à extraire par l'interlocuteur se trouve localisé spatialement.

– le geste se compose d'une forme de main, permettant de fournir l'information désambigüisant la consonne, et d'une position de la main par rapport au visage, désambigüisant la voyelle. Ainsi, le geste manuel du LPC se compose de deux informations : celle de configuration et celle de position. Ces deux informations sont théoriquement disponibles en même temps pour chaque phonème, et un seul geste code ces deux informations de manière orthogonale. Un codage continu du LPC se caractérise donc d'un mouvement manuel contenant deux flux d'information codés en parallèle. Afin de bien faire la distinction sur les schémas en couleur de ce document, nous réservons la couleur rouge à l'information de configuration et la couleur bleue à celle de la position (cf. Figure I-3).

Le Cued Speech n'est pas une langue, mais un code, ou une manière particulière d'articuler un enchaînement de phonèmes. Ainsi, il est possible de l'adapter à presque n'importe quelle langue. Actuellement, ce système a été dérivé dans plus de cinquante langues. Dans le cas du français, son adaptation date de 1979, et porte le nom de **Langue française Parlée Complétée**, abrégé en **LPC** ou **code LPC**. La Figure I-3 présente l'intégralité du codage gestuel du LPC.

Un même geste est utilisé pour coder différents phonèmes parfaitement différentiables aux lèvres. Ainsi, il est aussi difficile de comprendre la lecture labiale seule que le code gestuel sans lecture labiale. C'est aussi pour cette raison que le code est relativement compact : 5 positions et 8 configurations pour un total de 40 combinaisons. A titre d'exemple, pour coder la phrase "Que manges-tu ?", il faut produire le mouvement labial correspondant à la

prononciation orale de cette phrase et compléter chacune des 4 syllabes /k/-/Ø/, /m/-/ã/, /ʒ/-/_/, /t/-/y/ par un geste. La succession de ces 4 gestes est représentée sur la Figure I-4. Des exemples de vidéo LPC sont disponibles sur [19].

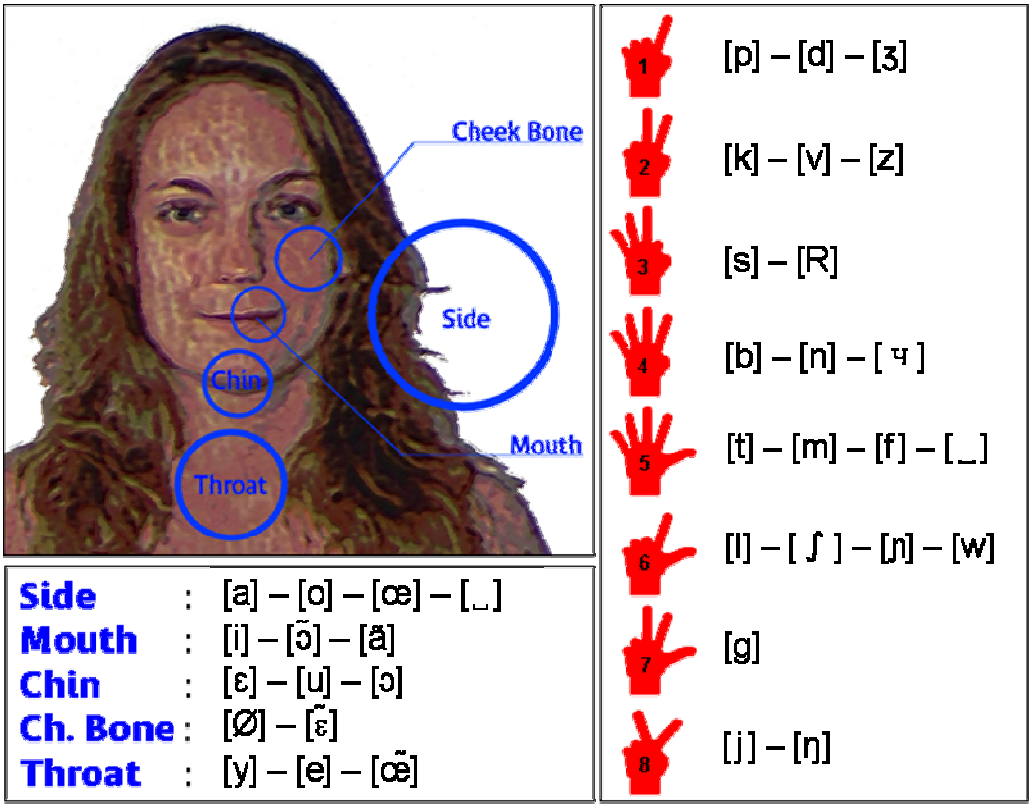


Figure I-3 : détails du code LPC : à gauche, les 5 Positions de la main par rapport au visage (codage des voyelles); à droite, les 8 Configurations de la main (codage des consonnes). Les doigts sont représentés écartés pour plus de lisibilité sur les 7 premières configurations (alors qu'ils doivent normalement être resserrés).

Figure I-4 : succession des gestes à réaliser pour le codage de la phrase "Que manges-tu ?".

Le LPC est encore jeune, dans le sens où seulement des personnes de moins de 25 ans ont été précocement exposées au LPC. Il est couramment utilisé par :

– Les malentendants oralisés, pour communiquer entre eux.

– Les orthophonistes pour permettre aux malentendants d'accéder au français oral.

– Les proches d'un malentendant pour lui parler. En revanche, il est très difficile à une personne bien-entendante de comprendre le LPC non vocalisé. Ainsi, les malentendants répondent généralement par oral, chose qu'ils font plus facilement puisque le LPC leur permet de mieux maîtriser le français.

Jusqu'à récemment, le LPC n'était pas ou peu connu. Aujourd'hui, il se popularise (à tel point que cette évolution a été remarquable à l'échelle du déroulement de ces travaux), et ses utilisateurs le revendiquent. Il en découle que l'intégration des malentendants ne passe pas simplement par la reconnaissance de la LSF, mais aussi par celle du code LPC. Il y a actuellement environ 30.000 personnes françaises, belges ou suisses qui utilisent le LPC, à mettre en rapport (mais à ne pas opposer) avec les 80.000 utilisateurs de la LSF (le reste des personnes ayant une déficience auditive se contentant d'utiliser le français oral).

I.2 Le projet RNTS TELMA

Ces travaux s'inscrivent dans le cadre du projet TELMA (**Tél**éphonie à l'usage des **mal**entendants). Il s'agit d'un projet financé par le Réseau National des Technologies pour la Santé (RNTS). Nous reprenons ici son intitulé de présentation :

"Ouvrir le domaine des Nouvelles Technologies de l'Information et de la Communication aux handicapés est un souci de plus en plus présent dans notre société. Ce projet vise à l'étude et au développement algorithmique de fonctionnalités audio-visuelles tout à fait originales à l'usage des personnes malentendantes, et à l'étude de faisabilité de leur intégration dans un terminal autonome de télécommunication téléphonique. Le projet a pour objectif technique précis d'exploiter la modalité visuelle de la parole, d'une part pour améliorer les techniques du débruitage du son de parole (la minimisation du bruit environnemental permettant une meilleure exploitation des restes auditifs des malentendants), et d'autre part, en mettant en œuvre des techniques d'analyse/synthèse de lecture labiale et de gestes de la Langue Française Parlée Complétée (LPC). [...]." [15].

Ainsi, le but du projet TELMA est la mise en place de briques technologiques modulables autour d'un terminal de téléphonie classique, afin de le rendre accessible aux malentendants, tout en conservant la modalité audio pour le transport de l'information. Ces briques sont au nombre de trois (Figure I-5), et permettent différentes configurations d'utilisation, auxquelles s'ajoute l'utilisation classique du terminal dans le cas de locuteurs bien-entendants :

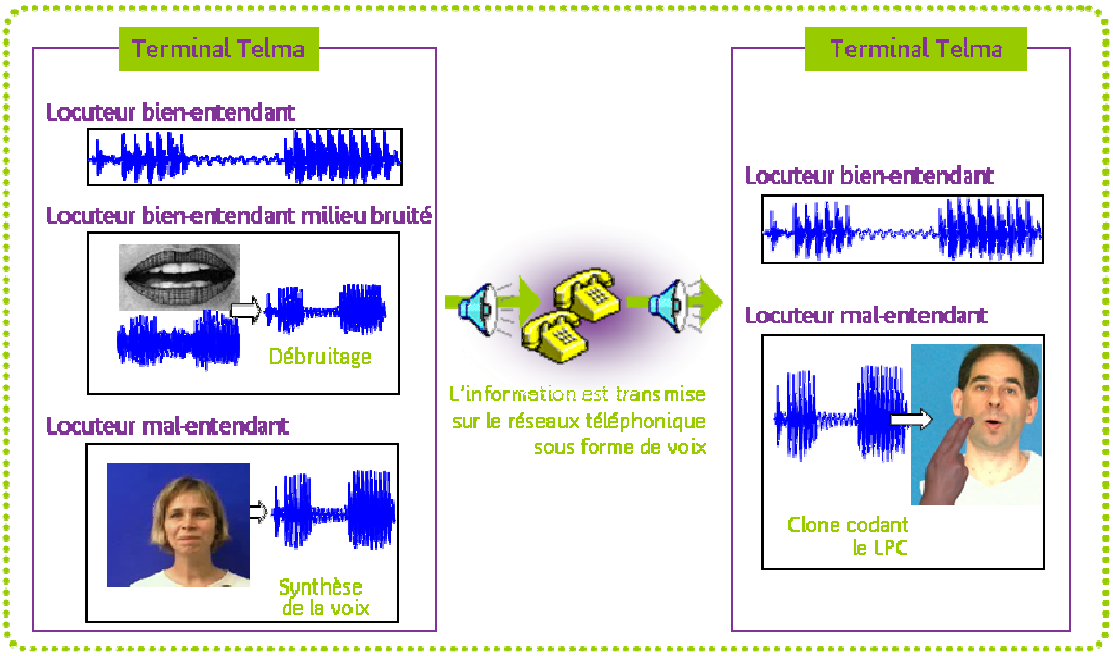


Figure I-5 : détails d'un terminal TELMA.

– L'utilisation de la modalité visuelle pour le rehaussement de la modalité audio. Il s'agit de (1) filmer la partie inférieure du visage du locuteur, (2) d'analyser le mouvement des lèvres et du visage, (3) d'utiliser l'information ainsi extraite pour rehausser la qualité de l'enregistrement audio qui est transmis sur le réseau téléphonique. Cette modalité permettra d'améliorer la communication en milieu bruité et/ou la communication avec (ou entre) malentendant(s) s'appuyant sur des restes auditifs. Sur la Figure I-5, la brique algorithmique correspondante est située à gauche et est intitulée "**Locuteur bien-entendant en milieu bruité**".

– La synthèse vocale automatique à partir de l'analyse des gestes labiaux et manuels du LPC. Il s'agit donc de filmer un codeur, de reconnaître la succession de gestes manuo-labiaux qu'il produit, d'y associer une chaîne de phonèmes et de la faire prononcer par un synthétiseur vocal. Cela permet à un codeur LPC n'oralisant pas correctement (et/ou n'étant pas compréhensible à cause des déformations vocales téléphoniques) d'être compris d'une personne ne maîtrisant pas le LPC. Dans le cas où une application de type visiophonie n'est pas disponible, elle permet aussi à deux malentendants de communiquer via le LPC. Sur la Figure I-5, la brique algorithmique correspondante est située à gauche et est intitulée "**Locuteur mal-entendant**".

– La synthèse d'un clone virtuel animé codant le LPC à partir du flux vocal issu du réseau téléphonique. Il faut ici (1) décomposer la phrase en une chaîne de phonèmes, (2) y associer une chaîne de gestes manuo-labiaux, et (3) animer le clone en conséquence. Ainsi, un malentendant comprenant le LPC peut recevoir le message de quelqu'un ne maîtrisant pas ce code. Sur la Figure I-5, la

brique algorithmique correspondante est située à droite et est intitulée "**Locuteur mal-entendant**".

Suivant la combinaison de personnes désirant communiquer, zéro, une ou deux des trois briques présentes sont utilisées.

Par choix, le projet TELMA repose sur le LPC. Il y a de nombreuses raisons à cela : tout d'abord, en termes d'usage, la demande a énormément augmenté ces dernières années. Ensuite, l'adaptation des technologies basées sur le LPC étant généralisables à d'autres langues, la démarche d'accessibilité correspondante se retrouve d'autant plus porteuse de retombées potentielles.

Ce projet trouve son origine dans plusieurs coopérations antérieures :

– le projet TEMPO-VALSE (financement RNRT – Réseau National de la Recherche en Télécommunications) basé sur l'étude et l'animation de mouvements labiaux, qui était une collaboration entre plusieurs partenaires industriels et universitaires, dont entre autre le GIPSA-Lab/DIS et le GIPSA-Lab/DPC (laboratoire de l'INPG - Institut National Polytechnique de Grenoble) et France Telecom R&D.

– le projet ARTUS de doublage des films avec des clones de synthèse codant le LPC, issu d'une collaboration entre le groupe de la chaîne de télévision Franco-allemande ARTE et le GIPSA-Lab/DPC.

– Enfin, les "près-projets" TELMA, avant le lancement officiel du projet dans le cadre d'un soutien RNTS (Réseau National des Technologies pour la Santé).

Actuellement, les institutions participant au projet sont les suivantes :

– Partenaires universitaires : l'ENST, le GIPSA-Lab et le Laboratoire d'Informatique de Grenoble,

– Partenaire industriel : France Telecom R&D,

– Représentant de la communauté malentendante : le service ORL du CHU de Grenoble.

I.3 La reconnaissance du geste manuel du LPC

Dans le cadre du projet TELMA, les travaux présentés ici concernent une partie de l'étude de la brique "codage LPC vers voix". Comme indiqué sur le diagramme de la Figure I-6, il y a différentes tâches à accomplir au sein de cette fonction :

– Réaliser une acquisition vidéo de qualité acceptable du codeur.

– Analyser l'image, et reconnaître à la fois les mouvements réalisés avec les lèvres, ainsi que la succession de gestes réalisés avec la main.

– Associer à chacun de ces flux visuels une signification gestuelle, c'est-à-dire, reconnaître un geste labial ou manuel particulier et être capable d'y associer un sens partiel.

– Effectuer la fusion de ces deux canaux d'information, afin de fournir la chaîne de phonèmes constitutive du message.

– Synthétiser le son correspondant à la prononciation de la phrase constitutive.

– Transmettre cette information vocale sur le réseau téléphonique.

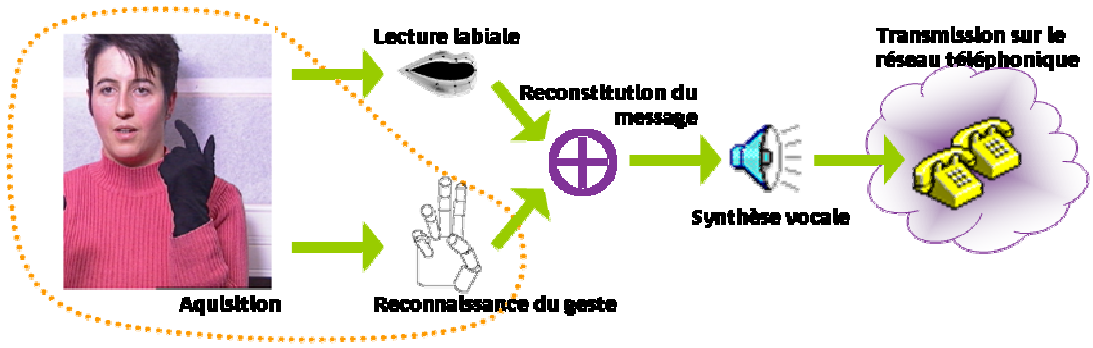


Figure I-6 : champ d'étude par rapport au projet TELMA.

Bien sûr, l'ensemble de ces traitements doit être effectué en temps réel, c'est-à-dire avec les ressources de calcul nécessaires pour ne pas prendre de retard sur le flot conversationnel. De plus, ces traitements ne doivent pas engendrer un décalage trop important. En effet, même si le temps réel est respecté, une latence supérieure à 200ms gène la conversation téléphonique puisque la spontanéité de la conversation est perturbée. En revanche, une conversation audio-visuelle (multimodale) permet de savoir de la même manière qu'avec un interprète comment partager les temps de parole et l'alternat conversationnel. Ainsi, dans ce dernier cas, une latence supérieure peut être acceptable.

Ce travail se focalise sur **les problématiques d'acquisition et d'interprétation du geste manuel** (il s'agit des composantes entourées de pointillés sur la Figure I-6). Ainsi, à partir du flux vidéo, il faut produire le treillis des phonèmes possibles correspondant à la succession des gestes de la main (au sein duquel le mouvement labial viendra préciser la chaîne phonétique articulée). Cela veut dire que l'étude du mouvement labial, que sa fusion avec le mouvement manuel et la synthèse vocale sont au-delà des objectifs des travaux présentés ici. De plus, les contraintes de traitement temps réel et de latence de traitement sont repoussées ultérieurement dans le projet : leur importance impliquera à terme la mise en place d'architectures matérielles dédiées et celles-ci seront envisagées globalement, sur l'ensemble des modules du projet. Il faut malgré tout éviter de complètement oublier ces contraintes ultérieures, afin de pouvoir les gérer plus facilement par la suite. De plus, comme pour l'instant, les outils de synthèse

vocale ne peuvent fonctionner qu'en considérant la phrase dans son ensemble, (il n'est pas possible de faire de la synthèse "à la volée") on s'autorise à mettre en place des éléments algorithmiques correspondant à un transcodeur hors-ligne fonctionnant sur le principe d'une interprétation phrase à phrase en traitement décalé. Il faudra tout de même éviter :

– Des traitements de calculs prohibitifs. Ce souci d'économie et d'efficacité dès l'étape de recherche garantira plus facilement la possibilité d'une application en termes d'usage.

– De traiter la phrase dans sa globalité seulement : la connaissance de la phrase entière est incompatible avec le temps-réel dans la mesure où cela implique une connaissance du futur. Il faudra donc limiter au maximum la plage de connaissance du futur dans les traitements proposés.

De plus, dans la définition originale du projet, un minimum de contraintes sur les conditions d'utilisation du terminal TELMA a été fixé. Ces contraintes pour l'utilisateur sont acceptées car elles sont jugées comme mineures par rapport à la robustesse du système qu'elles apportent au système. Elles sont au nombre de deux :

– Le codeur porte un gant fin et de couleur unie, afin que la distinction entre la main et le visage puisse se faire plus simplement et plus précisément lors de la phase d'analyse vidéo, notamment dans le cas de contact manuo-facial.

– Une étape d'adaptation d'un temps raisonnable, ou phase d'apprentissage, pour permettre au système de connaître individuellement les spécificités de chacun des codeurs, afin d'adapter les méthodes de reconnaissance automatique est envisageable si nécessaire.

I.4 Contexte de la thèse CIFRE

Ces travaux ont été financés par France Telecom R&D, et ont été réalisés pour moitié au GIPSA-Lab/DIS et pour moitié au sein d'une équipe du Laboratoire IDEA, (basé à Meylan, dans le bassin grenoblois), spécialisé sur les terminaux et nouvelles interfaces au sens large. Les raisons qui ont permis cette collaboration entre un grand groupe des télécommunications et un projet initialement universitaire sont multiples :

– Tout d'abord, elle est le prolongement de nombreux partenariats (cités plus haut).

– Ensuite, les problématiques d'accessibilité en général sont en plein développement au sein de France Telecom, en raison des multiples possibilités que permettent les nouvelles technologies de l'information. Celles-ci concernent bien sûr la population malentendante, mais pas seulement. Un des objectifs officiels du groupe est de pouvoir proposer toute une palette de services

modulables permettant l'accès du plus grand nombre au réseau téléphonique, mais aussi aux nouvelles technologies.

– Enfin, les perspectives de réalisation d'un prototype de terminal TELMA sont intéressantes d'un point de vue scientifique (pour les universitaires) comme en termes de développement de services (pour France Telecom).

Cependant, une telle collaboration n'est pas pour autant évidente. Il s'agit de faire un compromis entre des objectifs, des méthodes de travail et des cultures parfois différentes. Cela a été particulièrement facilité par plusieurs éléments :

– La proximité géographique de tous les acteurs ;

– La complémentarité entre les aspects "traitements algorithmiques" universitaires et "terminaux et usages" à France Télécom R&D ;

– L'histoire de France Telecom. Les équipes de chercheurs ont participé à la mutation du groupe consécutive à sa privatisation, et à ce titre, sont capables de faire le lien entre les partenaires universitaires et les exigences industrielles. Cela constitue un avantage unique en termes d'innovation pour France Telecom face à ses concurrents.

I.5 Contexte scientifique

En parallèle de ces travaux, d'autres équipes impliquées dans le projet ont élaboré d'autres modules de TELMA. Nous n'avons pas forcément eu besoin d'échanger avec toutes à part égale, néanmoins, il y a certains travaux avec lesquels les interactions et les échanges ont été indispensables. En plus de la collaboration naturelle avec les autres membres du projet, le travail dont il est question s'appuie sur un contexte scientifique géographique fort. Le GIPSA-Lab/DIS (Département Images et Signaux du laboiratoire GIPSA) est un laboratoire ayant une expertise historique en traitement d'images, et a notamment mené une série d'expériences comparatives sur les études 2D et 3D de la main [12]. De plus, de nombreux travaux y ont été menés concernant l'extraction du mouvement des lèvres, [9], [10], [13], étape nécessaire à la lecture labiale. En parallèle, certains travaux préliminaires ont été menés au GIPSA-Lab/DPC (Département Parole et Cognition du laboiratoire GIPSA), sur :

– l'organisation temporelle du LPC [7]. Ces travaux, menés par Attina, sont les seuls en la matière; lors de notre étude du LPC, ils seront donc notre référence.

– des méthodes de classification adaptées aux formes labiales [A1], [C6].

– Le débruitage audiovisuel de la parole [11], et la synthèse de code LPC à partir de texte [8].

En plus de cela, le GIPSA-Lab/DPC mène des travaux sur la reconnaissance du mouvement des lèvres [2], [3], [4]. La fusion de ces travaux et des travaux

présentés ici, bien qu'indispensable, ne fait que commencer. La pleine maturité de chacun des travaux est un pré-requis indispensable à cette fusion. Nous proposons néanmoins en fin de document, un chapitre à ce sujet (cf. chapitre VII p. 216).

Enfin, c'est le GIPSA-Lab/DPC, et particulièrement Denis Beautemps et Christophe Savariaux qui ont dirigé les campagnes d'acquisition nous permettant de disposer de vidéos de qualité et en nombre suffisant pour nos expériences [4], [A2]. Dans certains cas, celles-ci ont été réalisées à l'occasion d'expériences de "Magicien d'Oz"; ces expériences permettent de jouer des scenarii d'utilisation d'un terminal TELMA dans des conditions réelles, en simulant l'existence d'un tel terminal. Ceci permet d'avoir des retours en termes d'usage et d'ergonomie avant même la création d'un prototype, et donc de cibler plus facilement les difficultés à surmonter. Les scenarii utilisés ont été fournis par les ergonomes de France Telecom R&D [14].

En outre, de nombreux développements scientifiques dans les domaines de la vision par ordinateur et la prise de décision sont issus des laboratoires de France Telecom R&D. Ils constituent une richesse et un vivier important pour ces travaux.

Le LPC, ou de manière plus internationale le Cued Speech, a été très peu étudié en termes de reconnaissance de geste. Cependant, une importante bibliographie concernant la reconnaissance et l'interprétation gestuelle en général existe : elle se focalise principalement sur l'ASL (**A**merican **S**ign **L**anguage) et sur les interfaces homme-machine. Elle constitue une source naturelle d'inspiration, pour :

– Les méthodes de segmentation d'images (extraction des objets d'intérêt tout au long des images de la vidéo, à savoir le visage et la main. Ces aspects sont détaillés dans le chapitre III, p. 58).

– Les méthodes de classification (qui permettent de reconnaître une configuration ou un état particulier à partir de la définition d'une série d'états potentiels, et d'une série de mesures. (Cf. chapitre IV p. 101, chapitre V p. 123, appendice C.1 p. 291 et appendice C.5 p. 301).

– Les méthodes d'étude de phénomènes temporels (Cf. chapitre VI p. 182 et chapitre VII p. 216) qui ont connues de récents développements.

Concernant ce dernier point, une littérature très importante et plus théorique, donc détachée de la problématique directe de l'interprétation gestuelle, est aussi disponible. Il en est d'ailleurs de même pour les méthodes de segmentation, mais le problème que nous adressons étant relativement restreint, il ne sera pas nécessaire d'y faire appel de manière exhaustive.

Dans la mesure où le Cued Speech reste une articulation "visuelle" d'un message oral, il y a aussi beaucoup à espérer de la bibliographie du traitement

de la parole. Dans ce domaine, les méthodes mises au point sont maintenant très efficaces. Cela est bien sûr un avantage, mais aussi un inconvénient car l'état de l'art est plus difficile à remettre en cause : cela rend difficile de s'inspirer d'une méthode propre au traitement de la parole sans embrasser la méthodologie complète qui va avec. Cela revient à traiter le problème original comme n'étant plus qu'un cas particulier du traitement de la parole. Dans notre cas, cela risque de nous perdre : la composante de traitement d'images y est déterminante, et ses imperfections modifient la répartition des données d'une manière inattendue par rapport aux standards du traitement de la parole. Les méthodes connues pour être efficaces peuvent alors se trouver prises en défaut. De plus, notre travail ne franchit pas le fossé sémantique de l'interprétation langagière, nous nous intéressons à la reconnaissance d'un nombre fini de gestes, alors que les méthodes issues du traitement de la parole s'intéressent plutôt à un dictionnaire restreint correspondant à un sous-ensemble du langage et cherchent à l'agrandir petit à petit. Ainsi, la complexité du travail n'est pas là où l'on pourrait l'attendre. C'est pourquoi, nous pensons qu'un certain détachement est nécessaire pour une réutilisation raisonnée des résultats de ce domaine.

Enfin, au rang des collaborations extérieures, mentionnons l'association ADIDA [16], auprès de laquelle il a été possible d'apprendre le LPC, et l'université de Boğaziçi (prononcez "Bohazitchi") d'Istanbul, dont une équipe nous a proposé un échange scientifique des plus enrichissants dans le cadre du réseau européen d'excellence Similar [18]. Celui-ci s'est concrétisé par un "PhD exchange" de deux fois un mois, par de nombreuses publications communes, et ainsi que par l'exploration de thèmes de recherche conjoints dépassant le cadre strict du projet TELMA.

I.6 Plan du rapport

Ce document propose de développer une méthode d'analyse et de reconnaissance du geste de la main en situation de codage LPC. Cette méthode a été développée et testée durant une collaboration entre le GIPSA-Lab/DIS et France Telecom R&D lors d'un travail de thèse CIFRE de 3 ans commencé en novembre 2004 et terminé en octobre 2007. Il s'agit d'un prototype, permettant d'une part, une étude de faisabilité et un questionnement sur des problématiques d'usage, et permettant d'autre part de répondre à des questions scientifiques de manière originale, même si leur application est moins directe. Cet équilibre entre recherche appliquée et application explique pourquoi dans ce document des considérations très proches des besoins des utilisateurs alternent avec des développements plus théoriques, tout en gardant un aspect inachevé, propre à un travail de recherche sur un sujet encore peu abordé.

Le deuxième chapitre (p. 35) présente une étude des spécificités du geste du code LPC. A partir de cette étude, nous proposons une stratégie spécifique et adaptée à la résolution du problème du décodage automatique du mouvement manuel.

Le chapitre III (p. 58) décrit la méthode mise en place pour traiter les images de séquences vidéo de codeur LPC. Nous commençons par présenter le protocole ayant permis d'acquérir ces données. Ensuite, nous exposons l'algorithme de segmentation de la main du codeur que nous avons développé. Ceci permet d'extraire la forme globale de la main et de déterminer où se trouve le doigt pointeur parmi les doigts déployés de la main du codeur.

Le chapitre IV (p. 101) présente un algorithme original de labellisation des images en fonction de leur importance dans la dynamique du codage LPC. Cette labellisation se fait avant la reconnaissance complète du contenu de chaque image. Celle-ci est entre autre rendue possible par l'analyse du mouvement de la main à partir d'un filtre inspiré du fonctionnement de la rétine des vertébrés.

Les méthodes de classification utilisées pour la reconnaissance des différentes composantes du code LPC sont présentées au chapitre V (p. 123). A cette occasion, nous proposons d'appliquer le formalisme des fonctions de croyance à la combinaison de classifieurs, et nous explorons certaines des possibilités que cela permet d'envisager.

Dans le chapitre VI (p. 182), nous présentons les méthodes d'intégration temporelle et de fusion multimodale nécessaires à la reconnaissance du geste à partir de la connaissance des informations de Positions et de Configurations disponibles sur chacune des images du flux vidéo. La qualité des résultats obtenus tient en partie au processus de décision utilisé. Celui-ci permet de prendre des décisions partielles en l'absence d'information suffisante, toute en effectuant un pari sur la solution la plus probable.

Enfin, le chapitre VII (p. 216) est une ouverture sur la manière de faire le lien avec d'autres parties du projet. Principalement, il s'agit de l'étude du mouvement labial, et de la manière dont celle-ci peut être couplée à celle du mouvement manuel, pour un décodage complet du LPC.

Notons que certains développements techniques ne sont pas donnés dans le corps du document afin de ne pas nuire à la continuité de la lecture. Ils sont rapportés en appendices. Dans l'appendice A (p. 255), nous présentons le formalisme des fonctions de croyance, et nous proposons un point de vue permettant de mettre en valeur sa ressemblance avec les méthodes de traitement de l'information classiques. Ce point de vue a pour origine le besoin d'avoir une compréhension cohérente des différentes théories utilisées dans le cadre d'un travail à finalité applicative, et ce, malgré les différentes axiomatiques desquelles relèvent ces différentes théories.

Dans l'appendice B (p. 277), nous donnons des explications et des preuves mathématiques à propos de la méthode de décision proposée au chapitre VI.

Enfin, l'appendice C (p. 290) permet de présenter certains détails d'implantation. Ceux-ci ne sont pas indispensables à la compréhension du document, mais seront utiles à quiconque cherche à implanter des traitements similaires.

CHAPITRE II
DESCRIPTION DES GESTES DU LPC

Dans ce chapitre, nous étudions le geste LPC proprement dit, dans le but d'en connaître suffisamment les spécificités pour mettre en place une stratégie de décodage adaptée. En effet, la reconnaissance de geste en général est un problème complexe et ouvert, mais il est envisageable de pouvoir reconnaître un type de gestes particuliers (tels que ceux du LPC) en tirant partie des contraintes associées. Ces spécificités représentent une connaissance *a priori* permettant de contraindre le problème. Dans la première section, nous définissons des éléments de vocabulaire et précisons quelques points importants. Ensuite chacune des trois sections suivantes traite d'un aspect particulier du code LPC. Enfin, dans la section 5, nous proposons la stratégie globale de décodage sur laquelle se fondent ces travaux.

II.1 Spécifications pour la reconnaissance du LPC

Ici, nous précisons quelques éléments supplémentaires en lien avec le LPC. Ceux-ci sont d'une importance secondaire dans l'usage courant du LPC. En revanche, ils sont nécessaires à l'automatisation de son décodage.

Plan d'acquisition : suite aux travaux parus dans [12], il a été décidé que le terminal TELMA serait basé sur des acquisitions avec une seule caméra. Cela signifie que toutes les informations visuelles doivent être visibles sur un seul plan. Nous appelons un tel plan, le **plan d'acquisition**. Au sein du volume de la pièce dans laquelle le codeur est filmé, le plan d'acquisition désigne le plan passant par le codeur et qui est perpendiculaire à l'axe de visée de la caméra. Les mouvements effectués par le codeur dans ce plan seulement sont enregistrés sans aucune déformation.

Mouvements du poignet : le poignet est une articulation qui possède deux degrés de liberté, permettant autant de types de mouvement : le premier est le mouvement de **flexion/extension** (Figure II-1a). Dans le cadre du codage du LPC, l'axe de l'avant bras est censé rester dans le plan de la paume de la main. Il arrive malheureusement que la flexion du poignet vienne perturber cela. Couramment, on parle d'un mouvement où le poignet est "cassé". Le second mouvement possible est le mouvement d'**adduction/abduction** du poignet (Figure II-1b), ou **inclinaison radiale/cubitale** : c'est le seul mouvement de poignet qu'il est possible de réaliser en conservant à la fois l'avant-bras et la paume de la main posés à plat sur une table. Ce mouvement est en fait une rotation de la main autour d'un point situé entre le **scaphoïde** et la tête cubitale ; il apparaît couramment dans le codage du LPC, sans que cela ne soit une entrave à son décodage (automatique ou non). Pour une description plus anatomique et plus précise du fonctionnement et de la morphologie du poignet, cf. [165], [164]. Enfin, il y a un dernier mouvement possible de la main que l'on voit fréquemment durant un codage LPC, mais qui n'est pas un mouvement du poignet proprement dit. Il s'agit du mouvement de **supination/pronation** (Figure II-1c), qui consiste en la rotation du radius et du cubitus l'un par rapport à l'autre, et qui permet la rotation de la main que l'on utilise couramment pour fermer un robinet de douche murale. Ce mouvement est à

proscrire durant le codage LPC, car il implique une rotation de la main de sorte que celle-ci ne se trouve plus parfaitement dans le plan de codage. Pour un tel mouvement, nous parlons parfois de rotation du poignet, ce qui est un abus de langage, mais qui est aussi plus imagé.

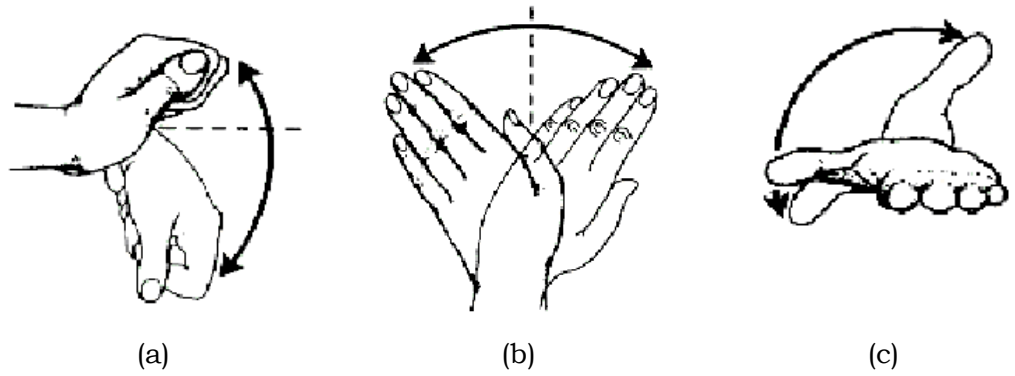


Figure II-1 : mouvement (a) de flexion/extension, (b) de adduction/abduction, et (c) de supination/pronation [164].

Ecartement des doigts en LPC : en théorie, parmi les 8 configurations du LPC, les 7 premières sont effectuées avec les doigts regroupés, et la 8ème est réalisée avec le majeur et l'index séparés. Sur le schéma de la Figure I-3, les 7 configurations sont toutes représentées avec les doigts légèrement écartés, et la dernière configuration avec les doigts complètement écartés. Nous sommes conscients de l'inexactitude de l'iconographie proposée, mais la séparation légère des doigts permet de plus facilement les compter, et donc de plus facilement faire la différence entre les différents logos de main. Nous avons donc à dessein décidé de garder une telle représentation.

Cependant, quand on regarde un codage rapide et courant du LPC, il apparaît que de nombreuses configurations sont réalisées avec des doigts plus ou moins écartés. Cela arrive même dans le cas de la configuration 2, qui alors devient semblable à la configuration 8. Nous pensons que cet écartement des doigts est dû à la vitesse de codage. Quand celui-ci doit être rapide, l'élan musculaire est plus important, la trajectoire des articulations moins précise, et le geste moins contrôlé. Cela apporte une plus grande variabilité aux formes de main, et complique beaucoup la tâche de reconnaissance automatique.

Doigt pointeur : étant donné un geste constitué d'une configuration et d'une position, parmi tous les doigts qui sont déployés pour la réalisation de la configuration, un seul est utilisé pour pointer précisément la position. C'est le doigt pointeur. Celui-ci est tout simplement le plus long parmi les doigts déployés. Ainsi, quand le majeur est déployé, il est le doigt pointeur, et sinon, il s'agit de l'index. Ceci est élémentaire et assez naturel, cependant, en raison des problématiques d'acquisition mono-caméra, il arrive souvent que le doigt qui semble le plus long à l'écran ne soit pas le doigt qui morphologiquement est le plus long. La définition du doigt pointeur devient donc problématique, et sera l'objet d'un travail à part entière.

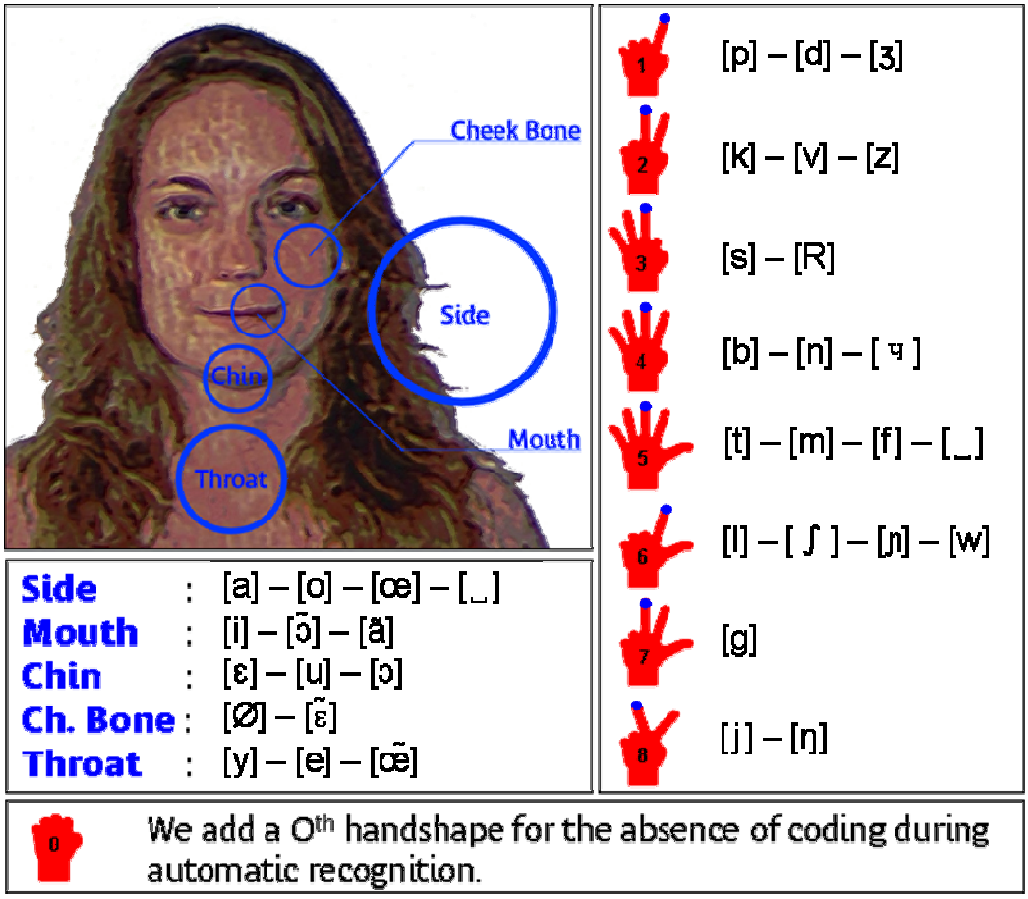


Figure II-2 : complément de définition du Code LPC.

Configuration 0 : afin de spécifier l'absence de codage sans pour autant que la main disparaisse du champ de la caméra, nous avons ajouté la **configuration 0**, désignant l'absence de codage (cf. Figure II-2). Cette configuration est toujours réalisée en position côté, qui par sa spécification moins grande et par son utilisation pour coder les voyelles muettes, est souvent définie comme la position neutre. Comme la configuration 0 ne peut pas être associée à une autre position que la position neutre, il n'est pas nécessaire de définir un doigt pointeur pour celle-ci. Ainsi, l'ensemble des gestes que nous considérons dans notre étude est de {5 positions} × {8 configurations} + {configuration 0 en position neutre} = 41 gestes.

En fonction de tout cela, nous pouvons reprendre la Figure I-3 et la modifier comme indiqué sur la Figure II-2.

D'une manière générale, il ne faudra pas confondre les éléments définis plus haut avec le sens courant que l'on donne aux mêmes mots. Pour bien marquer cette différence, nous parlons de **Configuration** (avec une majuscule) quand nous faisons référence à un des gestes standard du LPC, et le mot **configuration** (sans majuscule) pourra être utilisé dans son sens classique, indépendamment du LPC. Il en est de même pour **Position** et **position**. Ensuite, il ne faut pas confondre les significations morphologiques et les significations

sémantiques. Ainsi, une main dont tous les doigts sont repliés, à l'exception de l'annulaire et de l'auriculaire, n'est pas une Configuration du LPC, mais représente bien une configuration particulière des articulations de la main. Dans ce dernier cas, nous parlerons de "forme" de la main (signification morphologique), qui peut éventuellement correspondre à une Configuration du LPC (et donc avoir une signification sémantique). Nous proposons ce mot **forme** pour la simple raison que la plupart du temps, nous n'aurons pas besoin de faire la différence entre la forme telle que nous venons de la définir, morphologique (position angulaire de chacune des articulations), et la forme 2D à l'écran du contour de la main, tel qu'il est couramment défini en **reconnaissance de forme**. De même, nous parlons de **zones pointées** (signification morphologique), par opposition aux Positions LPC (signification sémantique), et le nom de chaque Position commence par une Majuscule (Menton, Côté, etc. ...) alors que ce ne sera pas le cas pour les zones pointées. Enfin, nous parlerons de **doigt pointeur** (signification sémantique) et d'**élément pointeur** (un morceau de chair, ou un pixel de l'image, pour la signification morphologique ou géométrique).

II.2 La nature statique du code LPC

Dans cette section, nous nous intéressons à une propriété très particulière du LPC. La manière la plus simple de présenter le détail du code LPC est d'utiliser un dessin, comme celui de la Figure II-2. Cela est d'autant plus facile que chaque geste, et même chaque composante (Configuration et Position) est intrinsèquement statique : elle est caractérisée par une position particulière des articulations de la main et du bras, et aucun mouvement n'intervient dans ce geste, si ce n'est sa mise en place et sa défection. C'est cette absence de mouvement, ou de composante dynamique, qui permet de représenter entièrement un geste à partir d'une photo. A l'inverse un hochement de tête, une négation, un salut de la main, la lettre 'X' en dactylologie (cf. Figure I-1), etc. font intervenir un mouvement dans leur définition, et la suppression de ce mouvement ne permet plus de reconnaître le geste en question. Ainsi, nous qualifions de **statiques** les gestes du LPC, et de **dynamiques**, les autres gestes mentionnés.

Bien sûr, le codage courant ressemble beaucoup plus à un mouvement continu de la main et des lèvres qu'à un mouvement continu des lèvres accompagné d'un enchaînement de gestes statiques. En effet, si l'on regarde le mouvement manuel d'un codage courant (cf. enchaînement de la Figure II-3 ainsi que les vidéos disponibles à l'adresse [19]), il est clair que l'ensemble des gestes produits, aussi statiques soient-ils pris individuellement, se retrouve fondu dans un mouvement continu. Evidemment, la main ne peut pas simplement apparaître dans un geste particulier (Configuration + Position), puis disparaître une fraction de seconde et réapparaître ailleurs dans une autre Configuration ou Position. Par conséquent, il y a forcément un mouvement continu. Ce geste est tellement **coarticulé** (il y a un équivalent manuel aux liaisons sonores que l'on effectue entre les phonèmes), que le geste statique n'apparaît pas en temps que

tel dans le continuum du mouvement manuel : au moment de la réalisation d'un geste, la main ne s'arrête jamais complètement. Il en résulte que le codage du LPC semble réellement dynamique.

Il n'empêche que le code lui-même est statique au sens où nous l'avons défini plus tôt. **Cela signifie qu'il existe, pour chaque geste réalisé, au moins une image l'exprimant complètement.** Cela veut dire qu'il est possible de reconnaître l'enchaînement des gestes produits par le codeur à partir d'un sous-ensemble d'images soigneusement choisi. Dans l'ensemble dynamique, nous appelons image de **transition** toute image qui ne correspond pas à la réalisation d'un geste statique. Pour l'image du geste noyé dans le flux dynamique et correspondant à la réalisation d'un geste statique, nous parlons d'image **cible**. Ainsi, dans un flux vidéo, deux images cibles (représentant un geste statique dans un continuum dynamique) sont séparées par un certain nombre d'images de transition (dues aux mouvements intermédiaires de la main pour passer d'un geste statique du LPC à un autre).

Evidemment, les linguistes sont en général en désaccord avec ce que nous venons d'affirmer. L'étude de la parole orale était initialement menée de la même manière, et elle n'a pas proposé de modèles acceptables tant que l'hypothèse de mouvements articulatoires modélisables par un enchaînement de cibles statiques a été maintenue. Ce n'est qu'en récusant cette hypothèse que cette science a pu progresser. Comme du point de vue des linguistes, le LPC n'est qu'un moyen différent d'articuler une même langue, il en est forcément de même pour le LPC. Cela n'est pas exact d'un point de vue phonémique, même si cela est probablement vrai à un niveau supérieur d'interprétation.

La parole est véhiculée par un son modulé en intensité et en fréquence. Si l'intensité peut être définie de manière statique par rapport au temps, la fréquence est intrinsèquement temporelle. L'hypothèse de trajectoire articulatoire orale statique est par conséquent difficilement défendable. En revanche, comme nous l'avons dit plus haut, l'information de niveau phonémique est représentée de manière statique dans le LPC.

Les linguistes ne s'intéressent pas au même type d'information : ils se placent au niveau linguistique, qui est un niveau beaucoup plus élevé. A ce niveau, l'information transmise est beaucoup plus riche que le simple enchaînement des phonèmes. Il y a l'intonation, le rythme, la ponctuation, l'émotion, et même certaines informations qui devraient rester d'ordre paralinguistique. Si l'on suppose que ces informations sont naturellement transposées au LPC, alors, évidemment, l'information complète codée en LPC ne peut plus être représentée par un échantillon d'images représentant un geste statique.

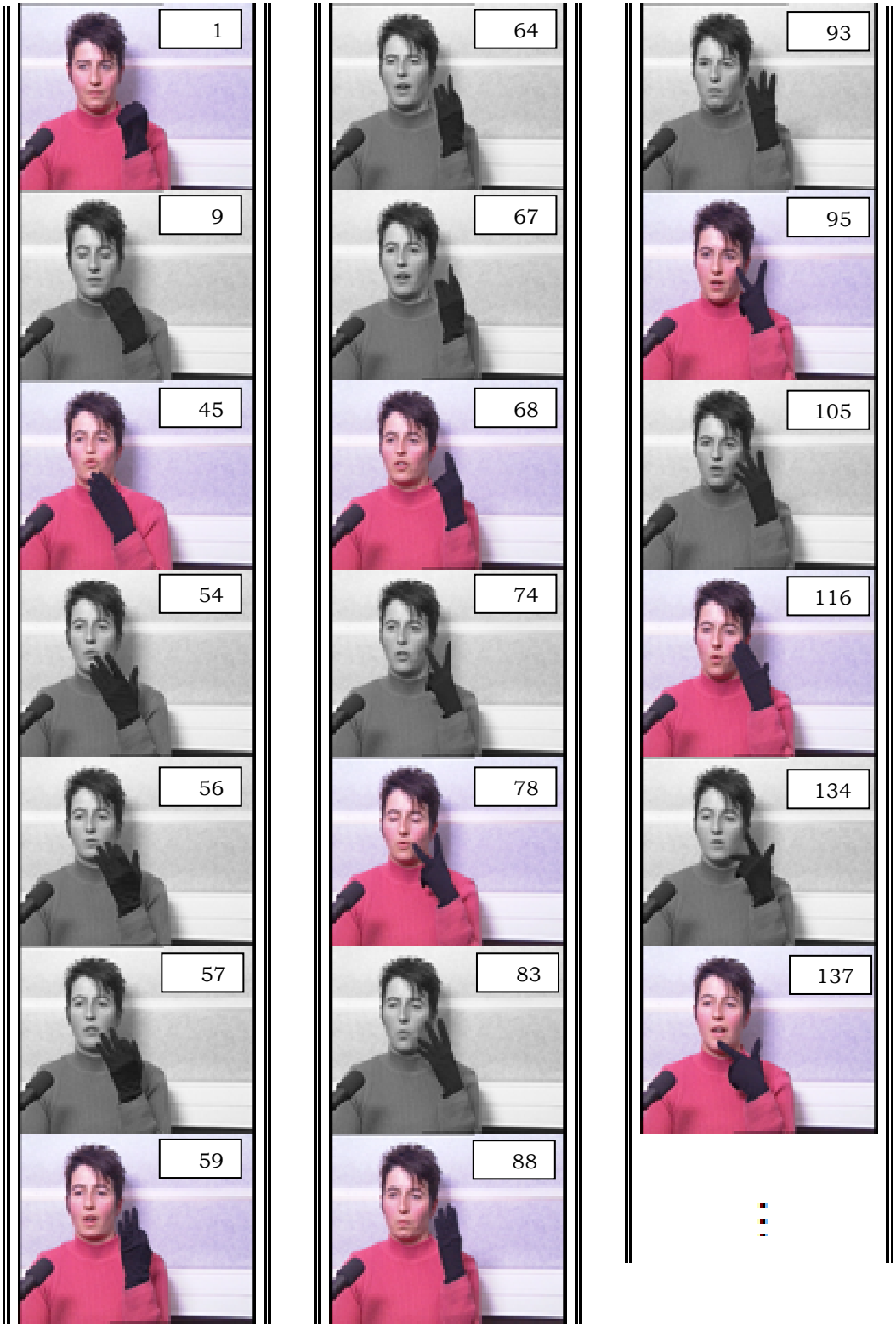


Figure II-3 : les gestes statiques sont fondus dans un mouvement continu. Les photos en noir et blanc représentent les images acquises pendant les mouvements de transition, alors que les images en couleur représentent les gestes statiques. Toutes les images de la séquence ne sont pas représentées. L'indice de l'image est indiqué dans le rectangle en haut à droite de chaque image.

Dans un premier temps, nous n'avons pas la prétention de décoder autre chose que le flux de phonèmes, et de transmettre cette information "réduite". Dans le cadre de cet objectif, l'hypothèse d'un codage statique noyé dans un flux continu est parfaitement acceptable ; nous en voulons pour preuve qu'accepter cette hypothèse simplifie grandement le problème, et que nous avons aussi travaillé sur des gestes pour lesquels celle-ci n'est pas valable. Il n'en ressort aucune simplification : des quelques travaux que nous avons menés en collaboration avec l'université Boğaziçi sur la Langue des Signes (soit Turque, soit Américaine), il apparaît que les modèles et méthodes à mettre en place sont beaucoup plus compliqués (les Langues des Signes sont réellement dynamiques), et beaucoup moins efficaces : le problème est donc réellement plus complexe. La nature statique du geste du LPC est donc une réalité.

Dans l'ensemble de nos travaux, nous pensons qu'il est indispensable d'utiliser cet aspect du LPC pour parvenir à de bons résultats. Ce sont donc les remarques de cette section qui vont guider la mise en place d'une stratégie complète pour la résolution de notre problème.

Bien sûr, si le traitement du LPC au niveau phonémique n'est pas parfait, il sera peut-être nécessaire de récupérer une information dynamique, d'un niveau supérieur (éventuellement un modèle de langage) pour régulariser les résultats obtenus sur une analyse statique. Cependant, cela ne remet pas en cause le fait que nous pensons trouver l'information nécessaire au niveau de l'image statique.

II.3 Les imperfections du codage humain

II.3.1 Codage réel et codage théorique

En théorie, le code LPC est assez simple. La quantité de gestes autorisés est restreinte, et leur variabilité sévèrement contrôlée. En se basant sur la définition théorique du LPC, le travail de décodage semble élémentaire. Malheureusement, le LPC est vivant au même titre que la langue orale qu'il sert à coder, et la communauté des codeurs se l'est approprié. Finalement, le codage réel s'éloigne parfois beaucoup de la définition théorique originale. De plus, chaque codeur se l'approprie à un autre niveau, et possède son propre "accent", sa propre dynamique, etc. de telle sorte que le code est imparfaitement et variablement effectué, avec un biais typiquement humain par rapport au codage théorique tel que peut le réaliser un avatar de synthèse [8].

Tout cela complique notre tâche à plusieurs niveaux :

– Tout d'abord, nous devons être capables de définir à quels types de codage nous acceptons de nous confronter pour cette tâche de décodage automatique. Il faut évidemment restreindre nos prétentions à un codage académique, même si par la suite, il sera toujours possible d'essayer de l'étendre. Il est nécessaire d'étudier et de comprendre les différents axes selon lesquels le code s'éloigne de sa production théorique. Pour chacun d'eux, il faut définir une limite acceptable entre un code perturbé mais humain, et un code qui perd de sa véracité. Cette

limite doit à la fois garantir que, dans le champ restreint que nous adressons, les performances seront bonnes, mais aussi que ce champ est suffisamment souple pour permettre à chacun de coder d'une manière qui lui semble suffisamment naturelle.

– L'étape suivante est d'être capable d'effectuer automatiquement une classification entre le code que l'on accepte de traiter et celui que l'on considère comme non gérable. Dans l'idéal, il serait même encore plus intéressant de pouvoir utiliser cet outil comme un tutoriel : cela permettrait de pointer les différents défauts afin que le codeur les corrige, mais cela dépasse de beaucoup les limites de ce travail.

– Ensuite, il faut mettre en place une stratégie de décodage adaptée aux humains. En effet, il est fort possible qu'une méthode étant initialement conçue pour traiter un codage théorique parfait, ne puisse pas être améliorée pour satisfaire les exigences d'un codage humain.

– Cette méthode doit garder une certaine souplesse permettant sa remise en cause partielle : au fur et à mesure que le LPC est et sera étudié, la connaissance de ses imperfections et ses spécificités humaines progressera.

Des études que nous avons menées sur le LPC, il apparaît plusieurs différences importantes entre un codage humain et un codage théorique. Nous détaillons cela dans le reste de cette section. Ensuite, dans la section suivante, nous nous intéressons à la dynamique dans laquelle sont immergés les gestes statiques. Cette dynamique est complexe en raison de la désynchronisation des différents articulateurs visuels (Position de la main, Configuration, et mouvement aux lèvres). Comme nous nous focalisons sur le décodage du geste manuel, nous abordons ici exclusivement les problèmes de synchronisation entre les flux de Positions et de Configurations. En ce qui concerne la synchronisation labiale et manuelle, elle aussi très complexe, nous renvoyons aux résultats de [7] et de [3] ainsi qu'au chapitre VII (p. 216).

II.3.2 Contraintes de codage au niveau linguistique

La première contrainte est la **coarticulation** : le rythme de codage de la main doit dans une certaine mesure suivre le rythme de la parole ; un certain nombre d'accélérations et de décélérations difficiles à cerner perturbent le codage. De plus, comme cela est expliqué dans la première partie, le code produit respecte une alternance de consonnes C et de voyelles V dans un codage CV. Dans le cas de diphtongues ou d'enchaînement de plusieurs consonnes, des consonnes ou des voyelles muettes (respectivement) sont ajoutées. Il en résulte que toutes les syllabes se codent à la main avec des mouvements de complexité équivalente, alors que leur articulation labiale peut être beaucoup plus simple dans certains cas que dans d'autres. Afin de ne pas trop ralentir le mouvement labial, la main accélère alors beaucoup, et un phénomène de **coarticulation** manuelle apparaît : les gestes s'enchaînent en un seul mouvement où chacun des gestes apparaît comme réellement transitoire. Il nous est même arrivé de voir une codeuse "mâcher" ses gestes de la même manière que l'on peut mâcher ses mots : alors qu'elle devait réaliser la séquence manuelle {(Conf_*i* ; Pos_*j*),

Transition, (Conf_*k* ; Pos_*l*), Transition, (Conf_*m* ; Pos_*l*)}, elle a effectué l'enchaînement {(Conf_*i* ; Pos_*j*), Transition de Position avec Configuration *k*, (Conf_*m* ; Pos_*l*)}. Un tel enchaînement est parfaitement compréhensible pour un humain, mais une machine n'est pas capable de reconstituer le bon enchaînement de gestes sans un niveau supérieur d'interprétation (sémantique ou lexical).

Dans le même ordre d'idée, quand quelqu'un parle, il peut lui arriver d'hésiter, de bafouiller, de s'arrêter ou de recommencer un peu en arrière et de répéter une ou deux syllabes. Cela est parfaitement compréhensible pour un humain qui corrige en fonction du sens de l'enchaînement de sons. Il en est bien évidemment de même en LPC. En revanche, un système de traduction automatique basé sur la gestuelle simple ne peut faire cela. Sans aller jusqu'à des corrections complètes et des répétitions, (qui sont évidemment ingérables sans niveau linguistique réel), un geste qui est un mélange temporel de deux Configurations ou Positions ne peut non plus être interprété correctement : il arrive en effet que le codeur commence par un geste, puis réalise qu'il fait une erreur de prononciation/codage, et sans reprendre son code en arrière effectue la correction en temps réel et modifie la trajectoire de ces articulations pour que le geste final soit bon, puis immédiatement enchaîne avec un autre geste.

A un niveau sémantique encore supérieur, il arrive qu'au sein du code, un geste pantomime ou aillant une forte connotation linguistique mais n'appartenant pas au code LPC, se glisse au milieu du codage. Dans certains cas, on aperçoit même certains gestes de langue des signes ! Tout mouvement paralinguistique des mains qui se confond au code ne peut évidemment pas être interprété.

Dans le cas de communauté de codage très restreinte, (groupe familial, enfant-parent, par exemple), d'importantes distorsions deviennent habituelles et ne posent pas de problème en terme de communication, puisqu'elles sont utilisées, connues et reconnues par tout le groupe. En revanche, celles-ci ne peuvent être apprises par la machine. Ainsi, par exemple, nous avons rencontré une jeune fille qui utilisait le LPC mais avait adopté, quand elle codait avec sa mère, la convention gestuelle anglaise pour le codage des diphtongues, qui est plus simple et plus efficace que sa version française. Il en résultait un mélange de code français et de code anglais.

Dans certains gestes, la rotation du poignet (mouvement de supination/pronation) est tellement importante que la main se retrouve de profil, et les doigts déployés se retrouvent les uns derrière les autres (cf. Figure II-4). Dans un tel cas, un humain est parfaitement capable de décoder le code à partir du mouvement qu'il voit [17]. En décodage automatique, ce genre de situation reste insoluble.

De même, le "cassé" du poignet, ou mouvement de flexion (cf. Figure II-4) ne permet plus de voir les doigts, et empêche toute reconnaissance de la Configuration. Ce genre de rotation (de même que la précédente) est parfaitement compréhensible pour des raisons de morphologie, d'efficacité et

d'économie d'énergie. De plus, celui-ci n'étant pas véritablement une entrave à la compréhension de l'interlocuteur entraîné, il devient habituel.

Figure II-4 : exemples de rotation rendant impossible la différenciation entre les Configurations 3 et 4 (en haut à gauche), ou entre les Configurations 2 et 8 (en haut à droite) et de flexion (en bas) du poignet déformant les doigts, de telle sorte qu'il est parfois impossible de reconnaître la Configuration (l'image en bas à droite représente une Configuration 3).

D'une manière générale, certains défauts peuvent être facilement corrigés par les codeurs en cause ; en revanche il est impossible, dans l'état des connaissances actuelles, de les prendre en compte dans le codage que la machine doit être capable de traduire de façon robuste. Ce sera donc à chacun des codeurs de faire attention. En revanche, il y a de nombreux artefacts mineurs qui eux sont inévitables pour un codage fonctionnel. Ils peuvent être appréhendés au niveau machine, moyennant quelques efforts supplémentaires. Dans un tel cas, c'est aux concepteurs de faire l'effort nécessaire.

II.3.3 Acceptation des variations morphologiques du codage

Dans le cas où deux syllabes consécutives se codent avec le même geste (Position + Configuration), celui-ci doit être produit deux fois, c'est-à-dire réalisé une première fois, puis répété. Chez certaines personnes, cette répétition apparaît de manière évidente, puisqu'elles font l'effort de marquer un mouvement d'aller retour vers la Position en question, soit en se redirigeant vers la Position Neutre (Position Côté), soit en effectuant un mouvement de zoom avec la main, c'est-à-dire un aller retour rapide en direction de leur

interlocuteur. En revanche, chez d'autres personnes, ce mouvement est trop faible pour être perceptible dans tous les cas, voire inexistant. Dès lors, seule la suppléance mentale permet à l'interlocuteur de comprendre qu'il n'y a pas eu une seule syllabe, mais deux. A partir du moment où le mouvement est perceptible par un humain, il doit être considéré comme suffisant et notre système devra pouvoir s'en contenter.

La rotation et le cassage du poignet ont déjà été abordés, et nous en avons déduit qu'ils ne pouvaient pas être appréhendés par la machine : il est nécessaire que les codeurs les limitent au maximum. Pour autant, il ne serait pas raisonnable d'espérer les voir complètement disparaître. Ainsi, une rotation minimale de quelques degrés doit pouvoir être incluse dans les modèles de reconnaissance, ce qui n'est pas sans poser de nombreux problèmes dans le cas d'une acquisition mono-caméra [12].

Dans le même ordre d'idée, la variabilité sur l'écartement des doigts pour la Configuration 8 (où les doigts doivent être écartés), mais aussi pour les autres Configurations, pour lesquelles les doigts doivent rester serrés, est un problème. Cette variabilité est aussi valable pour les doigts repliés, pour lequel le degré de repli et le degré de visibilité a une influence (par exemple, (1) des morceaux de doigts peuvent dépasser de derrière la paume, (2) certaines phalanges ne sont parfois pas assez dépliées, ou au contraire trop peu repliées).

(a) Le poignet est cassé : les autres doigts apparaissent plus court que le pouce.

(b) Le majeur n'est pas autant dans le plan de l'acquisition que l'index. Celui-ci semble donc plus long, et l'élément pointeur n'est pas le bon.

Figure II-5 : l'élément pointeur ne correspond pas à l'extrémité du doigt déployé le plus long.

En regardant des vidéos de codage effectuées par des codeurs professionnels, il apparaît que le doigt pointeur est lui aussi source de beaucoup de difficultés pour la Configuration 8. Tout d'abord, l'écartement des doigts est beaucoup plus important, mais comme la dynamique de Position de la main n'est pas particulièrement changée pour cette Configuration, le doigt pointeur (le majeur dans le cas de la Configuration 8) peut apparaître dans l'alignement du visage. Avec une acquisition mono-caméra, il semble donc que la zone pointée correspond à un élément du visage et non à la Position Côté (cf. Figure II-5b).

Le doigt pointeur est théoriquement le doigt le plus long parmi les doigts déployés. Comme il arrive souvent que la main ne soit pas parfaitement dans le plan de la caméra, il semble sur l'image que le doigt pointeur soit un autre doigt que le doigt le plus long (Figure II-5).

Il arrive aussi que ce soit l'annulaire et non le majeur qui serve à pointer la Position Bouche, ce qui morphologiquement est tout à fait compréhensible. Heureusement, quand les doigts restent groupés, l'erreur que cela implique devient minime (de l'ordre de l'épaisseur d'un doigt).

Si l'on analyse la manière de réaliser les Configurations, il existe une importante variabilité pour chacune d'elle, et parfois, les différents gestes deviennent similaires au point de se confondre. Ceci est parfois gênant, mais il est possible malgré tout d'espérer pouvoir faire la distinction. En revanche, dans certains cas, les Configurations se mélangent d'une autre manière : un geste ne ressemble à aucune Configuration, mais semble être un mélange de deux d'entre elles (cf. Figure II-6). Cela n'arrive pas uniquement pendant les phases de transition entre deux gestes et il semble difficile de pouvoir lever une telle ambigüité sans interprétation linguistique.

Figure II-6 : la configuration intermédiaire des doigts produite lors de la réalisation du geste statique est un mélange de la Configuration 8 et de la Configuration 7.

Dans le cas d'un ralentissement conséquent entre deux Positions ou entre deux Configurations très proches, ou dans le cas d'une hésitation, il arrive que les deux gestes soient produits avec une trop grande lenteur, de telle sorte que l'on ne fait plus la différence entre les deux gestes proprement dits et la trajectoire ou transition permettant de passer de l'un à l'autre. Le geste dérive progressivement, et dès lors, sa délimitation et sa reconnaissance deviennent délicates. Cette notion de **dérive** reviendra à plusieurs reprises dans l'analyse du LPC.

Enfin, la variabilité de la Position de la main est énorme : certaines personnes codent la Position Pommette en pointant la zone de la joue, d'autres en pointant l'os de la pommette proprement dit, d'autres le coin extérieur de l'œil. Il en est

de même pour la Position Gorge. Certains pointent la zone du cou, très près du menton, d'autres descendent jusqu'à la poitrine (en cassant généralement beaucoup le poignet). La variabilité est telle qu'il semble impossible de définir les Positions de codage de manière indépendante des codeurs sans un apprentissage préalable (cf. Figure II-7).

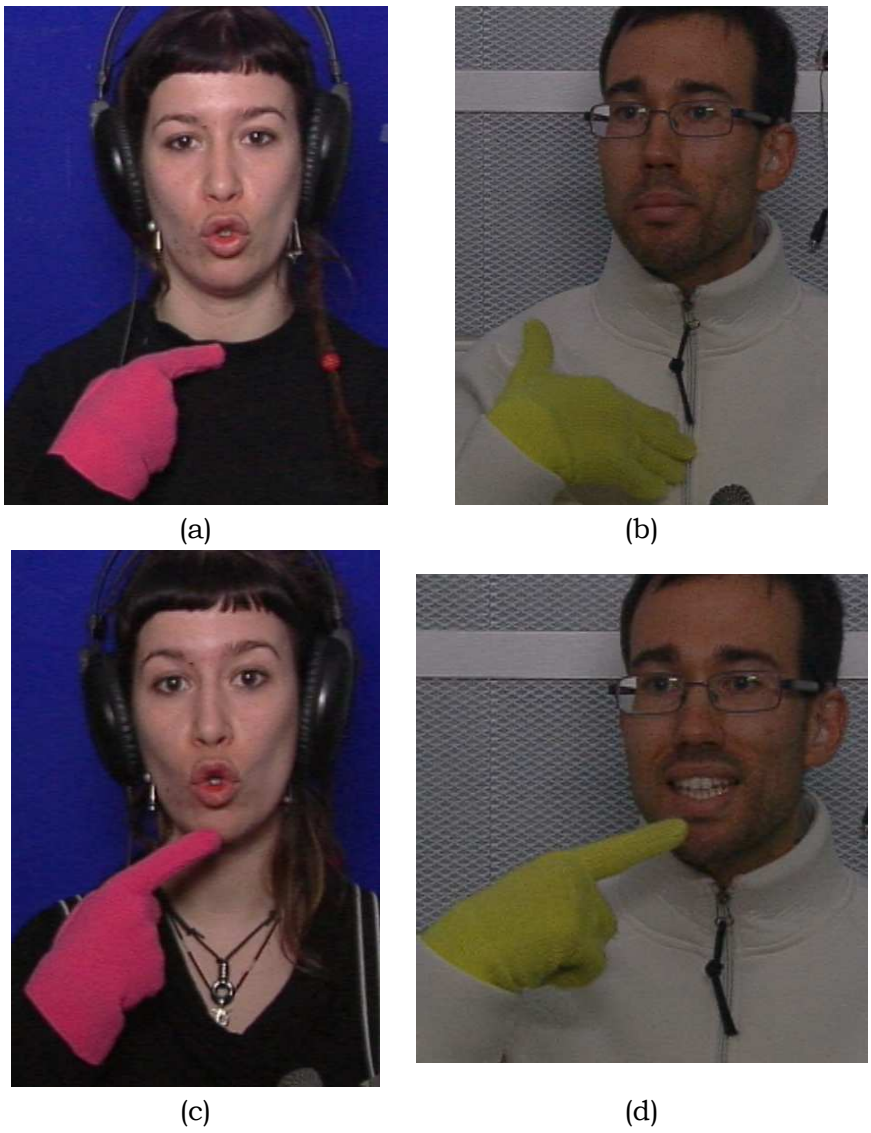


Figure II-7 : (a) et (b) représentent un codage de la Position Gorge, et (c) et (d) un codage de la Position Menton.

Notons enfin les éventuels problèmes d'occultation des lèvres par la main, mais nous ne nous sommes pas penchés sur ce problème, puisqu'il relève de l'analyse labiale.

II.4 Synchronisme des composantes manuelles du LPC

Dans cette section, nous détaillons nos conclusions quant au synchronisme des deux mouvements de la main (les mouvements de changement de

Configurations et de changement de Positions). Cela est issu de l'observation de nombreuses vidéos de codeurs LPC professionnels, ou de codeur LPC malentendants. Ces vidéos sont celles que nous avons utilisées pour la mise en place de nos algorithmes. Le détail du contenu de ces vidéos ainsi que leurs conditions d'acquisition sont décrits dans la section III.1 (p 59). Notons malgré tout que les codeurs LPC malentendants sont jeunes (cela n'est pas lié à un intérêt particulier pour une classe d'âge, mais simplement conditionné par le fait qu'il n'y a pas encore d'adulte ayant été exposé précocement au LPC, puisque son existence remonte à 1979), et que les codeurs professionnels sont des femmes (la plupart sont devenues des codeuses certifiées suite à la naissance d'un enfant malentendant). Cette étude est entièrement qualitative. Elle est cependant indispensable à une compréhension réelle de la manière dont le code théorique a été approprié par la population l'utilisant. De plus, à partir de la connaissance de la manière dont le LPC interagit avec le cerveau humain, il est possible d'inférer certaines structures dans le LPC et donc de travailler sur un flux d'informations plus cohérent. Enfin, il est dans certains cas possible de s'inspirer du fonctionnement du cerveau pour la tâche de décodage, et d'essayer de l'imiter de manière simpliste pour dégager une piste algorithmique.

Dans l'ensemble de ce document, quand nous discutons des problèmes de synchronisation, nous prenons toujours pour référence un **code théorique parfaitement synchronisé**, tel qu'il serait produit par un codeur respectant la définition du code LPC de **manière absolue** (ce qui est pratiquement impossible, en raison de ce qui est expliqué dans la section précédente) ou par un clone de synthèse animé par une machine.

La première observation que l'on peut faire est que d'une manière générale, la cible gestuelle correspondant à la pleine réalisation de la Configuration est atteinte avant la cible gestuelle de Position. Cela est facilement compréhensible pour des raisons mécaniques et morphologiques : dans le cas d'un contact entre le visage et le doigt pointeur, celui-ci doit être entièrement déployé avant le début du contact. Ainsi le flux de Configurations est légèrement en avance sur le flux de Positions.

Cela n'est cependant pas toujours vrai dans le cas de l'atteinte de la cible lors d'un codage en Position Côté. En effet, cette Position nécessite un pointage moins précis (il ne faut pas toucher le visage), et par conséquent, il n'est pas nécessaire que le doigt pointeur soit complètement déployé (donc la Configuration bien déployé n'a pas besoin d'être maintenue). Dans le cas contraire, quand la main quitte la Position Côté, la Configuration se retrouve naturellement en avance. Cependant, dans de nombreuses situations, la nouvelle cible de Configuration est déjà atteinte alors que la main n'a pas eu le temps de se rapprocher du visage suffisamment pour que la Position ne soit pas interprétée comme Côté. Ainsi, il apparaît des images sur lesquelles un geste supplémentaire "fantôme" apparaît.

Il y a une autre situation dans laquelle cette règle générale du retard de la Position sur la Configuration est mise en défaut : lorsqu'entre deux syllabes

consécutives, la Position reste inchangée, alors, la Configuration arrive bien évidemment avec du retard.

Enfin, cette règle générale est fortement modifiée par la prosodie du message à coder. De manière évidente, le mouvement labial peut être plus rapide que le mouvement de la main : l'amplitude du mouvement labial est moins grande, de sorte que les mouvements des lèvres peuvent être facilement enchaînés. Or, le code LPC est censé être produit à la vitesse de la diction courante, de sorte que le mouvement de la main n'est pas censé ralentir la dynamique labiale. En pratique, cela est toujours un peu le cas malgré tout, mais cet effet doit être minimisé. Sans rentrer dans le détail de la synchronisation main/lèvre, cela a plusieurs conséquences sur la désynchronisation Position/Configuration :

– En début de phrase, la main est en avance sur les lèvres d'un temps parfois supérieur à celui nécessaire pour coder la première syllabe. En fin de phrase, c'est le contraire. Cette modification du rythme du code manuel se trouve souvent accompagné de très forts décalages entre Position et Configuration.

– Le codage est perturbé par la charge cognitive de la personne. Si la phrase à prononcer est difficile, il semble que cela ne fasse pas intervenir les mêmes mécanismes cognitifs (l'attention ne doit pas être focalisée de la même manière), et ainsi, le codage perd de son naturel. Cette perte de naturel implique une synchronisation différente entre Position et Configuration. Il est aussi possible que cela ait été renforcé par le fait que les codeurs se savaient filmés lors des acquisitions.

– les différents messages à coder peuvent avoir une prosodie labiale très différente, mais doivent tous être complétés par un enchaînement de gestes manuels dont la vitesse ne varie que très peu (en fonction des différentes distances à parcourir entre les Positions de codage). Dans certains cas, la main doit donc accélérer fortement pour pouvoir suivre le rythme labial. Quand cette accélération devient trop forte, le code est perturbé. La synchronisation de la Position et de la Configuration s'en trouve altérée et certains cas, des gestes peuvent être "mâchés", comme cela a été illustré à la section précédente.

En conclusion, nous avons constaté que l'ordonnancement temporel du codage de la Configuration par rapport à celui de la Position n'est pas aussi synchrone qu'il n'y paraît au premier abord. Il se trouve qu'un synchronisme parfait entre Position et Configuration se rencontre parfois dans un codage très académique, mais qu'en pratique il est souvent remplacé par une avance de l'atteinte de la Configuration. De plus, ce décalage général n'est pas toujours respecté non plus, de telle sorte qu'il existe de nombreuses situations où les flux de Positions et de Configurations sont complètement désynchronisés d'une manière encore différente et qu'ils ne respectent pas la règle générale de désynchronisation que nous venons d'énoncer.

II.5 Proposition d'une stratégie de décodage

Dans les précédentes sections de ce chapitre, plusieurs points ont été mis en exergue.

Il y a d'abord des difficultés techniques intrinsèques aux problématiques de décodage automatique d'un geste : difficulté de l'acquisition, variabilité des codeurs, difficultés de traduire en instructions machine certaines tâches que le cerveau humain effectue avec simplicité (comme comprendre la notion de pointage, etc.). C'est la résolution de ces difficultés qui rend ces travaux intéressants dans le cadre de la vision par ordinateur ou de l'intelligence artificielle. Chacune de ces difficultés fera l'objet d'une partie de ce document. A titre d'exemple, le chapitre III (p. 58) traite de l'extraction de l'information utile au sein de chaque image, alors qu'au chapitre V (p. 123) est traité le problème de l'interprétation de ces informations à un niveau supérieur.

Il y a ensuite une liste non-exhaustive mais néanmoins longue de "défauts", ou de libertés que prennent couramment les codeurs par rapport au LPC théorique. On peut diviser ces défauts en deux sous-groupes :

– **Les imprécisions gestuelles** : il s'agit des mouvements indésirables du poignet (flexion et rotation), des Configurations mal formées, des Positions pointées de manière imprécise, des mouvements parasites, etc. Comme dans tout dialogue qui passe par une interface machine, le codeur se doit de chercher à les minimiser, même s'il est évident que leur disparition complète est impossible. Leur traitement correspond donc à autant de difficultés techniques auxquelles nous allons être confrontés, mais que nous ne résoudrons que partiellement. En effet, le jour où un codage bien contrôlé sera parfaitement interprété, le but sera de pouvoir relâcher la sévérité de certaines de ces contraintes.

– **Le langage naturel** : toutes les difficultés provenant du fait que l'on interprète un langage naturel plutôt que mécanique (ou produit comme un code théorique parfait), sont, au niveau où nous plaçons ces travaux, insurmontables : ainsi, l'utilisation de gestes pantomimes, de gestes issus du Cued Speech, d'onomatopées, la présence d'hésitations ou d'erreurs de prononciation, ainsi que tout raccourci s'appuyant sur la suppléance mentale de l'interlocuteur sont à proscrire. Ici, il s'agit donc de restreindre le champ d'investigation de notre travail, quitte à imposer des contraintes à son utilisateur.

Enfin, le dernier point a trait aux spécificités du LPC. Il est intéressant de les étudier afin de mieux déterminer la manière dont le décodage automatique doit être abordé. En effet, de l'originalité du LPC, du type particulier de mouvements qu'il implique, ou de sa structure, peuvent être extraites des informations pertinentes que l'on ne retrouvera pas dans d'autres gestes ; cela constitue un excellent guide pour élaborer une stratégie originale par rapport à l'état de l'art de la reconnaissance gestuelle en général, mais aussi une stratégie plus adaptée

et plus efficace. Ces spécificités sont principalement au nombre de trois : il y a tout d'abord l'aspect dynamique d'un codage intrinsèquement statique au travers de sa coarticulation. Il y a ensuite l'aspect lié à la désynchronisation Position/Configuration, et enfin, l'aspect lié à la désynchronisation Main/Lèvres. Pour faire référence à ces spécificités, nous parlons des trois **spécificités intrinsèques** du LPC, afin d'insister sur le fait que la plupart des gestes que l'on cherche à interpréter automatiquement diffèrent sur ces points.

Ayant admis comme première hypothèse que le geste manuel du LPC est un ensemble de gestes statiques noyé dans un flux dynamique, nous effectuons les deux hypothèses supplémentaires suivantes :

– Dans le cas où (1) le codage est effectué correctement (par rapport aux remarques de ce chapitre) et où (2) la cadence d'acquisition est suffisamment élevée, nous supposons qu'il existe pour chaque geste manuel une image sur laquelle il est intégralement représenté. Une telle image est appelée **Image Cible** (IC). De plus, la succession des IC est supposée suffisante au décodage du code manuel. On les distingue des **Images de Transition** (IT), qui ne sont pas nécessaires.

– Les IC sont reconnaissables à partir d'informations cinétiques de bas niveau, et leur extraction peut être effectuée avant l'analyse et la reconnaissance complète du contenu de chaque image.

Ces hypothèses sont le fondement et la justification de la labellisation précoce. Nous appelons ainsi la méthode originale que nous avons développée pour faire le lien entre la reconnaissance d'une image isolée, et la reconnaissance d'une séquence complète, avec l'aspect dynamique que cela implique. Cette labellisation a pour objectif d'effectuer une classification sommaire entre IC, qui sont des images correspondant à des instants clefs de la trajectoire gestuelle, et IT, qui *a priori*, ne sont pas utiles au décodage. Le terme de "précoce" traduit le fait qu'il n'est pas nécessaire de procéder à l'étude complète de toutes les images pour effectuer l'extraction des IC : cette labellisation peut se faire de manière anticipée par rapport à l'identification du contenu de chaque image. Ces deux aspects sont les raisons pour lesquelles cette méthode est originale par rapport à l'état de l'art.

D'une manière générale, le problème de la reconnaissance de gestes est d'associer une succession d'états aux différentes images d'une séquence vidéo. D'un point de vue théorique, il s'agit simplement de la reconnaissance d'une trajectoire au cours du temps, ce terme pouvant soit être pris au sens classique et géométrique du terme, soit en considérant une trajectoire dans un espace mathématique quelconque, discret ou continu, et de dimension arbitraire. En conséquence, la quantité de travaux faisant référence à cela est pléthorique, et les méthodes qui y sont décrites s'appliquent à un champ très large de problèmes (séquençage de l'ADN [141], robotique [27], reconnaissance de la parole [128], etc. ...). Ces méthodes sont pour la plupart dérivées de modélisation graphique sous l'hypothèse que le problème vérifie la propriété de

Markov, à savoir que tout ce qui peut conditionner l'instant futur dans le passé est résumé par l'instant présent [128], [129], [130], [148], [149], [150], [151] (chaîne de Markov, **Hidden Markov Model** ou HMM, **filtre de Kalman**, filtres particulaires ...). Ces méthodes sont tellement efficaces qu'elles constituent l'intégralité de l'état de l'art, et que leurs utilisations n'ont même plus besoin d'être justifiées. Elles souffrent cependant de quelques inconvénients [152] :

– La complexité des modèles croît plus vite que celle du problème, de sorte que dans bien des cas, celle-ci devient rédhibitoire. Afin de pallier cela, les modèles utilisés sont souvent moins généraux : les séquences d'apprentissage et la machine à état sont généralement simplifiées.

– L'apprentissage est exclusivement fait sur des exemples positifs, ce qui ne facilite pas la discrimination.

– Il est nécessaire de posséder des bases de données de très grande taille pour effectuer des apprentissages ayant un pouvoir de généralisation suffisant.

– Une modélisation probabiliste subjective du problème est souvent nécessaire ; cela va de paire avec soit (1) un modèle gaussien sous-jacent dont la justification est délicate, soit (2) des modèles dont la complexité calculatoire nous rend sceptique pour l'application visée (comme le filtre particulaire ou les méthodes de Monte-Carlo par exemple).

En pratique, ces inconvénients techniques peuvent amener à des situations que nous souhaitons éviter dans le cas de notre application. Par exemple :

– Une succession de gestes manuels qui n'est reconnue que lorsque celle-ci est réalisée par un codeur particulier, dont les particularités de codage et le dynamisme ont été sur-appris par hasard en raison de la distribution du corpus d'apprentissage.

– Les successions rares de gestes dans le corpus d'apprentissage sont considérées comme improbables, et rejetées systématiquement au profit d'hypothèses plus probables. De plus, cela nous amène à considérer le problème à un niveau d'interprétation supérieur à celui où nous nous trouvons.

Néanmoins, il est tout à fait possible d'appliquer une méthode de l'état de l'art à notre problème : traiter toutes les images (sans en réduire le nombre) afin d'extraire les informations importantes de celles-ci, puis extraire de l'ensemble de ces données la structure dynamique qui permet de spécifier le geste. En effet, de nombreuses méthodes ont été proposées pour éviter les défauts mentionnés plus haut, telles que [134], [147], [145]. Nous n'avons donc pas utilisé les méthodes classiques pour les deux raisons suivantes :

Tout d'abord, l'ensemble des remarques de ce chapitre sur les spécificités du LPC sont autant d'informations qu'il est possible d'injecter en tant que connaissances *a priori* dans le système, sans passer par un apprentissage

coûteux. Ainsi nous pensons simplifier la complexité de la solution à apporter. Par la même occasion le problème est abordé de manière originale et alternative par rapport à l'état de l'art, ce qui est toujours intéressant d'un point de vue scientifique.

La seconde est qu'il est très difficile d'avoir des données pertinentes et en quantité suffisante. Même si l'intérêt pour le LPC est grandissant, il n'est pas encore beaucoup diffusé : il n'est apparu qu'en 1979, et par conséquent, seuls les plus jeunes y ont été exposés depuis leur plus jeune âge. En conséquence, il est très difficile de rassembler suffisamment de séquences vidéo pour effectuer des apprentissages reflétant la diversité des codeurs français, de leur forme de main, etc. De plus, afin de garantir que nous travaillons sur un code effectué correctement, seuls les codeurs certifiés peuvent être utilisés, et ceux-ci sont encore plus rares. Plutôt que de mettre en place des apprentissages de mauvaise qualité, nous préférons chercher dans une autre direction. Malheureusement, il en résulte que nous ne disposons pas du matériel expérimental nécessaire pour mettre en place des méthodes plus classiques auxquelles comparer la nôtre.

Dans notre deuxième hypothèse, nous précisons que les IC existent aux conditions que (1) le code LPC soit parfaitement orthodoxe, et que (2) la cadence d'acquisition soit suffisamment élevée. Dans ce genre de cas théoriques seulement, la désynchronisation Position/Configuration n'apparaît pas. Dans les cas réels, celle-ci empêche de déterminer une IC : cette désynchronisation est cependant diminuée par un codage appliqué, et même si un léger décalage apparaît, il est possible à haute fréquence d'acquisition de trouver au moins une image pouvant être prise pour IC.

Afin d'être plus tolérant vis-à-vis du codage (accepter un codage réel) comme du matériel d'acquisition (accepter un matériel ayant une cadence maximum plus faible), nous proposons d'affiner le concept d'IC : plutôt que de ne chercher qu'une seule IC par geste, celle-ci devant résumer à la fois la Configuration et la Position, nous proposons d'utiliser deux images, pas forcément distinctes, mais pouvant l'être si nécessaire : une Image Cible de Configuration (ICC), et une Image Cible de Position (ICP). D'un point de vue pratique, cela permet aussi de faire plus facilement la distinction entre deux gestes consécutifs ne se différenciant que par une seule composante (Position ou Configuration), l'autre restant inchangée.

Dès lors, se pose le problème de la fusion de ces deux informations extraites en parallèle. En effet, à partir du moment où l'on admet l'existence de deux types d'IC différentes mais n'étant pas nécessairement synchronisées ni correctement ordonnées, l'association directe entre geste et IC n'est plus possible.

Afin de retrouver le geste complet, nous devons utiliser après la labellisation précoce, un autre processus permettant la fusion des informations issues des ICC et des ICP.

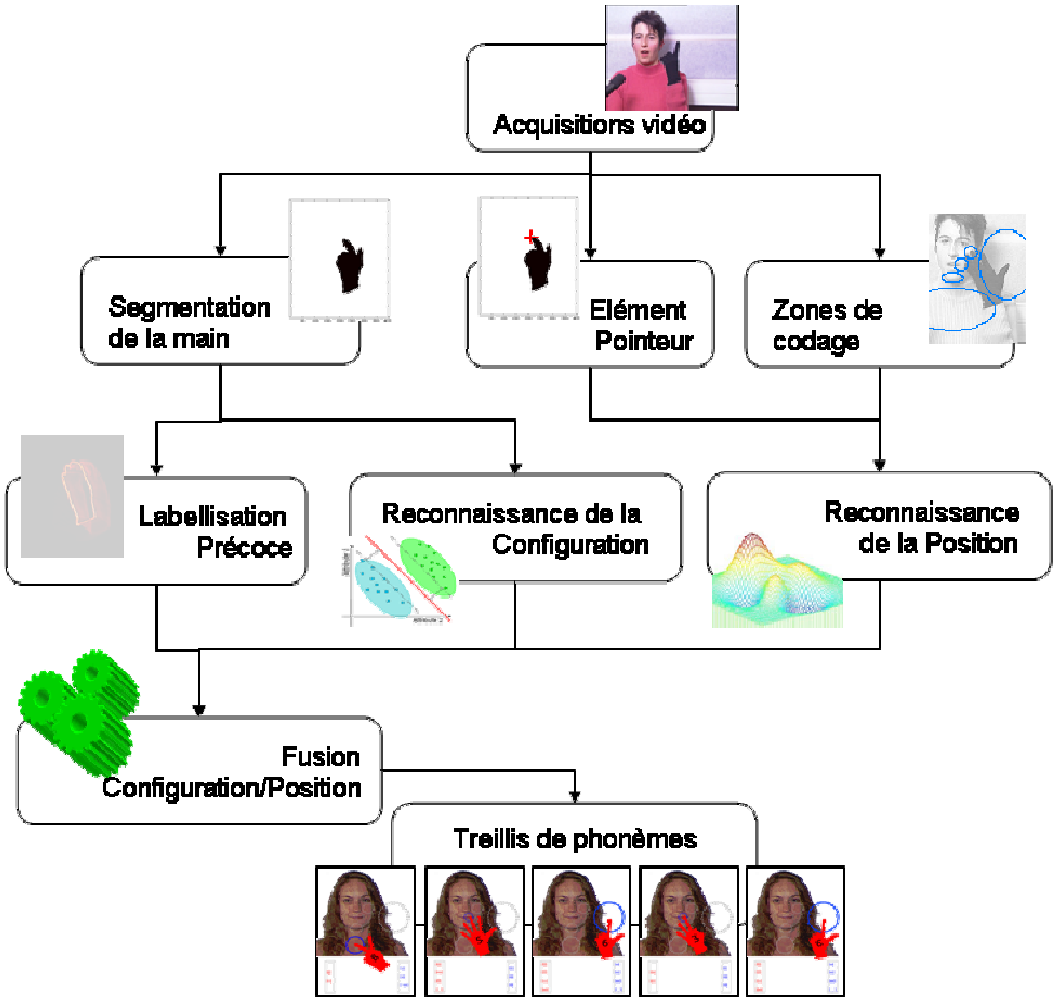


Figure II-8 : stratégie de décodage sous forme d'un schéma bloc.

Au regard de tout ceci, la stratégie de décodage automatique du LPC que nous proposons dans ce document est la suivante :

Etape 1 - Traitement des images : pour chaque image de la séquence vidéo, la forme de la main gantée est segmentée. La position du visage est détectée, et à partir de celle-ci, les zones de pointage sont inférées.

Etape 2 - Labellisation précoce : celle-ci est effectuée à partir d'informations cinétiques de bas niveau uniquement extraites de la forme binaire de la main.

Etape 3 - Reconnaissance : il s'agit d'effectuer des classifications sur les différentes composantes du geste, en s'appuyant sur des éléments issus des IC.

Etape 4 - Interprétation phonémique : un système d'intégration temporelle naïf, mais basé sur la labellisation induite par la Réduction Précoce, permet de récupérer la succession des gestes effectuée par le codeur. Un treillis des

phonèmes possibles (que le mouvement labial doit désambigüiser) est alors fourni.

En pratique, un très grand nombre de traitements de l'étape 1 ne sont pas nécessaires sur toutes les images. Cependant, ils seront essentiels à d'autres parties du projet. Par exemple, pour la lecture labiale, il est indispensable de repérer le visage et les lèvres (donc la bouche) sur toutes les images de la séquence. Ainsi, il n'est pas intéressant d'effectuer certains traitements uniquement sur les images qui nous intéressent, telles que les IC. C'est pour cette raison que nous avons décidé de les appliquer systématiquement à toutes les images des séquences vidéo. Au final, il en résulte que le processus global suit le schéma de la Figure II-8.

Le plan de ce document respecte globalement cette architecture. Ainsi les chapitres suivants se découpent de la façon suivante :

– Le **chapitre III** (p. 58) décrit l'ensemble des algorithmes de segmentation appliqués aux images : segmentation de la main, localisation du visage et des zones de pointage, et extraction de l'élément pointeur. De plus, nous y décrivons les corpus de données sur lesquels ces algorithmes sont élaborés puis testés.

– Le **chapitre IV** (p. 101) est consacré à la labellisation précoce, à savoir la classification des images de la séquence vidéo (en images cibles ou de transitions) à partir d'informations cinématiques de bas niveau. C'est à cette étape que la coarticulation, qui est la première spécificité intrinsèque au LPC, est traitée.

– Le **chapitre V** (p. 123) rassemble tous les algorithmes liés à la reconnaissance et à la classification des différentes composantes du geste (Configuration et Position).

– Le **chapitre VI** (p. 182) est consacré à la fusion des trois algorithmes de classification utilisés : la reconnaissance de la Configuration, la reconnaissance de la Position, et la labellisation précoce. C'est à cette étape que la deuxième spécificité intrinsèque du LPC (synchronisation Configuration/Position) est traitée.

– Le **chapitre VII** (p. 216) est l'occasion de s'intéresser à la dernière spécificité intrinsèque au LPC à savoir la désynchronisation de la main et des lèvres. Comme cet aspect n'est pas du ressort direct de ce travail, il s'agit plutôt d'une ouverture sur la suite de la présente étude.

Il se trouve que le plan des chapitres III à VII correspond dans l'ensemble à l'enchaînement des blocs de traitement de la Figure II-8. En conséquence, la Figure II-9 peut faire office de table des matières "graphique".

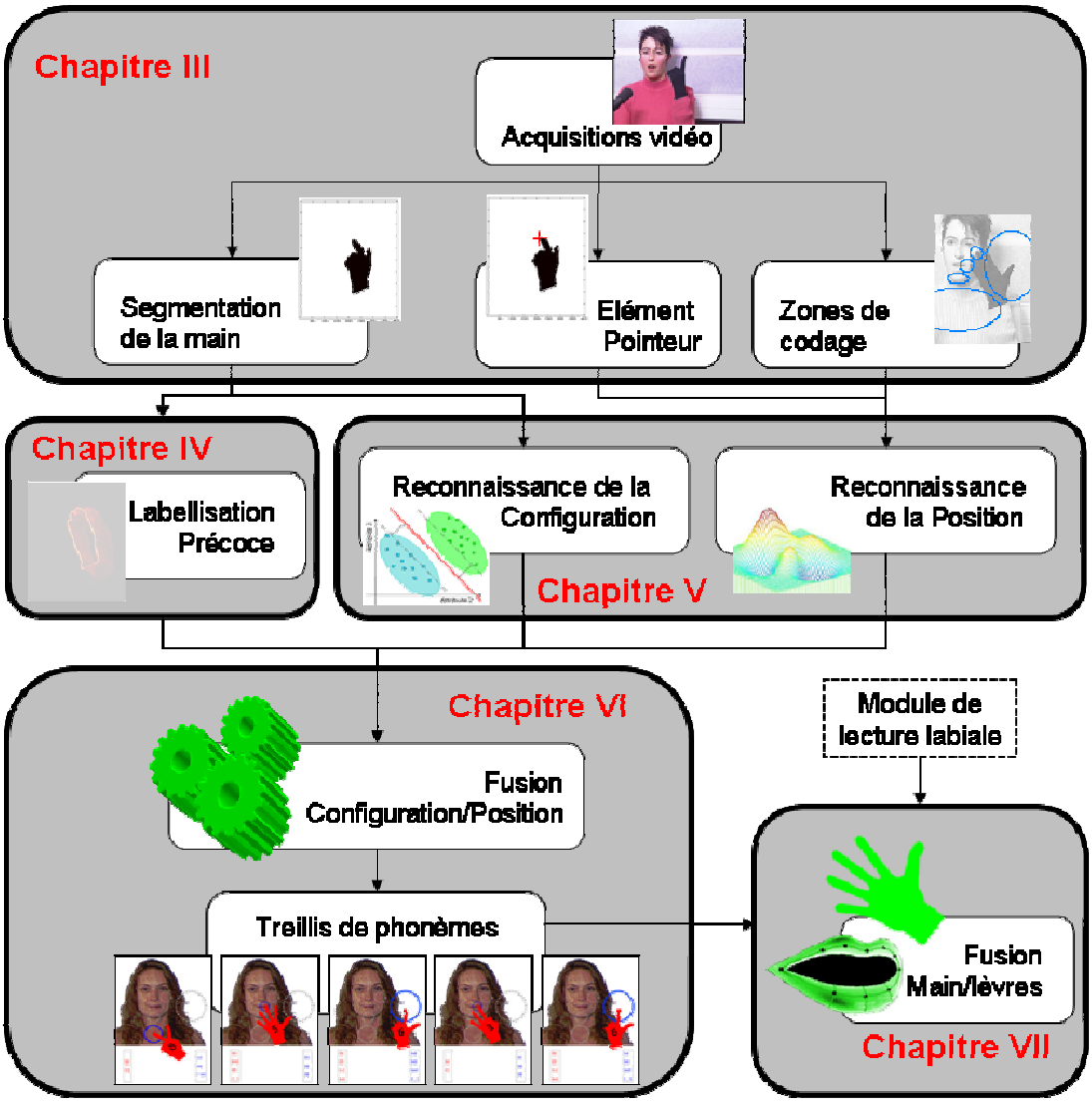
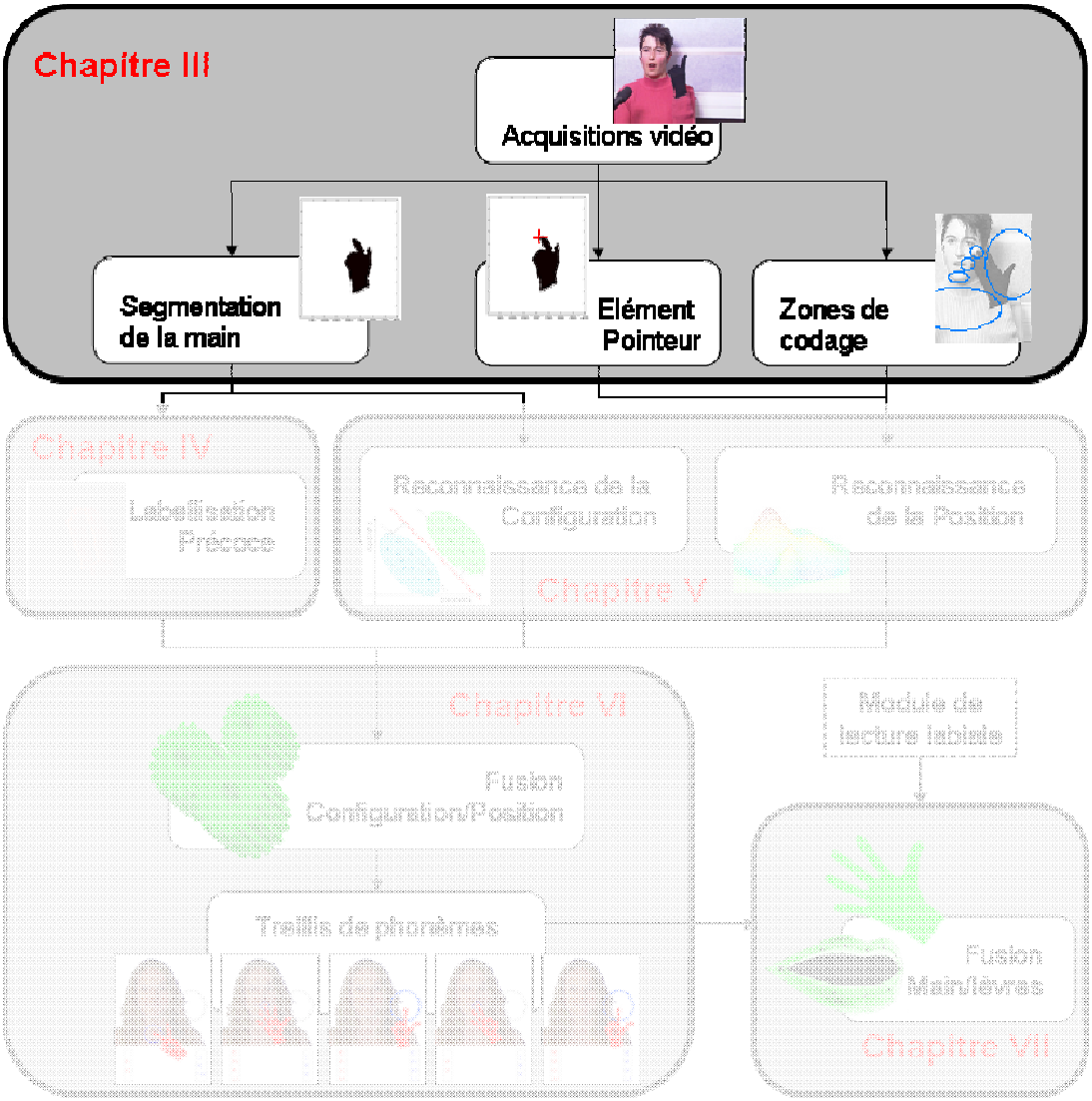


Figure II-9 : correspondance entres les différentes briques algorithmiques et le plan du document.

CHAPITRE III
ANALYSE & SEGMENTATION DES IMAGES



Dans ce chapitre, nous commençons par présenter les différents corpus de données vidéo sur lesquels les algorithmes sont mis en place, testés et évalués.

Ensuite, nous exposons les algorithmes de traitement de chacune des images du flux vidéo : (1) la segmentation de la main dans l'image, (2) la détermination du doigt pointeur sur la main, et (3) la détection du visage et de certains éléments du faciès qui sont désignés par le doigt pointeur afin de spécifier une Position. Une évaluation de chacun des algorithmes proposés est également présentée.

III.1 Acquisition et bases de données

Nous commençons par expliquer les différentes contraintes relatives à l'acquisition des vidéos de codeur LPC sur lesquelles nous allons travailler. Ensuite, nous décrivons les campagnes d'acquisition, et enfin les corpus qui en découlent.

III.1.1 Contraintes de l'acquisition

Il y a principalement trois contraintes d'acquisition à analyser : le nombre de caméras, le compromis résolution/cadrage, et la cadence vidéo.

Nombre de caméras : La première contrainte du projet est de travailler à partir d'acquisitions réalisées avec une seule caméra. Cette contrainte est avant tout motivée par des considérations relevant de l'objectif final de réalisation d'un produit commercialisable. Ainsi, pour des raisons de coût de matériel, de coût de calcul, et de facilité d'utilisation, il est préférable de limiter le nombre de caméra. Cela a plusieurs conséquences sur les choix algorithmiques.

La première est que toute acquisition en stéréovision, permettant de récupérer l'information de profondeur, est à bannir. Il est donc nécessaire que toutes les informations visuelles soient visibles sur un seul plan. Cette acquisition 2D peut avoir plusieurs inconvénients :

– Occultation du visage par la main,

– Occultation de la main par elle-même, (doigts repliés, doigts s'occultant mutuellement, sensibilité accrue aux rotations de la main, etc....)

– Perte des mouvements de la main dans la dimension de la profondeur,

– Difficulté de différencier contact et occultation entre la main et le visage.

Cependant, il a été vérifié par une étude préliminaire (cf. [12]) que la perte d'information 3D constituait une approximation raisonnable par rapport à la stéréovision.

La seconde est qu'une seule d'acquisition doit être suffisante pour récupérer toutes les informations visuelles avec une résolution et une cadence suffisante. Ainsi, il n'est pas possible d'utiliser une caméra en plan large pour le

mouvement de la main, et une autre en plan rapproché pour le mouvement labial. Cela signifie que les différentes briques algorithmiques du projet doivent fonctionner sur des images communes, et ce, malgré des contraintes qui sont différentes pour chacun des algorithmes. Ainsi, l'expertise commune des différents acteurs de TELMA laisse à penser que les acquisitions sont beaucoup plus contraintes pour permettre une lecture labiale automatique efficace que pour permettre une reconnaissance gestuelle efficace. C'est donc la lecture labiale qui va imposer des contraintes au reste des algorithmes :

– En termes de **compromis résolution/cadrage** : les méthodes d'extraction du contour des lèvres utilisées dans le projet TELMA nécessitent que la plus grande dimension de la bouche soit de l'ordre de 80 à 120 pixels. Une bouche plus grande est bien sûr possible, cependant, avec un tel zoom, il n'est pas évident de pouvoir cadrer tout le visage ainsi que l'espace dans lequel la main se déplace. La Figure III-1 présente un exemple de cadrage pour lequel l'ensemble de la main et du visage est filmé, tout en proposant une résolution suffisante pour les lèvres. L''image est de taille 720×576 pixels, et par conséquent, les dimensions des boites englobant le visage ou la main sont approximativement de 300×200 pixels.

– En termes de **cadence d'acquisition** : il apparaît dans [5], [6] que la reconnaissance labiale nécessite une cadence de 50 images/seconde. En revanche, le mouvement manuel est plus lent, et peut se contenter d'une cadence plus faible. Néanmoins, sur des vidéos acquises avec du matériel non professionnel, il apparaît qu'une cadence standard d'acquisition de 30 images/seconde n'est pas suffisante : dans le cas d'un codage rapide, il peut arriver que la pleine réalisation d'un geste n'apparaisse sur aucune image.

Il est donc inutile de chercher à faire fonctionner des algorithmes sur des vidéos qui se trouvent hors du champ des contraintes qui sont valables pour tous les éléments du projet, même si scientifiquement, il peut être intéressant de repousser ces limites pour seulement un des traitements. Ainsi, nous allons devoir chercher, au fur et à mesure des campagnes d'acquisition à uniformiser ce type d'exigences et à faire converger les contraintes d'acquisition. Ce n'est qu'après avoir fait cela, que nous auront la garantie qu'il est possible de mener le projet à son terme avec des acquisitions mon-caméra. Cet effort de convergence est donc indispensable.

Ainsi, les premières séquences d'images sur lesquelles nous avons travaillé résultent de la numérisation d'images analogiques, obtenues dans des conditions studios afin de garantir une qualité maximale. Par la suite, nous avons cherché à dégrader petit à petit la qualité du matériel, afin d'essayer de se rapprocher de la qualité d'une caméra embarquée, (matériel plus conforme au contexte du projet TELMA), sans pour autant aller au-delà des contraintes imposées par la lecture labiale.

Enfin, d'un point de vue beaucoup plus pratique, nous devons pouvoir considérer des codeurs gauchers comme droitiers. Afin que cela ne perturbe pas

le protocole d'acquisition, le plus simple est de centrer le siège du codeur dans l'image, de manière à ce qu'il puisse utiliser n'importe quelle main, puis, de retourner l'image par effet miroir au moment de l'acquisition si le codeur est droitier, afin de ne considérer que des codeurs gauchers.

Figure III-1 : exemple type d'un compromis cadrage/résolution.

III.1.2 Les différentes campagnes d'acquisition

Le but d'une campagne d'acquisition est d'avoir à disposition un nombre suffisant de données pour mettre en place, tester et évaluer les différentes composantes des algorithmes à développer. Comme toujours, il est difficile d'anticiper quelles sont les difficultés auxquelles le projet va être confronté. Par voie de conséquence, il est très difficile de déterminer le contenu intrinsèque des données à acquérir. C'est pourquoi, le nombre de données et leur répétition, le type de phrase à faire coder pendant l'enregistrement et la variabilité que l'on souhaite obtenir, sont autant de paramètres qui dépendent des types d'algorithmes choisis, et ces décisions ne peuvent être prises avant les premiers tests… Afin de pouvoir commencer à travailler, il est nécessaire d'accepter de faire une campagne d'acquisition préliminaire dans le but de commencer à défricher le champ d'investigations et de permettre de mieux spécifier les prochaines campagnes.

Pour des raisons de simplicité, il a été décidé de faire une campagne préliminaire commune à toutes les équipes de recherche intervenant sur le projet et se trouvant à ce stade initial. Nous nous en sommes remis à l'équipe du GIPSA-Lab/DPC afin de mettre en place le contenu de cette campagne. En effet,

forts de leur précédentes implications dans des projets sur le LPC et de leurs connaissances des acquisitions linguistiques, ils étaient les mieux placés pour spécifier cette expérimentation. Par la suite, nous avons effectué une seconde campagne d'acquisition, basée sur la même infrastructure logistique et matérielle. Celle-ci avait le double objectif de compenser les défauts de la première, tout en proposant des expérimentations plus ciblées aux besoins de chacune des équipes.

L'élaboration de nombreux corpus nécessite autant de campagnes d'acquisition. La mise en place d'une telle campagne est une charge de travail considérable et prend plusieurs semaines de travail : il faut définir l'intégralité du scénario de la campagne, les conditions d'acquisition et les variables que l'on va étudier ou faire varier. Il faut prendre contact avec des codeurs professionnels et des malentendants ; il faut que ceux-ci soient en nombre suffisant à chaque instant de la campagne en fonction de leurs emplois du temps professionnels ou scolaires respectifs, et des autorisations parentales pour les mineurs. La campagne d'acquisition elle-même dure environ une semaine, pendant laquelle le studio d'enregistrement doit être disponible et les techniciens et ingénieurs doivent être présents. Finalement, le dépouillement des données prend lui aussi plusieurs semaines. En conclusion, les coûts financiers et en temps sont tels qu'il n'est pas toujours possible de travailler avec autant de données que nous le souhaiterions d'un point de vue scientifique.

III.1.2.1 La campagne préliminaire

La campagne d'acquisition préliminaire a eu lieu en février 2005. Elle consiste au codage par une unique codeuse professionnelle d'une liste de 238 phrases.

Figure III-2 : à gauche le protocole d'acquisition à des fins linguistiques, et à droite, à des fins de traitement d'images.

L'enregistrement complet de cette séquence de codage est répété plusieurs fois dans des conditions expérimentales variables afin de satisfaire les exigences des différentes équipes de recherche. En effet, certaines exigences sont incompatibles. Par exemple, les équipes travaillant sur des problématiques de

linguistique, doivent faire appel à des marqueurs et du maquillage bleu saturé, qui rendent les vidéos non-exploitables à nos fins de traitement d'images. Il résulte de ces nombreuses heures d'acquisition un grand nombre de bandes magnétiques qu'il n'est pas possible de numériser automatiquement en raison de la taille mémoire prohibitive à laquelle cela correspond. En conséquence de quoi, toutes les données doivent être triées à la main afin d'éliminer les silences, les erreurs et les phases d'ajustement. Cela permet de supprimer plus des deux-tiers des enregistrements.

Tableau III-1 : exemples de phrases à coder durant la campagne préliminaire.

1	MA CHEMISE EST ROUSSIE.
2	VOILA DES BOUGIES.
3	DONNE UN PETIT COUP.
4	TIENS-TOI ASSIS.
5	IL A DU GOUT.
6	ELLE M'ETRIPA.
7	UNE REPONSE AMBIGUE.
	...
54	J'AI UN SCORPION SEC DANS MON TALON AIGUILLE .
55	NOS DALMATIENS CAMPAIENT AU CAMPING A LA MONTAGNE .
56	LES GANGS INFLIGENT DES BINGS ET DES BANGS PERILLEUX SUR UNE ILE
57	VEND-ON UN CAKE INTACT A HONG-KONG ?
58	NOAM CHOMSKY BALAIE ENCORE LE CLUB CE SOIR .
59	L'AVOUE A BESOIN D'UN JOINT SOUS HUITAINE .
60	LA SUEUR SUINTE DU THON HUILEUX .
61	LE BEAU OUISTITI SUIT LE RICHE HUISSIER A WATERLOO .
62	TOUT WINIPEG ATTEND WENDY SUR LE PARKING OUEST .
63	HUIT JESUITES TRES HUILEUX SE FONT UN BRUSHING YOUGOSLAVE .
64	BUD ET BUCK FONT UN BON WHIST A MAUBEUGE .
65	YOURI FOUETTE L'AIL IONIQUE DE KOHOUTEK .
66	BEUNG J'AI HEURTE LE PUITS DANS LA LUEUR .
	...
93	CE FOU ORDINAIRE FICHE LE TURBAN INDIEN DANS LE BAIN OPTIONNEL .
94	UNE AGRAPHE GEANTE A PU HEURTER SON BEAU HORS-BORD .
95	DE MAUVAISES GENS PRIVENT VICTOR DE SA COIFFE BRETONNE .
96	LA GRIVE PERCHEE SUR L'IF NOIR COUVE TOUJOURS CE CANIF CHINOIS .
97	POSE CALMEMENT TA DAGUE POINTUE SUR CETTE ETOFFE CARREE .
98	LE VASE ZEN A PERDU AUSSI UN ANNEAU EN ROCHE GRISE .
	...
194	CHAQUE ZERO EST UN LOOPING TORDU .
195	LA MEILLEURE OMELETTE DU LARZAC PEUT RIVALISER AVEC LE YACHTING NORMAND .
196	UN NAIN HEURTA UNE BOGUE CHARNUE UN ONZE JANVIER .
197	UNE TOMBE MING NE PASSE JAMAIS POUR UN KARTING BELGE .
198	UN HOMME JEUNE NE TOMBE PAS PENDANT CETTE JAVA .
199	DES RIDES CHARMANTES AERENT CETTE ROBE CHOISIE DANS LES PAGES JAUNES .
	...
210	MOREAU ETALE IMMANQUABLEMENT UN DEFICIT COMMUN A LA QUEUE DE L'UE .
211	ALADIN ELEVE CHACUN EN SYMBIOSE AVEC LE VIEUX OUZBEK .
212	UN COUP HEUREUX ET IMPETUEUX MODIFIE UN VULGAIRE PAIN ONCTUEUX EN GNOME .
213	CHACUN IGNORE SON C.E. UN~PEU UN MOMENT .
214	AVEC UN APLOMB IMPARABLE NOUS AVONS CHACUN UN C.E. ENERGIQUE .
215	CETTE ENERGIE INSENSEE GREVE UN QUINZIEME DE UGINES .
216	SUR LE ZING CHACUN INTERPRETE L'ATLAS HUMBLEMENT POSE SUR L'ANCIEN JABOT .
	...
237	QUANTUM SUEDOIS OU RITUEL WOLOF .
238	LA SECOUEUSE FAIT DES PERCINGS LINGUAUX .

La signification des 238 phrases à coder est parfois difficile à saisir, voire même inexistante, car cet ensemble de phrases a la particularité de contenir l'ensemble de toutes les transitions possibles entre tous les phonèmes du français (en conséquence, les noms des corpus de données issus de cette campagne commence par ETTRAN pour "ETude des TRANsitions"). Le manque de sens de certaines phrases les rend difficiles à prononcer et à coder (cf. Tableau III-1). Par conséquent, les enregistrements ne sont pas très fluides : il y a des hésitations

et des erreurs que l'on retrouve dans un phrasé qui n'a pas été préparé (parole naturelle). Les phrases sont de longueur variable (de 5 à 25 syllabes) et certaines ont des structures relativement élaborées. Ainsi, durant l'acquisition des séquences correspondant sans maquillage que nous utilisons, certaines phrases sont répétées suite à des erreurs de diction.

La codeuse est française d'origine francophone, bien-entendante, diplômée d'interprétation LPC, et elle travaille régulièrement comme telle, soit en conférence, soit à l'école, dans le cadre de l'aide à l'intégration de malentendants ou sourds oralisés. L'acquisition est réalisée dans une chambre sourde, dans des conditions d'éclairage de type "studio d'enregistrement professionnel". La codeuse est assise en face de la caméra, et porte un gant de soie noir et fin. Ce gant n'est pas un gant fourni pour l'expérience, mais son propre gant, qu'elle a porté pour se protéger du froid durant le trajet vers le lieu d'expérimentation. En conséquence, il s'agit vraiment d'un gant de type indéterminé (mais de couleur unie) et dont le choix relève bien du codeur. Lors de cette campagne d'acquisition, c'est la première fois qu'elle code avec un gant, mais après un court échauffement, elle n'est plus gênée par sa présence. Le choix du gant est déterminant à plusieurs niveaux tels que (1) le confort d'utilisation, (2) la difficulté de segmentation de la couleur du gant par rapport aux autres couleurs présentes dans l'image, (3) la difficulté de reconnaître un gant inadapté à la main du codeur. Il se trouve que son gant est d'une couleur qui est très proche de celle de ses cheveux. Afin d'avoir quelques repères par rapport à l'intérêt du choix du gant selon ces divers axes, nous avons fait quelques acquisitions avec un autre gant bleu sombre, plus épais, et trop grand pour la codeuse d'au moins deux tailles. L'enregistrement est au format analogique entrelacé BetaCam, à une vitesse de 25 images par seconde, réalisé par une caméra professionnelle de haute qualité (le détail du dispositif d'enregistrement et les références matérielles sont décrite en appendice C.6 p. 303). Les trames A et les trames B sont séparées, et chacune d'elle est utilisée pour reconstruire une image complète par la méthode d'interpolation moyenne. Ainsi, après numérisation, on obtient une séquence à 50 images par seconde de taille 720×576 (format PAL), d'une résolution effective deux fois plus faible que le nombre de pixels réels (l'interpolation n'apporte pas d'information), mais cela n'est pas problématique, tant la qualité des images originales est élevée.

III.1.2.2 La campagne d'expérimentation du Magicien d'Oz

Ensuite, il y a une deuxième campagne d'acquisition de grande envergure, effectuée en février 2007 dans des conditions matérielles similaires à la campagne préliminaire, et couplée à une expérience de **Magicien D'Oz** [A2].

Le protocole expérimental du Magicien d'Oz consiste à simuler l'existence d'un système automatique avec un opérateur caché des sujets de l'expérience. Ainsi, les personnes pensent utiliser un système TELMA qui n'existe pas encore, et cela permet de mettre en place des tests d'usage, de besoin ou d'ergonomie avant même la création d'un prototype. Cette expérience permet de mieux cibler les axes de réalisation, de développement ou d'amélioration. Cependant, un tel

protocole de Magicien d'Oz est particulièrement contraignant. Il est nécessaire d'avoir à disposition un grand nombre de salles équipées des matériels d'acquisition et de télécommunication permettant de simuler une communication à distance par des terminaux TELMA. Il est aussi nécessaire de pouvoir garantir que les différents participants aux tests ne vont pas se croiser lors de leur arrivée, de leur attente, de leur participation et de leur départ. Enfin, les rôles et les scénarii de chacun doivent être très bien préparés afin de garantir le maintien de "l'illusion", quelque soit les imprévus qui se produisent.

Cette expérience a été réalisée autour de la simulation d'un terminal TELMA auprès d'un groupe de malentendants en situation d'usage du terminal. Les aspects liés à l'expérience elle-même [A2] n'ont pas d'intérêt direct pour les algorithmes présentés ici, cependant, comme l'intégralité de l'expérience a été filmée dans des conditions similaires à la première campagne, elle a permis l'acquisition de base de données supplémentaires.

De ce fait, cette seconde campagne est aussi réalisée avec des caméras analogiques dont les images sont détramées de manière similaire à la campagne d'ETTRAN, afin d'obtenir une cadence de 50 images/secondes.

Les principales différences entre les vidéos issues de cette campagne et celles provenant de la campagne préliminaire sont les suivantes :

– Huit adolescents ou jeunes adultes ont participé à l'expérience. Ceux-ci connaissent le LPC, mais ne sont pas des codeurs professionnels. Ils utilisent forcément le LPC régulièrement en perception, mais ne le codent pas toujours très bien.

– Une codeuse professionnelle, membre de l'équipe de réalisation de l'expérience, a aussi été filmée tout au long des différents scénarii. Il s'agit d'une codeuse différente de celle de la première campagne d'acquisition, mais recrutée selon les mêmes critères.

– Les conditions d'acquisition sont globalement les mêmes, mais celles-ci sont réglées avec moins de précision. Cela concerne le zoom, le cadrage, le fond du décor dans les salles d'acquisition, les vêtements portés, etc.

– Certaines séquences ont été artificiellement bruitées pour les besoins de l'expérience du magicien d'Oz.

– Un grand nombre de gants de taille, de couleur, de texture et de forme différentes sont utilisés par les sujets de l'expérience. La codeuse professionnelle utilise toujours le même.

Les enregistrements correspondent soit à des dialogues téléphoniques figés, soit à des dialogues téléphoniques libres selon un fil conducteur préconçu, tel qu'un appel à une agence de voyage pour la réservation d'un voyage organisé, ou tel qu'un appel au Service d'Aide Médicale Urgente (SAMU) suite à un accident.

III.1.3 Les différents corpus de données

Corpus ETTRAN N et ETTRAN BF (cf. Tableau III-2) : de la campagne d'acquisition préliminaire, nous tirons (suite aux répétitions de la codeuse lors de l'enregistrement correspondant à notre protocole expérimental) 267 occurrences de cette liste de 238 phrases. Sur le premier tiers, un gant bleu sombre et inadapté est utilisé afin d'obtenir rapidement des informations sur l'influence du gant et de la forme de la main. Cela permet de constituer le corpus ETTRAN BF (pour ETTRAN gant Bleu Foncé). Sur les deux autres tiers, le gant de ville de la codeuse est utilisé. Cela permet de constituer le corpus ETTRAN N (pour ETTRAN gant Noir).

Tableau III-2 : résumé des caractéristiques des corpus ETTRAN N et ETTRAN BF. Les images de la colonne de gauche sont tronquées autour de la zone de codage afin de fournir une meilleure visibilité.

Corpus	Spécificité	Utilisation
ETTRAN N 	Gant noir (N) et fin, de couleur proche des cheveux, codeuse professionnelle, 2/3 de l'enregistrement ETTRAN (environ 100 000 images), Séparation Apprentissage/Test fixée pour garantir l'indépendance inter-séquence. Résolution : 720×576 pixels	Test segmentation Test de description et de reconnaissance mono-codeur et multi-codeur Test de la labellisation précoce
ETTRAN BF 	Gant bleu foncé (BF) inadapté à la codeuse, codeuse professionnelle, 1/3 de l'enregistrement ETTRAN (environ 50 000 images). Résolution : 720×576 pixels	Evaluation de l'influence du gant à codeur identique

L'ensemble de ces deux corpus correspond à un total d'environ 150 000 images. Suivant le type d'algorithme que nous testons dessus, nous ne pouvons pas utiliser l'intégralité des images. En effet, le paramètre pertinent est parfois le nombre d'images total, parfois le nombre d'images cibles, parfois le nombre de phonèmes, parfois le nombre de séquences vidéo, et enfin, parfois le nombre d'image n'ayant aucune corrélation temporelle entre elles (deux images consécutives sont parfois presque identiques).

Puisque dans l'ensemble de nos algorithmes, aucun apprentissage n'est réalisé au niveau sémantique, il n'est pas nécessaire de réserver une partie de ce corpus à l'apprentissage et une autre au test. Le principal avantage de cela est que le pouvoir de généralisation de ce corpus est complet : d'autres phrases du même type, acquises dans la même campagne, seraient traitées exactement de la même manière que celles utilisées pour calibrer nos algorithmes. Néanmoins, nos algorithmes nécessitent d'autres apprentissages réalisés à des niveaux inférieurs au niveau sémantique. En conséquence de quoi, nous avons pris soin de séparer le corpus en deux. Cela n'a aucun intérêt pour le niveau sémantique proprement dit, mais comme il est difficile de prévoir l'influence des apprentissages de plus bas niveau sur la reconnaissance au niveau sémantique, il s'agit plutôt d'une précaution expérimentale. Cette précaution n'est cependant pas la seule à prendre : le contenu des phrases n'a que peu d'importance dans nos tests par rapport aux autres facteurs de variabilité (codeur, gant, condition d'éclairage, qualité d'acquisition, etc.). C'est donc surtout par rapport à ces autres facteurs qu'il est important d'être prudent et de marquer la différence sur le protocole expérimental.

Corpus MAGOZ R, MAGOZ B et MAGOZ J (cf. Tableau III-3) : ces corpus sont issus des séquences qui ont été filmées durant l'expérimentation du Magicien d'Oz. Ils représentent une quantité de données équivalente aux Corpus ETTRAN N et ETTRAN BF, mais une grande partie n'est pas utilisable parce qu'elle concerne des codeurs (1) n'ayant pas été suffisamment filmé pour constituer un corpus de taille équivalente aux autres; (2) n'ayant pas un codage suffisamment académique. Ainsi, nous n'avons conservé que les séquences pour lesquelles il était possible d'avoir une taille suffisamment importante de données de bonne qualité. A titre d'exemple, quand le codeur ne fait que des phrases très courtes, ou quand le codeur ne reste pas suffisamment face à la caméra pour que l'on puisse distinguer ses lèvres, les données ne sont pas conservées. Au final, seule la moitié des images acquises est utilisable.

MAGOZ R représente une codeuse professionnelle vêtue d'un gant rose. MAGOZ B représente une codeuse malentendante ayant un code très académique, vêtue d'un gant bleu roi. Enfin, MAGOZ J représente une codeuse malentendante ayant un code difficile mais néanmoins acceptable par rapport à nos contraintes d'académisme : le code est lent et sans précipitation, mais les mouvements ont très peu d'amplitude. Ces corpus permettent de mettre en place des tests dans des cas multi-codeurs, et multi-gants.

Tableau III-3 : résumé des caractéristiques des corpus MAGOZ, R, B et J. Les images de la colonne de gauche sont tronquées autour de la zone de codage afin de fournir une meilleure visibilité.

Corpus	spécificité	utilisation
MAGOZ R 	Gant épais et de couleur proche de la peau (R), mais beaucoup plus saturée. Codeuse professionnelle différente d'ETTRAN. Résolution : 720×576 pixels (environ 40000 images)	Test segmentation Test de description et de reconnaissance multi-codeur Test de la labellisation précoce
MAGOZ B 	Codeuse malentendante. Gant bleu roi (B) particulièrement facile à segmenter. Résolution : 720×576 pixels (environ 10000 images)	Test segmentation Test de description et de reconnaissance multi-codeur Test de la labellisation précoce
MAGOZ J 	Codeuse à la dynamique lente. Codeuse malentendante. Gant jaune (J) clair et vif. Résolution : 720×576 pixels (environ 10000 images)	Test segmentation Test de description et de reconnaissance multi-codeur

Dans le cas des corpus de type ETTRAN, la séparation entre apprentissage et test est fixée pour garantir qu'il n'y a pas une forte corrélation entre certaines images issues de la même séquence se trouvant dans l'apprentissage et dans le

test (en effet, cela a pour conséquence une augmentation artificielle des résultats). Dans les corpus de type MAGOZ, le choix d'images non corrélées permet d'éviter cette contrainte, et il n'y a donc pas de séparation fixe entre apprentissage et test : la séparation est réalisée aléatoirement et automatiquement pour chaque test.

Corpus complémentaire (cf. Tableau III-4) : afin de compléter nos bases de données, nous avons effectué des acquisitions d'envergure restreinte, et nous avons récupéré des bases de données existantes. Ces corpus complémentaires permettent des tests plus spécifiques que les 5 corpus ETTRAN et MAGOZ :

Tableau III-4 : résumé des caractéristiques des corpus complémentaires.

Corpus	spécificité	utilisation
CORPDIV 	Divers petits corpus ou acquisitions de diverse qualité. Résolution : très variable	Apprentissages et tests des algorithmes de segmentation
BioId 	Base de données publique, décrite dans [171]. 1521 images monochromes (résolution 384×286 pixels)	Evaluation de la définition des zones de pointage
UCI-MLR	Ensemble de bases de données publiques, décrites dans [62]	Evaluation des méthodes de classification en général

– Nous avons filmé différentes personnes avec différents gants, différentes formes de main, dans des conditions variées et avec des caméras de qualité variable (caméra digitale, WebCams, etc.) afin de pouvoir étalonner les résultats des méthodes de segmentation vidéo que nous avons mises en place. Pour un tel travail de segmentation, il n'est pas nécessaire que la personne à qui l'on fait revêtir le gant connaisse le LPC. L'aspect dynamique du mouvement a ici une faible importance. Cependant, de tels corpus ont une réelle importance et une réelle utilité, dans la mesure où il est très difficile d'obtenir un nombre important d'acquisition avec une grande variabilité. Nous les rassemblons sous le nom de CORPDIV (**Corp**us **Div**ers).

– Nous avons aussi utilisé les bases de données publiques **BioId** [171], (afin de pouvoir étalonner les résultats de certains algorithmes ayant trait à l'analyse du visage), et **UCI machine learning database repository** [62], (afin de tester les méthodes de classification automatique). BioId est une base de 1521 images monochromes au format 384×286 représentant des visages à diverse échelles, en diverses positions et sous divers éclairages. UCI machine learning database repository est une collection de plusieurs bases de données compilées sous forme numérique sur lesquelles il est possible de tester les différents moyens de classification et d'établir des benchmarks.

Finalement, l'ensemble des corpus ETTRAN, des corpus MAGOZ et des corpus complémentaires représente à peu près 700 Giga-octets de données.

III.2 Segmentation de la main

Le premier traitement que l'on effectue suite à l'acquisition des images, est l'extraction du contour (ou la forme) de la main gantée. Sur chaque image du flux vidéo, il faut séparer l'image en deux parties : ce qui appartient à la main (l'ensemble des pixels du gant), et le reste (cf. Figure III-3). C'est ce que l'on appelle la **segmentation**.

(a) image originale (b) main segmentée

Figure III-3 : segmentation de la main.

III.2.1 Etat de l'art en segmentation de main

La segmentation d'une main dans une image ou un flux vidéo est un problème récurent et encore ouvert en vision par ordinateur [24]. De nombreuses publications sont consacrées à passer en revue les différentes méthodes qui font état dans l'art. Parmi celles-ci, mentionnons les travaux de Martin [27] et de Pavlovic [21] orientés vers la reconnaissance de gestes en général. Ceux de Ong [137], et de Derpanis [22] sont plutôt axés sur la reconnaissance de langages gestuels. Dans la majorité des cas, une recherche des pixels ayant la couleur de la peau permet de simplifier le problème [23], [27], [24]. Malheureusement, cela ne suffit pas lorsque la main se trouve en contact avec une autre partie du corps

dont la peau est visible. En effet, segmenter un objet de couleur "peau" sur un fond lui aussi de couleur "peau" est encore un sujet ouvert.

Jusqu'à récemment, il était usuel de diviser les méthodes de segmentation en deux grandes familles : la **segmentation orientée contour**, et la **segmentation orientée région**. La première consiste à repérer les frontières entre les objets, alors que dans la seconde, on recherche plutôt les similitudes entre pixels voisins pour les agglutiner en objets. Cette dichotomie n'a plus beaucoup de raison d'être puisque la plupart des méthodes efficaces à l'heure actuelle intègrent ces deux types d'approches. De plus, la plupart des problèmes de segmentation de la main ne sont plus étudiés en tant que tel, comme analyse de bas niveau (au niveau pixel), mais en corrélation avec leur interprétation, dans le cadre de la vision par ordinateur et de l'interprétation de gestes.

Ainsi, dans [43], la segmentation en région par la recherche de la couleur de la peau que l'on trouve aussi dans [45], [46], [37] est renforcée par une approche contour originale : plutôt que d'utiliser le gradient de Canny [47] sur la composante de luminance de l'image, l'utilisation du gradient de Di Zenzo (travaillant sur les trois composantes de l'image à la fois) est proposée [48]. Cela permet de calculer directement un gradient sur une multi-image en généralisant l'expression convolutive d'un filtre de Canny classique. La matrice représentant le noyau de convolution devient alors un tenseur. Enfin un algorithme de filtrage particulaire (appelé **Condensation**, pour **Cond**itional **Dens**ity Propag**ation** [43]) est utilisé de même que dans [53], où une analogie curieuse entre le contour de la main et le signal vocal est proposée.

Notons qu'un tel problème de segmentation peut toujours être vu comme un problème de classification en deux classes : la main et le reste. Ainsi, un très grand nombre de méthodes originales sont régulièrement publiées en marge de la direction globale de l'état de l'art, mais s'inspirant des méthodes de classification. A titre d'exemple, mentionnons [56], où l'image est modélisée par un hypergraphe (cf. appendice A.2.2 p. 264) dans lequel il s'agit de trouver des hyperchaînes de longueur maximale. Toujours dans le domaine de la recherche opérationnelle, mentionnons [63], qui propose de modéliser les problèmes de classification par la suppression d'une arête dans un arbre couvrant de poids minimum associé à l'image. Dans le domaine des méthodes inspirées du vivant, citons [154] et [155] dont l'approche est l'imitation des traitements que l'on retrouve dans la rétine et le cortex visuel [157]. Actuellement, de nombreux travaux proposent de récupérer une information plus pertinente que celle fournie au niveau pixel, par l'étude de la texture [54], [55]. Enfin, notons l'utilisation de plus en plus courante du Modèle de Croyance Transférable à des fins de segmentation (qu'il s'agisse de la main ou d'autres types d'images comme en vidéosurveillance [124], ou pour la recherche de tumeur cérébrale [119], [120], ou encore du traitement de la couleur en générale [121]).

Cependant, d'une manière générale, les méthodes de référence utilisent actuellement des contours actifs ou des modèles déformables. Les variations de protocole sont très nombreuses sur ce thème, mais le principe reste toujours le

même : la déformation itérative d'un contour jusqu'à ce qu'il corresponde à la frontière complexe de la main. La déformation de ce contour peut-être guidée par un modèle appris, préconçu, ou être libre : [36], [41], [42].

La segmentation précise de la main nue reste cependant un problème non encore résolu, en particulier dans le cas qui nous préoccupe. La difficulté vient de :

– La complexité de la forme de la main, et de la précision qu'elle requiert, si l'on ne veut pas "perdre de doigt".

– La grande variabilité de la forme, de la couleur et de la texture de la main.

– La grande variabilité résultant de conditions d'éclairage non contrôlées.

– Du fait que la main peut venir au contact du visage, dont la couleur est identique.

– Les ressources en calcul que l'on souhaite allouer à ce problème sont restreintes, dans le cadre de la réalisation ultérieure d'un prototype temps-réel, ayant une cadence de traitement suffisamment élevée.

III.2.2 Principe de la méthode proposée

Tout ceci nous a amené à momentanément simplifier le problème de segmentation en autorisant le port d'un gant de couleur uniforme non spécifiée. Bien que la gêne soit équivalente pour le codeur, nous refusons d'utiliser un gant ayant des doigts de couleur différente ou des marqueurs [44]. En effet, nous souhaitons pouvoir un jour supprimer le gant, sans pour autant remettre en cause les algorithmes autres que celui de la segmentation. Il n'est donc pas envisageable d'utiliser un tel gant et de développer des algorithmes de reconnaissance de la Configuration ou du doigt pointeur basés sur l'identification des spécificités du gant. L'utilisation d'un gant ne doit donc être prise que comme un artifice simplifiant l'étape de segmentation seulement jusqu'à ce qu'une méthode robuste de segmentation d'une main nue, même en cas de contact avec le visage, ait été développée.

Malgré tout, le port du gant ne rend pas la segmentation triviale pour autant : bien que ce problème ne soit plus "ouvert", et que de nombreuses solutions présentées dans la littérature puissent donner des résultats satisfaisants pour un tel problème, la segmentation d'un objet de couleur uniforme dans une image possède quelques écueils. En effet, il faut tenir compte des distorsions différentes des couleurs en fonction des capteurs variés des modèles de caméra, des effets d'ombre et de saturation de lumière. La discrimination de la couleur du gant dépend aussi fortement de la couleur des autres objets de l'image, ou même de la texture du tissu du gant.

Nous reprenons ici la description de notre algorithme de segmentation tel qu'il a été sommairement publié dans [J4]. De manière synthétique, notre algorithme

repose sur une étude hiérarchique de la couleur des pixels de l'image en fonction de celle du gant dans l'espace de Mahalanobis.

Etape 0 : Apprentissage. En début d'utilisation, un apprentissage de la couleur du gant en fonction des conditions d'acquisition est effectué. Cet apprentissage est réalisé sur la première image de la séquence. Il est recommandé que sur celle-ci, la main soit bien déployée, afin d'avoir un maximum de variations locales de luminance/chrominances pour l'apprentissage de sa couleur.

L'apprentissage est réalisé de manière statistique dans un rectangle strictement inscrit dans la surface gantée (cf. Figure III-4). On nomme *Training* l'ensemble des pixels de ce rectangle. Une modélisation probabiliste de la statistique obtenue est réalisée en prenant l'hypothèse que chaque composante de la couleur du gant (dans l'espace YCbCr) varie selon une loi gaussienne. On note (m_c, σ_c), le modèle gaussien correspondant, où m_c désigne le vecteur des moyennes de chaque composante, et σ_c désigne la matrice de covariance associée. L'espace de couleur YCbCr est utilisé plutôt qu'un autre parce que (1) aucune supposition n'est faite sur la couleur du gant, (2) les informations de luminance et de chrominance doivent être séparées pour pouvoir compenser les variations brusques d'illumination, (3) c'est un espace classique dans lequel de nombreuses caméras peuvent coder directement leur acquisition. Par ailleurs, la transformation de l'espace RGB vers l'espace YCbCr est linéaire.

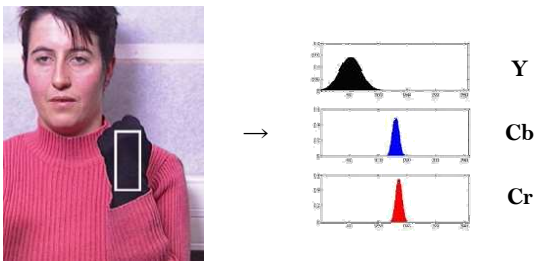


Figure III-4 : apprentissage de la couleur du gant sur l'ensemble *Training*.

Ensuite, une fois que la couleur du gant a été apprise, toutes les images du flux vidéo sont traitées tour à tour à partir de l'étape 1 jusqu'à l'étape 9.

Etape 1 : Correction de l'éclairage. La luminance de chaque image est mesurée et modélisée par une gaussienne tout au long de la vidéo, et une correction de la gaussienne est apportée afin qu'elle reste centrée sur la même valeur au cours du temps. Une telle correction est relativement simple, mais suffisamment efficace dans le cas d'une luminance qui reste globalement constante, comme par exemple avec un éclairage artificiel (sous lequel toutes nos vidéos ont été acquises). Dans le cas d'un éclairage variable, (éclairage naturel par exemple), une correction par filtrage de Kalman (cf. appendice C.2 p. 293) est plus adaptée.

Etape 2 : Image de Similarité. Pour chaque pixel *p*, la distance de Mahalanobis entre *p* et la couleur du gant est calculée ; cela permet d'associer à chaque image *I*, une Image de Similarité IS*(I)* (cf. Figure III-5). Cela correspond en fait à évaluer *p* sous le modèle gaussien (m_c, σ_c) de la couleur du gant. D'un point de vue mathématique, l'Image de Similarité est le résultat de la transformée de Mahalanobis de l'image originale :

$$IS\left(I(p)\right) = MT_{m_c,\sigma_c}\left(I(p)\right)$$

$$MT_{m_c,\sigma_c}(p) = 1 - \exp\left(-\frac{(p-m_c)\cdot\sigma_c\cdot(p-m_c)^{\mathrm{T}}}{2\cdot\det(\sigma_c)}\right)$$

I étant une image, *p* un pixel, MT_{m_c,σ_c} la transformée de Mahalanobis de noyau (m_c,σ_c) et $\det(\sigma_c)$ étant le déterminant de σ_c.

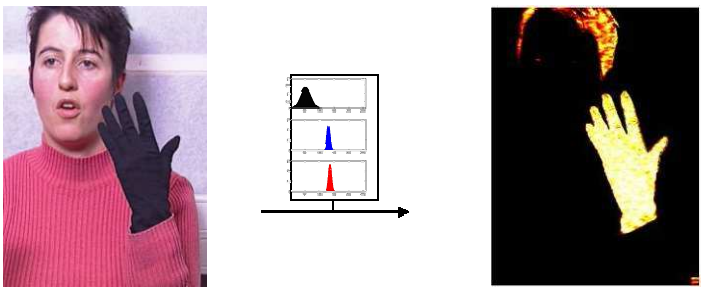


Figure III-*5* : pour chaque image courante, l'Image de Similarité est calculée. A gauche, image originale et à droite, image de similarité en niveaux de luminance : plus un pixel est sombre, plus ses caractéristiques de luminance/chrominance sont éloignées de celles du gant.

Etape 3 : Localisation de la main dans l'image. Un premier seuil très restrictif *T1* est appliqué à l'Image de Similarité, dans le but de repérer les pixels qui ont une couleur très proche de celle apprise (pixels appartenant très sûrement au gant). Cela permet de repérer les régions d'intérêt, c'est-à-dire les régions où il y a suffisamment de pixels proches de la couleur du gant pour que la main s'y trouve. *T1* est automatiquement calculé dès l'étape 0, en fonction de *Training* :

$$T1 = \frac{1}{2}\cdot\left(\underset{Training}{\mathrm{mean}}(IS) + \frac{\underset{Training}{\max}(IS)}{\underset{Training}{\min}(IS)}\right)$$

où mean(.), min(.) et max(.) désignent les fonctions moyenne, minimum et maximum sur l'ensemble indiqué en indice. Un exemple de résultat à l'issue de l'étape 3 est proposé Figure III-6b.

Etape 4 : Complétion de la connexité des objets. Un deuxième seuil *T2* est ensuite appliqué aux pixels qui n'ont pas encore été sélectionnés. Ce seuil est plus tolérant que le précédent, afin de récupérer plus de pixels, et donc espérer obtenir un objet connexe. Seulement, ce seuil est plus tolérant pour les pixels

dont un grand nombre de voisins ont été sélectionnés par le premier seuil (car de tels pixels sont vraisemblablement dans la zone correspondant au gant) que pour les pixels qui sont éloignés de tout pixel initialement sélectionné (car de tels pixels sont vraisemblablement hors de la zone d'intérêt). Ainsi, chaque pixel dans un voisinage de 5×5 du pixel *p* se voit attribuer un poids en fonction de sa position par rapport à *p*. Les poids des 25 pixels du voisinage sont présentés dans la matrice *GWM*. La somme de tous ces poids est ensuite utilisée pour pondérer *T1*. En pratique, GWM est simplement l'échantillonnage d'une fonction gaussienne 2D.

$$T2(x,y) = \frac{3 \cdot T1}{4} \cdot \left(\sum_{i=-2}^{2} \sum_{j=-2}^{2} \left(GWM(i,j) \cdot Nbgr_{x,y}(i,j) \right) \right)^{-1}$$

$$\text{avec} \quad Nbgr_{x,y} = \begin{pmatrix} IS(x-2,y-2) & \cdots & \cdots & \cdots & IS(x+2,y-2) \\ \vdots & \ddots & & \iddots & \vdots \\ \vdots & & IS(x,y) & & \vdots \\ \vdots & \iddots & & \ddots & \vdots \\ IS(x-2,y-1) & \cdots & \cdots & \cdots & IS(x+2,y+2) \end{pmatrix}$$

$$\text{et} \quad GWM = \begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}$$

IS(*x, y*) étant la valeur du pixel *p*(*x, y*) dans IS. Cette méthode permet d'extraire l'information liée à la cohérence spatiale de la couleur, tout en rendant les objets de l'image (qui ont la même couleur que le gant) connexes (Figure III-6c).

Etape 5 : Extraction d'un objet connexe. Un algorithme d'étiquetage et de recherche de composantes connexes permet d'extraire l'objet d'intérêt, à savoir la main gantée. Une mise en œuvre efficace de cet algorithme est détaillée dans appendice C.3, p. 295. Au niveau d'un traitement image à image, il est difficile d'extraire le bon objet dans le cas où plusieurs groupes de pixels connexes apparaissent. En revanche, lors de l'application de cet algorithme à toutes les images de la vidéo, il est beaucoup plus facile de repérer la main puisque celle-ci ne se déplace pas beaucoup d'une image à l'autre. Ainsi, au niveau vidéo, un algorithme de suivi naïf permet de choisir entre plusieurs objets le cas échéant. Cet algorithme se contente de choisir l'objet le plus près de celui ayant été identifié comme la main sur l'image précédente. Sur l'image initiale, la position de la main est connue par la position du rectangle *Training*. Cet algorithme est relativement simple, mais suffit en raison de la haute cadence d'acquisition. Encore une fois, il pourrait avantageusement être remplacé par un filtre de Kalman afin de gagner en robustesse sur des vidéos acquises à des cadences plus basses.

Etape 6 : Lissage. L'image de similarité est ensuite lissée par application d'un filtre de convolution de noyau gaussien (*GK*). Cela permet de réduire le bruit provenant des approximations permettant d'obtenir l'image de similarité (celle-ci utilise une fonction exponentielle, simplifiée au moyen de développements en séries entières) et des autres erreurs d'arrondis lors des calculs en virgule fixe. L'image ainsi obtenue est appelée GKSM (Gaussian Kernel Similarity Map). *GK* est défini comme suit :

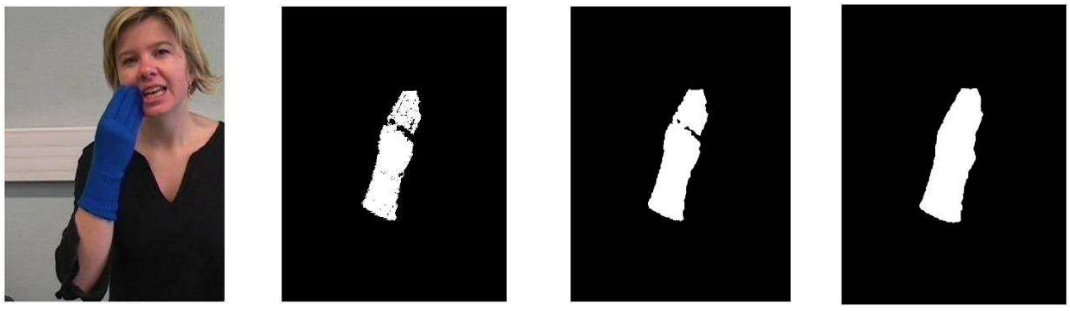
$$GK = \frac{1}{159}\begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}$$

Etape 7 : "Holes filling" ou démitage. L'image obtenue est souvent celle d'une main "mitée" : il y a de nombreux petit trous qui n'ont pas été bouchés par les précédents traitements (Figure III-6c). Pour pallier cela, nous appliquons en post-traitement un dernier seuil *T3* aux pixels qui ne font pas partie de l'objet dans GKSM, mais qui se trouvent dans un voisinage 15×15 d'un pixel de l'objet. *T3* est aussi calculé de manière automatique :

$$T3 = \frac{\max_{Training}(IS)}{\min_{Training}(IS)} - 0.1$$

Etape 8 : Filtre médian[1]. Sur une image binaire, un filtre médian de taille 5×5 convertit un pixel à la couleur des pixels de la majorité des 25 pixels du voisinage, afin d'améliorer la connexité de la main.

Etape 9 : Suppression du bruit. Un second étiquetage en composante connexe élimine les fausses alarmes induites par les deux étapes précédentes.



(a) (b) (c) (d)

Figure III-6 : (a) image originale, (b) premier seuillage (étape 3) (c) deuxième seuillage et post-traitement (étape 4 à étape 6) (d) troisième seuillage et post-traitement (étape 7 à étape 9).

[1] Conformément à ce que préconise l'état de l'art, notons qu'une morphologie mathématique permet de réaliser ce type de traitement de manière plus efficace qu'un filtre médian. Cependant, il s'agit d'un traitement plus lourd, et dans notre cas, l'efficacité du filtre médian est suffisante.

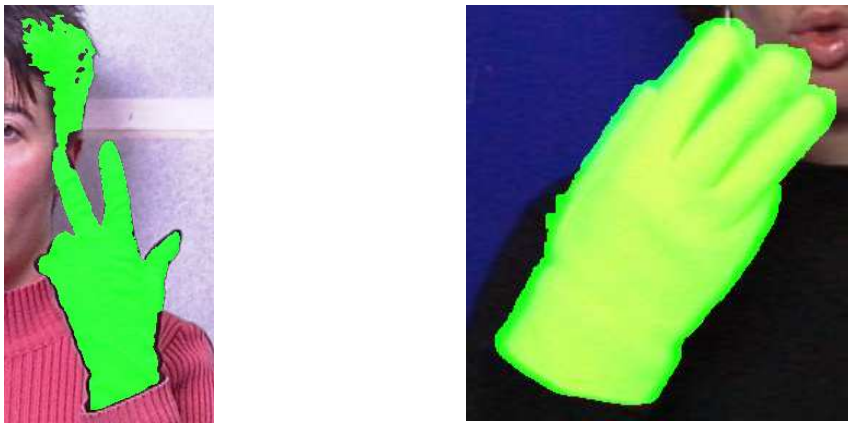
III.2.3 Evaluation de la méthode proposée

Afin d'évaluer la qualité de la segmentation, plusieurs mesures peuvent être considérées. La méthode la plus objective est de définir une vérité terrain pour chaque image du corpus de test et d'effectuer une comparaison. Celle-ci peut ensuite être chiffrée à l'aide de deux indices, à savoir la proportion de pixels manquant à la segmentation automatique pour correspondre à la vérité terrain, et la proportion de pixels ayant été segmentés automatiquement alors qu'ils n'auraient pas dû l'être. Cette méthode objective comporte l'inconvénient majeur de n'être pas très significative par rapport à notre problème de reconnaissance du geste manuel. En effet, on peut voir sur la Figure III-7 et sur la Figure III-8, que selon la répartition des pixels d'erreur, le résultat peut être très bon en termes de reconnaissance de la Configuration comme très mauvais. Ainsi, sur la Figure III-7a les pixels manquants suite à la segmentation vont conduire à une interprétation erronée de la Configuration, alors que ce n'est pas le cas sur la Figure III-7b et sur la Figure III-7c, où pourtant le nombre de pixels manquants est du même ordre de grandeur. Il en est de même avec des pixels ajoutés par erreur sur la Figure III-8.

(a) (b) (c)

Figure III-7 : une même proportion de pixels manquants peut empêcher toute reconnaissance de la Configuration dans certains cas (a), et ne pas être gênante dans d'autres (b), (c).

En conséquence, ce critère n'est pas adapté. Nous en proposons deux autres : le premier consiste à prendre le problème de reconnaissance comme un tout, et de considérer que la segmentation est de qualité suffisante lorsqu'elle permet une bonne reconnaissance du geste manuel. Ainsi, il s'agit d'évaluer globalement la segmentation et la classification. Dans la section V.2 (p. 132), les évaluations de la classification des formes de main sont donc aussi considérées sous cet angle.



(a) (b)

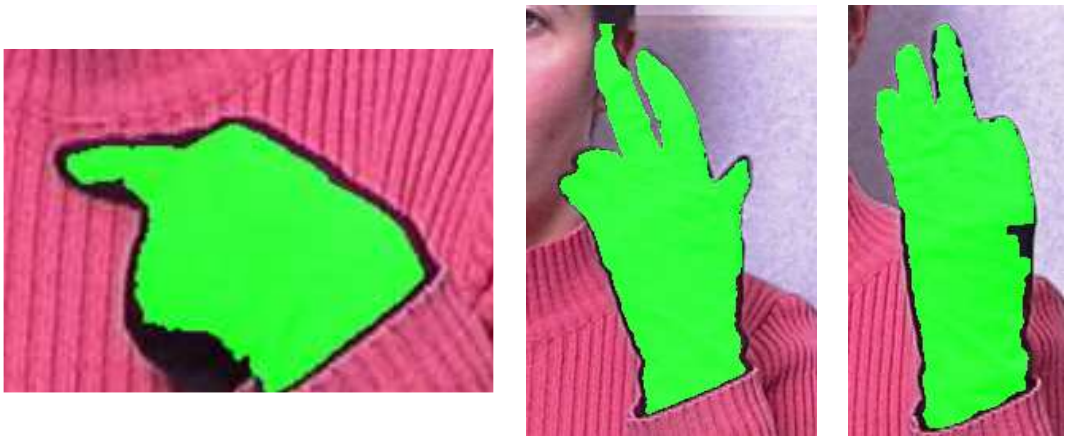
Figure III-8 : une même proportion de pixels ajoutés par erreur peut empêcher toute reconnaissance de la Configuration dans certains cas (a), et ne pas être gênante dans d'autres (b).

La seconde méthode, que nous appliquons ici, consiste à évaluer subjectivement la forme de la main segmentée, en déterminant qualitativement si elle correspond à la forme globale de la main réelle. Ainsi, si un nombre conséquent de pixels manque sur le bord de la main sans pour autant en changer l'apparence, la segmentation sera considérée comme bonne. C'est le cas par exemple de la Figure III-9a où (1) le contour est réduit, et où (2) un doigt replié en arrière plan mais qui ne participe pas à la production de la Configuration manque. A l'inverse, si un faible nombre de pixels manque ou est rajouté et que cela change l'apparence de la main, la segmentation sera considérée comme défectueuse. Ainsi la segmentation est défectueuse lorsque par exemple (1) un trou apparaît sur le contour de la forme (Figure III-9c), (2) un doigt manque (Figure III-7a), (3) un morceau de la chevelure est agglutiné à la main (Figure III-8a), etc. Notons qu'en fonction de la déformation que cela implique, un même défaut peut ne pas avoir de conséquence ou au contraire entraîner une mauvaise segmentation. Ainsi, le morceau de chevelure ajouté sur la Figure III-9b n'est pas problématique à l'inverse de ce qui est présenté sur la Figure III-8a.

Afin d'avoir une idée de la qualité de la segmentation, nous avons utilisé plusieurs types de vidéos. Les premiers tests ont été effectués sur les vidéos de ETTRAN et MAGOZ, où nous avons pu tester l'influence du gant et du changement de codeur. Ensuite, des tests à plus petite échelle, issus de données de CORPDIV, ont permis de tester d'autres conditions d'éclairage et d'autres résolutions.

Tout d'abord, il semble que la qualité de l'apprentissage varie beaucoup. Son influence sur la qualité de la segmentation est importante. D'une manière générale, les personnes ayant participé à la conception de l'algorithme, ou les gens l'ayant utilisé un grand nombre de fois finissent par avoir une bonne intuition sur la manière dont doit être défini l'ensemble d'apprentissage *Training*,

et sont capables de toujours proposer un bon apprentissage au système. En revanche, un néophyte ne parvient pas toujours à spécifier la zone d'apprentissage de telle sorte que la segmentation soit aussi bonne. Cela signifie que la sélection de la zone d'apprentissage n'est pas naturelle au premier abord. Cette forte dépendance de la segmentation vis-à-vis de la qualité de l'apprentissage devra par la suite être compensée. Pour l'instant il est difficile de définir de manière exhaustive les conditions d'un bon apprentissage. Ainsi, dans une problématique d'usage, nous pensons qu'une étude ergonomique et la mise en place d'une interface adaptée seraient nécessaires pour garantir qu'un codeur qui n'est pas familier avec les méthodes de segmentation puissent enseigner à la machine la couleur de son gant facilement et convenablement. Bien sûr, cela dépasse le cadre de ce travail de recherche.



(a) (b) (c)

Figure III-9 : malgré la suppression du contour et des doigts en arrière plan, l'image (a) est considérée comme bien segmentée. Malgré la suppression du contour et l'ajout de pixel des cheveux, l'image (b) est considérée comme bien segmentée. En revanche, la forme de (c) n'est pas correcte.

Sur ETTRAN et MAGOZ, l'évaluation subjective que nous proposons donne de bons résultats à partir du moment où l'apprentissage de la couleur du gant est effectué correctement. Voici les résultats obtenus :

– Sur ETTRAN N, les séquences vidéo testées contiennent un total de 1162 images. Le pourcentage d'images étant correctement segmentées dans chacune des séquences va de 95.8% à 100% avec une moyenne de 99.4%.

– Sur MAGOZ R le pourcentage d'images bien segmentées par séquence vidéo est de 99.56%, pour un total de 1372 images testées. Seules 139 images sont segmentées de telle sorte qu'au moins un trou apparaît dans la forme de la main (segmentation "mitée"). De tels trous ne sont pas toujours problématiques dans la réalité, puisqu'il arrive que des espaces fermés apparaissent entre les doigts (cf. Figure III-7b). Cependant, cela est très rare et il est plus intéressant de supprimer tous les trous qui mitent la main que de les accepter tous. Comme le seuil *T3* a pour objectif de les supprimer, ces images ont mis en défaut l'algorithme de segmentation.

– Sur MAGOZ J, le taux de bonne segmentation est de 98.98% avec 61 trous pour 1185 images.

Dans le cas où les apprentissages sont effectués de manière naïve, en positionnant le rectangle au hasard et sans chercher à maximiser sa taille, comme pourrait le faire un utilisateur néophyte, les résultats sont moins bons : dans le cas d'ETTRAN N, le taux moyen de bonne segmentation passe de 99.4% à 94.85%, avec une séquence pour laquelle ce taux descend à 77.9%. Dans le cas de MAGOZ R, celui-ci passe de 99.56% et 61 trous à 94% et 492 trous (les images sont donc très mitées). Enfin, dans le cas du gant jaune, il n'y a pas de différence notable dans les résultats puisque la forme de la main est globalement suffisamment respectée pour que l'on puisse reconnaître la Configuration. Cependant, la partie inférieure du poignet (qui ne nous intéresse pas) est souvent très mal segmentée (Figure III-10), ce qui finalement, souligne aussi l'instabilité d'un mauvais apprentissage, et l'influence de sa qualité sur celle de la segmentation.

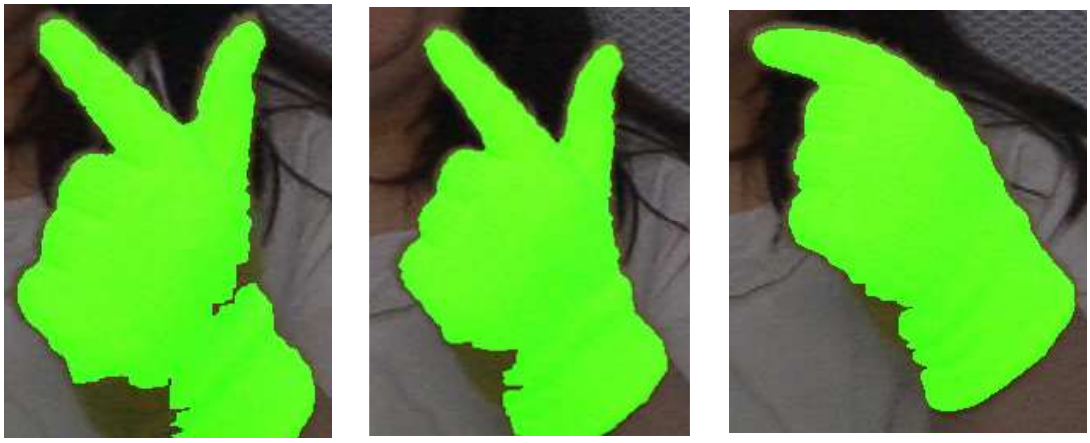


Figure III-10 : mauvaises segmentations du poignet en fonction de l'apprentissage. Néanmoins, la forme de la main est tout à fait respectée.

Les évaluations sur CORPDIV laissent apparaître que :

– une bonne segmentation n'est pas garantie avec des caméras de type WebCam, même de qualité supérieure. Ainsi, sur des images de la qualité de celle de la seconde rangée de la Figure III-15, il arrive que des doigts soient perdus. Nous comptons néanmoins sur la rapide évolution de ce matériel pour rendre nos algorithmes portables. A titre indicatif, les images de la Figure III-6 ont été acquises avec une caméra digitale portable grand public (caméra numérique non professionnelle, mais de qualité supérieure aux WebCam actuelles), en lumière naturelle, et dans un environnement intérieur non contrôlé. Qualitativement, la segmentation que nous fournissons est encore suffisamment correcte pour notre application.

– La qualité de la segmentation n'est pas perturbée par une évolution gaussienne de la luminance ou par l'ajout d'un bruit blanc avec un rapport

signal sur bruit inférieur à 60% [14] (cf. Figure III-11). Ces derniers résultats sont prévisibles de par la modélisation gaussienne de la couleur.

– Si la cadence du "shutter speed" de la caméra est suffisamment élevée, et qu'aucun effet de flou n'apparaît en raison de la vitesse de déplacement de la main et des doigts, alors, il n'y a pas de différence de qualité dans la segmentation d'éléments statiques et d'éléments dynamiques dans l'image.

(a) (b)

Figure III-11 : (a) modification de la luminance et (b) ajout d'un bruit blanc.

La segmentation est donc dans l'ensemble satisfaisante. Des exemples de segmentation sont donnés sur la Figure III-12 et sur la Figure III-13.

Une étude qualitative des cas d'erreur laisse apparaître plusieurs informations quant au choix de la couleur du gant par le codeur :

– Un gant de couleur similaire à un autre élément de l'image (peau, cheveux, vêtement, objet du fond, etc.) doit bien sûr être évité.

– Un gant de couleur trop claire, pour lequel il y a trop d'effets d'ombre (Figure III-14), est aussi à proscrire. Il en est de même pour les gants de couleur trop sombre, pour lesquels seule la luminance est discriminante (Figure III-8).

– Les résultats issus des caméras ayant des capteurs de couleur de mauvaise qualité ne peuvent pas être compensés par des couleurs de gant faciles à segmenter.

– La forte texture des éléments autres que la main (tels que les vêtements) a une influence. A cause de l'utilisation intensive de convolution dans la segmentation, la frontière entre la main et les zones texturées n'est pas précise (Figure III-7).

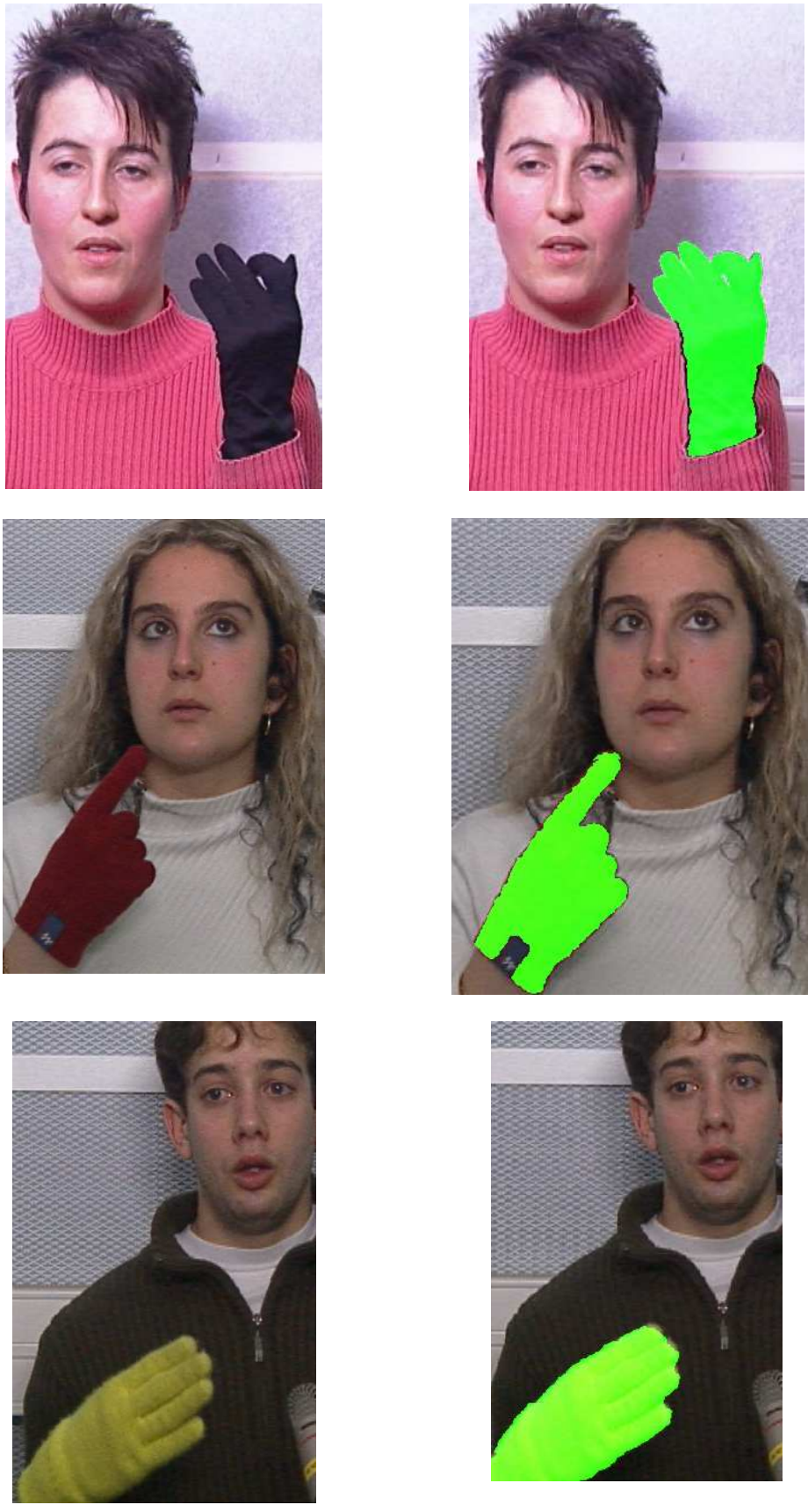


Figure III-12 : exemples de segmentation.

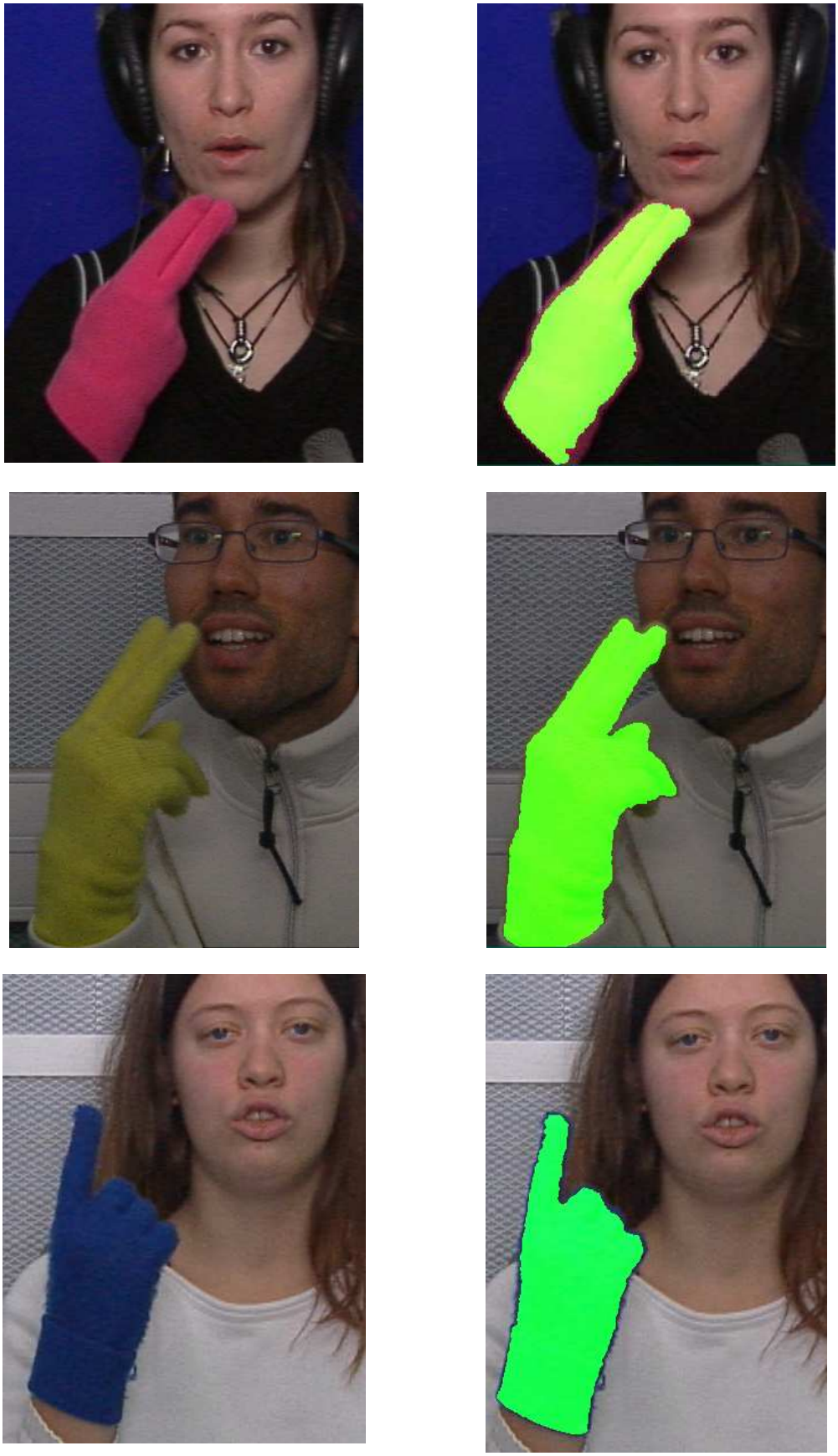
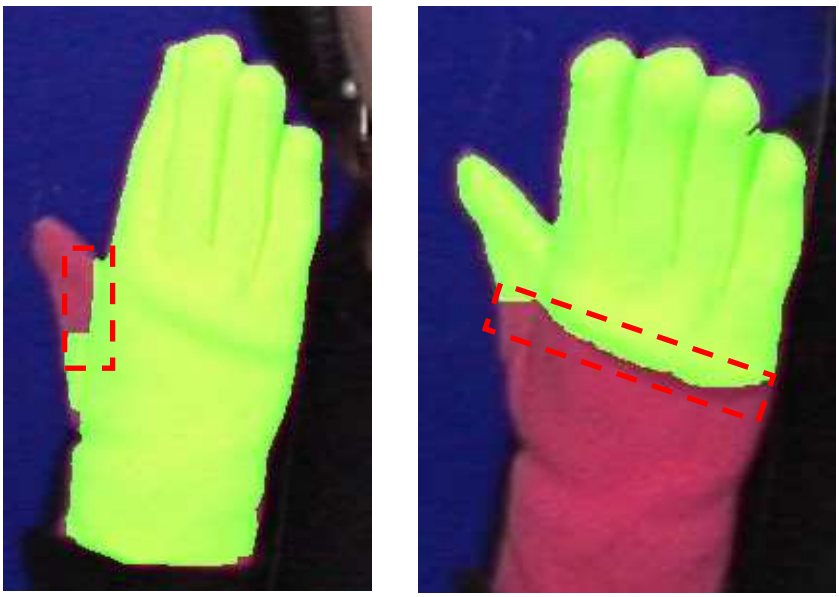


Figure III-13 : exemples de segmentation.

– En terme de coût calculatoire, ces algorithmes ne sont pour l'instant implantés qu'en code MatLab/C/C++, et sans aucune optimisation au niveau micro-processeur. Ils tournent sur des stations de travail classiques Intel Pentium® IV sous Microsoft Windows® XP. Une cadence de traitement de 5.75image/sec, pour des images de taille 480 × 370 pixels, est prometteuse pour de futures implantations complètes en C ou sur du hardware dédié.

Ces résultats indiquent globalement que, malgré le nombre important d'étapes impliquées dans la segmentation (et dont le nombre des erreurs se cumulent), le module de segmentation est à la fois robuste et efficace par rapport aux besoins du projet TELMA. En effet, celui-ci est robuste à (1) de légères variations d'éclairage, (2) des conditions d'acquisition variables, (3) des qualités d'acquisition variables, (4) l'utilisation de gants différents (dont la couleur reste uniforme). De plus, ce module est d'une complexité calculatoire raisonnable. Enfin, l'algorithme s'appuyant principalement sur des convolutions, sur de la recherche de composantes connexes et sur des seuillages, son implantation sur un système hardware dédié est tout à fait envisageable.



(a) (b)

Figure III-14 : les effets d'ombres (repérés par les rectangles en pointillés rouges) empêchent une bonne segmentation.

En termes d'efficacité, l'intérêt de cet algorithme réside dans la succession de plusieurs seuillages appliqués à l'image. Cela permet d'avoir une tolérance variable sur la sélection de chacun des pixels en fonction de sa position par rapport aux autres pixels dignes d'intérêt. Afin d'illustrer cela, nous pouvons comparer notre méthode à un algorithme similaire plus intuitif, mais moins "raffiné", pour lequel un seul seuil fixe est appliqué sur l'*IS*. Même si ce seuil est réglé manuellement de manière à être optimal pour chaque séquence, il existe toujours des images pour lesquelles la segmentation n'est pas efficace. A contrario, notre algorithme et son paramétrage automatique traitent parfaitement ces cas, comme cela est illustré sur la Figure III-15.

(a) (b) (c)

Figure III-15 : comparaison illustrée de l'intérêt des trois seuils (c), par rapport à une méthode à un seul seuil (b), pour une image donnée (a).

III.3 Définition de l'élément pointeur

III.3.1 Introduction

A partir de la forme de la main, il est nécessaire de définir un élément pointeur qui se rapproche autant que possible de la pulpe du doigt utilisé pour spécifier une Position durant le codage LPC. D'un point de vue théorique, le doigt pointeur est toujours le doigt le plus long parmi les doigts déployés, c'est-à-dire que, quand la Configuration de la main est telle que le majeur est déployé, il est le doigt pointeur, et dans les autres cas, il s'agit de l'index. Durant les transitions, il n'y a pas de doigt pointeur. Cependant, nous pouvons tout de même définir un élément pointeur durant les transitions (en utilisant le même algorithme durant le geste et durant les transitions). Il est intéressant de faire cela afin de vérifier que la trajectoire du doigt pointeur est la plus continue possible, d'une Position à une autre (cela est en effet très utile pour vérifier la stabilité de la définition du doigt pointeur lors de l'évaluation).

La détermination du doigt pointeur est un problème que nous traitons au niveau de l'image, sans aucune considération d'analyse dynamique de la vidéo. Ce choix peut être discuté, mais nous le soutenons avant tout pour des raisons de simplicité. De plus, et aussi par choix, cette étape est effectuée de manière indépendante de la reconnaissance de la Configuration. Même s'il y a une forte

dépendance entre le doigt pointeur et la Configuration, nous souhaitons garder une certaine indépendance entre les deux traitements qui permettent d'extraire ces deux informations. En effet, celles-ci sont orthogonales du point de vue du LPC, et par conséquent, maintenir leurs extractions indépendantes est un gage de robustesse.

Nous avons proposé plusieurs algorithmes pour répondre à ce problème [170]. Ils ont ensuite été méticuleusement évalués, et il est apparu que chacun d'eux donnait de bons résultats dans certains cas, mais de mauvais dans d'autres. Ainsi, la méthode retenue et présentée dans ce document est une combinaison de l'ensemble de ces algorithmes originaux. On peut résumer leurs différentes tendances de cette manière :

– Les algorithmes basés sur l'extraction du sommet le plus excentré de l'image (soit par des calculs d'enveloppe convexe, soit par utilisation de moments). C'est cette méthode que nous avons présentée dans [J4]. Ils permettent toujours de récupérer la pointe du doigt qui est le plus long sur l'image, mais en raison des imprécisions dues à la parallaxe, ce doigt n'est pas forcément le plus long morphologiquement. Cela arrive particulièrement dans le cas de la Configuration 8, comme cela est illustré sur la Figure III-16. Ainsi, ces algorithmes sont très efficaces quand les doigts sont rapprochés (car la parallaxe ne suffit pas à déformer leur taille respective), mais sont souvent mis en échec, lorsque les doigts sont écartés.

Figure III-16 : cas d'école dans lequel la pulpe la plus loin de la paume de la main est celle de l'index.

– Les méthodes qui cherchent à repérer le nombre de sommets sur le contour de la forme de la main, et à compter le nombre de doigts. En fonction du nombre de doigts et de leur hauteur respective, un sommet particulier est désigné. Ces méthodes fonctionnent bien quand les doigts sont séparés car le comptage est facile. En revanche, cela fonctionne mal lorsqu'ils sont rapprochés : en effet, en fonction du type de gant, il est parfois impossible de discriminer des doigts au

contact les uns des autres (Figure III-17). Il faut aussi noter que ce comptage, très lié à l'aspect de la forme de la main, est sujet au même type de variabilité que l'étude de la Configuration. Ainsi, il introduit un biais et une corrélation entre les erreurs sur la reconnaissance de la Configuration et de la Position.

Figure III-17 : dans un tel cas (doigts resserrés, segmentation un peu large, gant épais), il est difficile de compter automatiquement le nombre de doigts sans erreur à partir de la forme segmentée de la main.

S'ajoute à ces problèmes de resserrement des doigts et de parallaxe, celui du respect de la convention de choix du doigt pointeur par le codeur : pour des raisons morphologiques évidentes, il est dans certains cas plus facile d'utiliser l'annulaire que le médium pour pointer. C'est par exemple le cas pour coder la syllabe /b/-/i/ (Configuration 4 et Position Bouche) ; il est en effet plus naturel de pointer la commissure des lèvres avec l'annulaire qu'avec le médium tout en gardant les lèvres visibles, surtout quand les doigts sont resserrés. Ainsi, on observe parfois une légère dérive du doigt pointeur.

III.3.2 Principe de la méthode proposée

La méthode retenue pour la sélection de l'élément pointeur est décrite dans la suite de cette section. Tout au long de celle-ci, **CP** (**C**entre de la **P**aume de la main) désigne un point particulier de la forme de la main qui correspond au centre d'un cercle de rayon *R* servant d'approximation de la paume. Le calcul de ce point à partir de la forme de la main est détaillé au § V.2.1 (p. 132), qui traite de la caractérisation de la forme de la main. Nous admettons que CP est connu lors de la description de la méthode de détection du doigt pointeur.

Etape 1 : Parcours du contour de l'image binaire. Le contour de la forme de la main est décrit de manière polaire paramétrique (Figure III-18), par rapport au CP puis lissé par un filtrage passe-bas et un sous-échantillonnage, comme décrit dans [169]. Par souci de commodité, nous travaillons sur une image qui a subi une rotation de telle sorte que l'axe principal de la main soit vertical (cf. section IV.3 p. 106 pour la description de la rotation).

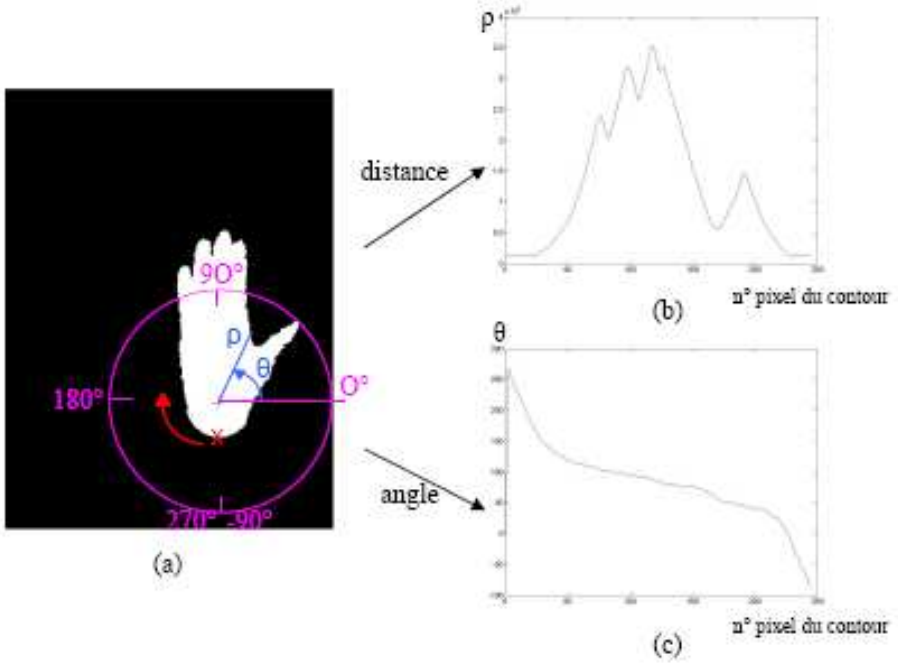


Figure III-18 : extraction du contour de la main et représentation paramétrique en coordonnées polaires.

Etape 2 : Détermination des doigts. De la représentation paramétrique du contour de la main est extrait l'ensemble $S = \{s_1, s_2, \ldots s_n\}$ des sommets dont la distance au CP est supérieure à $Q_1 \times R$, où Q_1 est un seuil. De même, on définit $C = \{c_1, c_2, c_3 \ldots c_m\}$ l'ensemble des creux dont la distance au CP est inférieure à $Q_2 \times R$, avec Q_2 un ratio préalablement défini. En pratique, Q_1 et Q_2 sont déterminés à partir de considérations morphologiques utilisées en dessin (Figure III-19). Ils représentent la borne supérieure et la borne inférieure des distances entre les doigts et CP en fonction de R. Ainsi, nous avons $Q_1 = 5/3$ et $Q_2 = 4/3$.

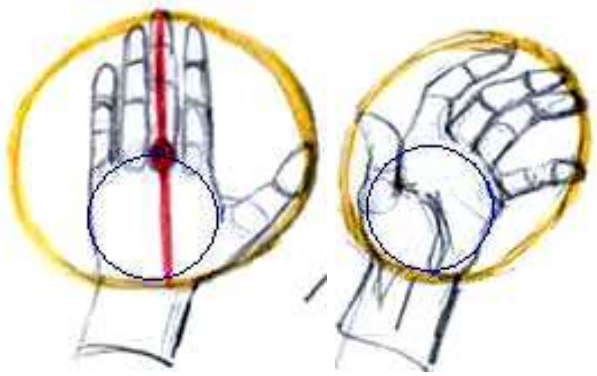


Figure III-19 : les proportions de la main [166] : la main est circonscrite à un cercle dont le rayon fait Q fois la taille de celui du cercle inscrit dans la paume, avec $Q_2 < Q < Q_1$.

Etape 3 : Regroupement des doigts en fonction de leur proximité. Pour tous les couples d'éléments consécutifs (s_i, s_{i+1}) de la liste associée à l'ensemble S des sommets, l'angle (s_i, CP, s_{i+1}) est mesuré (Figure III-20). S'il est inférieur à un seuil noté a_{min}, les deux sommets sont regroupés. On procède de même lorsqu'entre 2 éléments successifs (s_i, s_{i+1}), il n'existe pas d'élément de C. On a :

$$a_{min} = K \times \left| \frac{s_i.dist - s_{i+1}.dist}{\max(s_i.dist, s_{i+1}.dist)} \right|$$

où K correspond à l'angle à partir duquel on considère que 2 doigts sont séparés, c'est-à-dire 20°, et où *pixel.dist* désigne la distance de *pixel* à CP.

Etape 4 : Détermination du groupe de doigts comprenant l'élément pointeur. C'est le groupe de doigts le plus près de l'auriculaire (donc le plus à gauche sur des vidéos de gauchers) qui contient l'élément pointeur. Désormais, on ne considèrera plus que le sous-ensemble $S_{gauche} = \{s_1, s_2 \dots s_k\}$, $k \leq n$ de S, correspondant à ce groupe de doigts.

Etape 5 : Affinage de l'emplacement des sommets. Afin que l'élément pointeur soit morphologiquement plus crédible, on le décale d'une distance d vers l'intérieur du doigt : celui-ci est ainsi positionné sur la pulpe du doigt plutôt que sur le contour proprement dit. d s'étalonne en fonction de la taille de la main dans l'image. En pratique, un décalage de 6 pixels est suffisant ; cela correspond approximativement au rayon de la pulpe d'un doigt, pour des images sur lesquelles la résolution est suffisante pour qu'il soit possible de segmenter correctement les lèvres (cf. section III.1 p. 59).

Etape 6 : Détermination d'un premier élément pointeur. Soit S_{SIB} le sous-ensemble de S_{gauche} tel que :

$$\forall s \in S_{SIB}, s.ligne > \max_{s_i \in S_{gauche}} (s_i.ligne) - b$$

où *pixel.ligne* désigne la coordonnée verticale de *pixel* et où b est un seuil au-delà duquel l'élément pointeur peut se trouver. En pratique, $b = 12$ pixels sur des images de la résolution qui nous intéresse. P_1 est l'élément le plus à gauche de S_{SIB} (c'est-à-dire, l'élément de S_{SIB} dont le numéro de colonne est le plus petit sur l'image de la main, comme indiqué sur la Figure III-21).

(a)

(b)

(c)

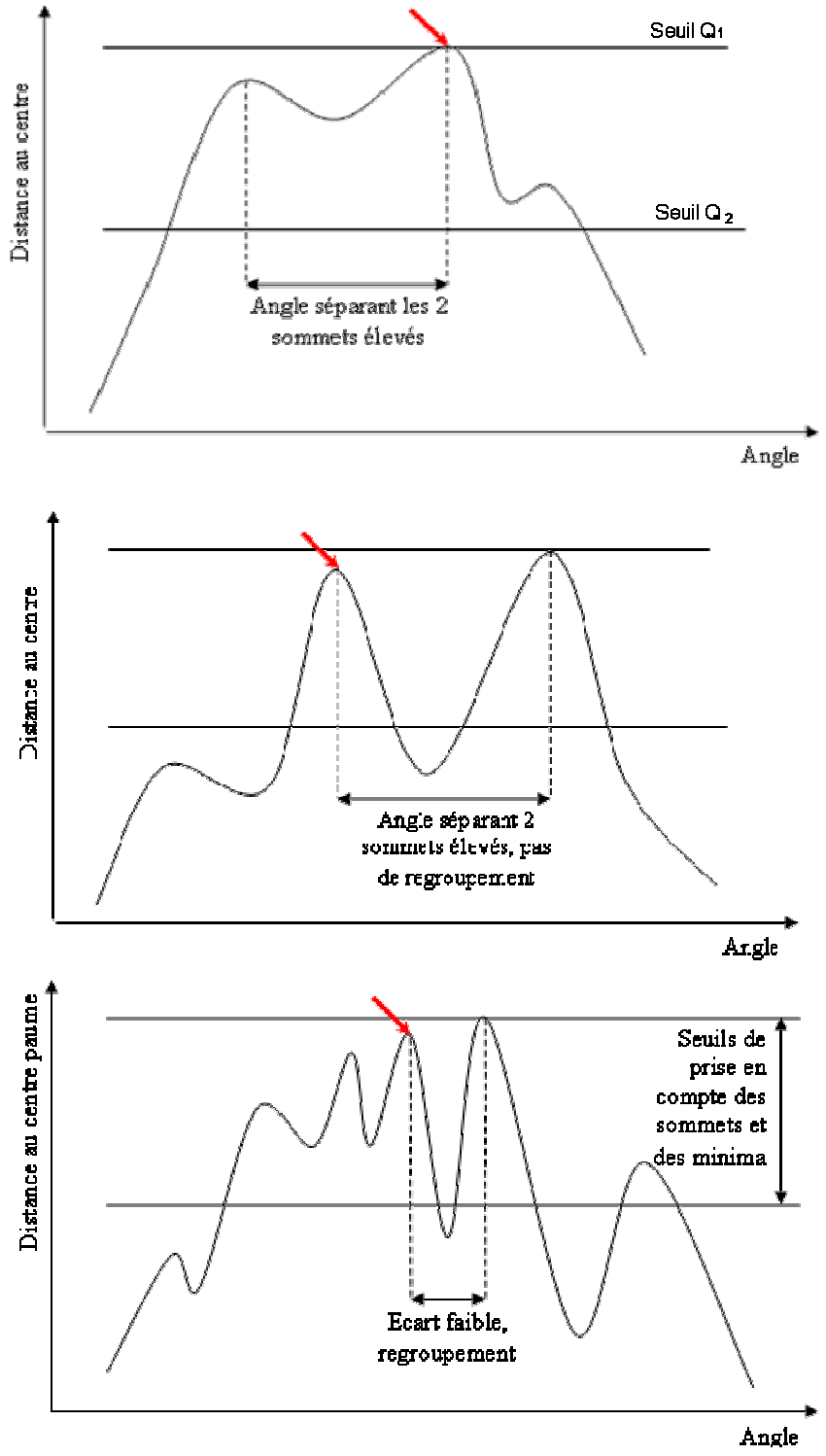


Figure III-20 : regroupement des sommets en fonction de leur écartement et des seuils Q_1 et Q_2. La flèche désigne le doigt pointeur. Le cas (a) représente une Configuration 3 où l'angle séparant les sommets est important. En revanche, la différence de hauteur avec la vallée entre eux est faible. Ainsi les deux sommets sont regroupés. Le cas (b) représente une Configuration 8, où la hauteur par rapport à la vallée et l'angle séparant les deux sommets sont tous les deux importants. En conséquence, les deux sommets ne sont pas rassemblés. Enfin, le cas (c) représente une Configuration 5. Les deux sommets sont regroupés malgré la profondeur de la vallée en raison du faible angle entre les deux sommets.

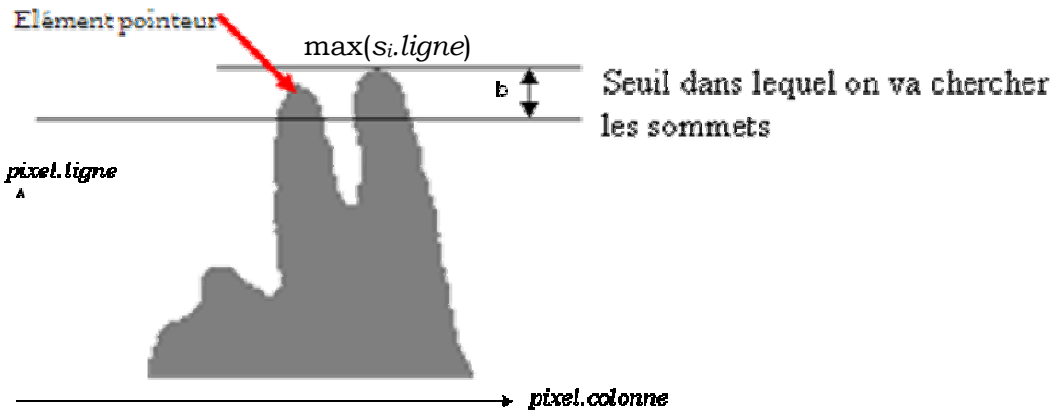


Figure III-21 : détermination de P_1.

Etape 7 : Détermination d'un second élément pointeur. Soit P_2 le sommet de Sgauche dont la distance au CP est la plus grande.

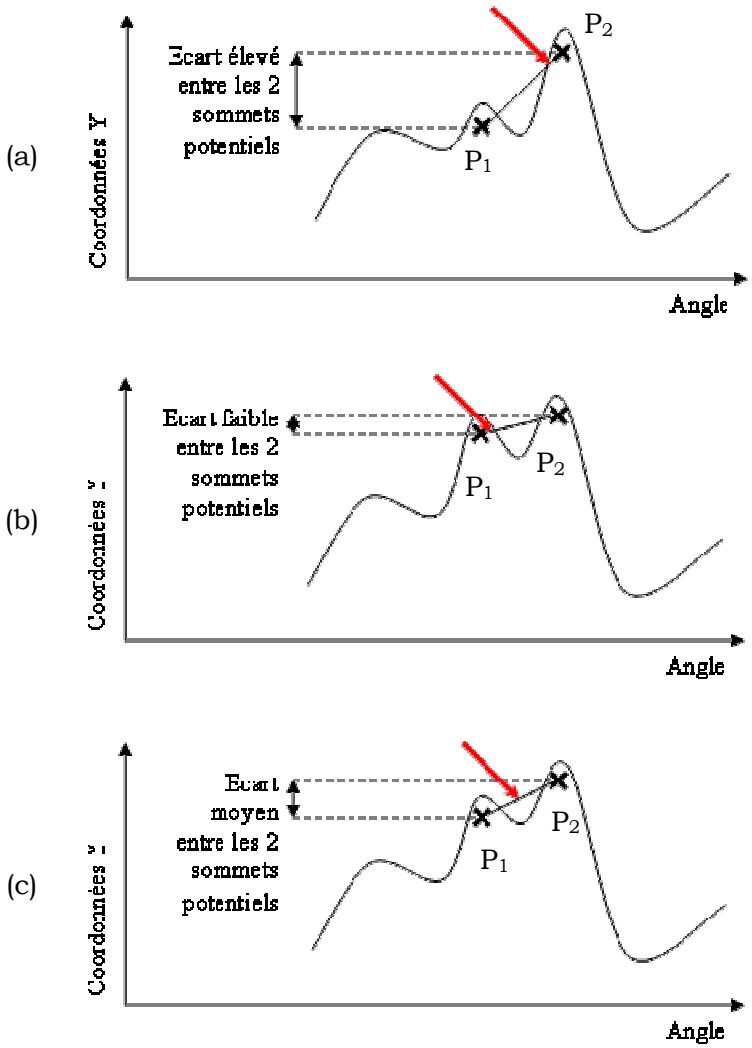


Figure III-22 : illustration du choix de l'élément pointeur (la flèche) dans plusieurs cas typiques.

Etape 8 : Détermination final de l'emplacement du doigt pointeur. Soit P_{final} le pixel correspondant à l'élément pointeur. Il est défini comme un "juste milieu" entre les deux doigts pointeurs P_1 et P_2. P_{final} est tel que :

$$\overrightarrow{P_2P_{final}} = \left| \frac{P_1.ligne - P_2.ligne}{C_{comb}} \right| \overrightarrow{P_2P_1}$$

avec $C_{comb} = 2.b$. C_{comb} permet de définir l'importance relative que l'on accorde à chacun des deux éléments pointeurs naïfs. On peut éventuellement le régler en fonction des habitudes du codeur. La conséquence directe de cette pondération est que l'élément pointeur n'est pas toujours un pixel de la forme de la main (Figure III-22). En pratique cela fonctionne mieux malgré tout ; la précision du pointage est plus élevée de cette manière, comme si le codeur, dans le cas d'un pointage très précis, utilisait l'enveloppe convexe de l'ensemble de sa main pour pointer, plutôt que la pulpe du doigt pointeur. La motivation du codeur pour pointer une Position de cette manière peut simplement être de ne pas trop masquer le visage, et notamment les lèvres.

III.3.3 Evaluation de la méthode proposée

L'évaluation de la définition du doigt pointeur est réalisée comme suit. Les 4 corpus ETTRAN N, MAGOZ B, MAGOZ J et MAGOZ R (représentant 4 gants différents et 4 codeurs) sont utilisés. Pour chacun d'eux, nous comptabilisons (1) le nombre d'erreurs, (2) le nombre de décalages, et (3) le nombre de déterminations précises (cf. Figure III-23) :

(a) (b) (c)

Figure III-23 : (a) le doigt pointeur est déterminé avec précision; (b) un décalage de l'élément pointeur vers l'annulaire est bénéfique, mais il peut arriver que ce ne soit pas le cas. Dès lors, il est difficile d'évaluer la pertinence des décalages pour l'élément pointeur ; (c) ici, l'erreur ne prête pas à discussion.

– une erreur a lieu quand l'élément pointeur est sur le mauvais doigt.

– un décalage a lieu quand nous avons délibérément choisi de décaler l'élément pointeur pour qu'il corresponde mieux au codage réel, mais que nous

pensons que ce décalage est trop important pour permettre une bonne reconnaissance de la Position.

– Le doigt pointeur est précisément déterminé quand il n'y ni erreur ni décalage.

Bien sûr, le fait que la Position soit mal reconnue dépend de l'élément pointeur, mais aussi de la définition des zones de pointage. En conséquence, l'évaluation des décalages ne peut être faite indépendamment et de manière rigoureuse. Ainsi, nous ne donnons pas un chiffrage précis des décalages. Quand il y a ni erreur, ni décalage, la détermination de l'élément pointeur est correcte.

Tableau III-5 : évaluation de la détermination de l'élément pointeur.

Config.	MAGOZ B		ETTRAN N		MAGOZ J		MAGOZ R	
	Nb images	Bonnes détections (%)	Nb images	Bonnes détections (%)	Nb images	Bonnes détections (%)	Nb images	Bonnes détections (%)
1	81	100,00	94	100,00	54	100,00	154	100,00
2	56	100,00	61	100,00	32	100,00	158	100,00
3	104	99,04	84	100,00	57	100,00	211	100,00
4	40	100,00	71	100,00	18	100,00	98	100,00
5	108	100,00	192	100,00	55	100,00	200	100,00
6	69	100,00	86	100,00	34	100,00	122	100,00
7	3	100,00	17	100,00	2	100,00	6	100,00
8	11	100,00	32	100,00	9	100,00	47	97,87
Total	**472**	**99,79**	**637**	**100,00**	**261**	**100,00**	**996**	**99,90**

Cette méthode donne des résultats concluants sur l'ensemble des corpus de test, comme cela est indiqué dans le Tableau III-5. Ces évaluations ne tiennent compte que des erreurs de pointage (moins de 1%). L'évaluation des décalages est plus difficile car elle est subjective. Cependant, il ressort que ces derniers correspondent à 3% à 7% des cas. Ainsi, le taux de détermination correcte de l'élément pointeur est compris entre 93% et 97% pour une évaluation stricte de toutes les erreurs ou décalages possibles.

III.4 Définition des zones de pointage

III.4.1 Principe

Maintenant que l'élément pointeur est défini pour chaque image, il est nécessaire de définir les zones qui peuvent être pointées. Celles-ci étant des éléments constitutifs du visage, il est nécessaire de repérer au préalable le visage dans l'image.

Pour cela, nous utilisons le **Convolutional Face Finder** ([49], CFF) un algorithme de détection de visage dont le noyau est un réseau de neurones convolutifs couplé à un Perceptron Multi-Couche, et dont les apprentissages

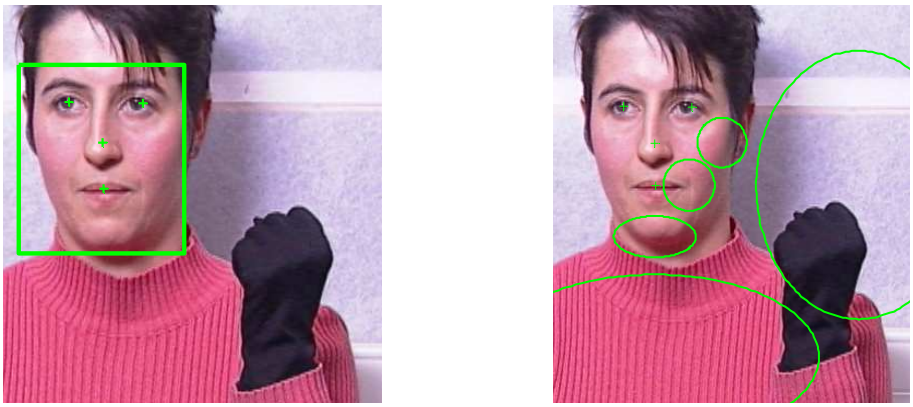
sont effectués de manière conjointe (cf. appendice C.1 p. 291). Cet algorithme, développé par Christophe Garcia à France Telecom R&D, prend en entrée une image en niveau de gris et fournit en sortie les coordonnées de la boite englobante correspondant au visage (Figure III-24a). Il se caractérise par une précision importante par rapport au faible nombre de fausses alarmes [49], tout en permettant un fonctionnement temps-réel dans ses implantations optimisées [50]. D'une manière générale, les images sur lesquelles nous travaillons nécessitent une qualité suffisante pour distinguer le contour des lèvres et le reconnaître. Sur des images d'une telle qualité, le CFF n'est jamais mis en défaut, et sa performance est donc certaine (les standards de test pour la détection de visage sont définis sur des bases de données bien plus difficiles que les images qui nous intéressent). En pratique, les seules fois où nos bases de test ont mis le CFF en défaut correspondent à des séquences où le codeur tourne trop la tête et qu'il n'est plus assez face à la caméra. Mais il n'est pas possible de lire sur les lèvres dans de telles conditions.

Après avoir détecté le visage, il est nécessaire de déterminer les zones de pointage proprement dites. Parmi celles-ci, certaines sont vraiment difficiles à déterminer, pour la simple raison qu'elles ne correspondent pas à une partie précise du visage que l'on peut détourer. Par exemple, il est impossible d'établir une vérité terrain pour la pommette. De plus, certaines zones n'ont pas d'élément suffisamment discriminant pour les reconnaître facilement, quelque soit le visage considéré (par exemple le menton). Il n'est donc pas raisonnable d'espérer les repérer directement. La méthode proposée consiste donc à repérer des éléments caractéristiques du visage, comme les yeux, le nez et la bouche, et à définir les zones de pointage par rapport à ceux-ci. Cette démarche, géométrique a le mérite de la simplicité, mais il est difficile de prédire sa capacité de généralisation. Son principal inconvénient est que les zones de pointage sont très variables d'une personne à l'autre d'un point de vue morphologique : même si la position globale des traits du visage est relativement invariante, il y a de grands écarts de codage. Ces écarts sont dus :

- aux erreurs de parallaxe lors de l'acquisition. Elles rendent difficile la distinction entre d'une part, une superposition de la main avec le visage (sans contact, quand celle-ci passe devant le visage du codeur), et d'autre part, l'atteinte d'une cible réelle.

- A la variabilité inter-codeur, beaucoup trop importante pour être appréhendée dans un seul modèle. Par exemple, certaines individus codent la Position Gorge au niveau de la poitrine, ou encore la Position Pommette près de l'œil.

Afin de détecter les yeux, le nez et la bouche, (qu'on désigne par "**traits permanents**"), nous utilisons une surcouche du CFF. Quand celle-ci est couplée à ce dernier, le système est nommé **Convolutional Face & Features Finder**, ou C3F [51]. Son fonctionnement est similaire à celui du CFF et produit des résultats de même qualité. En sortie, il fournit les coordonnées de quatre points correspondant respectivement aux centres des iris, du nez et de la bouche (Figure III-24a).



(a) Détection du visage et de ses traits permanents

(b) Définition des zones de pointage par rapport aux traits permanents

Figure III-24 : définition des zones de pointage.

A partir des coordonnées des traits permanents, de considérations morphologiques et de calculs géométriques, nous pouvons définir 5 régions correspondant aux 5 zones de pointage du LPC (Figure III-24b) :

- La **zone côté** est définie par une ellipse dont le centre est positionné horizontalement à côté du visage (de telle sorte que celle-ci ne recouvre ni le visage, ni le cercle de la zone de pointage Pommette) et est verticalement centré sur le nez.

- La **zone gorge** est une ellipse dont le centre est verticalement positionné sous le visage (de telle sorte que la bouche soit équidistante du centre des deux yeux et du bord supérieur de l'ellipse) et est aligné verticalement sur les centres de la bouche et du nez.

- La **zone pommette** est définie par un cercle qui est verticalement centré sur le nez, et horizontalement, tangent à la ligne verticale passant par le centre de l'œil placé du même côté du visage. Son rayon est de 2/3 de la distance entre les yeux et le nez.

- La **zone bouche** est un cercle équivalent à celui utilisé pour la position Pommette, mais centré sur la commissure des lèvres. En l'absence temporaire d'une routine de segmentation du contour des lèvres (pour le module de lecture labiale), nous utilisons la méthode suivante pour déterminer cette commissure : d'une manière générale, la plus grande dimension des lèvres est approximativement équivalente à la distance inter-pupille. Il suffit donc de translater le centre de la bouche obtenu par le C3F de la moitié du vecteur défini par les centres des deux yeux.

- La **zone menton** est approximativement définie par une ellipse placée sous la bouche, de telle sorte que celle-ci soit équidistante du bord de l'ellipse et du centre du nez. Le petit axe de l'ellipse est superposé avec l'axe de symétrie du visage.

III.4.2 Analyse des résultats

La précision de la détection des traits permanents est relativement importante. Néanmoins, le résultat est soumis à un léger bruit (l'imprécision est inférieure à la taille de l'iris). Ainsi, sur plusieurs images consécutives, le résultat donne une impression d'instabilité. Celle-ci est bien sûr démultipliée lors de la définition des zones de pointage (qui est basée sur les coordonnées des traits permanents). En conséquence, il est nécessaire de lisser temporellement la position de la constellation des points "yeux-nez-bouche". Pour cela, nous utilisons un filtre de Kalman monodirectionnel, qui correspond au système S suivant d'équations :

$$S : \begin{cases} {}^{T}\left(x_{t+1} \quad y_{t+1} \quad \frac{d\,x_{t+1}}{dt} \quad \frac{d\,y_{t+1}}{dt} \right) = \begin{pmatrix} Id(8) & Id(8) \\ ZERO_{8\times 8} & Id(8) \end{pmatrix} \cdot {}^{T}\left(x_t \quad y_t \quad \frac{d\,x_t}{dt} \quad \frac{d\,y_t}{dt} \right) + \quad \propto \mathbf{N}(ZERO_{8\times 1}, Id(8)) \\ {}^{T}\left(X_t \quad Y_t \quad \frac{d\,X_t}{dt} \quad \frac{d\,Y_t}{dt} \right) = {}^{T}\left(x_t \quad y_t \quad \frac{d\,x_t}{dt} \quad \frac{d\,y_t}{dt} \right) + \quad \propto \mathbf{N}\left(ZERO_{8\times 1}, \mathrm{cov}\left(\frac{d\,Z}{dt} \right) \right) \end{cases}$$

où :

- x_t et y_t sont les vecteurs-colonnes des coordonnées des 4 traits permanents (la vérité terrain que l'on cherche), à l'instant t, et X_t et Y_t les observations correspondantes (observations faites ou déduites par le C3F).

- *Id(i)* est la matrice identité de taille *i*.

- $ZERO_{i\times j}$ et la matrice nulle de taille $i \times j$.

- $\propto\mathbf{N}$*(param1, param2)* est une variable aléatoire qui suit une loi gaussienne de moyenne *param1* et de covariance *param2*.

- dZ/dt est un ensemble d'apprentissage pour la variabilité de la précision du C3F par rapport au temps.

Comme cela apparaît sur la Figure III-25, les coordonnées des traits permanents sont largement plus stables après leur traitement par le filtre de Kalman. Il en est naturellement de même pour les zones de pointage qui en sont déduites.

Cependant, ce filtre a été mis en place sous l'hypothèse que la vitesse de déplacement du visage est constante. Ainsi si le codeur effectue des mouvements de tête trop importants et trop saccadés, cette hypothèse ne sera plus raisonnable. En conséquence, la constellation de points obtenue en sortie du filtrage de Kalman ne correspondra plus aux traits permanents du visage.

Comme nous l'avons dit plus haut, il y a deux types de variabilité qu'il faut tester afin d'évaluer la qualité de la définition des zones de pointage. La première est la variabilité des zones morphologiques utilisées pour le codage. La seconde concerne celle de la morphologie proprement dite, indépendamment du codage. Illustrons cette différence sur un exemple : il y a une différence entre se demander si (1) l'ellipse détourant le menton se trouve bien autour du menton, et avec quelle précision, et (2) si l'ellipse trouvée correspond bien à l'endroit du menton qui est utilisé dans le codage pour désigner la position Menton (certains

utilisent le centre du menton, d'autres le bord du menton, d'autres l'arête de la mâchoire). Bien évidemment, nous ne pouvons pas nous poser la question de la précision de notre détection par rapport au codage, et ceci pour deux raisons : la première est qu'il nous est impossible d'appréhender la variabilité de tels habitudes de codage en raison du nombre trop faible de codeurs représentés sur nos corpus ; la seconde est que nous avons éliminé ce type de variabilité de notre étude dans le section II.1 (p. 36) en nous restreignant à un codage relativement académique. En revanche, il est possible d'évaluer la précision morphologique des zones détourées.

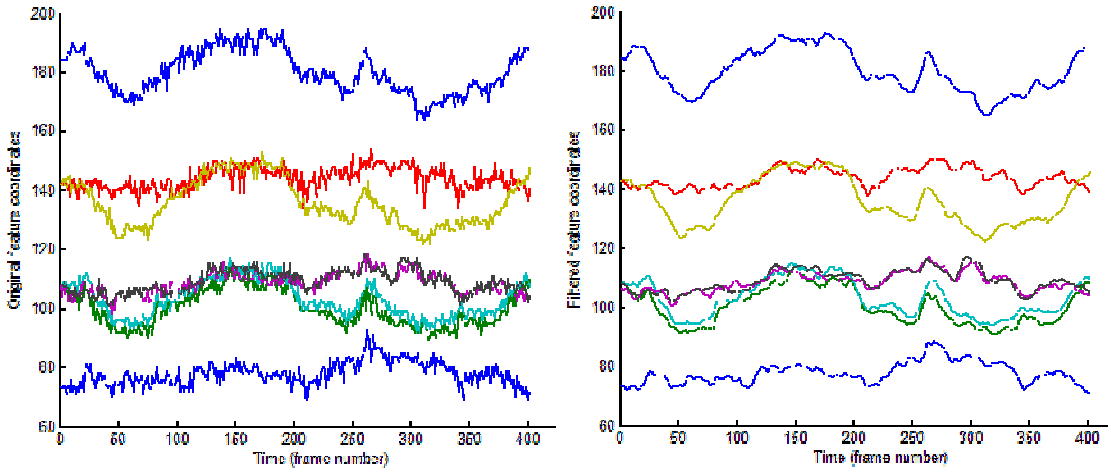


Figure III-25 : évolution de la sortie du C3F avant (à gauche) et après (à droite) filtrage de Kalman.

Afin d'évaluer cela, nous proposons donc de tester la qualité de la spécification morphologique des zones de pointage indépendamment de tout codeur. Cela a un énorme avantage : ce test s'effectue sur la base d'un grand nombre d'images de faciès qui ne représente pas forcément des codeurs, et qui sont donc beaucoup plus simples à collecter. De plus, tous les âges, ainsi que la gente masculine y sont représentés, alors que les codeurs certifiés qui participent à nos expériences sont principalement des jeunes femmes. BioId [171] est une base de données publique contenant un grand nombre de visages dans des orientations et des éclairages variables. Nous appliquons donc simplement les algorithmes du C3F et de tracé des ellipses qui nous servent de référence pour la définition des zones de pointage. Nous en apprécions ensuite la qualité sur 82 images. Evidemment, cette appréciation de la définition des ellipses est subjective, puisqu'il n'existe pas de vérité-terrain. Ainsi, plutôt que de donner une précision chiffrée dont le protocole d'obtention peut toujours être discuté, nous préférons fournir en illustration dans la Figure III-26 les situations les plus critiques. Avec un ordre de parcours en ligne, nous avons, pour ces exemples : la zone du Menton est trop basse pour tous les exemples à l'exception des images 8 et 9. La zone Pommette des exemples 6, 8 et 10 (petite taille et excentrée), 7 (excentrée), 9 (trop grande taille et recouvrement de la zone Bouche) et 12 (petite taille) est aussi inadaptée.

Il faut cependant reconnaître que ces résultats sont majoritairement bons. Cela signifie simplement que les zones de pointage, qui ont été géométriquement déterminées à partir des vidéos d'une seule codeuse se sont très bien généralisés (cf. Figure III-27), et ce, malgré la présence d'écharpe, de lunette, de barbe, la fermeture des yeux, ou de différents types ethniques. Notons enfin que certains visages sont correctement traités dans certains cas (Figure III-27.3) mais pas dans d'autres (Figure III-26.6), ce qui semble indiquer que le principal problème résulte de l'orientation du visage par rapport à la caméra.

Figure III-26 : quelques cas ne donnant pas entièrement satisfaction pour la détermination des zones de pointage.

On peut cependant noter que la zone Menton pose souvent des soucis de précision, et que suivant la manière dont la Position Gorge est codée, il pourrait y avoir confusion.

De plus, une fois que ces briques algorithmiques seront couplées à leur alter-ego sur la reconnaissance labiale, le contour des lèvres sera connu avec suffisamment de précision pour raffiner encore énormément les zones de pointage correspondant aux Positions Menton et Bouche. Notons aussi qu'il existe des travaux sur la segmentation de l'arc de la mâchoire [52].

Finalement, un **apprentissage actif** [69] est un moyen d'amélioration tout à fait envisageable, même si les résultats obtenus ne le rendent pas obligatoire dans l'élaboration d'un premier prototype. A titre indicatif, voici un possible fonctionnement pour un tel apprentissage : Dans un premier temps, les zones de codages sont spécifiées de manière géométrique, de telle sorte que pour tout codeur, le système est capable de performance minimal, éventuellement suffisante si la morphologie de l'individu se trouve correspondre à la spécification géométrique, mais pas nécessairement. Ensuite, Au fur est à mesure que le codeur pointe différentes zones de codage, celles-ci sont spécifiées et leurs localisations géométriques affinées. Le principal avantage d'une telle méthode par rapport à une méthode s'affranchissant tout de suite de la définition géométrique pour la mise en place d'une définition par apprentissage classique tient dans (1) la souplesse d'utilisation, car le temps d'apprentissage peut être long avant de pouvoir commencer à fonctionner, (2) la capacité de généralisation à différentes morphologies, (3) le rapport qualité de résultat/simplicité.

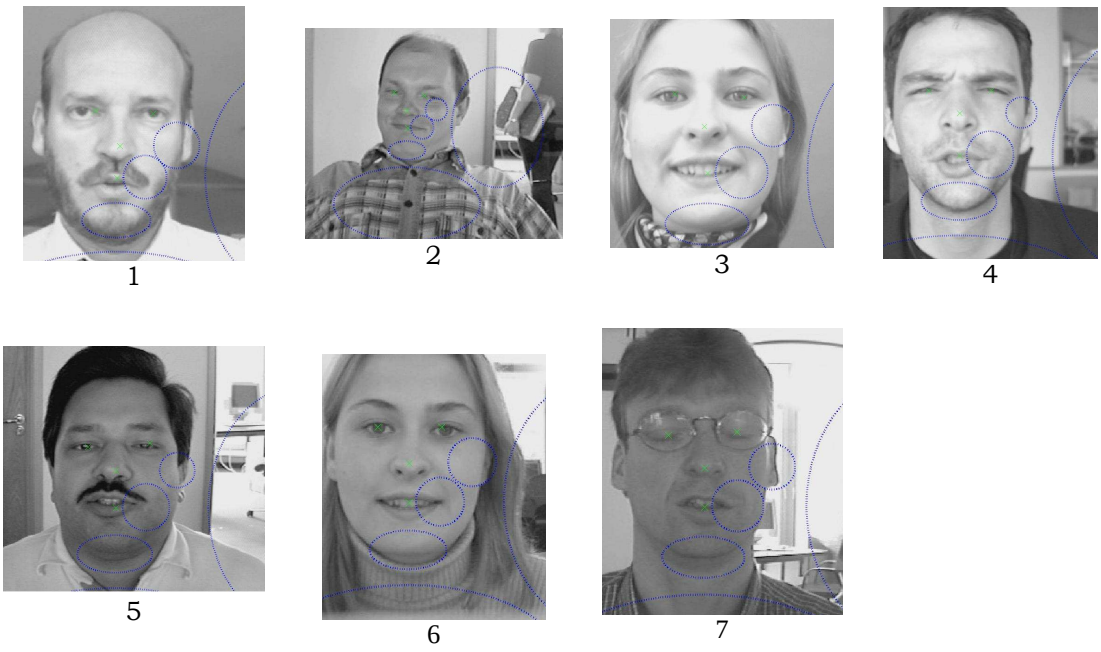


Figure III-27 : quelques cas où la détermination des zones de pointage est efficace.

Enfin, il convient de rappeler que ces résultats sont obtenus sur un test plus sévère que les situations pour lesquelles l'algorithme est sollicité. En effet :

– Le but de ce test est de confronter l'algorithme à un grand nombre de morphologies distinctes. C'est pourquoi nous avons utilisé une base de données publique. D'une manière générale, ce genre de base de données est constitué d'images. Or dans son utilisation courante, nous le confrontons à des séquences vidéo. En conséquence, le filtrage de Kalman, qui stabilise de manière importante les résultats en les conditionnant avec l'information des trames

passées de la vidéo ne peut pas être utilisé. Le test sur la base de données BioId donne une borne inferieure du résultat réel.

– La base de donnée BioId contient des visages pour lesquels l'orientation/la résolution/ l'éclairage sont moins contraints que les séquences sur lesquelles nous travaillons. A cet égard aussi, le test est sévère.

III.5 Conclusion du chapitre

Dans ce chapitre, nous avons présenté les différents corpus de données auxquels nous ferons référence dans le reste de ce document. Ensuite, nous avons discuté et évalué les algorithmes de traitement d'images que nous proposons pour premiers traitements d'une séquence d'entrée sur un terminal TELMA. Chaque image de la séquence est traitée individuellement afin de :

– Segmenter la main au moyen d'un triple seuillage dans l'espace de Mahalanobis.

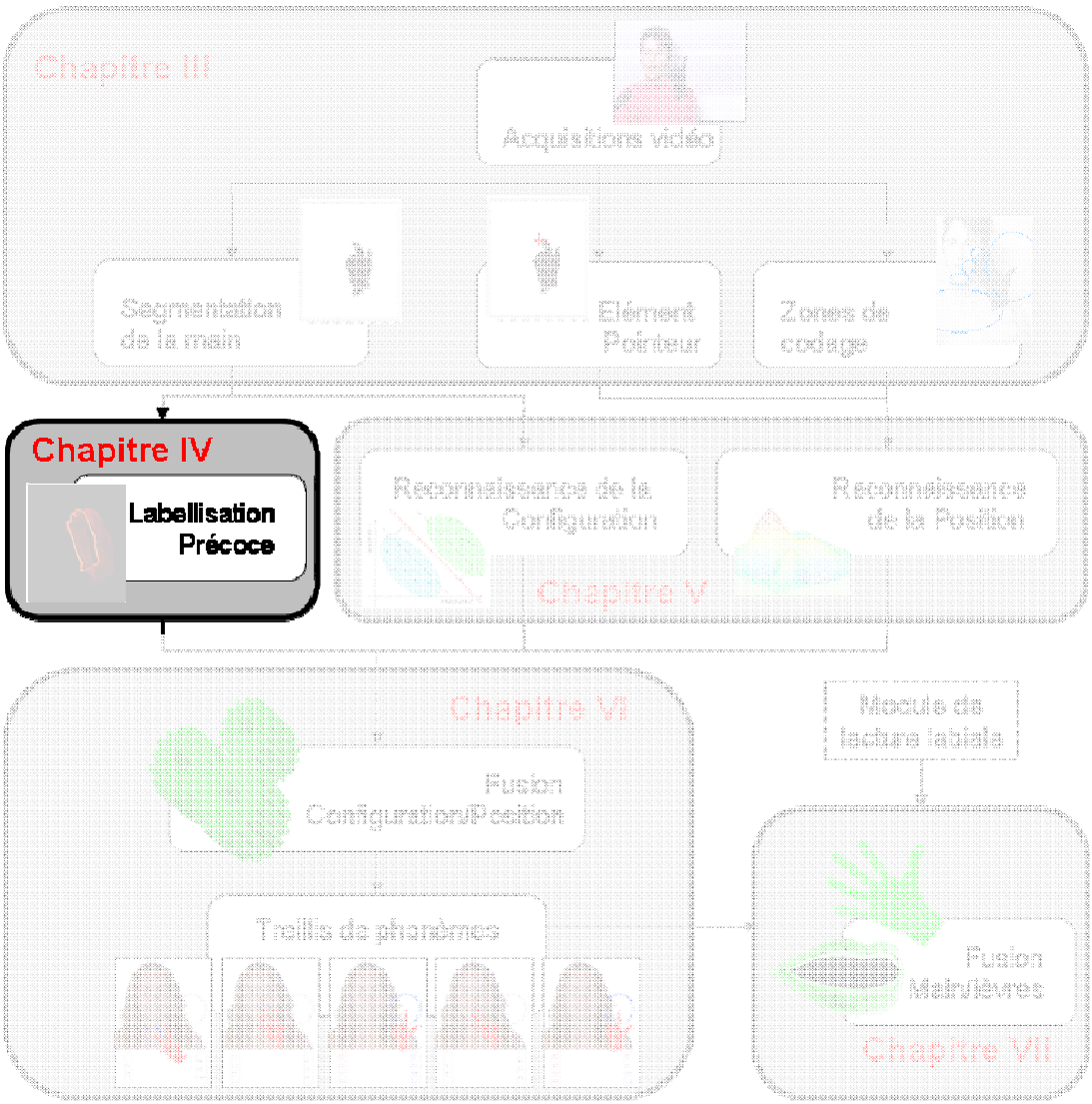
– De déterminer un point du masque binaire de la main segmentée correspondant au doigt pointeur. Ceci est effectué en utilisant un système expert basé sur (1) des considérations de codage du LPC et (2) des considérations morphologiques.

– De déterminer les zones de pointage des Positions du LPC au moyen d'un système expert basé sur la géométrie du visage. La description géométrique du visage se fonde sur la connaissance de la position dans l'image de sa boite englobante, des yeux, du nez et de la bouche. Ces positions sont fournies par l'algorithme du C3F et elles sont régularisées au cours du temps par un filtre de Kalman unidirectionnel.

L'algorithme de segmentation de la main que nous proposons remplit sa fonction dans les conditions d'acquisition des vidéos que nous avons fixées. Néanmoins, ce type de problème est encore ouvert dans le cas de conditions moins contrôlées. A terme, il serait intéressant de pouvoir remplacer l'algorithme de segmentation de la main par un autre, capable de performances équivalentes sur des images de qualité plus faible, ou même sur des vidéos sur lesquels les codeurs ne portent pas de gants. De tels algorithmes devront de plus être suffisamment efficaces pour permettre un traitement temps-réel après optimisation.

La morphologie du visage a été sommairement étudiée afin de pouvoir permettre la spécification des zones de pointage, dans le cas d'un codage académique. Une étude plus complète permettrait de (1) paramétrer un système adaptatif et (2) d'élaborer une méthode permettant un apprentissage semi-supervisé. Cela serait intéressant pour spécifier la morphologie et la manière de coder et ainsi traiter un LPC beaucoup plus naturel. Nous pensons que ce sont les deux aspects qu'il serait le plus intéressant d'améliorer afin de mettre en place un système de décodage plus robuste.

CHAPITRE IV
LABELLISATION PRECOCE



L'objectif de la **Labellisation Précoce** est de permettre une classification des images de la vidéo en **I**mages **C**ibles (IC) d'une part, et en **I**mages de **T**ransition (IT) d'autre part. Les IC correspondent à l'atteinte d'une cible gestuelle nécessaire au décodage du LPC alors que les IT sont les images de tous les gestes intermédiaires pour passer d'un geste statique à un autre, et ne sont *a priori* pas utiles au décodage.

Nous rappelons que pour des raisons mécaniques, les changements de Configurations et de Positions ne sont pas toujours synchronisés. En effet, dans l'action de déployer les doigts, la Configuration est souvent réalisée avant l'atteinte de la Position (surtout quand celle-ci se matérialise par un contact avec le visage : Positions Pommette, Bouche et Menton). Il y a donc de nombreux cas pour lesquels l'atteinte de la Configuration est en décalage avec l'atteinte de la Position, ce décalage pouvant être plus important que celui séparant l'acquisition de deux images de la vidéo.

Dès lors, nous avons introduit à la section II.5 (p. 51) les ICC (Images Cibles de Configuration), images correspondant à l'atteinte de la cible gestuelle dans le mouvement de changement de Configuration seulement, et les ICP (Images Cibles de Position), images correspondant à l'atteinte de la cible gestuelle dans le mouvement de changement de Position seulement. Les ICC et les ICP ne coïncident pas toujours et nous avons défini les IC comme l'ensemble des ICP et des ICC. Dès lors, par souci de simplicité, appelons **ITC** (**Image de Transition par rapport à la Configuration**) toute image qui n'est pas une ICC et **ITP** (**Image de Transition par rapport à la Position**) toute image qui n'est pas une ICP. L'ensemble des IC est l'union des ICP et des ICC, l'ensemble des IT est constitué des images qui sont à la fois des ITC et des ITP :

$$\{IC\} = \{ICC\} \cup \{ICP\}$$

$$\{IT\} = \{ITC\} \cap \{ITP\}$$

D'après la stratégie que nous avons mise en place au chapitre II, nous allons nous intéresser à la classification entre ICC et ITC d'une part, puis à la classification entre ICP et ITP d'autre part. Ce n'est qu'après les études séparées de ces deux flux que nous allons chercher à reconstruire le geste complet (cf. chapitre VI, p. 182). Comme les traitements que nous proposons sont souvent similaires pour les flux de Configurations et de Positions, nous les décrirons sans spécifier le flux. Ainsi, nous désignons par ICX les images cibles de Position ou de Configuration (sans préciser, mais sans faire référence à l'union des deux, à savoir les IC). Et nous ferons de même pour les images de transition avec la notation ITX :

$$\{ICX\} = \{ICC\} \text{ ou } \{ICP\}$$

$$\{ITX\} = \{ITC\} \text{ ou } \{ITP\}$$

La difficulté est de trouver une méthode permettant d'effectuer ces deux classifications avant d'avoir analysé l'intégralité du contenu de chaque image

ainsi que leur séquençage temporel. Ceci est possible en se basant sur trois hypothèses :

– Les images correspondant aux ICX sont suffisantes pour résumer l'ensemble du mouvement de chacune des composantes gestuelles du LPC.

– Pour chaque geste, il existe une image sur laquelle sa **Configuration** est parfaitement représentée. Pour chaque geste, il existe une image sur laquelle sa **Position** est parfaitement représentée.

– Le mouvement de la main ralentit de manière significative **mais sans forcément s'arrêter complètement** lors de l'atteinte de la cible. L'ICX est donc caractérisée par une diminution du mouvement de la main. Selon qu'il s'agit des ICC ou des ICP, le mouvement de la main en question n'est pas le même : dans le premier cas, c'est le mouvement des doigts qui nous intéresse, alors que dans le second, c'est le mouvement de la main dans son ensemble.

Le but de ce chapitre est de décrire le processus permettant d'extraire les informations cinématiques de bas niveau qui permettent de repérer ces ralentissements significatifs du mouvement de chacune des composantes gestuelles. A partir de là, il est possible d'inférer les instants (ou les images) correspondants aux ICX (et de définir les ITX comme les images n'étant pas des ICX). Pour cela nous approfondissons les travaux que nous avons publiés dans [C4] et [J4].

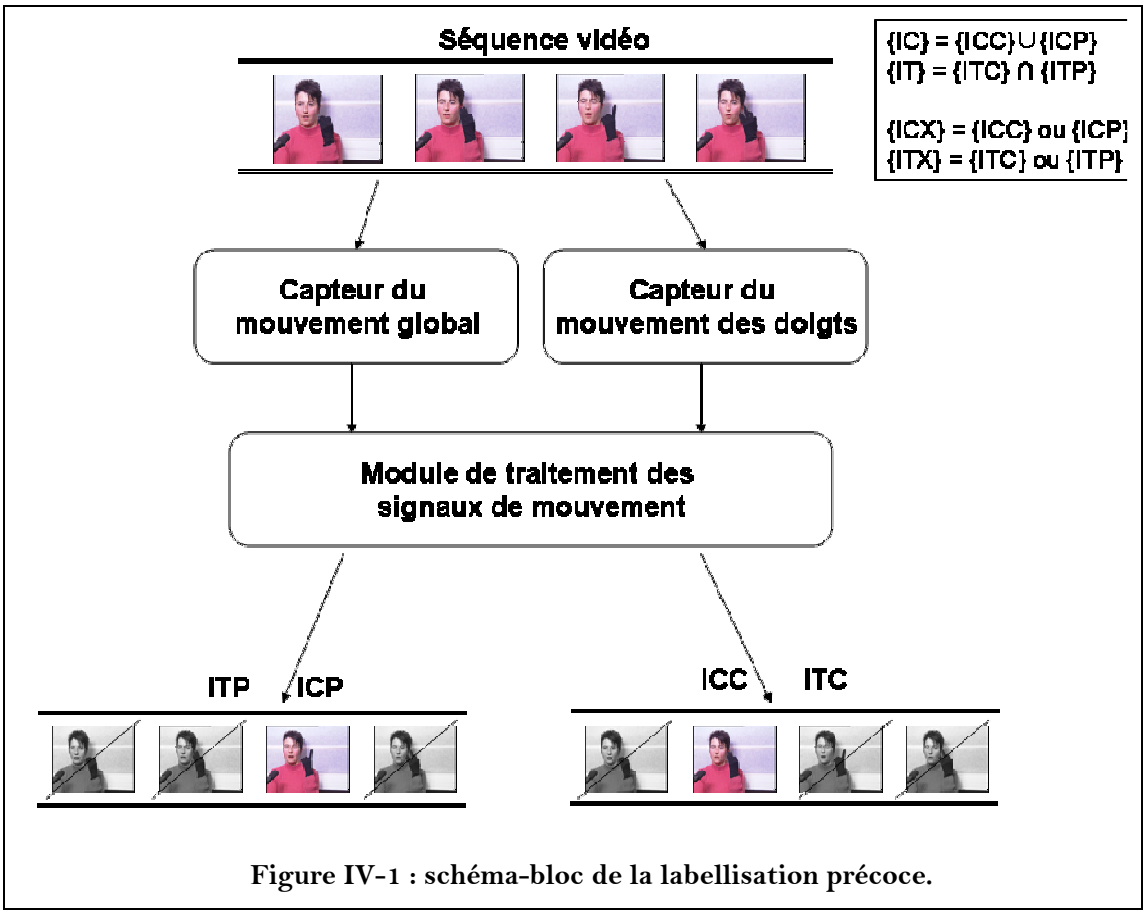


Figure IV-1 : schéma-bloc de la labellisation précoce.

Examinons maintenant le double mouvement de la main : le mouvement lors d'un changement de Position est global, alors que le mouvement lors d'un changement de Configuration est localisé aux doigts. Ce dernier, étudié sur une image binaire, se traduit par une déformation (mouvement non rigide) du contour de la main. Nous allons commencer par étudier séparément chacun de ces deux mouvements. Cependant, nous utilisons la même méthode d'analyse pour chacun des deux. Seul le "capteur" de mouvement proprement dit est différent afin d'être adapté à leur nature différente, mais le traitement appliqué aux signaux issus de ces deux capteurs est identique (cf. Figure IV-1).

Le reste de ce chapitre est structuré de la manière suivante. Dans la prochaine section est détaillé le capteur permettant l'analyse du mouvement global de la main, associé au changement de Position. La partie qui suit est consacrée à la description du second capteur, associé au mouvement des doigts et au changement de Configuration. Vient ensuite la description du traitement des signaux de mouvement afin de déterminer les ICX : quand ce traitement est appliqué au signal issu du capteur de mouvement global, cela permet d'extraire les ICP. En outre, quand le même traitement est appliqué à la sortie du capteur de déformation de la forme de la main (due au mouvement des doigts), cela permet d'extraire les ICC. Enfin, dans une dernière partie, nous évaluons et analysons les résultats de l'algorithme de labellisation proposé.

IV.1 Analyse du mouvement global de la main lors du changement de Position

Une première approche pour appréhender le mouvement global de la main associé au changement de Position est d'étudier la trajectoire du doigt pointeur dans le plan d'acquisition. Malheureusement, durant les changements de Configuration, il n'existe pas de doigt pointeur. De plus, celui-ci n'est pas connu avec précision ; il peut tout au plus être considéré comme confondu avec l'élément pointeur défini en III.3 (p. 85). Même s'il est toujours possible de déterminer un élément pointeur sur les images correspondant à une transition de Configuration, il n'est absolument pas garanti que le résultat obtenu soit cohérent. En effet, quand la forme de la main est proche d'une Configuration, l'élément pointeur pourra facilement être déterminé, et celui-ci a de grande chance d'être cohérent avec les images précédente et suivante. A l'inverse, quand la forme de main est relativement atypique, il n'est pas possible de garantir que l'algorithme déterminant l'élément pointeur donne un résultat cohérent, puisqu'il n'est pas destiné à cela. Ainsi, sur des images comme celles de la Figure IV-2, il n'est pas utile de chercher l'élément pointeur : dans de tels cas, il n'existe pas de vérité terrain, et même un humain n'est pas capable de le déterminer. De surcroît, il y a même des transitions pour lesquelles cette détermination est impossible puisque le doigt pointeur change (il passe du majeur à l'index, ou vice-versa).

La solution est donc de se servir d'un point de la main que l'on peut toujours déterminer, et qui correspond à une définition morphologique particulière.

Comme aucun doigt déployé n'est commun à toutes les Configurations, ce point doit appartenir à la paume de la main. Le problème c'est qu'elle est proche du poignet. Or, la plupart des mouvements de la main de faible amplitude (pour se déplacer entre deux Positions proches l'une de l'autre, comme les Positions Menton et Bouche ou Bouche et Pommette) sont des mouvements d'adduction/abduction de la main, c'est-à-dire des rotations autour d'un point situé à proximité du scaphoïde et de la tête cubitale. En conséquence, plus le point à partir duquel on étudie le mouvement est près de ce centre de rotation, moins les petits mouvements sont décelables. De plus, il est difficile de déterminer précisément un tel point.

(a) (b)

Figure IV-2 : cas où il n'est pas possible de déterminer un élément pointeur.

Nous pouvons aussi envisager d'utiliser plusieurs points, et de fonder la décision de mouvement sur leur ensemble. Bien que probablement plus stable, une telle méthode est plus compliquée et aucune garantie n'existe quand à l'amélioration des performances. En conséquence, nous nous contentons d'un seul point.

Finalement, nous utilisons la trajectoire du **c**entre de **g**ravité (**CG**) afin de déterminer le mouvement global de la main. Ce choix est discuté dans l'évaluation de l'algorithme. Afin de pallier certaines erreurs d'arrondi, ou d'imprécision des calculs, et afin de régulariser la trajectoire, nous appliquons un filtre de Kalman à cette trajectoire (cf. appendice C.2 p. 293). Cela permet d'obtenir les signaux représentés sur la Figure IV-3.

Ces signaux représentent la trajectoire de l'ordonnée (en haut) et de l'abscisse (en bas) du CG en nombre de pixels en fonction du numéro des images de la séquence. L'origine du repère associé à la mesure du déplacement est le coin supérieur gauche de l'image. La connaissance de cette trajectoire fournit toutes les informations cinématiques nécessaires pour déterminer les instants de ralentissement relatifs de la main à l'approche d'une cible de Position. En effet,

les images ou la position du CG évolue beaucoup sont des images où le mouvement est important, et inversement. Ainsi, nous constatons que les extrema de chacune des deux courbes apparaissent généralement pour les mêmes instants, c'est-à-dire que le ralentissement dû à l'atteinte d'une cible de Position est repérable en fonction du mouvement selon les deux coordonnées. Par exemple le long plateau central correspond à une pause dans le codage alors que la main est en Position Côté. De même le dernier maximum sur le mouvement vertical (associé à un minimum sur le mouvement horizontal) correspond à l'atteinte de la Position Gorge (c'est la Position qui correspond à la zone de pointage la plus éloignée du bord supérieur de l'image et la plus proche du bord gauche de l'image). Enfin, l'aller-retour rapide juste après le plateau central est symptomatique de la succession de deux gestes pour lesquels la Position est Bouche.

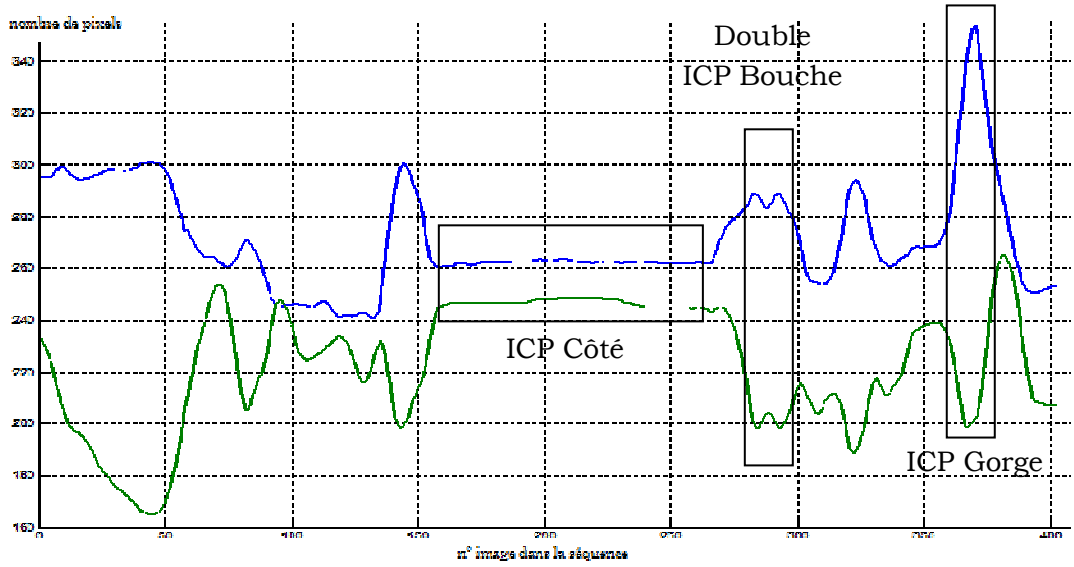


Figure IV-3 : décours temporel de x_{CG} (en bas) et de y_{CG} (en haut), coordonnées du CG de la main, lors du codage de la phrase 158 du corpus ETTRAN N : "*Nous traquions bien Euler durant son footing urbain*".

Comme la suite des traitements (la labellisation) nécessite de travailler sur un signal homogène à une vitesse, la dérivée de cette trajectoire est transmise au module suivant. D'une manière générale, ce signal de vitesse est beaucoup plus "haché", et donc relativement difficile à interpréter. En conséquence, nous représenterons toujours la trajectoire des coordonnées du CG à la manière de la Figure IV-3, même si l'information pris en compte par la suite est sa vitesse.

IV.2 Analyse de la déformation de la main lors du changement de Configuration

Mettre en place un capteur permettant de résumer l'équivalent d'une trajectoire pour un objet qui se déforme est une chose beaucoup plus difficile. Certaines méthodes classiques, telles que le **differential block matching** [38] ou encore

les **model based methods** [39] ne sont pas adaptées à la non-rigidité de la déformation considérée. De nombreux algorithmes de type **Deformable Templates** [36], [41], [42] sont plus efficaces mais souffrent cependant de trois inconvénients qui ne les rendent pas applicables à notre cas :

– La complexité calculatoire de ces algorithmes est trop importante.

– Ils nécessitent la mise en place d'un modèle de main et d'un modèle de déformation de celle-ci. Pour être généralisables, ces deux modèles nécessitent un corpus d'apprentissage de grande taille dont nous ne disposons pas.

– La mise en place d'un modèle prenant en compte la déformation projective de la main sur le plan d'acquisition (ce qui permettrait d'avoir un modèle seulement 2D de la main et de son mouvement) est là encore un problème complexe d'un point de vue scientifique comme calculatoire.

En conséquence, nous proposons de mettre en place notre propre algorithme d'analyse du mouvement des doigts. Cet algorithme est basé sur une approche bio-inspirée qui s'appuie sur certaines propriétés du fonctionnement du système visuel des vertébrés.

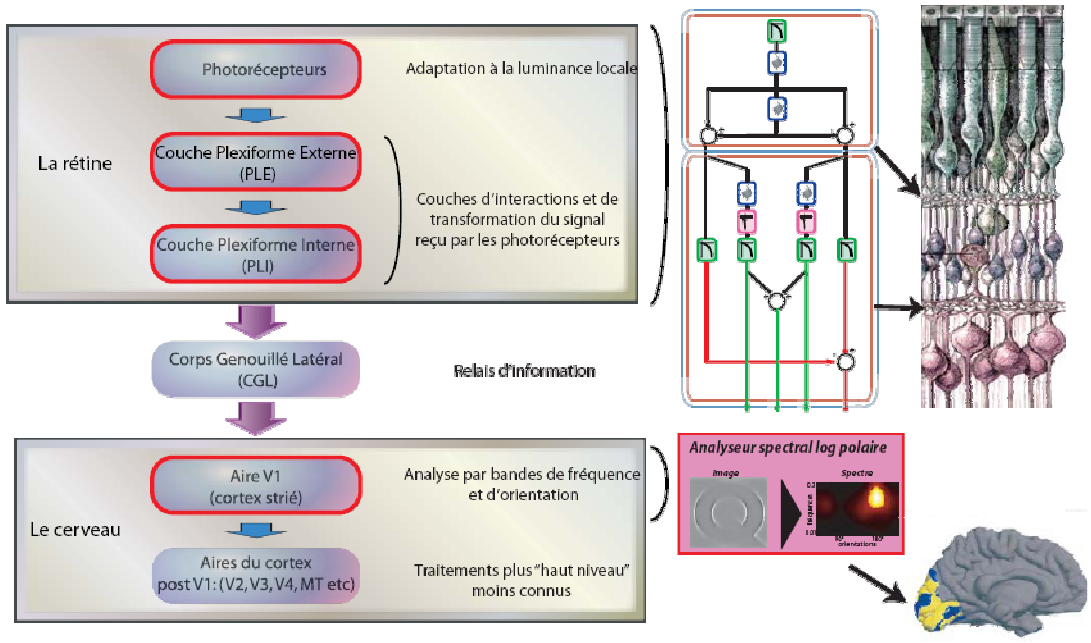


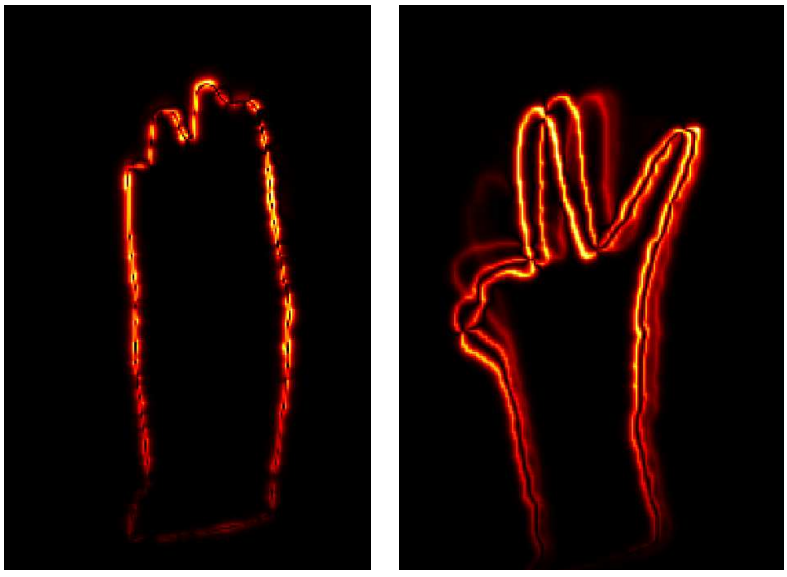
Figure IV-4 : structure simplifiée d'une rétine d'un point de vue traitement du signal, d'après [155].

La rétine des vertébrés est un système complexe et puissant qui est une source d'inspiration à plusieurs niveaux en vision par ordinateur [155], [156], [154]. D'un point de vue algorithmique, la rétine, en plus d'être un capteur d'images, réalise une succession de traitements vidéo [155]. D'un point de vue ingénierie, ces traitements sont autant de modules reliés entre eux et interprétables en termes de fonctions de transfert (Figure IV-4). Chaque module a une

fonctionnalité particulière, telle que lisser les variations d'illuminations, détecter les contours, extraire le mouvement, etc.

Parmi tous les modules de la rétine, nous nous intéressons particulièrement au filtre modélisant la **Couche Plexiforme Interne** (ou **IPL** pour **Inner Plexiform Layer**). Il s'agit d'un filtre qui renforce les gradients de mouvement, et particulièrement ceux qui sont perpendiculaires à la direction du mouvement. Ainsi, le filtre IPL réagit particulièrement aux contours des objets en mouvement, et y reste plus longtemps sensible. Cela peut facilement être interprété en terme de persistance rétinienne : plus un objet passe rapidement dans le champ visuel, plus celui-ci est vu de manière floue, mais aussi plus il est facile de se rendre compte qu'il y a eu un déplacement dans le champ visuel. En simplifiant, l'IPL peut être modélisé par un filtre temporel passe-haut (pour une description plus détaillée de son fonctionnement, se référer à [155]).

En évaluant la quantité de persistance rétinienne qu'il y a dans une "rétine artificielle" quand un objet se déforme à une vitesse variable dans son champ visuel, il est possible d'avoir une idée de la quantité de mouvement de l'objet. Cela peut parfaitement être appliqué à notre problème d'analyse du mouvement de la main. Comme cela apparaît sur la Figure IV-5, il est possible d'utiliser la persistance rétinienne pour décider si l'image est à peu près stable (elle a une chance raisonnable d'être une cible) ou non (il est vraisemblable qu'il s'agisse d'un mouvement de transition).



(a) (b)

Figure IV-5 : sortie du filtre IPL pour (a) une IC (très peu de mouvement) (b) une image de transition (mouvement important).

Notre objectif est d'utiliser cette fonctionnalité particulière de la rétine qu'est le filtre IPL afin de l'intégrer dans un algorithme dédié à notre problème. Nous appelons le capteur de mouvement qui en résulte **Filtre Rétinien Dédié (FRD)**.

Le FRD [J4] est constitué de plusieurs éléments mis en cascade, comme cela est indiqué sur la Figure IV-6.

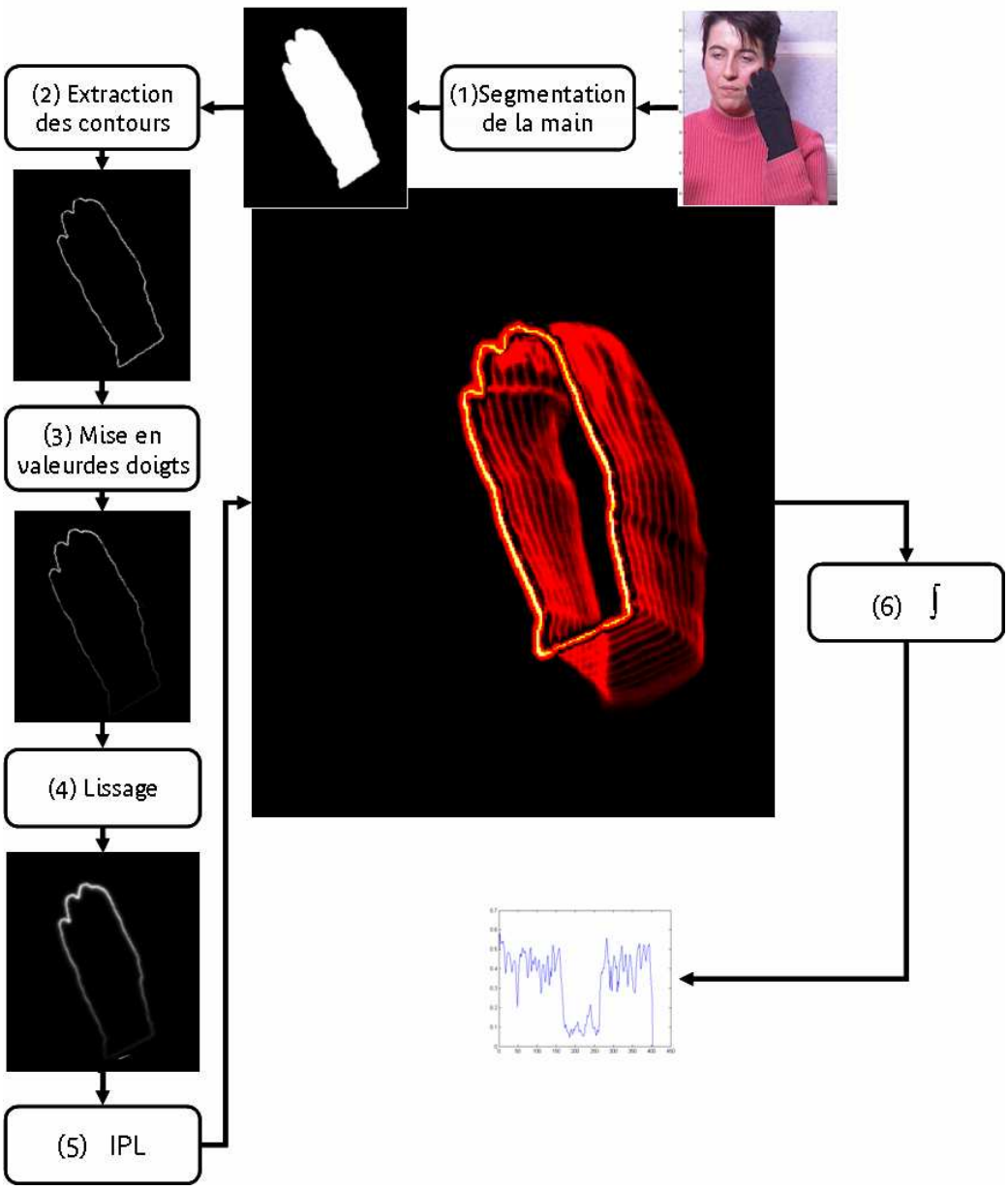


Figure IV-6 : schéma bloc du FRD.

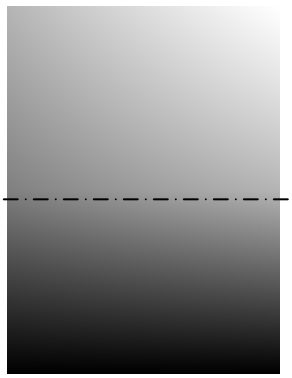
(1) **Segmentation de la main** : il s'agit des algorithmes détaillés dans la section III.2 (p. 69). A la fin de ce processus, une image binaire de la forme de la main est disponible. Cette image binaire est normalisée et retournée (la main est alors positionnée verticalement dans l'image comme sur la Figure IV-5) de manière à ce que la base du poignet soit immobile sur toutes les images. Ainsi, on s'affranchit du mouvement global dû au changement de Position. Evidemment, cette suppression du mouvement global n'est pas parfaite. En effet, la rotation de la main est inexacte et dépend de la répartition des axes principaux d'inertie (qui varie en fonction du mouvement du poignet et de la répartition des doigts) ;

de plus, l'élément invariant du mouvement du poignet est un point plus précis que la base du poignet que nous utilisons de manière approximative (ce point réel se trouve entre le scaphoïde et la tête cubitale et sa position est trop difficile à inférer sur l'image binaire). Cependant, grâce à ce système, il est malgré tout possible de s'affranchir du mouvement lié au changement de Position dans à peu près les mêmes proportions que l'étude du mouvement global est affranchie du biais induit par un mouvement de changement de Configuration.

(2) **Extraction de contours** : Il s'agit simplement de récupérer l'image du contour de la main à partir de l'image binaire issue de l'étape de segmentation. Nous utilisons un opérateur de soustraction comme préconisé dans [40]. Ensuite, *L*, la longueur du contour de la main est calculée et stockée.

(3) **Pondération digitale** : il s'agit d'un masque de poids que l'on applique sur l'image de contour, de manière à mettre en valeur les zones de l'image où il est plus probable d'avoir un morceau de contour associé à la zone des doigts. Ainsi cette zone sera plus sensible à la persistance rétinienne. Les valeurs numériques pour le masque de poids ne sont pas optimisées et n'ont pas fait l'objet d'une étude approfondie. Ceci est discuté dans la section dédiée aux évaluations. Les valeurs numériques du masque de pondération utilisé sont données sur la Figure IV-7b.

(4) **Lisseur[2]** : il s'agit d'une approximation d'un filtre gaussien. Un tel filtre apparaît comme prétraitement de l'IPL dans une rétine de vertébré [155].



(a) représentation en niveau de gris du masque de pondération (le foncé indiquant un poids faible, et le clair un poids fort)

Moitié supérieure : le poids $w(x,y)$ évolue en fonction de la racine carrée de la distance entre le projeté de (x,y) sur la droite de vecteur directeur ($Ymax.\vec{y}/2+Xmax.\vec{x}$) et le coin inférieur gauche.

$w(x,y)= 0.5$ si $y = 0$

$w(x,y)= 1$ si $(x, y) = (Xmax, Ymax)$

Moitié inférieure : évolution linéaire de $w(x,y)$ suivant le vecteur $\vec{y}$.

$w(x,y) = 0$ si $y = 0$

$w(x,y) = 0.5$ si $y = Ymax/2$

(b) Expression du masque pour chaque pixel $p(x,y)$. Le coin inférieur gauche sert de référence, et ($Xmax$, $Ymax$) sont les dimensions de l'image

Figure IV-7 : masque de pondération digitale.

[2] Pour des raisons de confidentialité liées à la propriété intellectuelle de ces algorithmes, nous ne les détaillons volontairement pas.

(5) **IPL** : le module de persistance rétinienne proprement dit, qui est introduit plus haut. Le détail de son implémentation est fourni dans [155].

(6) **Opérateur de Sommation** : il intègre la sortie du filtre IPL, dans le but de faire une évaluation globale de la quantité de "flou" sur l'image. Cette dernière peut directement être interprétée en termes de mesure de l'énergie du mouvement. En la divisant par la longueur du contour, on obtient une mesure normalisée que l'on peut comparer à une mesure de vitesse :

$$QuantificationMouvement\left(I_t\right)=\frac{1}{L}\cdot\sum_{x,y}SortieIPL_t\left(x,y\right)$$

où I_t représente l'image courante à l'instant t, L est la longueur du contour et $SortieIPL_t$ (x,y) représente la valeur du pixel de coordonnées (x, y) dans l'image résultant des traitements précédents. Le Signal *QuantificationMouvement* est représenté sur la Figure IV-8.

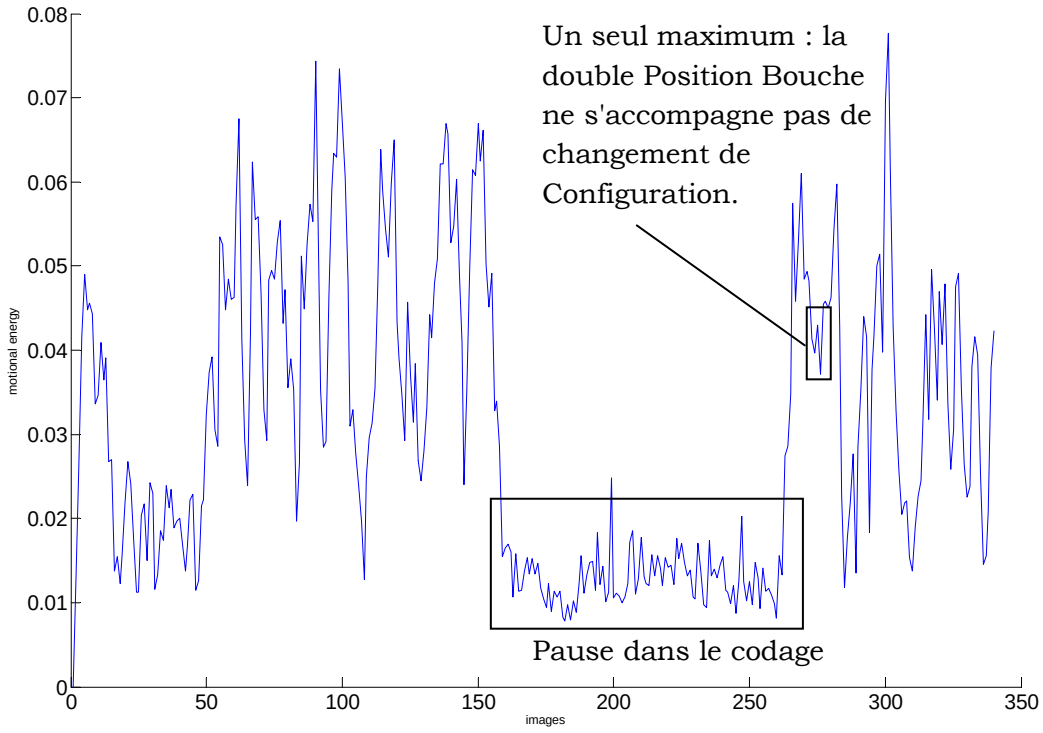


Figure IV-8 : extrait du décours temporel du signal *QuantificationMouvement* en sortie du FRD pour la phrase 158 du corpus ETTRAN N. L'axe des ordonnées n'a pas de signification physique précise, mais les valeurs sont positivement corrélées à une mesure de la quantité de mouvement. Ainsi, les maxima du signal correspondent à des vitesses de déformation maximales de la main, et les minima, à des déformations minimales.

Ce signal correspond au mouvement de changement de Configuration pour la séquence dont le changement de Position est illustré sur la Figure IV-3. Nous retrouvons donc la même pause dans le mouvement de codage au milieu de la

phrase. Durant ce plateau, la Configuration de la main reste inchangée, ce qui signifie que la forme de la main varie très peu. Il est donc naturel que la mesure de déformation soit elle aussi faible. De même, si nous analysons le mouvement au niveau des images pour lesquelles l'aller-retour vers la Position Bouche a été repéré sur la Figure IV-3, nous constatons qu'il n'y a qu'un seul minimum d'énergie. Cela signifie qu'il n'y a qu'une seule décélération importante de la main, donc une seule ICC, et ensuite, il n'y a qu'une seule accélération de la main (afin que celle-ci puisse le plus rapidement possible atteindre une autre forme correspondant à une autre Configuration). Nous pouvons donc en déduire que deux gestes ayant des Positions identiques mais des Configurations différentes se sont succédés. Cependant, avant de pouvoir reconstruire ces deux gestes, nous devons mettre en place une méthode permettant de faire automatiquement l'analyse que nous venons de faire. C'est l'objet de la section suivante.

IV.3 Analyse du mouvement et labellisation des cibles

Le déplacement du CG et la quantité de persistance rétinienne représentent les sorties de nos deux capteurs de mouvement global et de déformation de la main. Ce sont deux indices fournissant une certaine information sur un phénomène caché à savoir le mouvement de la main. Pour chacune des deux composantes du mouvement de changement de geste, notre objectif est de retrouver en fonction de ces indices, quelle image est une ICX et quelle image est une ITX.

L'hypothèse qui guide cette classification est que les forts minima de la sortie de chacun des deux capteurs de mouvement correspondent à des instants importants de la trajectoire de la main, alors que les minima de plus faible amplitude sont le résultat des différents bruits auxquels est soumis le système.

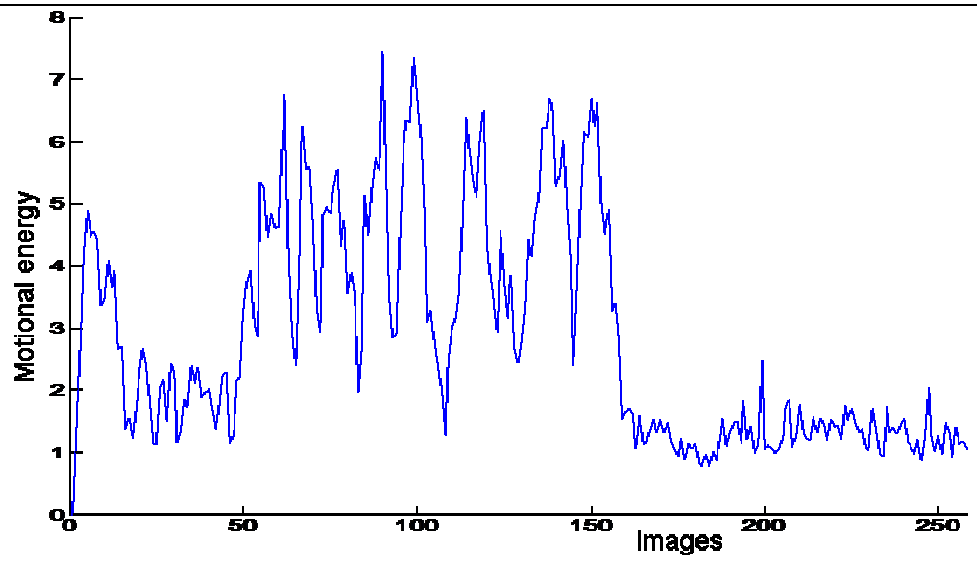


Figure IV-9 : exemple de signal (celui-ci est issu du FRD) traité selon l'étape 1.

Afin d'analyser la sortie de chacun des deux capteurs, nous proposons d'utiliser un seul et unique filtre. Celui-ci a été dessiné et paramétré sur la sortie du FRD,

qui est beaucoup plus irrégulière, et beaucoup plus difficile à traiter. Il a ensuite été utilisé pour filtrer la vitesse du CG de la main. Ainsi, le filtre proposé est assez général pour distinguer les ICX des ITX, quelque soit l'origine du capteur. Voici une description de son fonctionnement :

Etape 1 : La trajectoire est normalisée pour garantir que son amplitude est codée entre les valeurs 0 et 10 (cf. Figure IV-9).

Etape 2 : Le signal subit un filtrage par convolution d'un masque gaussien (Figure IV-10). Ensuite, les extrema du signal filtré sont ramenés à leur valeur d'avant filtrage. Cela permet de supprimer les extrema de faible amplitude (ils sont absorbés par la convolution) sans pour autant diminuer l'importance des extrema de plus grande amplitude. Ceci est répété une seconde fois. Pour les deux filtrages, les masques de convolution sont différents. Le premier masque correspond à un lissage assez fort, mais en pratique, il peut induire un léger déphasage des maxima. En conséquence, le dernier lissage est plus faible, et permet de garder une certaine homogénéité du signal :

$$Masque_1 = \frac{1}{10} \cdot [1 \quad 2 \quad 4 \quad 2 \quad 1]$$

$$Masque_2 = \frac{1}{10} \cdot [0 \quad 1 \quad 8 \quad 1 \quad 0]$$

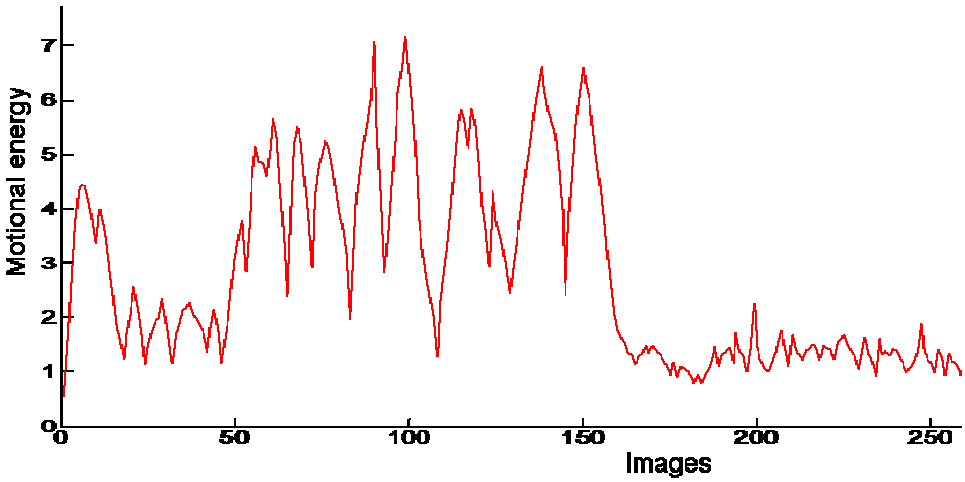


Figure IV-10 : signal de la Figure IV-9 après lissage de l'étape 2.

Etape 3 : Intuitivement, les ICX et les minima locaux sont très proches. Cependant, comme il peut rester des minima locaux n'ayant pas de signification réelle dans le signal, il n'est pas possible d'associer directement la liste des minima du signal à celle des ICX que nous recherchons : plusieurs minima peuvent ne correspondre qu'à une seule ICX (c'est par exemple ce qu'il se passe sur le plateau central illustré sur la Figure IV-8; il n'y a qu'un seul geste, mais une multitude de minima locaux). Ainsi, nous définissons des zones de stabilité correspondant à l'ensemble des instants de la trajectoire où

la quantité de mouvement est inférieure à une valeur seuil S^{immo}. Les moments de fortes transitions sont les instants de la trajectoire où la quantité de mouvement est supérieure à un seuil S^{motion} (cf. Figure IV-11). Il y a donc deux seuils à régler manuellement. Cela est discuté dans la section suivante. Afin de tenir compte de la rémanance du filtrage (retard de phase), nous définissons les zones de stabilités par hysteresis entre les deux seuils.

Etape 4 : Pour chaque zone de stabilité, **une unique** ICX est désignée. Il s'agit de l'image qui contient le moins de mouvement au sein de la plage de stabilité (cf. Figure IV-11). Finalement, nous obtenons une liste de zones de stabilité, (avec pour chacune d'elle une ICX), séparée par des images dont le mouvement est supérieure à S^{motion}. Ces images correspondent aux transitions qui séparent chacun des gestes. En plus de cette liste d'ICX, nous labellisons aussi l'ensemble des **images de transition par rapport à la Configuration/Position maximums** (ITXM), qui correspondent au maximum local de la quantité de mouvement lors de chaque transition entre deux zones de stabilité.

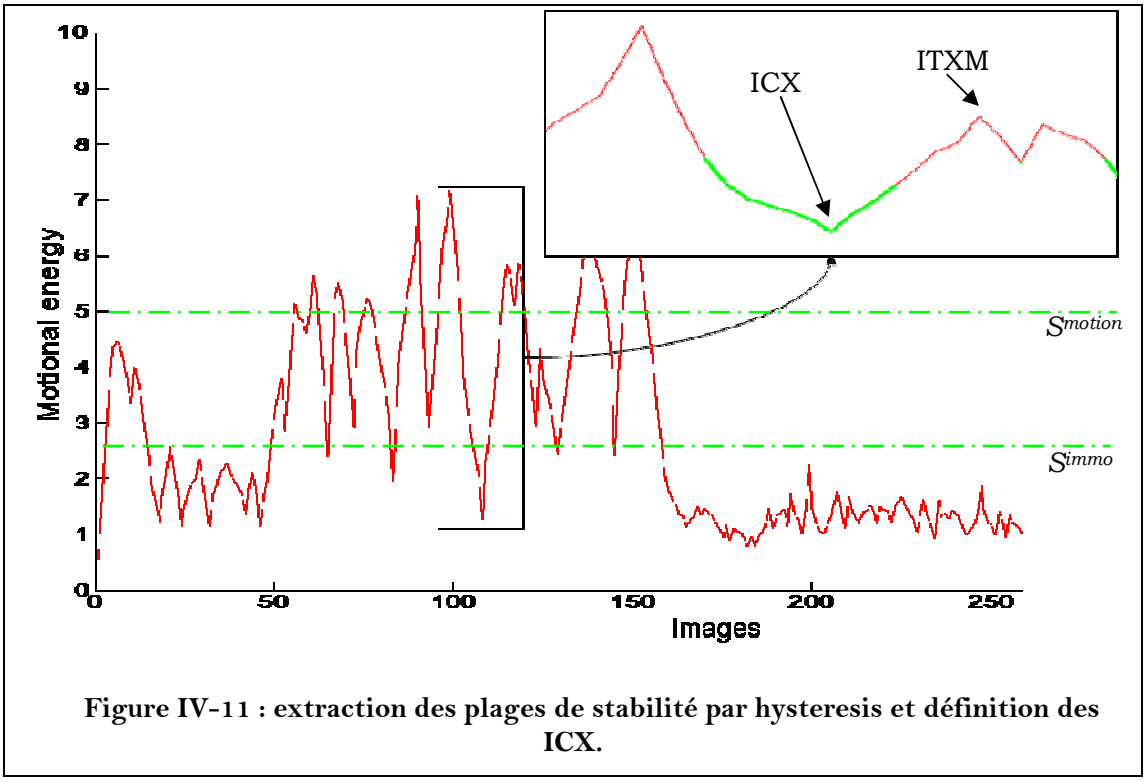


Figure IV-11 : extraction des plages de stabilité par hysteresis et définition des ICX.

Il est nécessaire de garder les images ayant le plus de mouvement (les ITXM), afin de vérifier l'alternance entre plages de stabilité et plages de transition ; il est en effet possible que la transition implicitement repérée entre deux zones de stabilité n'ait pas de raison d'être, et qu'en fait, les deux zones de stabilité ne doivent faire qu'une. Prenons un exemple. Deux ICP consécutives sont identiques et toutes les deux sont reconnues comme des Positions Bouche. Il peut s'agir :

– d'un seul geste pour lequel une transition inexistante a été repérée au milieu. Celle-ci peut être due à une erreur de segmentation déplaçant brutalement le

CG, ou a un mouvement réel qui est venu parasiter le codage, mais qui ne le modifie pas. Quelqu'en soit l'origine, si l'on considère l'ITXM associée à cette transition fictive, c'est-à-dire l'image du mouvement de transition contenant le plus d'énergie, il est probable que la Position reconnue soit sur celle-ci la Position Bouche.

– de deux phonèmes consécutifs qui doivent être codés de la même manière. Dans ce cas, le mouvement de transition qui est repéré entre les deux ICP est probablement réel. Le codeur produit un petit mouvement d'aller-retour latéral vers la zone de pointage dans ce genre de cas. Ainsi, il y a de fortes chances que l'ITXM de cette transition n'indique pas une Position Bouche, mais une Position Côté ou Pommette, ou encore une absence de Position. Dans tous les cas, il ne s'agit pas d'une cible parce que le mouvement est trop transitoire pour être considéré comme tel, néanmoins, il permet de distinguer les deux ICP semblables.

Ainsi, le contenu des ITXM est important. A terme, il peut permettre de faire la différence entre un geste artificiellement coupé par erreur en deux gestes, et une répétition. Nous ne pouvons donc pas nous limiter à l'extraction des ICX. Il faut aussi extraire les ITXM. En conséquence, nous les labellisons directement.

Comme il est beaucoup plus important de ne pas perdre de cible, que d'en détecter une là où il n'y en a pas, les seuils S^{immo} et S^{motion} sont particulièrement sélectifs. Cela peut sembler contradictoire, parce que des seuils sélectifs auront tendance à considérer qu'une image est plutôt instable, mais c'est malgré tout la stratégie la plus intéressante : toute image qui n'est pas un minimum de mouvement local n'est pas une ICX. Cependant, si la quantité de mouvement qu'elle contient est suffisamment faible, celle-ci peut malgré tout être d'une stabilité relative, et faire partie intégrante d'une zone de stabilité, telles qu'elles sont définies à l'étape 3 et pour lesquelles une unique ICX est définie à l'étape 4. En pratique, les signaux sont tels qu'il n'est pas possible de rater une plage de stabilité complète, c'est-à-dire que l'intégralité des images qui la constituent possède une trop grande quantité de mouvement, malgré leur stabilité d'un point de vue gestuel ; ou alors cela signifie que le code LPC n'est pas réalisé correctement, et qu'il est trop "mâché". La seule possibilité pour rater une zone de stabilité est de fusionner par inadvertance deux zones de stabilité successives, correspondant à deux gestes différents consécutifs, mais pour lesquels une seule ICX va être détectée. Cette erreur est simplement due au fait qu'aucune transition assez forte n'a été repérée entre les deux plages de stabilité. Il est donc très important d'être restrictif sur la définition des zones de stabilité. Cela a pour conséquence de ne pas faire perdre de cible, mais a pour inconvénient de couper en deux (ou plus) certaines zones de stabilité, (quand celles-ci contiennent un mouvement un tout petit peu trop important). Ainsi, il peut arriver que deux ou plusieurs plages consécutives où la main est relativement stable représentent le même geste. Dans un tel cas, il est possible de vérifier l'image correspondant au maximum de mouvement entre deux telles zones de stabilité, tel que nous venons de l'expliquer. Ainsi, le fait d'extraire les ITXM

permet de faire cette vérification et pallie le défaut d'un algorithme qui considère que le moindre mouvement est une transition. En revanche, l'analyse des ITXM ne permet pas de compenser le défaut contraire (un algorithme où le comportement par défaut est de considérer l'image comme stable). C'est pourquoi, nous préconisons d'avoir des seuils plutôt trop sélectifs que pas assez.

IV.4 Résultats, évaluation et discussion de la méthode

Sélection du représentant de la main : nous utilisons le CG comme résumé de la trajectoire globale de la main. Nous justifions cela par le fait que nous n'avons aucune garantie d'efficacité sur l'utilisation de plusieurs points, et par le fait que le point utilisé doit être sur la paume. Parmi tous les points possibles, il y a aussi le **c**entre de la **p**aume de la main (**CP**), déjà utilisé à la section III.3 (p. 85), mais dont nous ne détaillerons le calcul qu'au V.2.1.1 (p. 134) Ici, nous discutons de l'intérêt comparé de l'utilisation de CP ou CG pour résumer la trajectoire globale de la main.

D'un point de vue théorique, il semble que CP soit plus intéressant : en effet, CG correspond à un moment d'ordre 1, et donc, est équivalent à la moyenne pour une représentation statistique. Or la forme de la main change au cours du temps, et cela a une influence sur le déplacement du CG. Le mouvement qui nous intéresse est plus celui d'un point fixe de la main que le mouvement réel de la répartition massique dans l'image binaire. En conséquence, le mouvement du CG est biaisé. A titre d'exemple, si la main reste dans la même Position, mais que la Configuration change de telle sorte que le CG se déplace fortement, un mouvement global de la main peut être détecté, alors qu'il n'existe pas. De plus, le centre de gravité est influencé par rapport à la taille et la forme du gant. Tout cela nous laisse penser que l'étude d'un point fixe comme le CP, même si son calcul est imprécis, est plus efficace.

Pourtant, en pratique, c'est le contraire, et c'est pour cela que nous avons finalement gardé CG. En effet, les tests que nous avons effectués montrent que CP est moins intéressant, et ces résultats sont concordants avec les recommandations des travaux menés au GIPSA-Lab/DPC ; même si l'utilisation de ces deux points donne des performances à peu près équivalentes, la trajectoire du CG est plus stable. Les raisons, par rapport aux défauts que nous avons mentionnés peuvent être les suivantes :

– La variabilité du CG par rapport à la taille du gant, ou l'imprécision due à la proximité entre le point étudié et le centre de l'adduction/abduction n'a pas une influence significative dans la mesure où celle-ci est prise en compte séparément pour chaque codeur. En conséquence, l'influence de la variabilité inter-codeur sur la généralisation des résultats est faible.

– Les périodes de mouvement lors du changement de Configuration et de Position sont à peu près équivalentes. De plus l'influence du changement de Position sur le CG est beaucoup plus importante que celle du changement de Configuration. En conséquence, il est possible d'étudier le mouvement de la

main en termes de changement de Position à un léger biais et à une Configuration près.

Définition du masque de pondération : Le masque de pondération digitale, utilisé dans le FRD n'a pas fait l'objet d'une évaluation précise. Lors de la mise en place initiale du FRD, nous avons constaté que certains mouvements de grande amplitude (que nous ne voyions pas car ils ne correspondaient à aucune transition) étaient détectés. Il se trouve que ces mouvements existent bel et bien, et cela confirme le bon fonctionnement du FRD, mais un humain ne les voit pas au premier abord pour la simple raison qu'ils ne sont pas significatifs du point de vue du LPC ; en conséquence l'attention ne se focalise pas dessus. Il s'agit souvent de gestes parasites, comme des mouvements du poignet, ou tout autre geste ne concernant pas les doigts, et ne modifiant pas la Configuration en cours. De toute évidence, un FRD bien adapté au LPC ne doit pas être sensible à ces mouvements. C'est ainsi que l'idée d'un masque de pondération permettant de mettre en valeur les zones de l'image où les doigts sont présents est apparue. Les doigts évoluent dans le haut de l'image, et le pouce évolue sur la gauche de celle-ci (nous rappelons que nous ne travaillons que sur des images représentant un codeur gaucher). Un premier masque basé sur ces considérations a été proposé et inséré dans la chaîne de traitement. Comme d'après une évaluation qualitative, la plupart des défauts remarqués initialement avaient soit disparu, soit fortement diminué, il a été décidé de garder ce masque. Comme, relativement à la complexité du reste des algorithmes, il s'agit d'une étape simple, elle n'a jamais fait l'objet d'une évaluation quantitative, ni l'objet d'une optimisation quelconque. De telles améliorations seront toujours possibles par la suite.

Réglages des seuils : Les seuils S^{immo} et S^{motion} doivent être déterminés manuellement. En vision par ordinateur, il semble toujours gênant de faire intervenir un expert dans un processus de reconnaissance quelconque. Comme nous pensons qu'une expertise bien menée est plus efficace qu'un apprentissage sur un corpus non représentatif, nous choisissons avec pragmatisme cette première solution qui fournit les résultats les plus concrets le plus simplement. En l'occurrence, notre paramétrage manuel respecte plusieurs considérations :

– Tout d'abord, il respecte le type de décision à prendre : ces seuils doivent être très sélectifs pour la classification d'une image en tant qu'image stable. Par défaut, il vaut mieux classer une image comme contenant du mouvement (une ITX).

– Ensuite, nous avons veillé à éviter de trop grandes instabilités de seuil : ainsi, une fois que la tendance générale du comportement a été spécifiée, et qu'une plage de valeurs approximatives a été déterminée, une valeur grossière a été choisie au sein de cette plage, et nous avons vérifié que de petites variations autour de celle-ci n'avaient pas de conséquences sur le processus de décision attaché au seuil.

– Enfin, bien sûr, les valeurs de seuil ont été testées sur une autre base de données, afin de vérifier sa généralisation.

De tout cela, il résulte que le seuil S^{motion} prend la valeur 5 aussi bien dans le cas où le filtrage est appliqué à la recherche d'ICC que d'ICP, et que le seuil S^{immo} prend la valeur 2.5 pour la recherche d'ICC et 1 pour la recherche d'ICP.

Extraction des cibles : La labellisation précoce a pour but de classer les images en deux catégories, les ICX et les ITX. En pratique, il y a entre 5 et 30 fois plus d'ITX que d'ICX. En conséquence, le pourcentage de mauvaise classification pour chacune des classes n'est pas une mesure pertinente. En effet, une mauvaise classification (une ITX est prise pour une ICX, ou inversement) a entre 5 et 30 fois moins d'influence sur le pourcentage d'erreurs des ITX que sur celui des ICX. Il est donc plus intéressant de considérer le nombre d'erreur en fonction du nombre d'ICX, quelque soit le type d'erreur. C'est ce que nous appelons *TauxErreur*(ICX). Malheureusement, une ICX n'est pas toujours définissable objectivement : dans la plupart des cas, il existe plusieurs images à qui l'on peut donner le titre d'ICX. Nous proposons donc d'évaluer plutôt la stabilité de la plage associée à chaque ICX. Nous pensons en effet qu'un tel test est plus robuste à la subjectivité de l'opérateur impliqué dans la définition de la vérité terrain : chaque zone de stabilité doit être caractérisée comme stable par l'opérateur, pour ne pas être considérée comme une erreur. Pour cela, ce dernier vérifie qu'il n'y a pas de changement majeur dans la forme de la main pour les ICC, et qu'il n'existe pas de mouvement global pour les ICP. Il s'agit d'évaluer quand :

– Une zone de stabilité n'est pas repérée. C'est ce que l'on appelle une **erreur de type 1**. Ce type d'erreur est très problématique, puisqu'il implique la perte d'un geste dans le décodage. Il existe deux types d'erreurs de type 1 : dans le cas d'une **erreur de type 1a**, la zone de stabilité n'est pas repérée, et l'ICX non plus. Celle-ci est simplement vue comme une transition. Dans le cas d'une **erreur de type 1b**, c'est la zone de transition entre deux gestes qui n'est pas repérée. La seule zone de stabilité trouvée correspond en fait à deux gestes ou plus. Il devrait donc y avoir plusieurs zones de stabilité et autant d'ICX. En pratique, il n'est pas indispensable de faire la distinction entre les erreurs de type 1a et 1b. En effet, quelque soit la cause de la disparition d'un geste, celle-ci a les mêmes conséquences sur le décodage.

– Les **erreurs de type 2** sont les erreurs duales des erreurs de type 1 : un geste inexistant est repéré. Cela peut être dû à un mouvement de dérive trop long qui fait qu'une transition est cinématiquement stable (**erreur de type 2a**). Si ce n'est pas le cas, c'est qu'une transition est détectée au milieu d'une zone de stabilité : une zone de stabilité est coupée en deux ou plus. Il y a donc plusieurs zones de stabilité, et autant d'ICX qui sont détectées, au lieu d'une seule. C'est ce que nous appelons les **erreurs de type 2b**. Les erreurs de type 2a sont moins problématiques que les erreurs de type 1, puisqu'aucune information n'est perdue. Malgré tout elles sont la conséquence directe d'erreur

de décodage. En revanche, les erreurs de type 2b ne sont absolument pas problématiques. Il est toujours possible, après l'étape de reconnaissance, de "recoller" les plages coupées, via la reconnaissance des ITXM et leur comparaison avec les ICX. Ainsi, nous ne nous occupons pas pour l'instant des erreurs de type 2b. Nous évaluons donc seulement le pourcentage d'erreurs de type 1 et 2a dans un premier temps.

Le *TauxErreur*(ICX) est donc défini comme la somme de toutes les erreurs de type 1a, 1b et 2a, divisé par le nombre total d'ICX.

Des premières observations, il est apparu que la recherche d'ICC est beaucoup plus difficile que la recherche d'ICP. Ainsi, dans ce dernier cas, quasiment aucune erreur n'a été observée. En conséquence, nous avons focalisé notre évaluation sur la recherche d'ICC. Comme le taux d'erreur sur les IC correspond à la somme des taux d'erreurs sur les ICC et les ICP, nous savons que :

$$TauxErreur(IC) = TauxErreur(ICC) + TauxErreur(ICP) < 2 \ . \ TauxErreur(ICC)$$

Le corpus de test est composé de séquences issues de ETTRAN N et ETTRAN BF, ce qui correspond à près de 5000 images pour 167 zones de stabilité. Il y en a donc autant à reconnaître. Pour vérifier le fonctionnement de la labellisation précoce, nous procédons comme suit :

– L'ensemble des séquences est labellisé manuellement.

– L'ensemble des séquences est labellisé automatiquement.

– Les deux labellisations sont ensuite comparées.

La plupart du temps, les labellisations manuelle et automatique ne correspondent pas exactement. En effet, dans bien des cas, plusieurs images peuvent être manuellement labellisées comme des cibles, cependant une seule est choisie. Ainsi, celle-ci peut ne pas correspondre avec celle obtenue par la labellisation automatique, sans pour autant signifier que cette dernière est fausse. Afin de pallier cela, nous nous intéressons aux plages de stabilité, pour lesquelles cette variabilité est moins importante, plutôt qu'aux ICX. Ainsi, en observant la stabilité de la main sur les images vidéo correspondant à l'instant entourant la réalisation du geste, il est possible de vérifier que les deux zones de stabilité (obtenues manuellement et automatiquement), sans être rigoureusement identiques, sont similaires. Sur l'ensemble des 167 ICC, il y a un total de 13 erreurs :

– Il y a 4 erreurs de type 1a : 4 gestes n'ont pas été repérés. 2 pour des raisons de codage (hésitations très marquées rompant la dynamique de codage) que nous n'avons pas pour objectif de compenser à ce niveau. Il s'agit donc d'erreurs du point de vue de l'évaluation du système complet, mais elles ne correspondent pas à des situations ayant mis en défaut la labellisation précoce proprement dite. Enfin, les deux dernières erreurs sont dues à l'initialisation de

la rétine (qui nécessite un certain nombre d'images ne contenant aucun mouvement), en début de séquence. En effet, sur certaines séquences, le codeur commence tout de suite, de sorte qu'il manque au début de la séquence les quelques images ne contenant aucun mouvement et qui sont nécessaires à l'initialisation de l'IPL. Donc, ces erreurs ne doivent pas être prises en compte pour notre évaluation.

– Il y a 1 erreur de type 1b : elle est due à un codage beaucoup trop précipité. Il est à noter que de nombreuses zones de stabilité sont ainsi "mâchées" par le codeur (une petite dizaine sur les 167 du corpus), mais que malgré cela, elles sont toutes récupérées, à l'exception de celle indiquée. En ce sens, la labellisation précoce est relativement robuste aux erreurs de type 1b. Cela est dû en grande partie à la manière dont les seuils S^{immo} et S^{motion} ont été définis. Ainsi, la labellisation précoce permet de traiter des séquences dont le codage est d'une qualité inférieure à celles que nous avons retenues au chapitre II (p. 35).

– Il y a 8 erreurs de type 2a : 8 zones de stabilité comportent une transition réelle : parmi celles-ci, 3 transitions sont stables cinématiquement, et même si elles ne sont pas interprétables en termes de codage, elles ne constituent pas une erreur au niveau cinématique. De même, 2 transitions sont dues à de très fortes hésitations de codage où la rythmique est changée de manière tellement brutale que la détection de cible est mise en défaut. Il y a 2 transitions dues à un mouvement de dérive particulièrement lent. Enfin, il existe une transition non pertinente que l'on ne peut expliquer.

– Les erreurs de type 2b, non comptabilisées à ce niveau-là, sont beaucoup plus fréquentes. Cependant, par l'étude des ITXM, il est possible de les supprimer. Ainsi, la manière de les prendre en compte dépend des résultats de la reconnaissance de la Configuration et de la Position. Comme nous le verrons par la suite, le nombre d'erreur de type 2b est nul pourvu que la classification ne se trompe pas. En effet, si à chaque fois qu'une zone de stabilité est coupée en deux, il est possible de la repérer grâce à l'étude des ITXM, il est aussi possible de les "recoller". Le taux d'erreur de type 2b est donc directement lié à celui de la classification (cf. chapitre V p. 123).

Sur les 167 ICC, il y en a 2 dont la détection est délicate pour des raisons d'initialisation du système. Celles-ci ne doivent pas être prises en compte. Cela fait 165 cibles. Notre système est donc robuste à 97.6% face aux erreurs de type 1a, à 99.4% face aux erreurs de type 1b, et à 95.2% aux erreurs de types 2a, pour ce qui est de l'évaluation la plus sévère. En effet, quelque soit la cause de l'erreur (la labellisation proprement dite ou un des algorithmes précédents) elles sont comptabilisées dans leur ensemble.

Enfin, il serait intéressant d'évaluer sur ce même corpus l'adéquation entre les IC définies d'un point de vue cinématique d'une part, et les cibles phonétiques d'autre part. Il s'agit de vérifier la concordance entre les gestes cinématiques détectés, et ceux effectivement réalisés par le codeur. Cependant, cela est difficile pour deux raisons :

– Tout d'abord, il est difficile de déterminer quand un geste a réellement été fait ou pas. Certaines hésitations ou erreurs sont manifestes, mais ce n'est pas le cas de toutes.

Ensuite, en termes de codage phonétique, le codeur réalise des gestes, et non des Configurations et des Positions de manière indépendante. Comme nous n'avons pas encore synchronisé les deux flux d'ICC et d'ICP pour déterminer des cibles gestuelles complète, une telle évaluation ne peut pas être menée de manière objective. En effet, dans certains cas, le décalage entre les ICC et ICP est trop important pour pouvoir effectuer cette synchronisation de manière triviale (cf. Figure IV-12).

Les taux d'erreurs que nous avons obtenus valident l'intérêt et le principe de la labellisation précoce, même si, comme cela est illustrée sur la Figure IV-12, il est indispensable que nous ajoutions d'autres modules de traitement de la dynamique temporelle (cf. chapitre VI p. 182). Cependant, cela requiert de connaître d'abord le contenu de chacune des ICX, mais aussi celui des ITXM (afin de détecter et corriger les erreurs de types 2b). Cette tâche de reconnaissance est traitée au chapitre suivant.

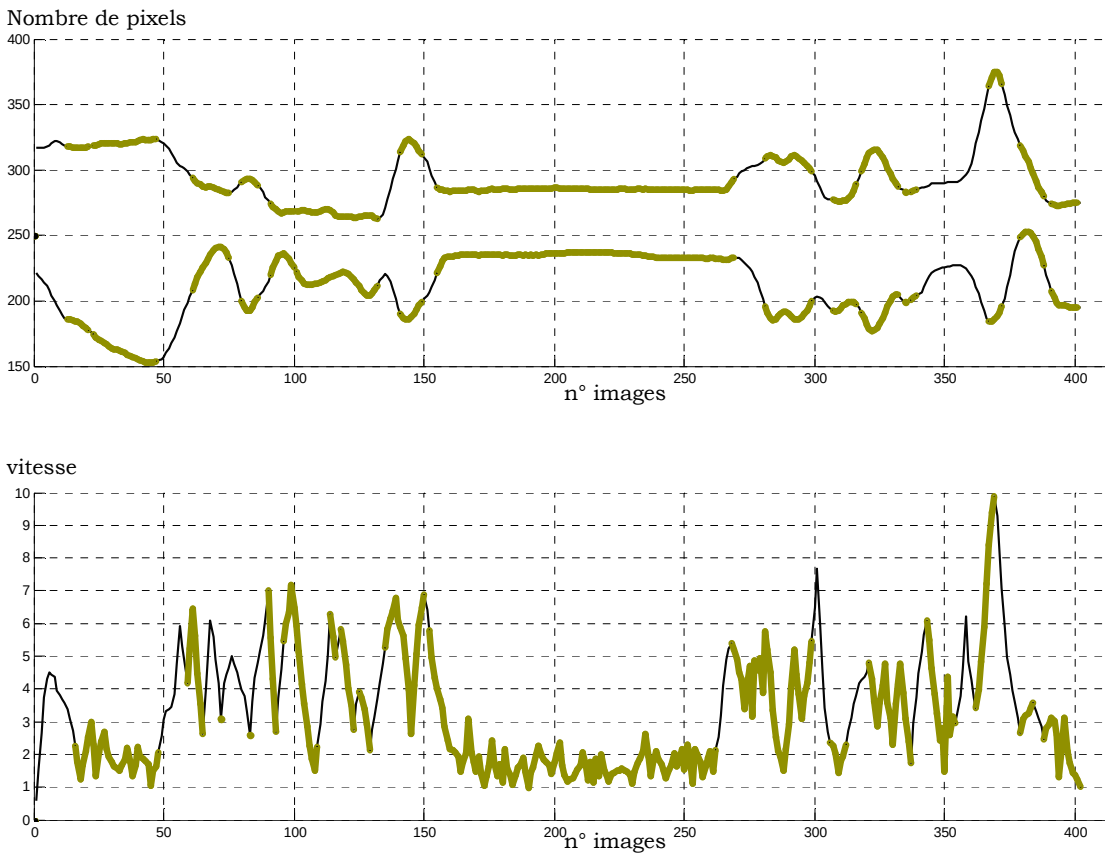


Figure IV-12 : en haut la représentation graphique du mouvement de la Position, et en bas, de la Configuration. En superposition à la représentation de la mesure de mouvement, les zones de stabilités sont indiquées en gras.

IV.5 Conclusion du chapitre

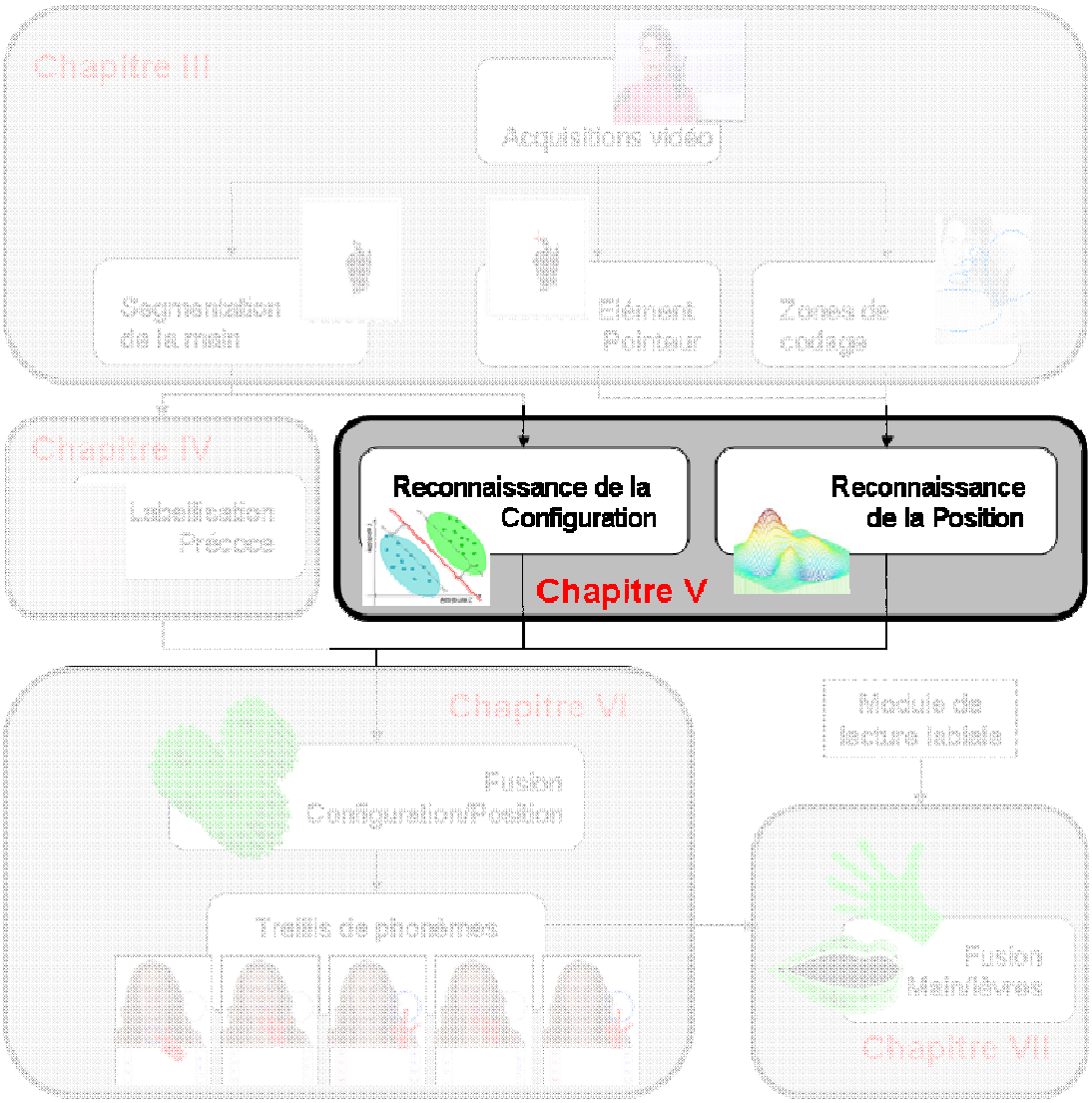
Dans ce chapitre, nous avons présenté un système original permettant de labelliser les images d'une séquence vidéo avant leur étude complète. Cette labellisation permet de faire la différence entre les images cibles de la trajectoire et les images de transition, afin de faire apparaître la structure de la séquence et faciliter l'étude de sa dynamique temporelle. Comme l'a montré son évaluation, le système proposé est relativement robuste. Cela est principalement dû à l'utilisation d'un algorithme biologiquement inspiré par le système visuel des vertébrés. En poussant l'analogie biologique plus loin, nous pensons qu'il est possible d'améliorer encore le système, et ce, selon deux directions :

– Tout d'abord en appliquant une méthode proche du FRD à l'étude du mouvement globale de la main, et non plus restreindre son utilisation à l'étude du changement de Configuration. Ainsi, il serait intéressant de chercher à utiliser les propriétés du filtre IPL pour déterminer la quantité de mouvement global de la main au cours d'une séquence, de manière plus efficace qu'en étudiant la trajectoire du CG.

– Ensuite, en utilisant la propriété de l'œil à s'adapter automatiquement à différents niveaux de mouvement. Cela permettrait non seulement d'éviter d'avoir à définir les seuils S^{immo} et S^{motion}, mais en plus, cela nous aiderait à résoudre les erreurs dues aux hésitations et aux mouvements de dérive qui sont actuellement les principales sources de confusion.

Enfin, il serait intéressant de pouvoir évaluer l'intérêt d'un tel algorithme dans des cas différents de ceux du LPC, et ainsi déterminer l'apport d'une telle méthode pour l'analyse de trajectoire en général.

CHAPITRE V
RECONNAISSANCE DU GESTE STATIQUE



En raison de la structure particulièrement ramifiée de ce chapitre, nous nous permettons d'en reprendre le sommaire détaillé :

Dans ce chapitre, nous traitons de l'ensemble des algorithmes ayant trait à la reconnaissance de la Position sur les ICP et les ITPM et à la reconnaissance de la Configuration sur les ICC et sur les ITCM. Dans le premier cas, il s'agit de déterminer si le doigt pointeur est dans une zone de pointage, et le cas échéant reconnaître laquelle, alors que dans le second cas, il s'agit de déterminer quelle est la Configuration la plus vraisemblable pour une forme de main donnée. D'une manière générale, les étapes d'un processus de reconnaissance automatique sont toujours les mêmes :

– **Définition des classes** : soit $C = \{C^1, \ldots, C^N\}$ l'ensemble des N **classes** auxquelles peut appartenir x un élément soumis à **classification**. Si les N classes sont **exclusives**, x ne peut appartenir qu'à une classe au plus. Si les N classes sont **exhaustives**, x appartient au moins à une classe. Dans le cas ou les classes ne sont pas exhaustives, x peut très bien n'appartenir à aucune classe. On définit alors généralement une **classe de rejet**, qui contient le **reste du monde**, c'est-à-dire tout sauf les N classes. L'objectif est ici de bien définir le problème de telle sorte que celui-ci puisse être résolu. Afin de faciliter la classification, les classes peuvent éventuellement être décomposées en sous-classes, si cela permet de mieux opérer la discrimination par la suite. Il est aussi possible d'effectuer quelques prétraitements permettant de limiter la variabilité intra-classe.

– **Définition de l'espace des attributs** : soit x un élément à classer, appelé un **item**, et $(x_1, x_2, \ldots, x_M)$ l'ensemble des M **attributs** numériques qui le décrivent au regard du problème de classification. Il s'agit donc de définir un ensemble de critères selon lequel la classification va être réalisée. Il peut s'agir de descripteurs calculés automatiquement, ou d'informations issues de systèmes plus complexes.

– **Choix d'une méthode de classification** : il s'agit de déterminer de quelle manière la séparation entre les différentes classes va être réalisée dans l'espace des attributs. La plupart du temps, il s'agit de choisir parmi les différentes méthodes disponibles de la littérature celle qui se prête le mieux au problème.

– **Prise de décision** : en fonction du résultat de la classification, une décision est prise sur l'espace des classes.

Cependant, la frontière entre chacune de ces étapes est parfois floue. Ainsi, dans certains cas, la prise de décision est triviale et découle directement du processus de classification. Dans d'autres cas, certains post-traitements sont ajoutés tels que dans [98]. De même, la méthode de classification ne consiste parfois qu'en une fusion de données. Dans de tels cas, les attributs sont souvent des informations de haut niveau sémantique, issues de systèmes complexes, tels que d'autres classifieurs. Ce processus apparaît alors plus comme un système de combinaison de classifieurs naïfs.

Si la séparation entre ces différents niveaux est floue, elle est aussi de peu d'importance. En effet, la résolution d'un même problème peut être effectuée de

plusieurs manières. Dans certains cas, les attributs de classification sont des descripteurs calculés automatiquement, et ils ne sont pas toujours discriminants pris individuellement. Dès lors, le classifieur utilisé doit être plus sophistiqué. A l'inverse, quand les attributs sont de plus haut niveau, la classification peut devenir facile (et ainsi se rapprocher d'un simple système de fusion de données). Dans un tel cas, la définition des descripteurs devient plus délicate et c'est pour cela que l'on peut l'assimiler à un processus de classification ou d'inférence. Finalement, tout est une question de choix. La manière de procéder dépend fortement du concepteur. C'est lui qui décidera si l'intelligence du système est placée au niveau de la description, de la classification, ou au niveau de la prise de décision.

Dans ces travaux, différentes stratégies ont été utilisées :

– Pour la reconnaissance de la Position, la classification est naïve puisque l'ensemble de la difficulté du problème est traité auparavant, au niveau de la description des classes (zones de pointage) et des attributs de classification (coordonnées du doigt pointeur). L'intelligence est donc focalisée sur les étapes en amont de la classification proprement dite.

– A l'inverse, pour la reconnaissance de la Configuration, il y a une méthode parmi celles proposées pour laquelle l'intelligence est répartie en divers endroits. Comme cela est détaillé par la suite, des descripteurs de bas niveau sont utilisés avec un système de classification élaboré. Seulement, le résultat de cette classification n'est pas utilisé en temps que tel, mais fusionné avec un autre attribut de haut niveau issu d'un système de classification expert. Celui-ci permet de repérer l'éventuelle présence du pouce dans la forme de la main à reconnaître, et donc vient compléter la classification initiale. La fusion étant effectuée avec un outil de classification élaboré, celui-ci sert à la fois de système de classification et de fusion de classifieurs.

– Enfin, la détection du visage, des yeux, du nez et de la bouche du codeur (détaillé à la section III.4 p. 93) est aussi un système de classification : pour chaque zone de l'image, celui-ci détermine s'il s'agit d'un visage ou non. Comme cela est expliqué en appendice C.1 (p. 291), il n'y a dans ce système aucune différence entre le processus de description des attributs et celui de classification. L'ensemble est réalisé conjointement.

Dans une première section, nous présentons l'algorithme de reconnaissance de la Position. En raison de ce que nous avons annoncé plus haut, celui-ci est simple. Ensuite, dans une deuxième section, la reconnaissance de la Configuration est détaillée. Cette méthode fait intervenir de nombreux algorithmes, et dans tous les cas, leur évaluation est présentée suite à leur description. Enfin, dans une troisième section, nous élargissons le champ de ces travaux appliqués à la reconnaissance des gestes manuels du LPC, en proposant diverses améliorations théoriques. Celles-ci n'ont pas forcément d'application directe pour la reconnaissance de la Configuration ou de la Position, mais elles sont cependant utiles dans les étapes ultérieures du

système (cf. chapitres VI et VII). Nous les présentons ici plutôt que dans les chapitres où elles ont un intérêt pratique, pour la simple raison qu'elles découlent logiquement du même formalisme et qu'elles en sont la continuité directe. Leur compréhension et leur justification s'en trouvent donc facilitées.

V.1 Reconnaissance de la Position

V.1.1 Méthode

Contrairement à ce que l'on pourrait penser, la reconnaissance de la Position n'est pas plus simple que celle de la Configuration. Seulement, la difficulté de chacune des deux tâches ne se trouve pas au même endroit. Pour ce qui est de la reconnaissance de la Position, l'ensemble des difficultés a :

– Soit été traité dans les chapitres précédents (par exemple, la définition des zones de pointage, ou encore la définition du doigt pointeur),

– Soit été repoussé à une amélioration ultérieure de l'algorithme initial présenté ici (par exemple un apprentissage semi-supervisé permettant d'affiner la définition des zones de pointage en fonction des habitudes du codeur).

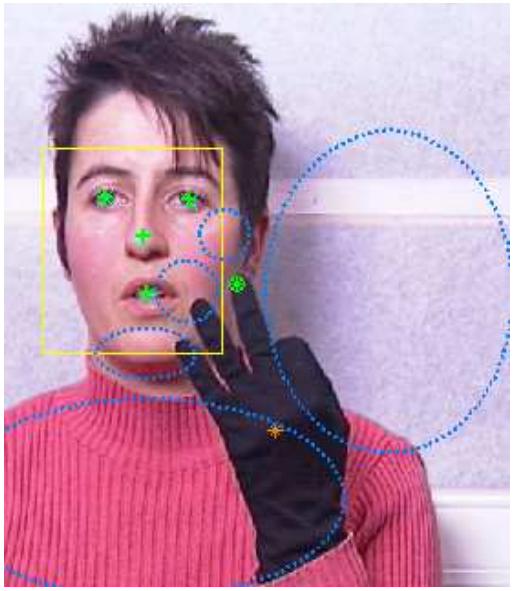


Figure V-1 : illustration d'une image de transition : le doigt pointeur, bien que défini, ne passe dans aucune zone de pointage.

Ainsi, à la fois l'espace de codage (espace géométrique du plan d'acquisition), les classes (frontière géométrique de chacune des zones de pointage) et les descripteurs (coordonnées spatiales de l'élément pointeur) sont déjà définis. Il ne reste plus à cette étape qu'à déterminer le processus de sélection de la classe. Quand l'espace de codage est un espace géométrique 2D discrétisé sur les pixels de l'image numérique, un tel processus est trivial et la plupart des méthodes donnent des résultats similaires. Nous proposons une méthode simple qui pourrait être remplacée par une autre, mais au final, les résultats seraient

globalement semblables tant que les éléments constitutifs importants (définition des zones de pointage et de l'élément pointeur) restent inchangés.

Nous proposons d'associer à l'élément pointeur la Position qui correspond à la zone de pointage dans laquelle il se trouve. Dans le cas où l'élément pointeur n'est dans aucune des zones de pointage prédéfinies, on suppose simplement qu'il s'agit d'une ITPM et qu'aucune Position n'est codée (Figure V-1).

Une autre méthode consisterait à choisir en permanence (c'est-à-dire pour chaque image) une Position en associant à chaque zone de pointage une gaussienne dont le tracé de l'ellipse représente un iso-écart-type. Cela a deux inconvénients majeurs :

– Tout d'abord, il serait impossible de traiter les ITPM sans les associer à un pointage particulier.

– Ensuite, la modélisation gaussienne empêcherait que les classes soient connexes (Figure V-2). La conséquence directe serait que la Position Côté (la Position ayant le plus de variabilité, et par conséquent dont la zone de pointage est la plus large) deviendrait une classe de rejet, comme cela est illustré sur la Figure V-2.

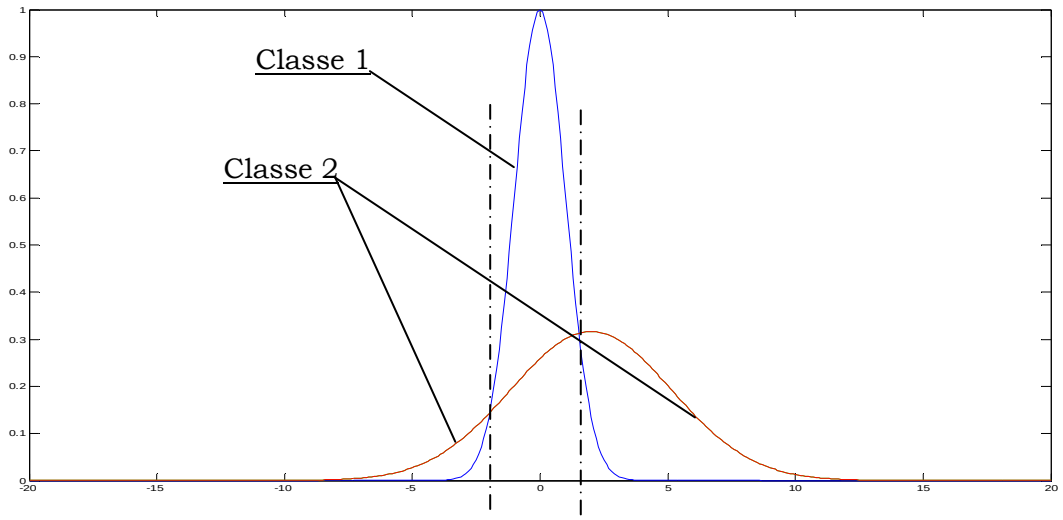


Figure V-2 : illustration au cas 1D de la non-connexité des classes dans un classifieur gaussien : la classe 1 est définie par la zone où la gaussienne la plus étroite est la plus grande, et la classe 2, par la zone où la gaussienne la plus étalée est la plus grande. Par définition, la classe 2 n'est pas connexe.

V.1.2 Evaluation

Afin de justifier la pertinence de la méthode de reconnaissance de la Position, il y a 4 points à vérifier :

– La pertinence de la définition des classes, à savoir les zones de pointage.

– La pertinence de la définition des attributs de classification.

– La pertinence de la description des items à classer.

– La pertinence de la méthode de classification.

Le premier et le troisième points ont été abordés au chapitre III (p. 58). En particulier le premier point est dénoncé comme étant le point faible de la généralisation de la méthode au cas multi-codeur, et des pistes d'amélioration sont proposées.

Les attributs de classification sont simplement les coordonnées géométriques du doigt pointeur dans le plan de l'image. En l'absence d'un autre jeu d'attributs constituant une alternative valable, il n'est pas question d'évaluation de ce point.

Enfin, le dernier point concerne la méthode de classification naïve des classes en fonction de leur modèle géométrique. Comme nous l'avons dit, elle est à peu près équivalente à une modélisation gaussienne, mais s'affranchit de certains inconvénients de cette dernière.

Nous proposons donc de réaliser une évaluation globale de tous les éléments impliqués dans la reconnaissance de la Position à partir de vidéos non traitées. Ainsi, les erreurs issues de prétraitements tels que la segmentation seront comptabilisées. Pour cette évaluation, nous utilisons des séquences du corpus ETTRAN N. L'ensemble de ces séquences correspond à un total de 3346 images pour 150 gestes. Comme la notion de pointage de Position n'existe pas sur la plupart des images de transition, nous ne pouvons considérer que les images sur lesquelles des gestes sont réellement codés. Toutes ces images ne sont pas forcément des ICP. En effet, pour chacun des gestes, il y a plusieurs images pour lesquelles la Position est identifiable, alors qu'il n'y a qu'une seule ICP. Une méthode d'évaluation de la reconnaissance de la Position est de comptabiliser l'ensemble des images pour lesquelles celle-ci est juste. Cependant, en raison de la forte dépendance temporelle qu'il y a entre des images successives, nombre d'entre elles, qui sont presque identiques, se retrouvent comptabilisées. Cela augmente artificiellement les scores obtenus. Ainsi, nous proposons de n'effectuer qu'une seule comptabilisation par geste. Cependant, il ne faut pas considérer que les ICP, car cela entrainerait la comptabilisation des erreurs de la labellisation précoce (qui permet de définir les ICP). Ainsi, nous visualisons l'ensemble d'une séquence et nous considérons la qualité de la reconnaissance de la Position dans son ensemble pour chaque geste, et non pas sur des ICP exclusivement. Cela revient en pratique à se placer à un niveau supérieur d'interprétation, et cela ne devrait être pris en compte et évalué que plus tard dans ce document. Cependant, c'est la méthode d'évaluation qui nous semble la plus fiable.

Le protocole d'évaluation est le suivant. Les séquences sont visionnées tour à tour, avec en superposition, l'affichage du doigt pointeur et des zones de pointage, (cf. Figure V-1). Pour chaque geste, la position du doigt pointeur dans

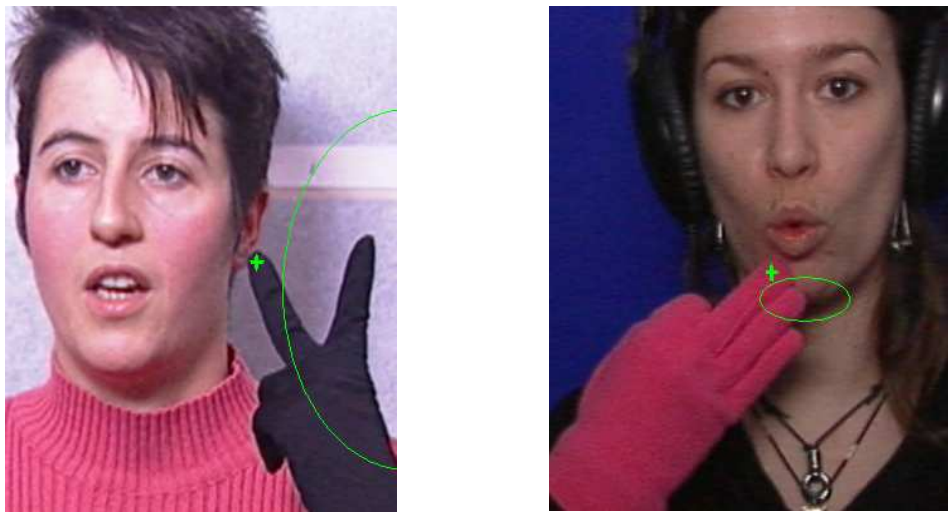
l'image est comparée à la zone de pointage afin de déterminer si la Position est correctement interprétée. Nous avons réparti les images dans 3 catégories :

– Position reconnue correctement dans toutes les images d'un même geste. Il y a 137 tels gestes, ce qui correspond à une précision de 91.3%.

– Position reconnue correctement dans certaines images du geste, mais pas dans toutes. Il y a donc une certaine instabilité. Cela correspond à 9 gestes. Si cette instabilité est gérée par la suite dans le processus de reconnaissance globale, la précision de la reconnaissance est alors de 97.3%.

– Position non reconnue. Il y a 4 gestes correspondant à ce cas.

Il est intéressant de faire une analyse qualitative des situations menant à une erreur ou à une imprécision dans la reconnaissance de la Position. En effet, celles-ci sont caractéristiques.



(a) (b)

Figure V-3 : (a) la Position Côté n'est pas reconnue parce que le doigt pointeur est en dehors de la zone de pointage en raison du fort écartement des doigts; (b) les imprécisions cumulées de la zone de pointage et de l'élément pointeur entraînent une mauvaise classification.

Pour les Positions non reconnues, il y a 3 cas où la Configuration 8 est codée avec les doigts tellement écartés que le doigt pointeur semble trop près du visage, même si les positions globales de la main et du poignet dans l'image sont celles d'une Position Côté (cf. Figure V-3a). De telles situations doivent pouvoir être gérées par une étude de la corrélation entre la distribution des zones de pointage et la Position : ainsi, dans le cas particulier de la Configuration 8, il doit être possible de proposer une définition des ellipses des zones de pointage qui tienne compte de ce décalage. Comme cela doit d'abord être confirmé sur un grand nombre de codeurs, nous ne proposons pas à ce niveau de compenser ce type d'erreur, mais c'est une piste d'amélioration indéniable. Le dernier cas

correspond à une flexion du poignet tellement importante pendant le codage d'une Position Gorge que les doigts ne sont plus visibles à l'exception du pouce, et que celui-ci est désigné à tort comme doigt pointeur. Dans un tel cas, nous ne prétendons pas pouvoir compenser un codage de cette qualité.

Parmi les 9 imprécisions (Positions reconnues dans certaines images du geste, mais pas dans d'autres), il y en 3 qui correspondent au même cas d'erreur que précédemment. La Configuration 8 est codée avec des doigts tellement écartés que le doigt pointeur se trouve trop près du visage. Il y a ensuite 1 cas correspondant à l'erreur déjà mentionnée et due à une flexion trop importante du poignet dans la Position Gorge. On constate donc une forte concentration d'erreurs du même type. Les 5 imprécisions restantes sont des cas où l'imprécision de la zone de pointage du menton (que nous avons déjà mentionnée à la section III.4 p. 93) et celle du doigt pointeur se cumulent : ces deux imprécisions ne sont pas toujours sources d'erreur prises indépendamment, mais le sont plus souvent lorsqu'elles sont cumulées. En effet, le doigt pointeur utilisé en pratique est de temps en temps le majeur, de temps en temps l'annulaire, et de temps en temps l'index, quand plusieurs d'entre eux sont déployés (cf. sections II.3 p. 42 et III.3 p. 85) de sorte que le choix que nous proposons n'est pas toujours le bon. Quand ces deux imprécisions se cumulent (cf. Figure V-3b), il arrive que l'élément pointeur ne corresponde plus à la zone de pointage de la Position réelle. Une manière de limiter cela est de mieux définir la zone de pointage du menton ; cela sera possible une fois la segmentation des lèvres incluse dans l'algorithme final. Une autre manière est de demander aux codeurs de limiter au maximum ces imperfections de pointage.

Finalement, il y a donc seulement 3 types d'erreur. Cela signifie qu'en les surmontant, il est possible d'améliorer le système. Afin de vérifier cela, nous devons être sûrs que ces erreurs sont aussi les plus courantes pour d'autres codeurs, et que d'une manière générale, les performances sont équivalentes. Cela est d'autant plus important que la définition des zones de pointage n'est pas automatiquement adaptable à l'individu.

Afin de vérifier cela, nous effectuons les mêmes tests sur des vidéos du corpus MAGOZ J, acquis dans des conditions différentes (codeur, gant, éclairage, etc.). Ce corpus a été choisi par ce que c'est celui où le codage est le plus différent de celui de ETTRAN N. En effet, le codeur est un malentendant, et n'a pas de certificat de codage. Celui-ci est donc de moins bonne qualité : les mouvements ont moins d'amplitude et les Positions sont pointées avec beaucoup moins de précision. Le gant est épais, ce qui rajoute de l'imprécision. Cette expérience est donc en conditions limites par rapport aux types de codage auxquels le système est censé être confronté. Comme il ne s'agit que d'un complément permettant d'en tester les limites, nous utilisons un échantillon plus petit du corpus : nous considérons 41 gestes, pour lesquels 82.9% ont une Position bien reconnue. Il y a donc 17.1% de gestes pour lesquels il y a soit une imprécision, soit une erreur (sur ce corpus, il est beaucoup plus difficile de bien séparer ces deux cas). Les zones de pointage semblent qualitativement bien correspondre. En revanche,

les imprécisions du codage et du doigt pointeur sont toutes les deux plus importantes, et l'intégralité des erreurs est due au cumul de ces deux imprécisions. Notons, qu'il est difficile de dire dans quelle proportion l'imprécision vient du codage ou de la détermination du doigt pointeur. Nous savons juste que des décalages peuvent survenir dans 3% à 7% des cas, lors de la détermination du doigt pointeur (cf. tests du paragraphe III.3.3 p. 92).

Les doigts étant plus petits, et l'amplitude des mouvements plus faible, des erreurs dues à la Position Gorge et à la Configuration 8 n'ont pas été repérées, mais dans de nombreux cas, le doigt pointeur se retrouve en périphérie de la zone de pointage, ce qui confirme la tendance de ces facteurs d'erreur. Cependant, d'autres tests à ce sujet seraient nécessaires, sur un grand nombre de codeurs, en fonction de leur morphologie et de leurs habitudes de codage.

Dans tous les cas, nous renvoyons à la section III.4 (p. 93) pour tout ce qui concerne plus précisément la description des zones de pointage en fonction de la variabilité des visages et des morphologies.

V.2 Reconnaissance de la Configuration

Nous traitons du problème de la reconnaissance de la Configuration en abordant tour à tour les quatre différentes étapes énoncées en introduction. La première concerne la définition des classes. Celles-ci sont ici définies de manière élémentaire : nous associons à chaque Configuration une unique classe. Il ne reste donc que trois étapes. Dans un premier paragraphe (V.2.1), nous indiquons comment diminuer la variabilité intra-classe en supprimant la partie de l'image binaire de la main qui correspond au poignet. Ensuite, au paragraphe V.2.2, nous nous intéressons à l'espace d'attributs dans lequel effectuer la classification. Au troisième paragraphe (V.2.3), la classification proprement dite est abordée. Enfin, au (0), un résumé global de la stratégie retenue pour la reconnaissance de la Configuration est présenté. Toute au long de cette section, une évaluation de chacun des algorithmes est présentée immédiatement après sa description.

V.2.1 Réduction de la variabilité par suppression du poignet

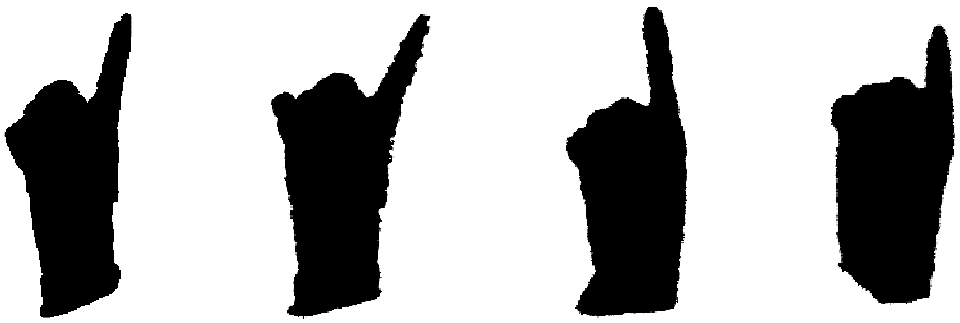


Figure V-4 : influence de la variabilité du poignet sur la forme générale de la main dans le cas d'une Configuration 1.

Quand nous cherchons à reconnaître la Configuration, notre attention se focalise sur la zone de l'image binaire dans laquelle nous devinons que les doigts se trouvent, et nous sommes capables de nous affranchir de la variabilité induite par le reste de la forme. Ce n'est évidemment pas le cas d'un système automatique (cf. Figure V-4). Ainsi, nous proposons d'ajouter certains prétraitements permettant de diminuer cette variabilité. Conformément à [37], la principale opération est de supprimer le poignet ou l'avant-bras. Pour cela, nous sommes partis de la méthode préconisée dans cet article, puis nous l'avons améliorée, afin qu'elle réponde à plusieurs critères :

– **Stabilité temporelle**. La forme de la main tronquée doit être stable au cours du temps : le poignet doit être découpé de la même manière tout au long d'une séquence.

– **Stabilité par rapport au codage**. La forme de la main tronquée doit être stable indépendamment de la Configuration et de la Position : le poignet doit être découpé de la même manière dans tous les cas.

– **Stabilité inter-codeur**. La découpe de l'image binaire doit toujours correspondre à peu près à la même zone morphologique, afin qu'elle soit cohérente d'un codeur à l'autre, et qu'il soit possible d'utiliser des bases d'apprentissage communes pour la classification multi-codeurs.

– **Efficacité.** Le prétraitement doit augmenter la précision de la classification. Parmi plusieurs méthodes possibles et satisfaisant visuellement à ces critères de stabilité, nous avons retenu celle qui permet la meilleure classification.

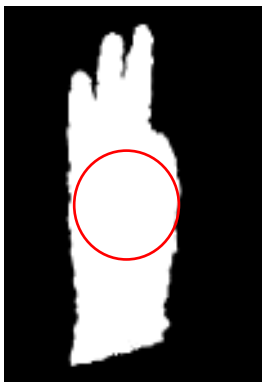


Figure V-5 : emplacement de la paume de la main.

Originellement et conformément à [37], le principe est de repérer la paume de la main et de supprimer les parties de la forme de la main qui se trouvent en dessous de la paume. En pratique, [37] propose de repérer la paume en l'approchant par le plus grand cercle inscrit dans la forme de la main (cf. Figure V-5). Le centre de ce cercle ainsi que son rayon sont donnés par la recherche du maximum de la **transformée de distance** de l'image binaire (cf. appendice C.4 p. 298).

La transformée de distance d'une image binaire associe à chaque pixel de l'objet la distance au pixel du fond le plus proche, et associe à chaque pixel du fond la valeur 0 (cf. Figure V-7). En raison de la grande variabilité de la forme de la main dans notre cas et en raison du port du gant qui déforme légèrement la main (cf. Figure V-6), cette méthode ne donne pas des résultats satisfaisants. Nous gardons le principe général de la méthode, à savoir la détermination de la paume, suivie de la suppression du poignet induit par la définition de la paume. Cependant, nous proposons d'améliorer chacun des deux aspects de cet algorithme.

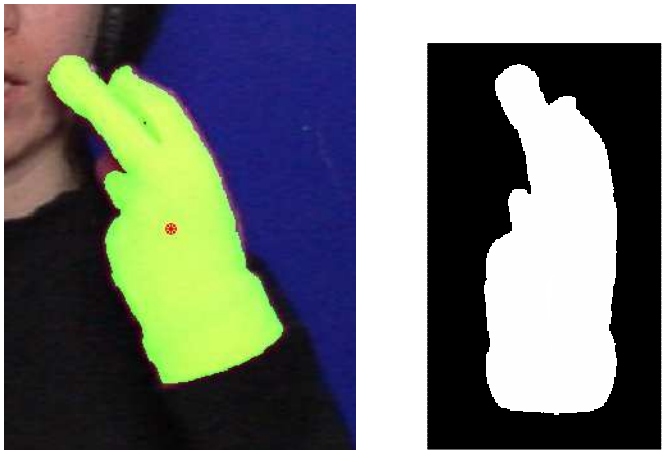


Figure V-6 : cas où la forme de la main est modifiée par le gant. Dans un tel cas, la méthode décrite dans [37] ne permet pas d'extraire la paume.

V.2.1.1 Détermination de la paume de la main

Nous décrivons ici l'algorithme d'approximation de la paume par un cercle inscrit dans le contour de la main.

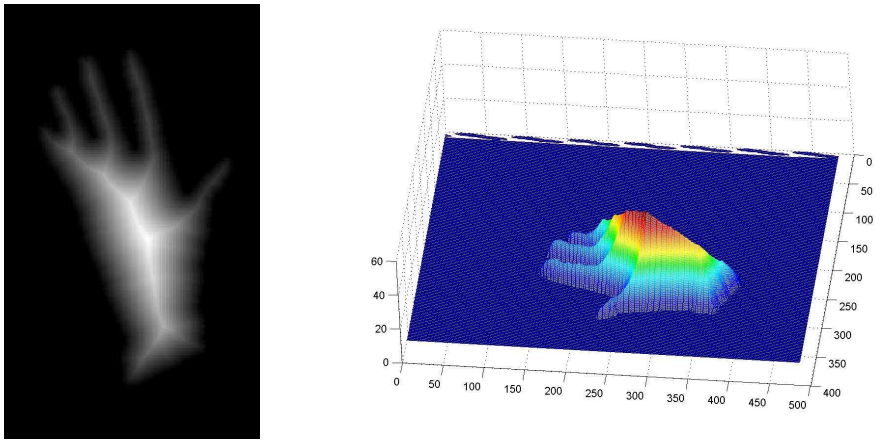


Figure V-7 : illustration de la transformée de distance en niveau de gris (le noir correspond au fond, et les pixels sont d'autant plus clairs qu'ils sont éloignés du fond) et en représentation tridimensionnelle.

Etape 1 : Calcul de la transformée de distance 2D-Euclidienne de l'image binaire représentant la forme de la main. Intuitivement, le centre de la paume de la main est un point qui est parmi les plus éloignés du contour de la forme de la main (cf. Figure V-7). La transformée de distance permet d'évaluer cela. Soit R la valeur maximale de la transformée de distance.

Etape 2 : Extraction de l'ensemble E des pixels qui sont à une distance supérieure ou égale à 95% de la distance maximale au bord (cf. Figure V-8).

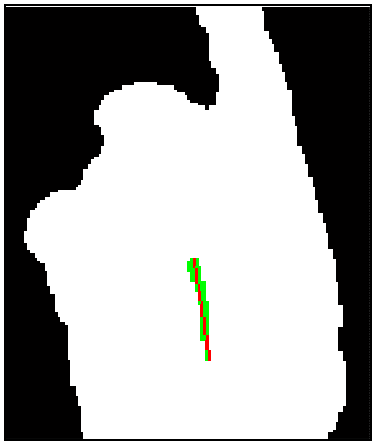


Figure V-8 : extraction de *E*, représenté en vert, et calcul de S, représenté en rouge.

Etape 3 : Régression linéaire sur E. Soit *d*, la droite ainsi trouvée et S le segment de *d* correspondant à la zone de projection de E sur d (cf. Figure V-8).

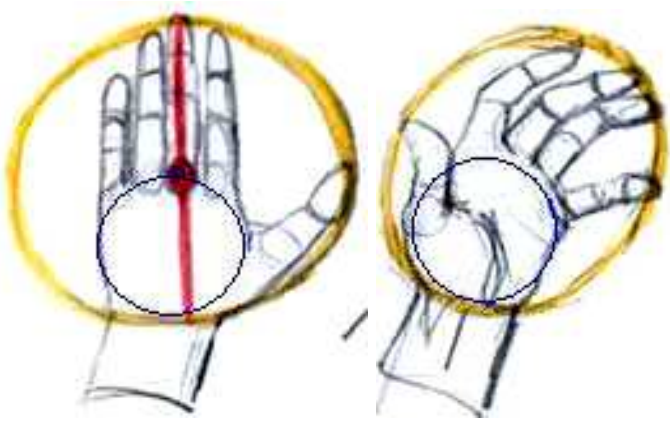


Figure V-9 : proportions de la main [166].

Etape 4 : D'après les observations morphologiques que retiennent les dessinateurs pour respecter les proportions de la main, il apparaît que le cercle inscrit de la paume est entre 1.5 et 2 fois plus petit que le cercle circonscrit à la main (cf. Figure V-9), suivant la manière dont elle est déployée. Soit A le point le plus haut de l'image binaire, et B l'intersection la plus proche de S entre le cercle de centre A et de rayon 1.5×R et *d* (cf. Figure V-10).

Etape 5 : Soit CP (le Centre de la Paume de la main) le point de S le plus proche de B. Si CP appartient à S, CP = B, sinon, CP est une des extrémités de S. La paume de la main est le cercle de centre CP et de rayon R.

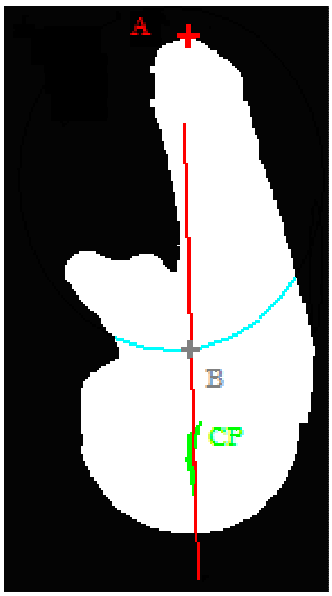


Figure V-10 : définition de A et de B.

V.2.1.2 Suppression du poignet

Dans ce paragraphe, nous proposons d'améliorer la manière de supprimer le poignet (c'est-à-dire la zone de la main se trouvant sous la paume). La méthode originelle de [37], que nous avons reprise dans [J4] et [C2], préconise simplement de découper en suivant le contour du cercle de centre CP et de rayon R. Voici l'algorithme que nous proposons :

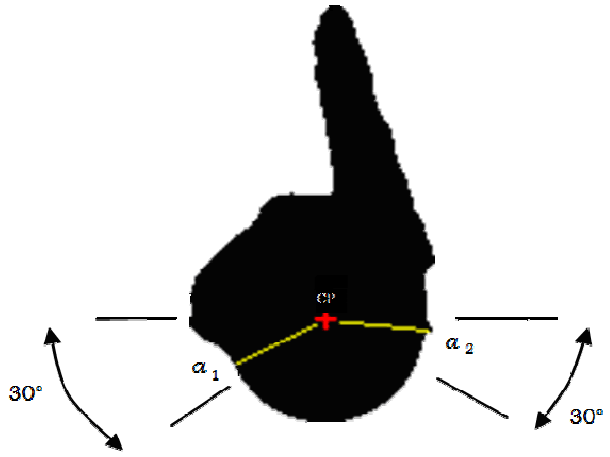
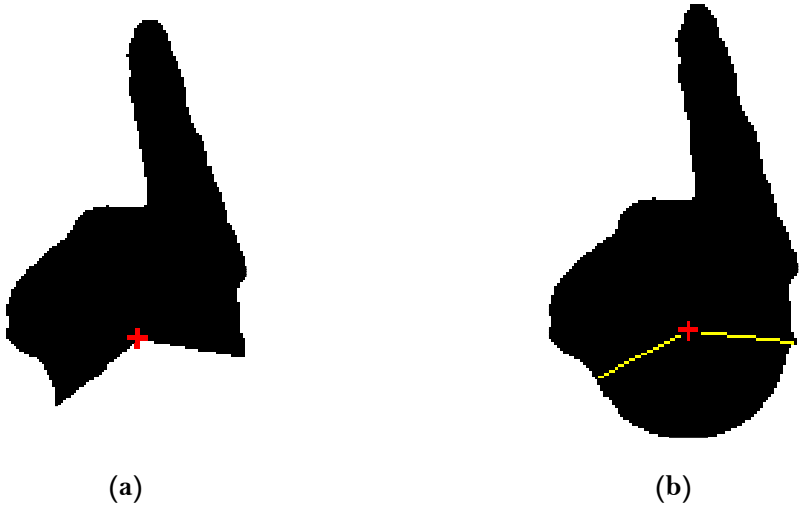


Figure V-11 : méthode de délimitation de la paume de la main.

Etape 1: Morphologiquement, il y a un rétrécissement dans la forme de la main au niveau du poignet. Ce rétrécissement correspond au segment qui se trouve entre la tête radiale et la tête cubitale de l'avant-bras. Le principe est de supprimer la partie de la forme de la main qui se trouve en-dessous de ce resserrement. En pratique, il est plus avantageux de couper la main plus haut et de remplacer la paume de la main par un cercle : cela diminue les variabilités dues (1) à la position du poignet, (2) à la Configuration, (3) aux différentes morphologies de poignets. Ainsi, nous commençons par rechercher les deux points du contour de l'image qui sont les plus près de CP dans une zone angulaire se trouvant entre l'horizontale par rapport à CP et 30° (cf. Figure V-11). Soit a_1 et a_2 ces deux points.

Etape 2: La zone angulaire comprise entre les vecteurs $\overrightarrow{CPa_1}$ et $\overrightarrow{CPa_2}$ est supprimée comme indiquée sur la Figure V-12a. A ce niveau le poignet et une partie de la paume de la main sont supprimés ; il résulte de la découpe brutale et de la forme en "V" une certaine instabilité : en fonction des descripteurs, la description de la main n'est pas forcément radiale, et même si c'est le cas, son centre de description n'est pas forcément CP. Enfin, l'angle que forme le "V" n'est pas constant, ce qui est donc source de variabilité.



(a) (b)

Figure V-12 : (a) La suppression en "V" est instable. (b) Il convient donc de remplir le "V" avec un disque de rayon variant linéairement entre les deux points de découpe. En effet, les distances entre CP et a_1 d'une part, et CP et a_2 d'autre part, ne sont pas toujours égales.

Etape 3: Afin de pallier ce défaut, le "V" est rempli à nouveau par un cercle de centre CP et de rayon variant linéairement avec l'angle afin de relier a_1 et a_2 (cf. Figure V-12b).

V.2.1.3 Evaluation de la réduction de la variabilité

Pour vérifier le bien fondé de l'hypothèse selon laquelle la suppression du poignet facilite la reconnaissance, nous proposons d'effectuer une série de tests via le programme **ImTrAc**. **ImTrAc** est un des éléments ayant servi à

l'élaboration de **DocMining** [65], un logiciel de reconnaissance d'écriture et de document [31]. **ImTrAc** est une applet java/XML qui permet de mettre très facilement en place des corpus et des scénarii de test pour la classification d'images binaires. Dans le cas présent, il nous est utile pour évaluer la reconnaissance que l'on effectue sur un même corpus en fonction des traitements appliqués. Cela permet d'évaluer la qualité des traitements au regard du problème de reconnaissance. Ainsi, nous pouvons facilement tester notre méthode de suppression du poignet, et la comparer avec un grand nombre d'autres méthodes.

Pour cela, nous utilisons les méthodes de classification décrites plus loin, et nous faisons jouer par **ImTrAc** l'ensemble des scénarios de tests sur chacune des méthodes de suppression du poignet. A ce niveau-là, il n'est pas nécessaire d'avoir une compréhension complète du système de classification. Il suffit d'interpréter le taux de bonne classification comme un indice positivement corrélé à l'efficacité du prétraitement.

On s'intéresse donc à la classification comparée d'un corpus en fonction de la manière dont on supprime la variabilité issue du poignet. Pour cela, on utilise les **Descripteurs de Fourier-Mellin** (**DFM**) (cf. V.2.2.4 p. 145) et un classifieur de type K-NN (cf. V.2.3.1 p. 148). L'intérêt du K-NN est que la classification ne s'effectue qu'en fonction des exemples d'apprentissage les plus proches de l'item à classer. Il n'y a quasiment aucun processus d'inférence au niveau global. Ce classifieur est donc particulièrement sensible à la variabilité locale (entre les items proches dans l'espace des descripteurs) qui apparaît dans le corpus. Ceci semble donc bien adapté pour mettre en valeur la manière dont la variabilité intra-classe est diminuée.

Le corpus utilisé est une base de 443 images issues de la base ETTRAN N réparties inégalement entre les 9 classes[3] (de 17 à 60 items par classe) pour lesquelles 4 traitements distincts ont été appliqués, générant ainsi quatre corpus distincts :

– **Corpus 1** : il s'agit du corpus témoin. Les images ne sont pas traitées au-delà de la segmentation et le poignet n'est pas supprimé.

– **Corpus 2** : découpe naïve du poignet telle qu'elle est décrite dans [37] : le centre de la paume et son rayon sont repérés par la transformée de distance et l'ensemble du poignet est découpé du reste de la main en suivant le contour inférieur du cercle approchant la paume. Nous avons cependant artificiellement descendu le cercle d'un maximum de 5 pixels, pour (1) obtenir une découpe plus lisse, (2) tenir compte de l'épaisseur rajoutée par le gant, (3) être sûr de ne pas perdre le pouce par une découpe trop haute, (4) et enfin avoir un meilleur rendu visuel.

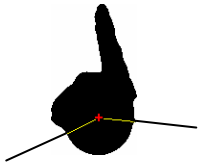
[3] Les huit Configurations du LPC ainsi que la Configuration 0, rajoutée pour les besoins de ce travail.

– **Corpus 3** : dans ce corpus, le centre de la paume de la main est déterminé par la méthode que nous avons proposée, mais le poignet est découpé comme dans le cas du précédent corpus.

– **Corpus 4** : dans ce corpus, l'ensemble de la méthode proposée est appliquée.

Le protocole de test mis en place avec **ImTrAc** est le suivant. A chaque itération, chaque corpus est divisé aléatoirement en deux ensembles d'apprentissage et de test, de taille égale, et un test de classification est effectué. 20 itérations sont produites afin de comparer la médiane, la moyenne et l'écart-type du taux de classification. Les résultats du Tableau V-1 laissent clairement apparaître que la méthode que nous avons proposée est la plus efficace.

Tableau V-1 : comparaison des méthodes de suppression du poignet.

	Corpus 1	**Corpus 2**	**Corpus 3**	**Corpus 4**
				
Médiane	68%	77.8%	88.3%	91.8%
Moyenne	67%	77.9%	88.3%	91.2%
Ecart-type	2.5%	1.5%	2%	2%

Par la suite, nous avons appliqué le même protocole à 3 autres corpus issus de la campagne MAGOZ afin de comparer les diverses méthodes sur d'autres gants/conditions d'acquisition. Les résultats sont identiques : la méthode que nous proposons reste la meilleure avec une moyenne des taux de reconnaissance comprise entre 89.9% et 92.7%, suivant les corpus.

Maintenant que la variabilité de la forme de la main a été réduite, il s'agit de déterminer (1) des descripteurs pouvant servir d'attributs de classification, (2) une méthode de classification. Le paragraphe suivant traite donc de la définition de l'espace des attributs de la forme de la main.

V.2.2 Définition de l'espace des attributs de classification de la forme de la main

Après une brève description des diverses méthodes à notre disposition, nous présentons trois jeux d'attributs dignes d'intérêt pour notre problème de reconnaissance de Configuration. Enfin nous évaluons chacun d'eux en les

appliquant à diverses formes de main dont le poignet à été supprimé, et nous comparons leur efficacité.

V.2.2.1 Généralités sur les méthodes de description

En vision par ordinateur, tout objet doit passer par une étape de description avant de pouvoir être appréhendé par la machine. Il existe donc toute une littérature sur ce que doit être une bonne méthode de description [30], [27], [33]. Le nombre de publications à ce sujet a encore augmenté avec le développement des travaux sur la compression et le codage d'images, notamment vidéo, au travers du format MPEG 7 [25]. De l'ensemble de ces travaux, il résulte une liste de critères permettant de déterminer l'intérêt d'une méthode de description [26]. Celle-ci doit respecter les propriétés suivantes :

– **Invariance** : 2 objets identiques doivent avoir la même description. Il est à noter que l'identité d'un objet est bien souvent invariante par similitude directe (translations, changements d'échelle et rotations) et parfois indirecte (réflexions ou effets miroir).

– **Unicité** : 2 objets différents ont des descriptions différentes.

– **Stabilité** : la différence de description de deux objets doit être corrélée positivement avec la différence perçue entre ces deux objets.

– **Efficacité** : la représentation de l'objet doit être aussi compacte que possible en mémoire, et son calcul doit être polynomial de la taille de l'objet (tout calcul dont la complexité est non-polynomiale en fonction de la taille de l'objet est évidemment à bannir).

– **Facilité d'implantation** : cela permet une plus grande portabilité et un risque plus faible d'erreur.

– **Complétude** : il doit être possible de reconstruire l'objet à partir de sa description. Cette reconstruction doit être **hiérarchique** afin qu'il soit possible de tronquer la représentation pour s'affranchir d'un certain niveau de détail. Il doit être possible aussi de ne reconstruire que partiellement l'objet, et notamment de faire ressortir certaines propriétés de l'objet (telles que la symétrie par exemple).

La description la plus simple de la forme de la main est d'utiliser l'image binaire issue de la segmentation, en tant que telle. Il est cependant évident que l'image elle-même ne satisfait pas à l'ensemble de ces critères.

Jusqu'à présent, la notion d'item à classer est assez floue. Il peut s'agir d'un morceau d'image, d'une forme (image binaire type masque ou type contour), etc. Certaines méthodes de description sont restreintes à un certain type d'items, d'autres sont plus générales. Nous nous intéressons ici à la description d'une image binaire représentant une forme de main. Il s'agit donc d'un type d'image particulier : binaire, ayant des contours fermés, contenant un seul objet

connexe, ayant un unique contour (aux erreurs de segmentation près qui "mitent" la forme de la main), ainsi que quelques propriétés géométriques intrinsèques à la main (le CG est contenu dans la main, les doigts sont orientés vers le haut, etc.). Pour de telles images, on décompose classiquement les méthodes en deux groupes :

– Les descripteurs orientés **Région** : il s'agit de décrire la répartition de la "masse" de l'objet par rapport à un point de référence. Par effet d'inertie, ces descripteurs sont robustes à un léger bruit lors de la définition du masque de l'objet (erreur de segmentation, calcul en virgule fixe, etc.). Voici quelques descripteurs classiques correspondant à ce paradigme : moment de Zernike [25], invariants de Hu [32], [12], descripteurs de Grille [25], descripteurs de Fourier-Mellin [28], etc. …

– Les descripteurs orientées **Contours** : il s'agit de décrire la trajectoire du contour dans l'espace. Bien que plus sensibles au bruit, ces descripteurs ont un avantage : ils sont plus proches de la manière dont le système visuel humain appréhende et compare les formes. Dans cette catégorie, on trouve les descripteurs de Fourier [25], les Curvature Scale Space Descriptors (descripteurs CSS) [26], etc.

D'une manière générale, il est possible de passer d'un type de descripteurs à l'autre en convertissant une image de contour en un masque binaire, et réciproquement.

Une dernière possibilité est de définir des descripteurs *ad hoc*, à partir de l'extraction d'invariants sur les classes de notre problème. Cette solution est bien sûr beaucoup plus difficile à mettre en place, mais elle ne doit pas être écartée de prime abord ; elle peut en effet être nécessaire dans le cas où les descripteurs classiques ne fourniraient pas de bons résultats. Pour trouver de tels descripteurs *ad hoc*, il est possible :

– d'utiliser l'apprentissage, tel que cela est décrit dans l'appendice C.1 (p. 291),

– d'effectuer cela de manière théorique. D'une manière générale, la définition et la recherche d'invariants mathématiques est un problème théorique complexe. Cette théorie est décrite dans [34], [35], mais la thèse de Schmid [33] propose un résumé clair et synthétique appliqué à la vision par ordinateur.

– d'utiliser la réponse de classifieurs moins évolués comme autant de paramètres de classification. Contrairement aux deux cas précédents, il s'agit alors généralement d'attributs de haut niveau.

V.2.2.2 Attribut de haut niveau : indicateur de présence du pouce

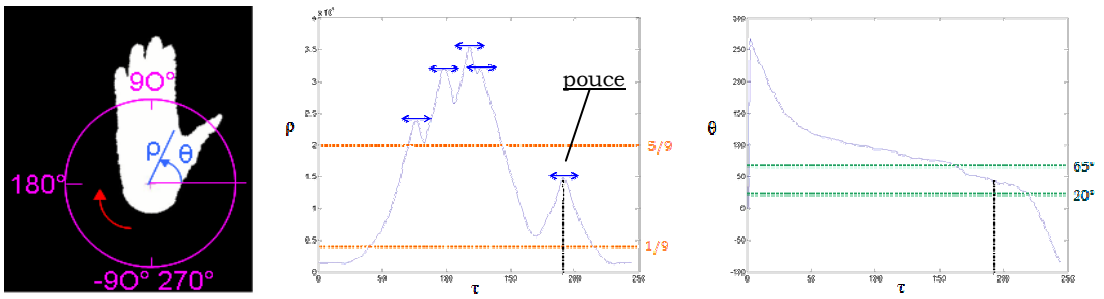
Il est aussi possible de déterminer des descripteurs de plus haut niveau d'interprétation (par exemple, de niveau morphologique ou sémantique), mais une telle détermination s'appuie en général sur un calcul à partir d'éléments de plus bas niveau. Il s'agit donc d'un descripteur *ad hoc*, basé sur un système de

classification ou sur un ou plusieurs systèmes experts. Dans le cas de la reconnaissance de la Configuration, nous proposons d'utiliser un tel descripteur de haut niveau.

Dans [C2], nous envisageons l'usage d'un descripteur indiquant la présence du pouce. Il se déduit de sa présence ou de son absence, un algorithme de classification triviale entre les Configurations {0, 1, 2, 3, 4, 8} d'une part, et les Configurations {5, 6, 7} d'autre part. Ce descripteur est calculé de la manière suivante :

Etape 1 : description du contour. Par un suivi de contour sur la forme de la main redressée verticalement, la représentation paramétrique {ρ(τ), θ(τ)} en coordonnées polaires est calculée (cf. Figure V-13).

Etape 2 : détection des pics. Après avoir lissé les fonctions paramétriques du contour (filtre passe-bas et sous-échantillonnage), les maxima locaux de la fonction ρ sont détectés. Comme nous l'avons vu pour la détection de l'élément pointeur, ils correspondent aux doigts potentiels de la main (cf. Figure V-13.b).



(a) image binaire de la forme de la main.

(b) représentation polaire-paramétrique de la main en fonction de l'abscisse curviligne τ (distance ρ et angle θ). Le pouce se trouve dans un intervalle particulier d'angle et de distance (matérialisés par les lignes horizontales).

Figure V-13 : détection du pouce.

Etape 3 : adaptation des seuils. Des seuils sont définis sur les valeurs d'angles et de distances afin de délimiter la région où il est possible de détecter un pouce. Les angles qui décrivent la région du pouce sont tirés de statistiques sur la morphologie de la main [164]. En pratique, l'angle que forme le pouce avec l'horizontale (cf. Figure V-13.b) est compris entre 20° et 65°. Les distances minimum et maximum de la région du pouce sont extraites des mêmes considérations de dessin que celles utilisées précédemment [166]. Leurs valeurs sont déterminées de manière approximative (et par excès, afin de ne pas perdre le pouce) à 1/9 et 5/9 de la longueur maximale de la main.

Etape 4 : mesure du pouce. Si un doigt est détecté dans la zone définie par ces seuils, il s'agit du Pouce. La hauteur du pic correspondant est mesurée par rapport à la hauteur du minimum local qui le précède, c'est-à-dire, le creux

entre le pouce et l'index (cf. Figure V-13.b). Cette mesure permet de définir un indicateur chiffré de la présence du pouce, dont la valeur est 0 si aucun pouce n'est détecté et dont la valeur est celle de la hauteur du pouce sinon (cf. Figure V-13.b).

Notons qu'une méthode suffisamment élaborée pour permettre de compter les doigts présents dans l'image est aussi un descripteur de haut niveau. Il s'agit simplement d'un nombre, et le classifieur associe une classe à ce nombre [C5]. Couplé avec l'indicateur de la présence du pouce, il est possible de faire une classification complète sur les 8 premières Configurations (0 à 7, à l'exclusion de la 8ième), puisque qu'au sein de chacune des superclasses AVEC_POUCE et SANS_POUCE, aucune Configuration ne possède le même nombre de doigts déployés. Cette méthode a été proposée et testée dans [C5]. Elle souffre de deux inconvénients majeurs :

– la Configuration 8 possède le même nombre de doigts déployés que la Configuration 2 et par conséquent la classification n'est pas complète.

– Le repérage des doigts est très difficile sur les gants de couleur sombre. Le pouvoir de généralisation d'une telle méthode est donc trop faible pour qu'elle soit adoptée dans notre système.

Au chapitre des descripteurs de plus haut niveau, un grand nombre ont été proposés dans [136], et sont particulièrement adaptés à la description de formes de mains.

V.2.2.3 Attributs bas niveau : invariants de Hu

Dans [12], l'intérêt des invariants de Hu [32] pour la description des formes de la main est démontrée. Il s'agit de décrire la répartition de la masse δ de la forme de la main par la combinaison de plusieurs moments d'inertie, de telle sorte que la combinaison obtenue soit invariante aux similarités. La masse est définie par $\delta(x,y) = 1$ si le pixel de coordonnées (x,y) appartient à la main, et $\delta(x,y) = 0$ sinon. Soit $\bar{x}$ et $\bar{y}$ les coordonnées du CG de la main. Les moments d'inertie centrés (afin d'être invariants aux translations) d'ordre $p+q$ sont définis par :

$$m_{pq} = \iint_{x\,y} (x-\bar{x})^p (y-\bar{y})^q \, \delta(x,y)\, dx\, dy$$

Afin que ces moments soient invariants à l'échelle, ils sont normalisés [32] :

$$n_{pq} = \frac{m_{pq}}{m_{00}^{\frac{p+q}{2}+1}}$$

Enfin, 6 **invariants de Hu** peuvent être calculés ; ils sont invariants aux rotations et aux réflexions [32] :

$$S_1 = n_{20} + n_{02}$$

$$S_2 = (n_{20} + n_{02})^2 + 4 \cdot {n_{11}}^2$$

$$S_3 = (n_{30} - 3 \cdot n_{12})^2 + (n_{03} - 3 \cdot n_{21})^2$$

$$S_4 = (n_{30} + n_{12})^2 + (n_{03} + n_{21})^2$$

$$S_5 = (n_{30} - 3 \cdot n_{12}) \cdot (n_{30} + n_{12}) \cdot \left((n_{30} + n_{12})^2 - 3 \cdot (n_{03} + n_{21})^2\right) - (n_{03} - 3 \cdot n_{21}) \cdot (n_{03} + n_{21}) \cdot \left(3 \cdot (n_{30} + n_{12})^2 - (n_{03} + n_{21})^2\right)$$

$$S_6 = (n_{20} + n_{02}) \cdot \left((n_{30} + n_{12})^2 - (n_{03} + n_{21})^2\right) + 4 \cdot {n_{11}}^2 \cdot (n_{30} + n_{12}) \cdot (n_{03} + n_{21})$$

Il existe un 7ème invariant. Son signe permet de faire la distinction entre deux images miroirs, et de supprimer l'invariance aux réflexions :

$$S_7 = (3 \cdot n_{21} - n_{03}).(n_{30} + n_{12}).\left((n_{30} + n_{12})^2 - 3 \cdot (n_{03} + n_{21})^2\right) - (n_{30} - 3 \cdot n_{12}) \cdot (n_{03} + n_{21}) \cdot \left(3 \cdot (n_{30} + n_{12})^2 + (n_{03} + n_{21})^2\right)$$

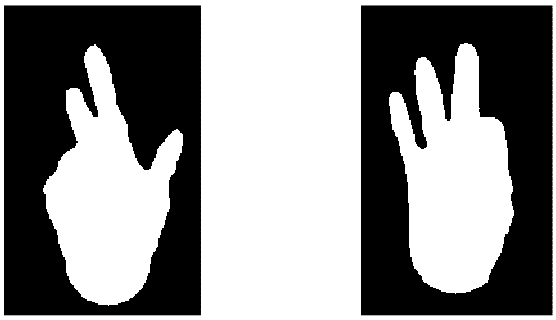


Figure V-14 : Configuration 7 et Configuration 3 très proches à une symétrie près.

Comme les problèmes liés à l'effet miroir ont été supprimés en ne traitant que des images de gauchers, il ne nous est pas nécessaire de discriminer des images miroirs. Néanmoins, ce septième invariant est intéressant pour deux autres raisons :

– Tout d'abord, sa valeur numérique absolue est un élément de description supplémentaire. Son utilisation est donc justifiée indépendamment de son signe et de la chiralité de la main.

– Ensuite, contrairement à ce que l'on pourrait croire, le signe en lui-même est intéressant : il se trouve que pour un certain type de codage, les formes de main correspondant aux Configurations 3 et 7 peuvent être très proches, à leur symétrie près (cf. Figure V-14).

Ainsi, l'ensemble des 7 descripteurs $\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$ est considéré pour la discrimination des Configurations du LPC.

V.2.2.4 Attributs bas niveau : descripteurs de Fourier-Mellin

Dans ce paragraphe, nous présentons de façon succincte l'utilisation de la *"Transformée de Fourier-Mellin (TFM) pour la mise en place de descripteurs multi-orientés et multi-échelles*" [28], telle qu'elle est décrite dans [28], et dans le brevet France Telecom correspondant [29].

La TFM d'une fonction f correspond à la représentation par coefficients de Fourier de la transformée de Mellin de f. Elle est définie pour toute fonction réelle positive $f(r,\theta)$ en coordonnées polaires (il s'agit de la forme à décrire), de telle sorte que la transformée de Mellin soit 2π-périodique :

$$M_f(q,s) = \int_{r=0}^{\infty} \int_{\theta=0}^{2\pi} r^{s-1} e^{-iq\theta} f(r,\theta)\, dr d\theta \quad \text{avec} \quad q \in \mathbb{Z}, s = \sigma + i\upsilon \in \mathbb{C}, \text{ et } i = \sqrt{-1}$$

Le module de la TFM à laquelle il a été appliqué le théorème du retard permet d'obtenir des descripteurs indicés par q et s invariants par rotation, et la normalisation par $M_f(0, \sigma)$ permet d'obtenir l'invariance d'échelle. L'invariance par translation est naturellement dérivée de la définition du repère (r,θ) pour la TFM (c'est ce que l'on appelle le centre de développement de la TFM) : il suffit que celui-ci soit attaché à la forme (on utilise le CG par exemple).

Appliquée aux images numériques, cette méthode nécessite la discrétisation de la TFM et la conversion de l'espace cartésien discret en un espace polaire. En pratique, $M_f(0, \sigma)$ est approchée par :

$$M_f(q,\sigma+i\upsilon) \approx \sum_{\substack{k,l \\ 0 \le (k^2+l^2) \le r^2_{\max}}} h_{p,q}(k,l) \cdot f\left(\bar{k}-k, \bar{l}-l\right)$$

avec
- $(\bar{k}, \bar{l})$ centre de développement de la TFM (ici, le CG de la main)
- $r_{\max}$ borne supérieure de r
- $h_{p,q}(k,l) = \dfrac{1}{(k^2+l^2)^{1-\frac{\sigma}{2}}} \cdot \exp\left(i.\left(\dfrac{p}{2} \ln(k^2+l^2) - q.\arctan\left(\dfrac{l}{k}\right)\right)\right)$

Les invariants obtenus satisfaisant la propriété de symétrie hermitienne, il est possible de diviser par presque deux le nombre de calculs. Ainsi, 33 descripteurs (invariants aux similitudes directes) sont déduits de 18 filtres $h_{p,q}$, avec p et q appartenant à A :

$$A = \left\{ (p,q) \middle| (q = 0; 0 \le p \le P) \cup (1 \le q \le Q; -P \le p \le P) \right\}$$

et avec σ = 1, $Q = 3$ et $P = 2$. Les choix des valeurs σ, Q et P, ainsi que du nombre de descripteurs (33) sont explicités dans [28]. Ces 33 descripteurs sont les Descripteurs de Fourier-Mellin (DFM). Pour plus de détails, se reporter à [28].

V.2.2.5 Evaluation des attributs de classification

Nous évaluons ici l'intérêt des jeux attributs précédemment décrits. Ceux-ci sont au nombre de trois : l'indicateur de présence du pouce, les invariants de Hu et les DFM.

L'indicateur de présence du pouce est évalué sur 960 ICC du corpus ETTRAN N. Elles sont choisies de telle sorte que leur distribution soit à peu près représentative de leur fréquence d'apparition en français et qu'il n'y ait aucune corrélation entre elles (cf. Tableau V-2). Ce corpus a été initialement séparé en deux corpus d'apprentissage et de test, comme cela est indiqué à la section III.1 p. 59. Cependant, cet algorithme indiquant la présence du pouce n'est basé sur aucun apprentissage. Son évaluation peut donc être réalisée sur l'ensemble des 960 ICC. Pour chaque image, la vérité terrain est déterminée en fonction de l'allure globale de la forme binaire de la main, sans tenir compte du codage réel. De cette manière, si en visionnant la vidéo de manière dynamique, ou si en regardant les images originales pour lesquelles il ne peut y avoir d'erreur de segmentation, il apparaît que la vérité terrain établie est fausse (la Configuration reconnue par l'opérateur se révèle ne pas correspondre au geste réalisé), celle-ci n'est pas remise en cause. Cela signifie que l'on évalue la pertinence des attributs en fonction de ce que l'expert perçoit, et que c'est la seule référence. Cela permet de n'évaluer que l'indicateur de présence du pouce, à l'exclusion des autres algorithmes, qui pourrait introduire un biais supplémentaire. Ainsi, en comparant l'indication de présence du pouce qui est fournit par l'algorithme, et ce que perçoit l'opérateur, il est possible de déterminer la qualité de l'attribut. Les résultats sont les suivants : 94% de bonnes détections pour 2% de fausses alarmes.

Tableau V-2 : détails des 960 ICC issues du corpus ETTRAN N.

Configuration	Corpus 1 (Apprentissage)	Corpus 2 (Test)	Total
0	37	12	49
1	94	47	141
2	64	27	91
3	84	36	120
4	72	34	106
5	193	59	252
6	80	46	126
7	20	7	27
8	35	23	58
Total	**679**	**291**	**970**

Pour évaluer la pertinence de chacun des jeux d'attributs correspondant à des descripteurs de bas niveau, nous utilisons encore une fois l'algorithme de classification (avec sa phase d'apprentissage). Celui-ci est entièrement décrit au paragraphe V.2.3 (p. 147). Les taux de classification sont les suivants : 91.8% pour les invariants de Hu et 96.8% pour les DFM. Le résultat est clairement en

faveur des DFM. Il faut cependant noter que ceux-ci sont au nombre de 33 contre 7 pour les invariants de Hu. Il est donc difficile de comparer le pouvoir discriminant par attribut.

Du résultat comparé de l'indicateur de présence du pouce et de ces deux jeux de descripteurs, la combinaison des DFM et de l'indicateur de présence du pouce n'a pas grand intérêt. En effet, les DFM seul sont plus discriminants sur un nombre plus grand de classes. En conséquence, l'utilisation combinée des DFM et de l'indicateur de présence du pouce risque de donner un taux de classification inférieur. Cela n'est pas garanti d'un point de vue théorique (cela dépend de la capacité informative de l'indicateur de présence du pouce par rapport aux DFM), mais c'est en pratique ce que nous avons constaté. En revanche, la combinaison des invariants de Hu avec l'indicateur de présence du pouce a de grandes chances d'être intéressante. La méthode mise en place à cette fin est décrite au V.2.3.6 (p. 168). Cette méthode est faite de telle sorte, qu'il est cohérent de comparer les résultats qu'elle donne avec les scores que nous venons de présenter. Il en résulte que l'ensemble {invariants de Hu + indicateur de présence du pouce} permet un taux de reconnaissance de 92.8%, supérieur à celui fourni par les invariants de Hu seuls. Cependant, il apparaît clairement que les DFM seuls restent plus performants. De plus, ces derniers sont suffisamment performants à l'égard de notre problème de reconnaissance, et par conséquent, c'est ce jeu d'attributs qui est conservé.

V.2.3 Méthodes de classification

Relativement au problème de la reconnaissance de la Configuration, nous avons jusqu'à présent détaillé :

– les classes (il s'agit des Configurations),

– la manière de réduire la variabilité de la forme de la main (par suppression du poignet),

– le jeu d'attributs de classification à utiliser (les DFM).

Dans ce paragraphe, nous abordons le dernier aspect, à savoir la méthode de classification. Nous commençons par une description des principales méthodes de classification existantes (V.2.3.1). Ensuite, nous détaillons la méthode retenue (V.2.3.2). Il s'agit des **Support Vector Machines** (**SVM**). Cet algorithme est très efficace, cependant, il souffre de certains inconvénients dans le cas de problème à plus de deux classes. Nous proposons de les résoudre partiellement en considérant les SVM dans le formalisme de fonctions de croyance (appendice A p. 255). Pour cela, nous décrivons brièvement ce formalisme (V.2.3.3) ainsi que l'état de l'art au sujet de son application aux SVM (V.2.3.4), puis nous présentons nos propres travaux à ce sujet (V.2.3.5) et nous en évaluons expérimentalement l'intérêt (V.2.3.6). Finalement, nous montrons que la méthode résultante est aussi intéressante pour la fusion de plusieurs classifieurs hétérogènes. C'est grâce à cela que nous avons pu combiner

l'indicateur de présence du pouce et les invariants de Hu. Cette méthode est présentée et évaluée au V.2.3.7.

V.2.3.1 Inventaires des méthodes de classification

Un classifieur est un objet qui connaît la manière dont les attributs d'un item doivent être corrélés pour appartenir à chacune des classes de *C*. Cette connaissance vient de l'accumulation d'expériences similaires dont des règles générales sont tirées par un processus d'inférence. Cette inférence peut être réalisée par le programmeur lui-même (on parle alors de **système expert**), ou par la machine selon un processus défini par le programmeur (on parle alors d'**apprentissage** ou de "**machine learning**"). Des descripteurs de haut niveau ou des seuillages sont des systèmes experts. Il existe de nombreuses méthodes d'apprentissage :

– **Le K-NN** (K-Nearest Neighbors ou K plus proches voisins) : il s'agit de repérer la position de chaque item dans l'espace des attributs et de trouver dans le corpus d'apprentissage, les K plus proches items de celui à classer. Une règle de décision sur la classe de ces K plus proches voisins est utilisée pour associer une classe à l'item.

– **Les réseaux de neurones** : il s'agit de petites entités (les neurones) ne pouvant effectuer que des opérations très simples, mais dont la mise en réseau permet de faire des tâches plus complexes (pour une revue complète sur les réseaux de neurones, cf. [66]). Parmi toutes ces tâches, celles d'apprentissage et de décision sont les plus utilisées. Parmi tous les types de réseaux de neurones, les plus simples et les plus répandus sont les **perceptrons multi-couches** (PMC). Un perceptron est un automate permettant de calculer un barycentre pondéré de plusieurs entrées, et de comparer sa position à un seuil. Si chaque entrée représente un attribut, le calcul d'un certain barycentre permet la mise en valeur d'un certain type de corrélation. Ensuite, sa comparaison avec un seuil permet l'émission d'une décision par rapport à une règle associée au seuil. Cette décision est simple puisqu'elle ne correspond qu'à une décision entre deux classes. En revanche, l'utilisation d'un banc (ou d'une couche) de plusieurs perceptrons permet d'effectuer une classification entre plus de deux classes (**multi-classification**). De plus, cette séparation est linéaire dans l'espace des attributs ; mais la mise en réseaux de trois couches en un treillis de perceptrons permet d'effectuer des séparations non-linéaires, comme cela est illustré schématiquement sur la Figure V-15. Toute la difficulté de la mise en place d'un PMC réside dans la pondération des entrées de chaque perceptron en fonction des données à partir desquelles l'apprentissage doit être inféré. En effet, ce sont ces poids d'entrées (appelées **coefficients synaptiques**) qui permettent d'effectuer une séparation correcte. En pratique, les items du corpus d'apprentissage sont présentés itérativement au PMC qui effectue une classification sur chacun d'eux. En fonction de la qualité de celle-ci, une information de modification des coefficients synaptiques est rétro-propagée sur le réseau. Cette itération est poursuivie jusqu'à remplir un critère de convergence. Le second point délicat de l'utilisation du PMC est le choix d'une

architecture particulière (nombre de perceptrons par couche). Une alternative à l'utilisation de PMC est l'utilisation d'une seule couche précédée d'un algorithme de redéfinition de l'espace des attributs à l'aide de fonction noyau (cf. [70] et paragraphe suivant p. 150).

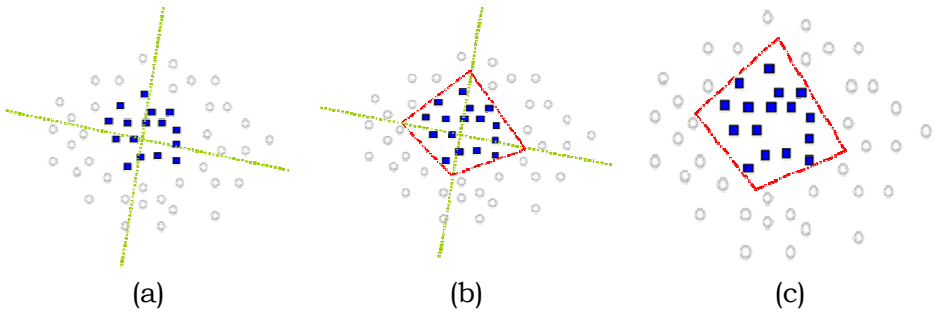


Figure V-15 : représentation schématique permettant simplifier le fonctionnement d'un PMC : (a) séparations linéaires par la première couche de perceptrons, (b) Suivant chaque classification de la première couche, une séparation linéaire différente est appliquée, (c) les classes sont reconstituées par une dernière couche.

– **Boosting** : il s'agit de méthodes permettant la fusion de plusieurs classifieurs naïfs [64]. Voici une méthode classique : un premier classifieur naïf meilleur qu'une classification aléatoire est entraînée sur une partie de l'ensemble d'apprentissage. Ensuite, un deuxième classifieur est déterminé de telle sorte qu'il soit le plus informatif par rapport au premier classifieur. Pour cela, on définit son ensemble d'apprentissage de telle sorte que la moitié des items qu'il contient soit bien classée par le premier classifieur et l'autre moitié mal classée par ce même classifieur. Enfin, un troisième classifieur est déterminé pour prendre une décision dans les cas où les deux premiers classifieurs ne sont pas d'accord. Il existe bien sur de nombreuses sophistications de cet algorithme, dont la plus célèbre est AdaBoost [82].

– **Les méthodes d'optimisation combinatoire** : il s'agit de voir le taux de classification que donne un séparateur comme la valeur d'une fonction objective qu'il faut maximiser. Comme cette maximisation (sous contraintes de l'apprentissage) est fortement non convexe, une méthode linéaire type **simplex** est inefficace. On utilise donc de nombreuses heuristiques dont le but est de parcourir la surface de la fonction objective (dont la topologie est inconnue) à la recherche de son maximum. Parmi ces méthodes, on trouve le **tabou** (ou le **recuit-simulé**) [67], les **algorithmes génétiques** [67], et les **colonies de fourmis** [67].

– **Les modèles génératifs, ou classifieurs unaires** : il s'agit de confronter chaque item à une série de modèles génératifs de chaque classe. La classe retenue est celle correspondant au modèle qui a la plus forte probabilité de générer l'item à classer. Les HMM [128] sont l'exemple le plus classique.

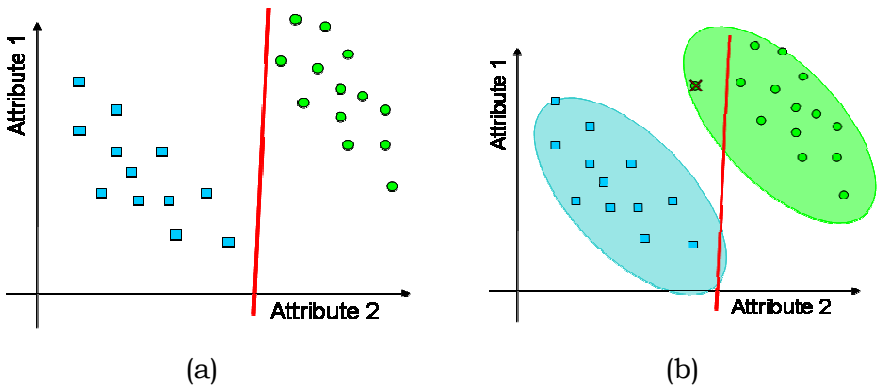
– **Les Séparateurs à Vastes Marges (SVM – Support Vector Machine)** : il s'agit de baser la définition d'un **hyperplan** séparateur entre deux classes sur

un processus de double optimisation combinatoire [57], [58], [60]. Le principe est de maximiser la distance entre les deux classes et l'hyperplan séparateur pour éviter les problèmes de biais à l'apprentissage. Comme il s'agit de la méthode retenue, nous donnons les aspects calculatoires dans appendice C.5 p. 301, et lui consacrons le sous-paragraphe suivant.

D'une manière générale, les différents algorithmes présentés ici sont capable de performances équivalentes. Notre choix s'est porté sur les SVM pour la simple raison que nous possédions l'expertise et les programmes informatiques nécessaires à une mise en place rapide et efficace.

V.2.3.2 Séparateurs à Vastes Marges

Comme illustré sur la Figure V-16a et sur la Figure V-16b, la définition d'un séparateur efficace pour un apprentissage donné peut très bien se révéler inadaptée pour des situations inconnues de l'apprentissage : le pouvoir de généralisation est faible. Cela est dû au fait que toute représentation statistique de la connaissance implique un biais. Si celui-ci est plus important que la cohérence interne à la corrélation des attributs pour chaque classe, cela peut entraîner de mauvaises classifications.



(a) (b)

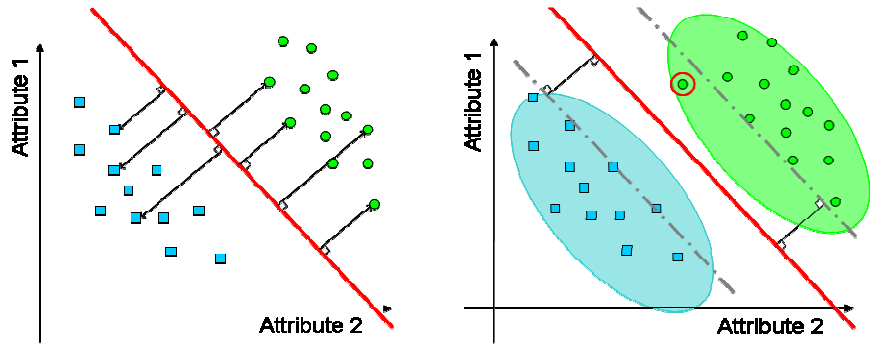
Figure V-16 : classifieur pour lequel le biais du corpus d'apprentissage entraîne une erreur de classification.

Comme il est impossible de prévoir à l'avance l'éventuel biais d'un corpus d'apprentissage, l'algorithme du SVM propose de définir l'hyperplan séparateur de telle sorte qu'il soit le plus loin possible de tous les items du corpus d'apprentissage (Figure V-17a). Ainsi, tant que la distance (appelée la **marge**) entre chacune des classes et l'hyperplan est supérieure au biais, il n'y a pas d'erreur de classification (Figure V-17b).

En pratique, il y a de nombreuses difficultés auxquelles il faut faire face lors de l'utilisation des SVM :

– Il arrive qu'il y ait un certain recouvrement entre les classes à discriminer (Figure V-18a). Dès lors, la mise en place directe d'un hyperplan séparateur sous contrainte d'éloignement est impossible. Cela peut être résolu par

plusieurs méthodes [61]. La plus connue est l'utilisation de **variables ressorts** dans le problème d'optimisation. Dans un tel cas, on parle de **C-SVM**.



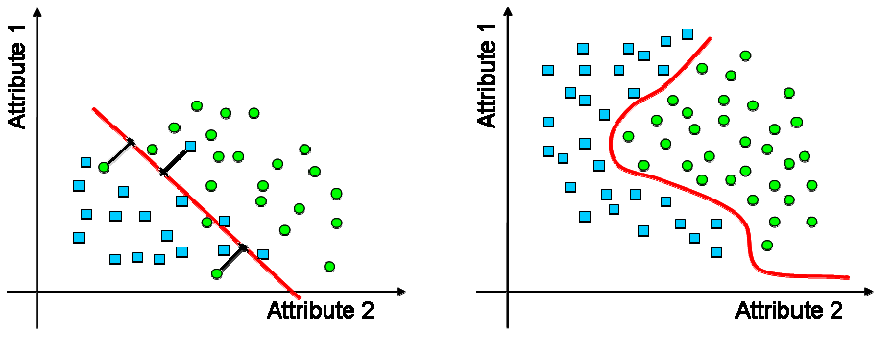
(a) (b)

Figure V-17 : (a) optimisation combinatoire de la position de l'hyperplan sous la contrainte du contenu du corpus d'apprentissage. (b) Le SVM permet une bonne classification malgré le biais de l'apprentissage.

– Dans les cas où les classes ne sont pas linéairement séparables dans l'espace des attributs (Figure V-18b), la définition directe d'un hyperplan séparateur n'est pas possible. Cela est résolu par la **méthode des noyaux**, dont le but est d'augmenter la dimension de l'espace des attributs par utilisation d'une application φ non linéaire dont l'espace image est plus grand que l'espace des attributs. La non-linéarité de la fonction ajoutée à l'augmentation de la taille de l'espace permet alors de trouver un hyperplan séparateur. Afin d'éviter un coût calculatoire trop important dû à l'augmentation du nombre de dimensions, on utilise une classe particulière $\mathcal{A}$ d'applications φ, telles que :

$$\forall \varphi \in \mathcal{A}, \quad \varphi(x) \cdot \varphi(y) = K(x, y)$$

où K est une fonction noyau. Le théorème de Mercer [71] énonce que toute fonction noyau continue, symétrique, et semi-définie positive peut s'exprimer comme un produit scalaire (forme bilinéaire symétrique et strictement positive) dans un espace de plus grande dimension. Ainsi, dans l'optimisation de l'hyperplan, φ n'intervient jamais, et seulement son produit scalaire.



(a) (b)

Figure V-18 : (a) classes non séparables (b) classes non linéairement séparables.

– Le dernier problème avec les SVM est qu'ils sont basés par définition sur une optimisation combinatoire qui ne se généralise pas bien dans le cas où il y a plus de deux classes à discriminer [59]. Il existe donc toute une littérature sur la manière de combiner des classifieurs binaires (et particulièrement des SVM) afin de pouvoir traiter des problèmes à plus de deux classes. Certaines méthodes proposent de décomposer le problème en plusieurs sous-problèmes pour chacun desquels il est possible d'appliquer une classification binaire. La décomposition peut soit être hiérarchique, en s'appuyant sur une arborescence ou un dendrogramme [2], soit être basée sur une résolution inférentielle du problème [60], par une classification qui suit un DAG (Direct Acyclic Graph, c'est-à-dire un arbre orienté). Cependant, la plupart des méthodes procèdent différemment : elles proposent de décomposer l'espace de classification plutôt que le problème. Soit C l'ensemble des classes. Il s'agit alors de projeter le corpus d'apprentissage sur un grand nombre de sous-ensembles de classes binaires $\{C^i, C^j\}$, avec C^i et C^j telles que :

$$\forall (i,j,k) \in \lfloor 1, |C| \rfloor^3, \quad C^i \cap C^j = 0 \quad \text{avec} \quad C^i, C^j \in 2^C \quad \text{et} \quad 2^C = \left\{ C^k / C^k \subseteq C \right\}$$

2^C est appelé le **powerset** de C. Pour chaque projection du corpus d'apprentissage (celui-ci n'est d'ailleurs pas forcément de taille plus petite que le corpus original), un classifieur donne une réponse partielle. Ensuite, l'ensemble des réponses partielles est fusionné et cela permet de fournir une décision sur la classe. Les différences au sein des méthodes suivant ce principe se trouvent à deux niveaux. On distingue le schéma suivant lequel les corpus sont projetés sur C, et la technique de fusion des réponses de l'ensemble des SVM. Concernant le schéma de projection, les deux méthodes les plus connues sont :

– **Schéma 1vs1** : soit N le nombre de classes ($N = |C|$). $N.(N-1)/2$ classifieurs sont utilisés. Chacun est entraîné sur un corpus ne contenant que deux classes C^i et C^j. Les autres items du corpus d'apprentissage ne sont pas utilisés.

– **Schéma 1vsAll** : soit N le nombre de classes. N classifieurs sont utilisés. Chacun d'eux est entraîné sur une projection du corpus d'apprentissage contenant le corpus en entier, mais re-labélisé en C^i et $C \backslash C^i$.

Concernant les techniques de combinaison des SVM, les plus utilisées sont les suivantes :

– **Vote** : dans cette technique, chaque SVM a droit à une voix. Il l'attribue à la classe correspondant à sa décision partielle. Dans le cas où cette classe n'est pas un élément singleton du powerset de C, la décision peut éventuellement subir une pondération en fonction du cardinal de la décision en question ; cela permet de donner un poids juste à chaque vote et d'éviter les égalités dans le processus de comptage de ces derniers.

– **Estimation des probabilités *a posteriori*** : dans cette technique, il s'agit d'associer à chaque SVM une probabilité conditionnée par l'ensemble des classes sur lequel il travaille : $p_{ij}(x) = p(\text{classe}(x) = C^i \mid \text{classe}(x) \in \{C^i, C^j\})$.

Ensuite, toutes les probabilités conditionnelles que l'item x soit de la classe i sont combinées :

$$p_i(x) = \frac{1}{NbSVM} \sum_{j \neq i} p_{ij}(x)$$

Où $NbSVM$ est le nombre de classifieurs mis en place dans le choix du schéma de projection. Pour déterminer $p_{ij}(x)$, on utilise une fonction f de la distance entre l'item à classer et l'hyperplan séparateur. En pratique, f est une fonction sigmoïde qui est paramétrée en parallèle du processus d'apprentissage et de la définition de l'hyperplan.

Dans [60], il apparaît que vote ou estimation de probabilité donnent des résultats équivalents, bien qu'il ne soit pas évident de les comparer en raison des différences au niveau de leurs apprentissages respectifs.

En plus de la décomposition du problème ou de l'espace des classes, il existe une troisième voie, assez récente, basée sur des travaux plus anciens, avec la promesse d'une grande efficacité selon le critère qualité/calcul. Il s'agit de la méthode **ECOC** (Error Correcting Output Code ou **CCE** pour Codes Correcteurs d'Erreurs) [76]. Dans cette approche, chaque classe est codée comme un mot binaire où chaque bit correspond à la sortie attendue d'un classifieur pour les items de la classe en question. Pour chaque item à classer, un mot de code correspondant à la sortie de chaque SVM est fourni et ce mot de code est comparé à celui de chaque classe, en utilisant la distance de Hamming. Cette méthode est résumée par une matrice CCE telle que celle de la Figure V-19. Dans une telle matrice, chaque colonne représente un classifieur et chaque ligne une classe. Une valeur v dans la cellule (i, j) de la matrice doit être interprétée de la manière suivante : durant l'apprentissage, la classe C^i correspond à la $v^{ème}$ classe pour le classifieur K^j ; durant la phase de test, un item de la classe C^i doit être labellisé v par le classifieur K^j. v peut prendre les valeurs 0, 1 ou ψ, cette dernière représentant l'absence d'utilisation d'une classe pour un apprentissage, et l'inintérêt d'un classifieur pour un test.

$$\begin{array}{c} \begin{array}{ccccccc} K^1 & K^2 & K^3 & K^4 & K^5 & K^6 & K^7 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{array} \\ \begin{array}{c} C^1 \rightarrow \\ C^2 \rightarrow \\ C^3 \rightarrow \\ C^4 \rightarrow \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Figure V-19 : exemple d'une matrice CCE qui représente la structure d'un banc de 7 SVM pour un problème à 4 classes.

Il se trouve que les schémas 1vs1 et 1vsAll utilisés avec une procédure de vote sont des cas particuliers de CCE ; leur matrices CCE sont décrites sur la Figure V-20. Le principal problème des CCE est la définition de la matrice. En effet, contrairement aux codes correcteurs d'erreurs quand ils sont utilisés pour la

détection ou la correction d'erreur de transmission, une bonne séparation des mots de code (correspondant aux lignes de la matrice) n'est pas suffisante. Il faut aussi que les définitions des classifieurs soient bien séparées (colonnes de la matrice).

$$\begin{array}{c} \quad K^1 \; K^2 \; K^3 \; K^4 \\ \quad \downarrow \; \downarrow \; \downarrow \; \downarrow \\ \begin{array}{l} C^1 \rightarrow \\ C^2 \rightarrow \\ C^3 \rightarrow \\ C^4 \rightarrow \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

(a)

$$\begin{array}{c} \quad K^1 \; K^2 \; K^3 \; K^4 \; K^5 \; K^6 \\ \quad \downarrow \; \downarrow \; \downarrow \; \downarrow \; \downarrow \; \downarrow \\ \begin{array}{l} C^1 \rightarrow \\ C^2 \rightarrow \\ C^3 \rightarrow \\ C^4 \rightarrow \end{array} \begin{pmatrix} 1 & 1 & 1 & \psi & \psi & \psi \\ 0 & \psi & \psi & 1 & 1 & \psi \\ \psi & 0 & \psi & 0 & \psi & 1 \\ \psi & \psi & 0 & \psi & 0 & 0 \end{pmatrix} \end{array}$$

(b)

Figure V-20 : matrices CCE des méthodes (a) 1vsALL + vote et (b) 1vs1 + vote.

D'après les expériences de [75], [76], [79], il semble que toutes les méthodes décrites dans ce paragraphe ont des résultats plus ou moins équivalents ; quelques-unes sont plus efficaces dans certains cas, d'autres sont moins coûteuses en temps de calcul, etc. En conséquence, il n'y a pas de méthode dominante en général et faisant consensus dans l'état de l'art. Le choix doit donc être guidé par l'application.

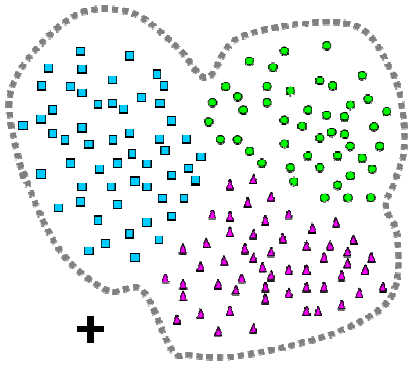


Figure V-21 : illustration de la notion de rejet.

D'un point de vue topologique, la multi-classification pose trois problèmes qui ne sont pas encore complètement résolus : le premier est la définition d'une **classe de rejet**. Dans le cas où il est nécessaire de faire la différence entre une classe et le reste du monde (détection de visage par exemple), ou dans le cas où les classes ne sont pas exhaustives, il doit être possible d'effectuer un rejet, ou de classer l'item dans la classe "autre". D'un point de vue géométrique, il existe une hypersurface faisant la séparation entre les classes définies et le reste du monde (cf. Figure V-21). En pratique, définir une telle hypersurface est difficile, puisque les SVM nécessitent des exemples d'apprentissage représentatifs (et peu biaisés) des deux côtés [73]. Récemment, des SVM à une classe sont apparus (les SVRDM – Support Vector Regression and Discrimination Machine) [72]. Ils permettent de déterminer une séparation entre un ensemble d'items et le reste de l'espace d'attributs. Comme le nom l'indique, c'est un puissant outil de

régression non linéaire, mais aussi un moyen efficace d'implanter une classe de rejet. Cependant comme toutes les méthodes à une classe (méthodes génératives), leur utilisation pour de la classification est parfois difficile à paramétrer.

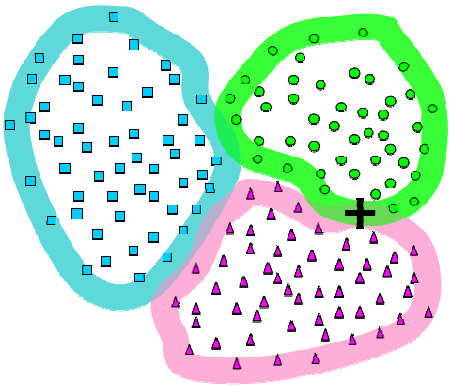


Figure V-22 : illustration d'une situation d'indécision.

Le deuxième problème est la **situation d'indécision** : cela arrive quand un item est très proche d'un des hyperplans (à l'intérieur des marges). Même si un banc de SVM propose une décision sur la classe, celle-ci est très peu fiable, comme cela est illustré sur la Figure V-22.

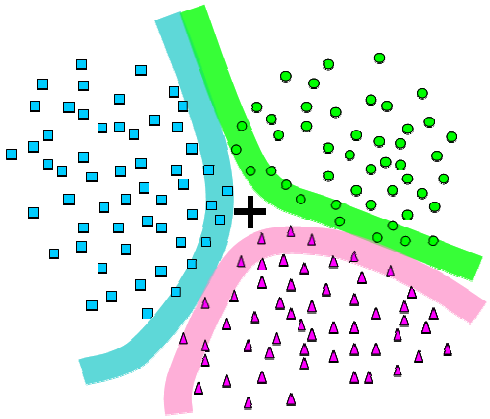


Figure V-23 : illustration d'une boucle d'incertitude.

Finalement, il y a le problème de la **boucle d'incertitude** : cela arrive quand 3 classifieurs ou plus ont des réponses contradictoires. Par exemple, si l'on utilise la notation suivante pour exprimer que le $i^{\text{ème}}$ SVM qui travaille sur la discrimination des classes j et k donne la préférence à la classe j pour un item donné :

$$C^j \succ_i C^k$$

alors, une boucle d'incertitude est une situation telle que (cf. Figure V-23) :

$$C^A \overset{1}{\succ} C^B$$

$$C^B \overset{2}{\succ} C^C$$

$$C^C \overset{3}{\succ} C^A$$

Dans le cas de notre application de reconnaissance de gestes, la classification des formes de main peut parfois être délicate. Comme la labellisation des ICC peut être sujette à erreurs, ou comme il est nécessaire de traiter certaines ITCM, il est possible que le processus de classification soit appliqué à des images de transition, contenant une forme de main qui ressemble à un mélange de deux Configurations. Pour de telles images, une décision classique, attribuant une unique classe est inadaptée. En conséquence, une réponse permettant une réponse partielle, c'est-à-dire un doute, serait intéressante. Le **formalisme crédal** (encore appelé **formalisme des fonctions de croyance**, ou **formalisme évidentiel**) permet une telle gestion du doute (une présentation complète de ce formalisme est donnée en appendice A p. 255, ainsi qu'une brève introduction, dans le sous-paragraphe suivant).

Nous proposons donc de modifier l'algorithme de classification, de telle sorte qu'il fournisse une réponse de type crédal. Cependant, dans le cas d'ICC correctement labellisées, il faut être capable de les reconnaître, et ce, sans ajouter une hésitation qui n'a pas lieu d'être. En conséquence, il doit être possible d'exprimer une incertitude, mais il doit aussi être possible de fournir une décision complète (unique et sans hésitation), et celle-ci doit être capable de performances au moins équivalentes à une procédure de décision classique.

Dès lors, il s'agit de mettre en place une méthode crédale ayant les avantages que nous venons de mentionner dans le cas d'images sujettes à hésitation, tout en assurant que cette méthode soit équivalente à celles décrites précédemment dans cette section, lorsque les images ne prêtent pas à confusion : la méthode crédale n'est intéressante que si elle satisfait ce double objectif.

V.2.3.3 Aperçu des fonctions de croyance

Ici, nous présentons sommairement et de manière peu formelle les fonctions de croyance ainsi que leur intérêt pour la reconnaissance de forme de main. Nous renvoyons le lecteur à [86], [92], ainsi qu'à l'appendice A p. 255 où de plus amples détails sont donnés, et où l'ensemble des concepts clefs sont rassemblés formellement. En outre, nous y proposons de nombreuses références vers les publications des auteurs ayant contribué à la mise en place de ce système théorique.

Soit X une variable et $\Omega_X = \{h_1, \ldots, h_N\}$ un ensemble de N hypothèses exhaustives et exclusives que peut satisfaire la variable X. Ω_X est le **cadre de discernement** de X, ou tout simplement, le **cadre** de X. Par exemple, X peut être la variable *forme de main* et Ω_X peut être l'ensemble des *Configurations possibles du LPC*. Une **masse de croyance,** ou une **fonction de croyance** (FC en abrégé) m associée à la variable X est une application de 2^{Ω_X} (le **powerset** de Ω_X) qui

représente la croyance que l'on peut placer sur les hypothèses de Ω_X, et dont la somme vaut 1 :

$$m : \begin{cases} 2^{\Omega_X} \rightarrow [0,1] \\ A \mapsto m(A) \end{cases} \qquad \text{avec} \qquad \begin{cases} \sum_{A \subseteq \Omega} m(A) = 1 \\ m(\varnothing) = 0 \end{cases}$$

$m(A)$ représente la croyance que l'on place exactement en A, et non en une partie plus grande ou plus petite du cadre.

Ainsi, une FC associée à la variable forme de la main est une application qui a toute ensemble non vide de Configuration associe un score correspondant à la croyance que l'on a que la forme de main soit une des Configurations de l'ensemble. Ainsi, il est possible d'exprimer facilement une situation d'ignorance entre les différentes Configurations de l'ensemble en question. Cette ignorance (une absence d'information) engendre une hésitation, qui est différente de l'hésitation due à une chance égale de réalisation. Cette dernière est une situation "d'équi-croyance", intuitivement équivalente à une situation d'équiprobabilité lors d'un raisonnement fréquentiel.

Remarquons la parenté conceptuelle qu'il y a entre une FC et une fonction d'appartenance telle que définie en logique floue. Ainsi, une FC peut-être intuitivement vue comme une fonction chiffrant l'appartenance d'une variable au groupe des hypothèses auxquelles on croit.

La combinaison de 2 FC provenant de 2 sources d'informations indépendantes permet de calculer la FC correspondant à la fusion des deux sources d'informations. Elle se calcule en utilisant la **règle de combinaison de Dempster**, définie de la manière suivante :

$$\begin{aligned} (\cap) : \mathcal{B}^{\Omega_X} \times \mathcal{B}^{\Omega_X} &\rightarrow \mathcal{B}^{\Omega_X} \\ m_1 \ (\cap) \ m_2 &\mapsto m_{(\cap)} \end{aligned}$$

avec $\mathcal{B}^{\Omega_X}$ l'ensemble des FC définies sur Ω_X et avec :

$$m_{(\cap)}(A) = \frac{1}{1-\mathcal{K}} \cdot \sum_{A=A_1 \cap A_2} m_1(A_1) \times m_2(A_2) \qquad \forall A \subseteq 2^{\Omega_X}$$

où la constante de normalisation,

$$\mathcal{K} = \sum_{\varnothing=A_1 \cap A_2} m_1(A_1) \times m_2(A_2)$$

quantifie l'incohérence entre les FC ainsi combinées. Certains auteurs proposent de ne pas normaliser le résultat de la combinaison de Dempster, et d'associer la valeur de la constante de normalisation à $m_{(\cap)}(\varnothing)$ (cf. appendice A p. 255). Ainsi, $m_{(\cap)}(\varnothing)$ traduit l'incohérence interne à la FC $m_{(\cap)}$.

Dans le cas où une décision unique doit être prise, il existe plusieurs méthodes. La **Transformée Pignistique** (**PT**) est l'une des plus courantes. Elle permet de

convertir la FC en une probabilité sur laquelle une décision est prise en choisissant l'hypothèse de probabilité maximum. La Transformée Pignistique PigT d'une FC m est définie de la manière suivante :

$$\text{PigT}: \quad \begin{array}{l} \mathcal{B}^{\Omega} \to \mathcal{P}r^{\Omega} \\ m(.) \mapsto \text{BetP}(.) \end{array} \qquad \text{avec} \quad \text{BetP}(h) = \frac{1}{1-m(\emptyset)} \sum_{h \in A,\, A \subset \Omega} \frac{m(A)}{|A|} \quad \forall h \in \Omega$$

avec $\mathcal{B}^{\Omega}$ et $\mathcal{P}r^{\Omega}$ désignant respectivement l'ensemble des FC et celui fonctions de probabilité sur Ω, et avec $|A|$ désignant le cardinal de A. BetP est appelée **Probabilité Pignistique**.

Un problème de multi-classification par SVM est résolu en considérant la combinaison de plusieurs SVM effectuant autant de classifications binaires. D'une manière générale, chaque classifieur binaire peut alors être vu comme une source d'information partielle par rapport au problème complet de la multi-classification. La combinaison des SVM est donc un problème de fusion de données. Comme nous venons de le voir, les fonctions de croyance constituent un cadre particulièrement riche pour effectuer une telle fusion. Cela permet d'obtenir une modélisation fine dans laquelle l'éventuel conflit entre les sources d'information est directement géré dans la combinaison de Dempster. Ainsi, considérer la combinaison de SVM dans un cadre crédal possède deux avantages :

– Tout d'abord, le résultat est une fonction de croyance, ce qui nous permet de gérer les cas douteux où la forme de main à reconnaître est un mélange de Configurations.

– Ensuite, la gestion des éventuels conflits entre les différents classifieurs est automatique, de sorte que de nombreux cas d'indécision sont résolus. Ainsi, même dans le cas où la décision est certaine et exempte de doute (via l'application de la PT), celle-ci a toutes les raisons d'être performante.

Au premier abord, il semble donc que l'utilisation de SVM dans un cadre crédal soit une solution particulièrement adaptée à la reconnaissance de la Configuration.

V.2.3.4 Etat de l'art sur l'amélioration proposée : combinaison de SVM dans un cadre crédal

L'idée d'associer une fonction d'appartenance à la sortie d'un SVM (les Fuzzy-SVM) est relativement répandue ([78], [76]). L'appartenance à une classe par rapport à une autre est en effet intuitivement modélisable par une fonction décroissante de la distance à l'hyperplan du SVM. Le principal intérêt de cela est de prendre une décision dans le cas des boucles d'incertitude. Il y a plusieurs stratégies, mais à notre connaissance aucune ne propose de considérer la fonction d'appartenance comme une fonction de croyance, comme nous l'avons proposé dans [C3], [C2]. Cela a pourtant deux avantages principaux : (1) la possibilité de prendre une décision partielle sur la classification, et (2) celle de

fusionner la sortie du classifieur avec d'autres classifieurs ou connaissances par l'utilisation d'une combinaison de Dempster, comme cela est préconisé par Appriou dans [112].

La combinaison de classifieurs en général est un sujet assez traité [74], [81], et parmi les possibilités de combinaison, l'utilisation du formalisme crédal est toujours en bonne place [112], [83], [84], [77]. La plupart du temps, la méthode consiste à utiliser la sortie d'un classifieur comme une source d'information pour créer une fonction de croyance, ou directement à convertir la sortie du classifieur (déterministe, probabiliste, etc.) en une fonction de croyance. Pour cela, notons l'existence des Transformées Pignistiques Inverses [116], dont chacune fournit une famille de solutions possibles en fonction d'un type de fonction de croyance à obtenir. Ensuite, au sein de chaque famille, il faut encore déterminer quelle fonction de croyance sélectionner. Cela peut sembler fastidieux, mais c'est tout à fait cohérent avec l'aspect fortement non injectif de la Transformée Pignistique. Une autre méthode est d'utiliser la matrice de confusion sur un corpus de validation pour déterminer le degré de fiabilité d'une réponse du classifieur, et donc définir la manière dont la masse de croyance doit être dispersée sur des éléments focaux de grande taille.

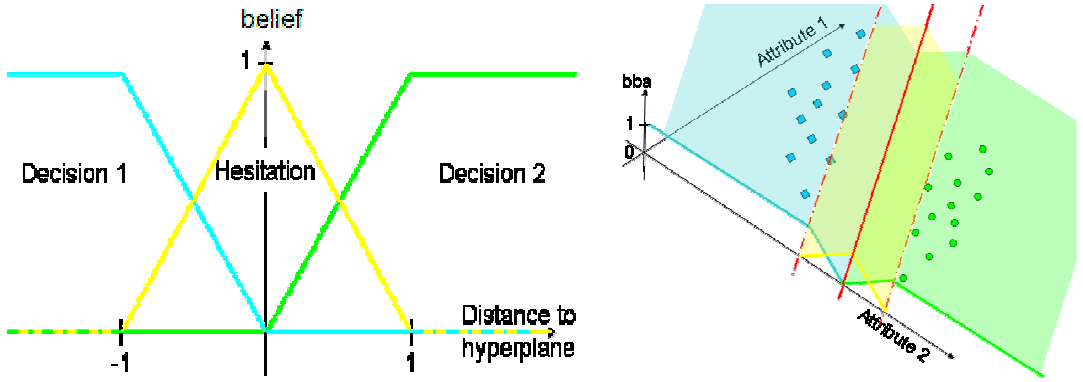
Pour ce qui est de la fusion de classifieurs binaires à des fins de multi-classification dans le cadre crédal, notons [106], dont l'inventaire précis de l'état de l'art est aussi intéressant que la méthode proposée. Sur le principe, Quost propose de considérer chaque sortie de classifieur binaire comme le résultat d'une transformée particulière (appelée Réduction) de la fonction de croyance finale que doit donner la combinaison de tous les classifieurs. Il serait d'ailleurs intéressant de comparer l'approche que nous avons proposée dans [C2] et [C3], et la sienne, et de voir si d'une manière ou d'une autre elle n'englobe ou ne généralise pas ce que nous proposons. Même si c'est le cas, la simplicité de la méthode que nous proposons la rend accessible aux néophytes des fonctions de croyance, et devient de ce fait, d'un intérêt comparable aux autres méthodes de multi-classification par SVM. Un autre élément original des travaux de Quost est l'utilisation de SVRDM afin d'améliorer la multi-classification par SVM selon un schéma 1vs1, mais sans pour autant implanter de classe de rejet. En effet, il s'agit d'utiliser un SVRDM avec chaque classifieur pour déterminer si celui-ci est compétent ou non pour émettre un avis sur l'item à classer. Les expériences décrites montrent une augmentation certaine du taux de classification au prix de l'utilisation d'un nombre deux fois plus grand de classifieurs.

L'alternative est d'utiliser directement des classifieurs donnant une réponse crédale, comme le classifieur décrit dans [97] basé sur une version crédale de l'algorithme Expectation-Maximization, les systèmes experts évidentiels [101], [118], les K-NN evidentiels [102], ou les réseaux de neurones evidentiels [104].

V.2.3.5 SVM et fonctions de croyance : la Combinaison Evidentielle

Dans ce paragraphe, nous reprenons les travaux que nous avons présentés dans [C3], [C2] sur la Combinaison Evidentielle des résultats d'un banc de SVM

pour la multi-classification. Ces travaux avaient les mêmes objectifs que ceux présentés ici, à savoir (1) permettre un certain doute dans le cas de forme de main difficile à classer, (2) permettre une reconnaissance efficace des formes de main quand celles-ci ne prêtent pas à discussion.



(a) (b)

Figure V-24 : (a) une fonction d'appartenance floue est définie par rapport à la distance à l'hyperplan ; (b) elle modélise le degré de croyance en l'appartenance à une classe dans l'espace des attributs.

Pour cela, nous proposons de (1) associer une **fonction de croyance élémentaire** (**FCE**) sur l'ensemble des classes (les Configurations du LPC) à chaque SVM, et de (2) les combiner selon la règle de Dempster en une **fonction de croyance finale** (**FCF**) sur laquelle une règle de décision peut être appliquée en vue de la classification ; le premier point étant le verrou à lever.

La marge d'un SVM est naturellement envisagée comme la région qui sépare les deux zones de certitude quant à la classification, et par conséquent, elle représente la zone d'hésitation dans l'espace des attributs. Un moyen relativement simple de modéliser cela est de définir une fonction d'appartenance définie sur le powerset des classes en jeu sur l'espace des attributs (Figure V-24). Il est alors possible de considérer cette fonction d'appartenance comme une FCE et de la manipuler comme telle. Dans le cas où un SVM ne travaille pas sur l'ensemble des classes, mais sur un ensemble plus restreint, il est nécessaire de veiller à ce que l'ensemble des FCE soit défini sur le même ensemble de Configurations afin d'être combinables. Pour cela nous recommandons d'appliquer une transformation particulière au FCE :

Notons $m^{[\Omega i]}(.)$ une fonction de croyance sur un cadre Ω_i. Soit Ω_1 et Ω_2 deux cadres tels que $\Omega_1 \subseteq \Omega_2$. Soit $\mathbf{T_R}$ de $m^{[\Omega 1]}(.)$ sur Ω_2 telle que $m_2^{[\Omega 2]}(.) = \mathbf{T_R}(m_1^{[\Omega 1]}(.))$, avec :

$$m_2^{[\Omega 2]}(A) = m_1^{[\Omega 1]}(A \cap \Omega_1) \quad \text{si} \quad A \setminus (A \cap \Omega_1) = \Omega_2 \setminus \Omega_1$$
$$m_2^{[\Omega 2]}(A) = 0 \quad \text{sinon}$$

A étant un élément de $2^{\Omega 2}$, ou, ce qui est équivalent, un sous-ensemble de Ω_2.

En pratique, cette Transformation est constituée de deux étapes. (1) Il s'agit de considérer que le classifieur binaire est un "rejeteur", c'est-à-dire qu'au lieu de choisir une classe, il rejette l'autre. (2) une opération appelée **raffinement** dans le formalisme des FC est appliquée (D'où le "$_R$" de $\mathbf{T_R}$), afin d'associer à la décision du rejeteur toutes les classes du powerset non rejetées (cf. appendice A p. 255 pour la définition formelle, et la Figure V-25 pour un exemple).

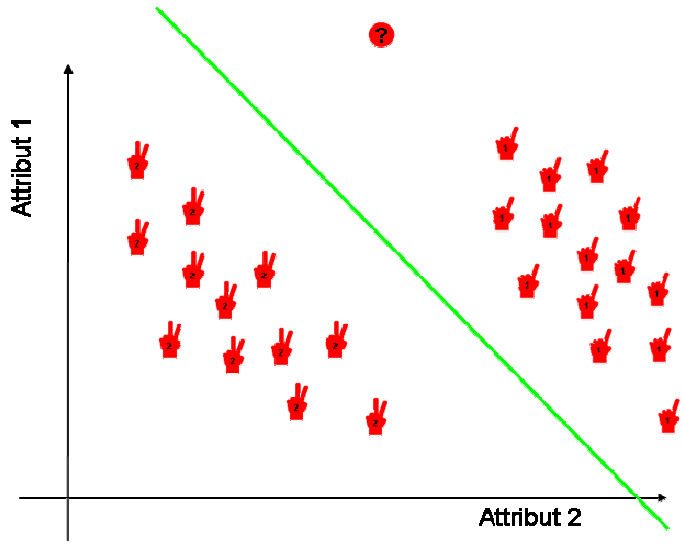


Figure V-25 : face à la présentation d'une forme de main inconnue (représentée par un point d'interrogation cerclé), le SVM discriminant la Configuration 1 (par rapport à la Configuration 2) va classiquement répondre "Configuration 1" alors que la forme de main peut potentiellement correspondre à une Configuration 3, 4, …, 7 ou 8. Une réponse du type "Tout sauf Configuration 2" serait plus adaptée. Il s'agit de la réponse de ce que nous appelons le "rejeteur". Ensuite, le raffinement R permet de reconstruire la réponse " Configuration 1, 3, 4, …, 7 ou 8" à partir de la réponse "Tout sauf Configuration 2".

Expliquons cela : soient Ω_1 = {Conf_1, Conf_2} et Ω_2 = {Conf_1, Conf_2, …, Conf_8} deux cadres et m(.), une fonction de croyance sur Ω_1. m(.) quantifie la croyance en {Conf_1}, {Conf_2} et {Conf_1, Conf_2} (que l'on appelle éléments focaux si l'on place en eux une croyance non nulle). Soit A un de ces éléments focaux. Pour étendre m(.) à un cadre plus large tel que Ω_2, il faut considérer le sens de m(A) dans Ω_2. En tout état de cause, cela a un sens complètement différent que dans Ω_1 pour la simple raison que l'absence de certains éléments de $\Omega_2 \backslash \Omega_1$ est maintenant explicite. A l'inverse, si m(A) est attribué à l'élément focal de Ω_2 qui contient A mais aussi tous les éléments $\Omega_2 \backslash \Omega_1$, alors la fonction de croyance aura la même signification. Appliqué à notre problème de fusion de SVM pour la reconnaissance de la main, cela permet d'éviter qu'un SVM ne travaillant que sur un groupe de Configurations ait une proposition qui discrédite systématiquement toutes les Configurations sur lesquelles il ne travaille pas (cf. Figure V-25).

Suite à ce raffinement, la combinaison de Dempster des FCE permet d'obtenir la FCF. Si l'ensemble des informations disponibles a déjà été pris en compte, il est possible d'appliquer une des nombreuses règles de décision disponibles pour

effectuer la classification. Ainsi, si tous les attributs de la forme de la main ont été utilisés par les SVM, il est temps de prendre une décision sur le powerset des Configurations. Suivant le type de décision, on pourra soit garantir la prise d'une décision unique, soit au contraire, autoriser un éventuel doute. Pour cela, nous proposons dans [C3] de ne pas normaliser la FCF et d'interpréter classiquement la masse de croyance associée à l'ensemble vide comme l'incapacité du classifieur à se prononcer. Ainsi, quand cette masse est trop élevée, le classifieur ne classe pas la forme de main : l'ensemble des SVM n'est pas d'accord pour lui associer une Configuration. Il y a donc de forte chance que l'image correspondante représente une transition entre deux gestes.

D'un point de vue plus théorique, la Combinaison Evidentielle permet de traiter les cas de boucles d'incertitude, et de rejeter certains items ne correspondant à aucune classe (cela n'est cependant pas suffisant à implanter une véritable classe de rejet). Ainsi, les situations d'indécision sont traitées différemment selon le mode de décision appliqué sur la FCF. Dans le sous-paragraphe suivant, nous montrons expérimentalement l'intérêt de la méthode que nous venons de décrire.

V.2.3.6 Evaluation de la Combinaison Evidentielle de SVM

Dans ce paragraphe, nous n'avons pas encore pour objectif de montrer que la Combinaison Evidentielle est la méthode la plus adaptée à notre problème de reconnaissance de Configurations. En effet, nous proposons d'abord de l'évaluer de manière plus générale par rapport aux autres méthodes de l'état de l'art. Une fois les performances de notre méthode établies, nous l'appliquerons à notre problème.

Ainsi, nous proposons d'évaluer les différentes méthodes de combinaison de SVM sur diverses bases de données, afin de diminuer le biais de la comparaison à l'égard de la particularité du problème. En effet, une disposition particulière des classes dans l'espace des attributs peut très bien être plus adaptée à une méthode qu'à une autre, sans que cela établisse une hiérarchie définitive entre ces deux méthodes.

Il est objectivement difficile de comparer deux méthodes quand celles-ci diffèrent trop l'une de l'autre. Ainsi, dans [60], [75], il est indiqué que les schémas de projection 1vs1 et 1vsAll sont à peu près équivalents, mais que le premier a un coût légèrement inférieur à l'apprentissage puis légèrement supérieur ensuite. Il y est aussi indiqué que les méthodes de combinaison par estimation de probabilités postérieures et le vote sont globalement équivalentes mais qu'il n'est pas utile de les comparer sur un problème en particulier pour la simple raison que leurs algorithmes d'apprentissage ne sont pas les mêmes, et que par "effet de bord", l'un ou l'autre peut se révéler localement plus efficace. Dans [76], il n'apparaît aucune supériorité générale des méthodes CCE sur les autres. Dans [68], un résultat équivalent est donné sur d'autres schémas moins populaires.

Nous proposons donc de comparer notre méthode avec la méthode du vote, puisqu'elle utilise le même algorithme d'apprentissage, et qu'individuellement, les SVM fonctionnent de la même manière (ce n'est qu'au moment de la combinaison des SVM que l'information de distance à l'hyperplan est utilisée). Ainsi, si une éventuelle différence apparaît dans les taux de classification, celle-ci ne peut être due qu'à la manière dont les mêmes informations de base issues des SVM sont organisées et combinées entre elles, et en aucun cas à une meilleure définition des hyperplans séparant les classes (ainsi, un problème parfaitement séparable ne sera pas mieux traité par l'une ou l'autre des méthodes). De ce fait, cette différence ne peut être imputable qu'à l'élément que nous souhaitons évaluer. Ainsi les deux protocoles de tests sont identiques :

– Nous utilisons LIBSVM [61], qui est une librairie complète et efficace pour l'implantation d'un banc de SVM en C/C++. Nous avons implanté les algorithmes de la Combinaison Evidentielle dans LIBSVM, de sorte que le code est maintenant disponible.

– Nous utilisons diverses bases de données, éventuellement publiques, afin de comparer les méthodes dans plusieurs situations. Celles-ci sont présentées plus loin.

– Les SVM sont de type C-SVM et la fonction noyau utilisée est une fonction à base radiale (**RBF**), dont l'équation est :

$$Ker_{\beta}(u,v) = \exp\left(-\beta \cdot |u-v|^2\right)$$

– Le paramètre β[4] utilisé est celui par défaut, ou le cas échéant, celui recommandé pour chaque base de test utilisée. Ce paramètre a peu d'importance dans la mesure où l'on ne cherche pas à avoir une reconnaissance la plus fiable possible, mais à effectuer des comparaisons relatives entre les capacités de reconnaissance des algorithmes dans des situations équivalentes. Dans tous les cas, le paramètre est donc identique d'une méthode à l'autre au sein d'une comparaison. De même, nous n'avons pas cherché à utiliser une fonction noyau particulièrement efficace ; nous avons choisi les RBF par défaut.

Pour ce qui est du taux de reconnaissance, il faut le définir de telle sorte qu'il soit compatible avec les deux méthodes, et qu'il puisse faire ressortir les points qui importent dans l'évaluation. Ces derniers sont au nombre de trois :

– Dans le cas où une décision est prise systématiquement, comment se comporte la Combinaison Evidentielle par rapport au vote classique ? Il est nécessaire que le résultat soit au moins aussi satisfaisant.

[4] Habituellement, ce paramètre est désigné par γ, mais nous utilisons déjà cette notation. Nous le renommons afin d'éviter toute confusion.

– Dans le cas où l'on accepte de ne pas classer un item en raison d'une masse de croyance trop forte en l'ensemble vide, comment la classification sur les items restant évolue-t-elle ?

– Dans le cas où il y a une croyance trop forte en des unions de classes, et que seule une décision partielle est prise, comment la reconnaissance globale évolue-t-elle ?

Ce dernier point est difficile à chiffrer en parallèle des deux premiers. De plus, dans nos corpus, nous n'avons que peu d'items dont la classification partielle est justifiée. Il est donc difficile d'en tirer des conclusions statistiques. Enfin, au paragraphe VI.2.1 (p. 201), nous proposons une nouvelle règle permettant d'effectuer des décisions partielles. Nous évaluerons donc cette possibilité à ce moment-là. Pour les deux premiers points à évaluer, nous proposons un jeu de trois taux de classification :

Le premier consiste en l'application de la Transformée Pignistique à la FCF, afin de toujours fournir une décision comparable à celle de la procédure de vote. Le taux de bonne reconnaissance est alors calculé par :

$$\%Reco = 100 \cdot \frac{\text{Nombre d'items classés correctement}}{\text{Nombre total d'items}}$$

Les deux mesures suivantes évaluent l'intérêt de la réjection de certains éléments. Soit m_{final} la FCF sur laquelle la classification est effectuée. Si $m_{final}(\emptyset) = \max_{2^\Omega}(m_{final}(.))$, alors le classifieur est considéré comme incompétent au regard de l'item, et celui-ci est rejeté (cela est en pratique d'un grand intérêt pour mettre en place une procédure d'**apprentissage actif** [69] basée sur l'étiquetage manuel des éléments rejetés). Dans le cas contraire, la classification a lieu normalement. On définit donc :

$$Acc_{Sup} = \frac{\text{Nombre de classifications correctes}}{\text{Nombre total d'items} - \text{Nombre d'items rejetés}}$$

$$Acc_{Inf} = \frac{\text{Nombre de classifications correctes}}{\text{Nombre total d'items}}$$

Ce qui veut simplement dire que Acc_{Sup} ne considère pas les items rejetés, alors que Acc_{Inf} les considère comme systématiquement faux. En conséquence, on définit un taux de rejet $\%R$ par :

$$\%R = 1 - \frac{Acc_{Inf}}{Acc_{Sup}}$$

Il est possible d'utiliser une autre distribution que celle présentée à la Figure V-24 afin d'avoir un autre comportement. Par exemple, nous utilisons aussi une distribution où le support de l'hésitation se résume à l'hyperplan, et où les

fonctions d'appartenance des classes sont des créneaux (cf. Figure V-26). Dans un tel cas, l'hésitation est nulle, le système est donc équivalent à celui d'un vote, à partir du moment où la FCF est normalisée. En l'absence de normalisation, il se trouve que c'est la méthode pour laquelle il y a le plus de conflit, et donc, qui provoque le plus haut taux de réjection $\%R_{max}$. Ainsi, suivant la distribution que l'on choisit pour la fonction d'appartenance, il est possible d'obtenir n'importe quel taux de réjection entre 0 et $\%R_{max}$.

Notons que le fait qu'il soit possible de retomber sur la combinaison par vote en prenant un support particulier pour la fonction d'hésitation est une justification de l'intérêt de la méthode : celle-ci est au moins aussi générale et puissante que la méthode à laquelle on la compare.

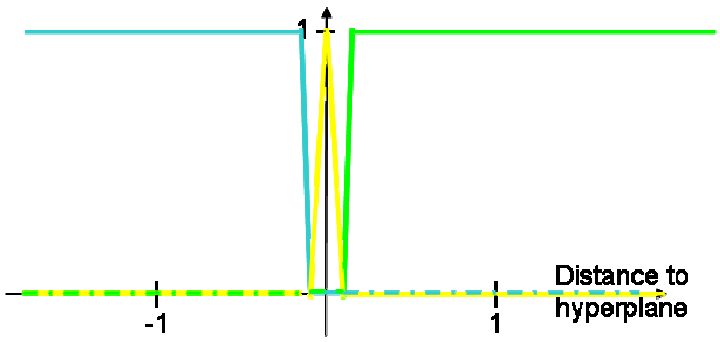


Figure V-26 : une fonction d'appartenance avec un support d'hésitation réduit est équivalente à une méthode de vote binaire.

Enfin, pour évaluer l'amélioration éventuelle de la Combinaison Evidentielle par rapport au vote, nous calculons le taux d'erreurs qui ont été évitées (ou rajoutées). *AvMis*, le pourcentage d'erreurs évitées, est défini par :

$$AvMis = \frac{\text{Nombre d'erreurs évitées}}{\text{Nombre original d'erreurs}}$$

$$= \frac{\text{Nombre original d'erreurs - Nouveaux nombre d'erreurs}}{\text{Nombre original d'erreurs}}$$

$$= \frac{\text{Taux original d'erreurs - Nouveaux taux d'erreurs}}{\text{Taux original d'erreurs}}$$

$$= \frac{1 - \text{Taux de reco. original} - 1 + \text{Nouveaux taux de reco.}}{1 - \text{Taux de reconnaissance original}}$$

$$= \frac{\text{Nouveaux taux de reconnaissance - Taux de reconnaissance original}}{1 - \text{Taux de reconnaissance original}}$$

Dans le Tableau V-3, divers corpus de données sont présentés. Le corpus Vowels est issu de [62] ; il est constitué de 528 items d'apprentissage et de 462 items pour le test. Il y a 10 attributs et 11 classes. Le corpus 5-letter est un extrait du corpus Letters de [62] qui contient 20,000 items correspondant aux 26 lettres de l'alphabet anglais. L'extrait que nous utilisons est constitué des 5 lettres S, T, U, V, et X. Texture est un autre corpus issu de la même source. Enfin, HCS est un corpus extrait des ICC de la base de données ETTRAN N,

auxquelles les algorithmes de segmentation présentés dans ce document ont été appliqués, suivi d'une description par les invariants de Hu [12], [32] (les DFM ne sont pas utilisés pour éviter un taux de reconnaissance trop élevé avec la procédure de vote, et ainsi permettre une véritable comparaison).

Tableau V-3 : description des corpus.

	Nombre classes	Nombre attributs	Taille corpus apprentissage	Taille corpus test
Vowels	11	10	528	462
5-letter	5	16	1950	1952
Texture	7	19	210	2100
HCS	8	7	732	196

Les résultats des expériences sont indiqués dans le Tableau V-4 : la colonne **vote** indique le taux de reconnaissance avec la méthode de vote, et **BetP** donne le même taux quand la Combinaison Evidentielle et la Transformée Pignistique sont utilisées. ***AvMis*** donne l'amélioration en pourcentage d'erreurs évitées. **Avec Hésitation** et **Sans Hésitation** correspondent à la présence ou à l'absence d'hésitation dans la fonction d'appartenance. Pour chacun des modèles, $\mathbf{Acc_{Sup}}$ et $\mathbf{Acc_{Inf}}$ sont calculés. $\%\mathbf{R_{max}}$ correspond à %R calculé dans le cas Sans Hésitation. Pour le corpus 5-letter, certaines valeurs ne sont pas calculées en raison de leur absence de signification pour un taux de classification original trop élevé.

Tableau V-4 : résultats en % pour différents corpus.

	vote	BetP	AvMis	Avec Hésitation		Sans Hésitation		$\%\mathbf{R_{max}}$
				$\mathbf{Acc_{Sup}}$	$\mathbf{Acc_{Inf}}$	$\mathbf{Acc_{Sup}}$	$\mathbf{Acc_{Inf}}$	
Vowels	55.8	57.4	3.6	58.1	56.5	60.4	52.8	12.6
5-letter	99.2	99.8	73.3					
texture	91.6	95.9	51.2	96.0	95.4	96.4	94.6	1.9
HCS	78.6	86.2	35.7	86.2	82.7	86.8	80.6	7.1

Illustrons la manière de lire ce tableau sur l'exemple de la première ligne. Avec le corpus Vowels, le taux de bonne classification est de 55.8% sur le corpus de test avec une procédure de vote classique. Les distances à l'hyperplan sont sauvegardées et réutilisées pour le processus de Combinaison Evidentielle. Avec l'utilisation de la PT, le taux de bonne classification atteint 57.4%, ce qui

représente une suppression de 3.6% des erreurs, par la seule amélioration du processus de combinaison. Si nous considérons la possibilité de rejeter tous les items source de trop de conflits, nous obtenons alors Acc_{Sup} = 58.1% et Acc_{Inf} = 56.5%. Si l'on fait de même en maximisant la source du conflit, on obtient alors Acc_{Sup} = 60.4% et Acc_{Inf} = 52.8%, ce qui correspond à un taux de rejet maximum de 12.6%. En conséquence, il est possible de paramétrer la distribution de la fonction d'appartenance de telle sorte que l'on ait un rejet compris entre 0% et 12.6%.

Sur l'ensemble des tests effectués, il est indéniable que la Combinaison Evidentielle est plus efficace que le vote, puisque l'ensemble des valeurs de la troisième colonne est positif. En comparant la première colonne et la cinquième, il apparaît que même un comportement prudent dans lequel les items conflictuels sont rejetés, (mais comptabilisés comme mal classés) le résultat global est meilleur qu'en utilisant une procédure de vote. Cela reste vrai dans deux cas sur trois lorsque l'on utilise une fonction d'appartenance entraînant un rejet maximal (avant dernière colonne).

En pratique, les performances sont similaire avec l'utilisation d'un schéma de projection de type 1vsAll.

Malgré la très grande variabilité des résultats obtenus (ceux-ci dépendent fortement de la cohérence interne de chacune des classes et de la difficulté de la discrimination), l'ensemble indique que la Combinaison Evidentielle est capable de diminuer fortement le nombre d'erreurs lors de l'étape de combinaison des SVM.

Si l'on compare le coût calculatoire de la Combinaison Evidentielle à celle du vote, la seconde est beaucoup plus légère. En effet, la combinaison de Dempster de $N.(N-1)/2$ fonctions de croyance définies sur le powerset d'un ensemble de cardinal N devient en pratique très lourd dès que N dépasse une valeur comprise entre 15 et 20, en fonction de l'application. Ainsi, il y a certaines applications, telle que la classification de pixels en vue de segmentation (qui nécessite une certaine rapidité), ou certains problèmes d'indexation de vidéos du web (où le nombre de classes est facilement de l'ordre de 200) où la méthode de Combinaison Evidentielle est inadaptée. D'une manière générale, voici la stratégie que nous préconisons en fonction de la valeur de N :

– Dans le cas où N est beaucoup trop grand ($N > 25$), la méthode n'est pas applicable d'un point de vue calculatoire, et n'apporte que très peu d'intérêt ; en effet, quand le nombre de SVM ayant droit à un vote est très important, l'influence de quelques SVM ayant une réponse inadaptée ne changera pas la tendance générale, et la décision sera globalement la même.

– Dans le cas où N est de taille intermédiaire ($25 > N > 15$), c'est-à-dire que le temps de calcul commence à s'en faire ressentir, mais que (1) le nombre de votants n'est pas suffisant pour espérer faire ressortir une tendance statistique vers la bonne décision, ou quand (2) l'indétermination de tous les classifieurs est

importante en raison de la distribution des classes, alors l'utilisation de la Combinaison Evidentielle peut être justifiée malgré le coût qu'elle implique. Dans de tels cas, plusieurs méthodes permettent de diminuer la complexité calculatoire. Tout d'abord, il est possible de prendre un schéma de projection qui diminue le nombre de votants par rapport au nombre de classes (1vsAll par exemple). Ensuite, il est possible d'utiliser les approximations et les méthodes proposées dans [100], pour effectuer le calcul plus rapidement.

– Dans le cas où N est de taille raisonnable ($N < 15$), les performances comparées de la Combinaison Evidentielle sont les meilleures par rapport au vote, et son coût calculatoire est le moins pénalisant.

Nous venons de montrer que la Combinaison Evidentielle est une méthode plus efficace que la procédure classique de vote. Maintenant, nous pouvons l'appliquer à notre problème de reconnaissance de Configuration, et par là même vérifier que c'est la méthode la plus efficace parmi celles que nous avons envisagées.

Cependant, tout ce que nous avons considéré jusqu'à présent concerne le cas où l'intégralité des attributs de classification est prise en compte par les SVM. Si en revanche, il y a encore de l'information disponible, il est trop tôt pour prendre la décision. Dans notre problème pratique, cela se matérialise par l'existence d'un attribut de haut niveau, issu d'un système expert, et permettant d'indiquer si le pouce est déployé ou non. Un autre intérêt de la Combinaison Evidentielle est de pouvoir facilement gérer ce type de cas : il est possible d'utiliser la FCF à l'entrée d'un autre système de classification permettant la fusion de plusieurs classifieurs hétérogènes.

Dans le paragraphe suivant, nous traitons ces deux points : nous commençons par évaluer l'intérêt de la Combinaison Evidentielle par rapport au vote sur le problème de la reconnaissance de la Configuration. Dans ce cas-là, et au contraire des expériences précédentes, nous cherchons à optimiser les taux de classification. Ensuite, nous utilisons une combinaison de Dempster pour fusionner les informations issues de la Combinaison Evidentielle et de l'indicateur de présence du pouce (que l'on a converti en FC).

V.2.3.7 Reconnaissance de la Configuration par Combinaison Evidentielle de classifieurs hétérogènes

En règle générale, une fois la classification effectuée, une décision est prise simplement sur le résultat de celle-ci. Quand elle fournit une réponse de type crédal, il est possible d'utiliser les différentes méthodes de décision de ce formalisme. Il est aussi possible de combiner cette information avec d'autres résultats de classification. En effet, il arrive souvent que les différents attributs utilisés pour modéliser un problème ne soient pas compatibles et qu'ils ne puissent pas être utilisés dans un unique classifieur. C'est une des nombreuses raisons motivant la recherche dans le domaine de la fusion de classifieurs. Dans le cas où un banc de SVM fournit une réponse crédale, il devient alors facile de

combiner cette décision avec d'autres sources d'information. En effet, ce formalisme est particulièrement bien adapté à cette situation. Il suffit de calculer la combinaison de Dempster des différentes informations ou FCF. C'est ce que nous avons montré dans [C3], par la fusion des informations d'un banc de SVM et d'un système expert fournissant une réponse sous la forme d'une fonction de croyance. Nous reprenons ici l'évaluation de ces travaux.

Les performances de classification sont comparées pour : (1) un banc de SVM seul avec une procédure de vote, (2) un banc de SVM seul avec la Combinaison Evidentielle, et enfin, (3) un banc de SVM et un système expert fournissant un indicateur crédal de la présence éventuelle du pouce combinés par la Combinaison Evidentielle.

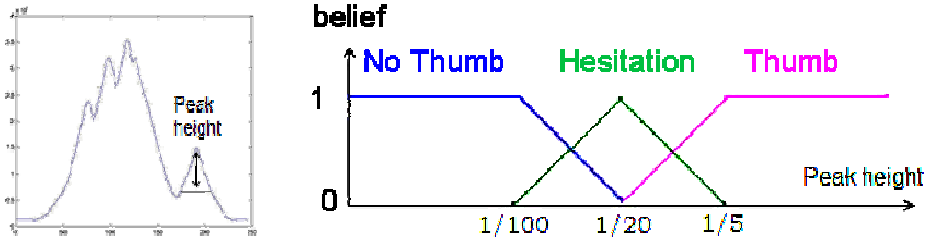


Figure V-27 : (a) la hauteur du pic détermine (b) la croyance en la présence du pouce.

Le système expert crédal indiquant la présence du pouce fonctionne sur le principe de l'attribut que nous avons présenté en début de chapitre. Cependant, au lieu de fournir une décision binaire, il fournit une décision de type fonction d'appartenance telle que celle illustrée sur la Figure V-27 : plus le pic de détection du pouce est haut, plus il est possible d'avoir confiance en l'indicateur de présence du pouce.

Tableau V-5 : détails des 960 ICC issues du corpus ETTRAN N.

Configuration	Corpus 1 (Apprentissage)	Corpus 2 (Test)
0	37	12
1	94	47
2	64	27
3	84	36
4	72	34
5	193	59
6	80	46
7	20	7
8	35	23
Total	**679**	**291**

Les 3 seuils qui permettent de définir la fonction d'appartenance de la Figure V-27b sont déterminés manuellement en fonction d'observations faites sur l'ensemble d'apprentissage du corpus ETTRAN N. Pour mémoire, nous rappelons dans le Tableau V-5, la composition de ce corpus. Grâce à l'aspect flou de la prise de décision, les valeurs de seuil n'ont pas besoin d'être définies avec précision. En pratique, elles sont déterminées par 3 ratios (1/5, 1/20 et 1/100) de la distance maximale entre CP et un élément du contour de la main[5].

Ensuite, la croyance obtenue sur le powerset de l'ensemble {POUCE, PAS_POUCE} est raffinée en une croyance sur l'ensemble des classes. La croyance en POUCE est naturellement associée aux Configurations {5, 6, 7} et la croyance en PAS_POUCE aux autres. Dans le cas où le système expert est indécis, aucune information n'est apportée puisque toute la masse est associée au cadre tout entier.

Les SVM sont entraînés sur le Corpus 1, et le Corpus 2 est utilisé pour effectuer tous les tests. Classiquement, nous utilisons la définition précédente du taux de reconnaissance :

$$\%Reco = 100 \cdot \frac{\text{Nombre d'items classés correctement}}{\text{Nombre total d'items}}$$

Quand il s'agit de comparer deux méthodes, nous utilisons toujours la même mesure *AvMis*. Nous procédons au même protocole expérimental que précédemment, avec les modifications suivantes :

– Le paramètre de coût est fixé une fois pour toute à 100,000 (il s'agit en quelque sorte de la constante de raideur de la variable ressort dans l'algorithme C-SVM) et le critère de convergence à 0.001 (il s'agit de l'erreur autorisée sur la finesse de définition de l'hyperplan séparateur).

– La fonction noyau utilisée[6] est une sigmoïde :

$$Ker_{\beta,R}(u,v) = \tanh\left(\beta \cdot u^T \cdot v + R\right) \quad \text{avec} \begin{cases} \beta = 0.001 \\ R = -0.25 \end{cases}$$

Les résultats sont présentés dans le Tableau V-6. Conformément à nos attentes, la Combinaison Evidentielle est digne d'intérêt pour le problème de la reconnaissance de la Configuration, puisqu'elle permet une augmentation de 1 point du taux de reconnaissance, ce qui correspond à un *AvMis* de 11.11%. Ensuite, la combinaison avec l'information de présence du pouce est elle aussi efficace, puisqu'elle permet une amélioration totale de 2.1 points par rapport à

[5] Attention, nous utilisons CG comme résumé de la trajectoire de la main, mais CP comme référence pour la mesure de la longueur des doigts.

[6] Encore une fois, ici, l'optimisation des paramètres des SVM n'a pas d'intérêt. Notre souci est surtout de comparer des méthodes équivalentes dans des conditions données, et non d'optenir les scores les plus performants.

la méthode de vote pour laquelle il n'y a pas de combinaison possible, ce qui représente un *AvMis* de 2.22% : entre 1/4 et 1/5 des erreurs sont donc évitées. Le Tableau V-7 représente la matrice de confusion pour ce dernier test. Nous pouvons faire trois remarques :

– la Configuration 3 est souvent mal classée,

– quand une forme de main est mal reconnue, elle est souvent considérée à tord comme une Configuration 1 ou 7

Il n'y a que 3 erreurs entre les superclasses POUCE et PAS_POUCE.

Tableau V-6 : résultats de l'évaluation de la combinaison de classifieurs.

	Vote	**Combinaison Evidentielle**	
		seule	Combinaison avec l'indicateur de pouce
%reco	90.7%	91.8%	92.8%

Ainsi, nous avons la confirmation expérimentale que (1) la Combinaison Evidentielle est adaptée à notre problème et que (2), elle permet la fusion d'informations issues de systèmes de classification hétérogènes. De manière formelle, nous n'avons considéré que deux types de classifieurs : les SVM et les systèmes experts. Nous allons par la suite généraliser encore ces résultats à d'autres systèmes de classification, mais comme cela dépasse le cadre de la reconnaissance de Configurations, nous allons d'abord clore ce point.

Tableau V-7 : matrice de confusion pour le classifieur final (SVM+ système expert). Le rectangles en tirets représentent la superclasse PAS_POUCE, et celui en pointillés, la superclasse POUCE. Les classes reconnues sont indiquées en colonne et les vérités terrain en ligne.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	12	0	0	0	0	0	0	0	0
1	0	46	0	0	0	0	0	0	1
2	0	2	23	2	0	0	0	0	0
3	0	2	0	29	2	2	1	0	0
4	0	0	0	1	32	0	0	0	1
5	0	0	0	0	0	58	0	1	0
6	0	0	2	0	0	0	41	3	0
7	0	0	0	0	0	0	1	6	0
8	0	0	0	0	0	0	0	0	23

Evidemment, en fonction des similarités de corpus, les apprentissages se généralisent de manière très différente. En conséquence, les paramètres aussi

sont très différents. Bien sûr, les résultats sont un peu plus faibles dans ce cas là, mais malgré tout, ils restent du même ordre de grandeur que dans les expériences précédentes. Après une sélection judicieuse des paramètres des SVM (par validation croisée), on obtient les résultats du Tableau V-11.

V.2.4 Stratégie retenue pour la reconnaissance de la Configuration

Dans ce paragraphe, nous nous intéressons au choix de la méthode de classification. En effet, il s'agit du dernier élément à spécifier par rapport au problème de reconnaissance des Configurations :

– Les classes sont connues (les Configurations),

– la manière de réduire la variabilité par suppression du poignet de la forme de la main a été spécifiée,

– le jeu d'attributs de classification le plus performants parmi ceux que nous avons proposé est connu : il s'agit des DFM.

Maintenant, le but est donc de prendre les meilleurs descripteurs, de paramétrer au mieux le banc de SVM, et de les combiner avec la méthode de Combinaison Evidentielle. Au passage, il est prudent de s'intéresser à la stabilité des résultats et à leur pouvoir de généralisation. Enfin, nous étendons au "cas multi-codeurs", c'est-à-dire que nous considérons la reconnaissance de plusieurs morphologies différentes.

Nous utilisons 4 corpus d'ICC, chacun correspondant à un gant particulier et un codeur particulier. Le premier corpus correspond à une base de 675 ICC issues de ETTRAN N. Les autres corpus sont constitués de 472 ICC de MAGOZ B, de 263 ICC de MAGOZ J, et de 998 ICC de MAGOZ R.

V.2.4.1 Paramètres des SVM et stabilité des DFM

L'objectif est de déterminer comment paramétrer le banc de SVM, afin de maximiser le taux de reconnaissance. Malheureusement, le meilleur moyen d'obtenir un taux de reconnaissance élevé est de sur-apprendre les données du corpus d'apprentissage. Dans un tel cas, le pouvoir de généralisation baisse. Il y a donc un compromis à trouver.

Les premiers tests que nous effectuons sont mono-codeurs : seul le corpus ETTRAN N est utilisé. Il est toujours divisé en deux parties pour l'apprentissage et le test. Ensuite, de nombreux apprentissages sont effectués avec divers paramètres de réglage du banc de SVM. Chacun des apprentissages réalisés est ensuite testé sur le corpus qui a servi à l'apprentissage, et sur le corpus de test jusque là inutilisé. Ensuite, l'apprentissage et le test sont réalisés sur le corpus de test, et finalement, l'apprentissage est réalisé sur le corpus de test et le test sur le corpus d'apprentissage. Cela permet de simuler des corpus de validation,

ce qui rend possible l'évaluation du pouvoir de généralisation du système. Tous les scénarii de tests de classification sont gérés via **ImTrAc**.

Les paramètres de réglages des SVM sont les suivants :

– La fonction noyau est du type sigmoïde. De nos observations, il est possible de conclure que cette famille de fonction noyau est un peu plus efficace sur notre problème que les RBF, mais beaucoup plus que les noyaux gaussiens ou polynomiaux.

– β et R sont des paramètres permettant de spécifier une fonction noyau particulière au sein de la famille des sigmoïdes. Il est très difficile, voire impossible, de connaître à l'avance l'influence de ces paramètres :

$$Ker_{\beta,R}(u,v) = \tanh\left(\beta \cdot u^T \cdot v + R\right)$$

– C représente toujours le coût marginal associé à une erreur dans le cas de non séparabilité complète. Si C est trop faible, l'apprentissage ne sera pas de bonne qualité, mais se généralisera tel qu'il est à un grand nombre de situations inconnues. Si C est trop grand, l'apprentissage risque d'être tellement adapté à l'ensemble d'apprentissage, qu'il ne sera pas assez souple pour appréhender les situations inconnues. C'est ce que l'on appelle un sur-apprentissage.

– e est un critère de terminaison de l'algorithme d'optimisation combinatoire quand celui-ci est implanté sur une machine et que l'optimisation n'a plus lieu dans $\mathbb{R}$ mais en virgule fixe ou en flottants. Plus e est faible, plus l'apprentissage est long mais précis.

Tableau V-8 : tests de reconnaissance en configuration mono-codeur.

β	C	R	e	Appr/Appr	Appr/Test	Test/Test	Test/Appr
0,0001	**5000**	**0,25**	**0,001**	**100**	**95,9**	**99,37**	**97,2**
0,0001	**1000**	**0,25**	**0,001**	**99,7**	**95,9**	**98,42**	**97,2**
0,0001	100	0,25	0,001	98,5	91,8	91,8	91,6
0,0001	100	1	0,001	97,64	89,27	88,96	88,78
0,0001	100	0	0,001	98,82	92,11	92,11	92,52
0,0001	**100**	**0**	**0,1**	**98**	**95,9**	**99,37**	**97,5**
0,0001	5000	0	0,1	100	91,8	92,11	91,9
0,0001	**100000**	**0**	**0,1**	**100**	**96,21**	**100**	**97,2**
0	5000	0	0,001	85,36	79,18	89,27	82,55
0,01	5000	0	0,001	97,82	94	98,42	94,7
0,001	**5000**	**0**	**0,001**	**100**	**96,21**	**100**	**97,2**
0,00001	**5000**	**0**	**0,001**	**98,13**	**96,85**	**97,79**	**97,2**
0,000001	5000	0	0,001	89,1	90,54	90,22	89,72

Ces premiers tests sont résumés dans le Tableau V-8. Les 4 premières colonnes indiquent les valeurs des paramètres et les 4 colonnes suivantes indiquent le

pourcentage de bonne classification en fonction de Corpus1/Corpus2 où Corpus1 est le corpus sur lequel l'apprentissage est réalisé, et Corpus2 celui que l'on utilise pour le test. Utiliser le corpus de test pour l'apprentissage et vice-versa est en effet un bon indicateur de l'intérêt du paramétrage. Il ressort que :

– Les DFM appliqués à la reconnaissance de la Configuration sont peu sensibles à la manière dont le banc de SVM est paramétré. Par ailleurs, d'importantes variations sur les paramètres de la fonction noyau, sur le paramètre de coût ou sur le paramètre de terminaison de l'algorithme ont beaucoup moins d'influence qu'avec d'autres descripteurs. A titre d'exemple, le Tableau V-9 montre que les invariants de Hu sont quand à eux beaucoup plus sensibles. Cela confirme donc l'intérêt des DFM pour notre problème. En effet, d'une manière générale, il est très difficile de paramétrer un banc de SVM : les données sont difficiles à visualiser et l'optimum de la fonction objective associée à l'hyperplan est tellement non convexe et non linéaire qu'en pratique, il n'y a pas d'autre méthode que d'utiliser un ensemble de validation. Néanmoins, il est toujours délicat de prévoir la capacité de généralisation d'un apprentissage. Le fait d'avoir des paramètres stables est un gage de l'efficacité des descripteurs et de la capacité de généralisation de l'apprentissage.

– L'inversion des corpus permet de vérifier que la qualité des résultats à paramètres constants (comparaison des colonnes du tableau) est maintenue, ce qui est une preuve supplémentaire de la relative stabilité des DFM pour ce problème. En outre, cela aussi est gage d'une bonne capacité de généralisation.

Tableau V-9 : illustration de l'instabilité du taux de classification en fonction des paramètres des SVM. Les 4 premières colonnes indiquent les valeurs des différents paramètres. La dernière colonne indique le taux de classification que l'algorithme permet quand le test est réalisé sur le corpus d'apprentissage. La première ligne en gras (que l'on prend comme référence) indique le réglage des paramètres permettant d'obtenir le taux de classification maximal (91.8% sur le corpus de test). Sur le corpus d'apprentissage le taux de classification est de 92.9%. Ensuite, chacune des lignes suivantes illustre comment la variation d'un seul paramètre (en gras) dégrade les performances sur le corpus d'apprentissage.

β	C	R	e	Taux de reconnaissance sur le corpus d'apprentissage
0,001	**100000**	**-0.25**	**0,001**	**92.9**
0,001	**10000**	-0.25	0,001	91.6
0,001	**1000**	-0.25	0,001	85.1
0,001	**100**	-0.25	0,001	75.7
0,01	100000	-0.25	0,001	90.9
0,1	100000	-0.25	0,001	60.1

Finalement, il ressort que le choix des paramètres du banc de SVM n'a pas grande importance, tant ceux-ci sont stables.

V.2.4.2 Cas de la reconnaissance multi-codeurs

Dans une seconde série d'expériences, nous évaluons la perte qui résulte de l'utilisation de plusieurs corpus simultanément, c'est-à-dire que nous évaluons de quelle manière la variabilité inter-codeur va venir diminuer la capacité du système à être discriminant. Pour cela, nous mélangeons plusieurs corpus et nous effectuons une séparation aléatoire pour partager en deux parties égales le corpus obtenu. Cela permet d'avoir un corpus d'apprentissage et un corpus de test. Grâce à **ImTrAc**, l'ensemble {séparation aléatoire, apprentissage, test} est répété 50 fois et la moyenne des 50 itérations est considérée. Tout cela est résumé dans le Tableau V-10, où certains résultats particulièrement représentatifs sont mis en caractères gras. De tout cela, il apparaît que :

– La robustesse à la variation des paramètres est toujours valable.

– Les résultats avec les DFM sur un corpus multi-codeurs sont meilleurs que ceux obtenus dans le cas d'un corpus mono-codeur avec les {invariants de Hu + descripteur de présence du pouce}. En effet, ils ne permettaient d'atteindre que 92.8% de classifications correctes.

Tableau V-10 : taux de reconnaissance dans le cas d'un apprentissage multi-codeur.

Corpus	β	C	R	e	Moyenne
MAGOZ B + ETTRAN N	0,0001	20000	0,15	0,0001	95,7
MAGOZ B + ETTRAN N	**0,001**	**20000**	**0,15**	**0,0001**	**93,0**
MAGOZ B + ETTRAN N	0,000001	20000	0,15	0,0001	92,8
MAGOZ B + ETTRAN N	0,0001	100	0,15	0,0001	90,4
MAGOZ B + ETTRAN N	0,0001	5000000	0,15	0,0001	94,3
MAGOZ B + ETTRAN N	**0,0001**	**20000**	**0**	**0,0001**	**95,8**
MAGOZ B + ETTRAN N	0,0001	20000	0,5	0,0001	95,3
MAGOZ B + ETTRAN N	0,0001	20000	0,15	10^{-11}	95,6
MAGOZ {B, J} + ETTRAN N	0,0001	20000	0,15	0,0001	95,4
MAGOZ {B, J} + ETTRAN N	0,0001	20000	0	0,0001	94,9
MAGOZ {B, J, R} + ETTRAN N	**0,0001**	**20000**	**0**	**0,0001**	**95,8**

Finalement, il est intéressant d'analyser la capacité de discrimination du système face à un codeur inconnu, c'est-à-dire, en utilisant un banc de SVM qui n'a été entrainé que sur des codeurs différents de celui sur lequel il est testé. Pour cela, nous effectuons des tests selon le même protocole que précédemment.

Tableau V-11 : taux de reconnaissance dans le cas d'un apprentissage multi-codeur et d'une utilisation face à un codeur inconnu.

Apprentissage	Test	β	C	R	E	Moyenne du taux de reconnaissance
MAGOZ {B, J} + ETTRAN N	MAGOZ R	0,0001	1000	0	0,0001	91,5
MAGOZ {B, R} + ETTRAN N	MAGOZ J	0,0001	50000	0	0,0001	91,3
MAGOZ {R, J} + ETTRAN N	MAGOZ B	0,0001	20000	0	0,0001	91,7
MAGOZ {R, J} + ETTRAN N	MAGOZ B	0,0001	5000	0	0,0001	91,7

V.2.4.3 Résumé des évaluations et de la méthode retenue

Finalement, dans le cas d'une application mono-codeur comme multi-codeur, nous préconisons l'utilisation d'un banc de C-SVM de type 1vs1, avec une fonction noyau sigmoïde, et une Combinaison Evidentielle, appliquée sur les DFM, eux-mêmes calculés sur une image binaire de la main dont le poignet a été supprimé. Les paramètres de la fonction sigmoïde sont relativement stables mais leur estimation précise et leur adaptation à la morphologie du codeur permet d'augmenter les résultats. De même, le réglage du paramètre C dépend de l'équilibre que l'on souhaite entre précision des résultats et pouvoir de généralisation.

Cela termine la partie relative à la reconnaissance de la Configuration. La reconnaissance de la Position ayant également été traitée, nous en avons terminé avec la problématique de classification des composantes gestuelles du LPC. Néanmoins, avant de passer au chapitre suivant, nous allons explorer les conséquences théoriques des méthodes de classification proposées.

V.3 Généralisations de la Combinaison Evidentielle

Dans cette section, nous généralisons les résultats sur la Combinaison Evidentielle, et nous nous détachons momentanément de la problématique du LPC. Néanmoins, ces résultats trouveront une application aux chapitres suivants.

Ainsi nous nous intéressons d'abord au cas où les SVM sont remplacés par d'autres classifieurs binaires non crédaux. En effet, la Combinaison Evidentielle est *a priori* impossible à utiliser sur des classifieurs binaires non crédaux autres que les SVM pour la simple raison qu'il n'y a pas de marge comme support prédéterminé pour la zone d'hésitation. Nous allons voir que cela peut cependant être contourné moyennant un apprentissage sur le support de l'hésitation.

Ensuite, nous considérons le cas où les classifieurs ne sont plus des classifieurs binaires, mais des classifieurs unaires. Par la même occasion, nous enrichissons le nombre de classifieurs de nature hétérogène que l'on peut combiner grâce à notre méthode. Mais surtout, cela nous permet de mettre en place une méthode

de conversion des fonctions de probabilité en fonctions de croyance. Cela est d'un grand intérêt, puisqu'ainsi il est possible de convertir la réponse de tout classifieur en une fonction de croyance, pourvue que sa réponse soit de type probabiliste.

V.3.1 Combinaison Evidentielle de classifieurs binaires non crédaux

La Combinaison Evidentielle de classifieurs n'est *a priori* pas utilisable sur des classifieurs binaires non crédaux autres que les SVM. La raison est qu'il n'y a pas de marge comme support prédéterminé pour la zone d'hésitation. Il est donc impossible de déterminer la distribution de la fonction d'appartenance associée à la sortie du classifieur. En pratique, cela peut être contourné de la manière suivante. On fait dépendre la fonction d'appartenance d'un paramètre dont la variation est associée au support de l'hésitation (normalement défini par les marges). Ensuite, il suffit de déterminer une valeur du paramètre pour laquelle la classification donne de bons résultats. Pour ce faire, nous recommandons l'utilisation d'une validation croisée sur le corpus d'apprentissage des classifieurs, afin de réaliser un bon compromis rejet/hésitation.

V.3.2 Combinaison Evidentielle de classifieurs unaires

Il est même possible de s'inspirer de cette méthode de détermination de la fonction d'appartenance au moyen d'une validation croisée pour étendre la combinaison crédale au cas où le système ne possède pas des classifieurs binaires, mais des classifieurs unaires : le système de classification consiste alors en la mise en compétition de systèmes génératifs n'ayant aucun pouvoir de discrimination (c'est-à-dire que chaque classe possède un modèle et que celui-ci fournit un score de vraisemblance entre le modèle et l'item à classer). Dans un tel cas, on peut considérer que le système de classification fournit une réponse sous la forme d'un tableau de grandeurs chiffrées (score associé à chaque classe) pour chaque item se soumettant à la classification. Si nous acceptons pour seule hypothèse que plus le score associé à une classe est élevé, plus celle-ci est crédible (ce qui correspond au premier axiome de Cox-Jaynes [125]), alors, il est possible d'inférer une réponse évidentielle avec tous les avantages que cela comporte. En considérant le résultat algébrique de la comparaison des scores de chacun des couples de classes possibles, on obtient une série d'indices pertinents très semblables aux précurseurs des FCE : ils indiquent en effet l'appartenance comparée de l'item pour chaque couple de classes, exactement de la même manière que chaque élément d'un banc de SVM de type 1vs1 fournit une réponse de type "plutôt telle classe que telle autre", assorti d'un score permettant de mesurer la quantité de préférence émise. La seule difficulté est donc toujours la même : à partir de ces seuls scores de distance ou de préférence relative comment déterminer correctement les FCE ? Là encore, nous proposons la validation croisée. Dans le cadre d'un partenariat avec l'université de Boğaziçi d'Istanbul, nous avons eu l'occasion de tester l'efficacité d'une telle méthode, dont nous reprenons les principaux résultats au chapitre VI (p. 182).

V.3.3 Transformée Crédale

Comme nous l'avons dit, le tableau de scores fourni par un système de classification unaire satisfait au moins le premier axiome de Cox-Jaynes (cf. appendice A.2.2 p. 270). Dans le cas où les 3 axiomes sont vérifiés, le tableau de scores est alors équivalent à une normalisation près, à une probabilité subjective. Dès lors, la méthode que nous proposons est aussi valable pour toute probabilité subjective, mais aussi pour toute probabilité objective puisque ces dernières obéissent à une définition plus stricte. Ainsi, par dérivation, nous avons à notre disposition une méthode de conversion des probabilités en fonctions de croyance. Cependant, la nécessité de devoir faire appel à une validation croisée pour déterminer là où se place la frontière entre le doute et la certitude n'est pas utilisable dans un tel cas. En théorie, la variation de cette frontière induit la définition d'une famille de fonctions de croyance correspondant à une seule probabilité, et cela n'est pas gênant pour la même raison que celle justifiant la multiplicité des Transformées Pignistiques Inverses (cf. p. 158 et [116]). Cependant, l'hypothèse selon laquelle la différence entre deux scores est liée à la quantité de doute par une seule fonction caractérisant l'incertitude (la fonction d'appartenance) est relativement restrictive, et cela devrait réduire la marge de manœuvre pour la détermination de la FCF. C'est pourquoi, nous proposons de (1) supprimer l'information redondante en ordonnant le tableau contenant les valeurs de la fonction de probabilité par scores décroissants et (2) de ne considérer que les comparaisons entre classes ayant des scores successifs dans le tableau ainsi ordonné. Ainsi, dans le cas de N classes, il n'y a plus $N\times(N-1)/2$ comparaisons, mais seulement N-1, celles-ci résumant toute l'information disponible. Dès lors, le même processus est utilisé : mise en place des FCE et utilisation de la transformation $\mathbf{T_R}$. Ainsi, la différence entre le score des $i^{\text{ème}}$ and $i+1^{\text{ème}}$ classes conduit à la création d'une FCE dont la masse de croyance est partagée entre les deux seuls éléments focaux suivants $\{C^1,\ldots, C^i\}$ et $\{\Omega\}$, avec la convention que C^1 est la classe ayant le plus haut score et C^N le plus faible. Comme la somme des masses de croyance sur ces deux éléments focaux vaut 1, définir la répartition entre le doute et la certitude pour une FCE à partir du résultat de la comparaison est équivalent à définir la masse de croyance en $\{\Omega\}$. Ce qui peut être fait en résolvant une équation du type :

$$\mathrm{ProbT}\left(m^{\mathrm{FCF}}\right) = p$$

où ProbT est une méthode de transformation d'une probabilité en fonction de croyance, et p la distribution de probabilité à convertir. Suivant la méthode que l'on accepte comme transformée de probabilité, des relations différentes entre probabilités et fonctions de croyance sont implicitement acceptées : comme cela est expliqué dans les appendices A (p. 255) et B (p. 277), il n'y a pas de vision unique sur le lien entre probabilité et croyance. Suivant le type de lien que l'on décide de considérer, l'une ou l'autre des nombreuses transformées probabilistes existantes sera adoptée. Lors de la définition d'une transformée inverse (transformée crédale), il est évident que les mêmes présuppositions doivent rester valables, et que la combinaison des deux transformées doit rester sans effet. D'où l'équation précédente. Comme la FCF est le résultat de la

combinaison de Dempster des FCE, il est possible de la réécrire de la manière suivante :

$$\mathrm{ProbT}\left(m_1 \textcircled{\cap} m_2 \textcircled{\cap} \ldots \textcircled{\cap} m_{N-1}\right) = p$$

La résolution de cette équation permet donc de définir une transformée crédale correspondant à l'inverse de la transformée probabiliste utilisée. Notons que par construction de l'ensemble des FCE sur un ensemble complètement ordonné, le résultat de la transformée crédale (la FCF) est (1) normalisé ($m(\emptyset) = 0$), et (2) **consonant** (cela signifie que les éléments focaux sont ordonnés par rapport à l'opérateur d'inclusion).

A titre illustratif, voici la transformée crédale que nous proposons pour convertir une probabilité en fonction de croyance sous l'hypothèse que les liens entre fonctions de croyance et probabilités sont les axiomes permettant de définir la Transformée Pignistique. Cela revient à considérer que la Transformée Pignistique est l'inverse de cette transformée crédale. On a donc :

$$\mathrm{PigT}\left(\mathrm{CredT}_{\mathrm{PigT}}(p)\right) = p$$
$$\text{avec } \mathrm{CredT}_{\mathrm{PigT}}(p) = m_1 \textcircled{\cap} \ldots \textcircled{\cap} m_{N-1} = m_{\textcircled{\cap}}$$

Par définition de la combinaison de Dempster, nous avons :

$$\begin{cases} m_{\textcircled{\cap}}\left(\mathcal{C}^1\right) = m_1\left(\mathcal{C}^1\right) \\ m_{\textcircled{\cap}}\left(\left\{\mathcal{C}^1, \mathcal{C}^2\right\}\right) = m_2\left(\left\{\mathcal{C}^1, \mathcal{C}^2\right\}\right) \cdot m_1(\Omega) \\ \vdots \\ m_{\textcircled{\cap}}\left(\left\{\mathcal{C}^1, \ldots, \mathcal{C}^i\right\}\right) = m_i\left(\left\{\mathcal{C}^1, \ldots, \mathcal{C}^i\right\}\right) \cdot \prod_{j=1}^{i-1} m_j(\Omega) \\ \vdots \\ m_{\textcircled{\cap}}(\Omega) = \prod_{j=1}^{N-1} m_j(\Omega) \\ m_{\textcircled{\cap}}(A) = 0 \quad \forall \text{ autre } A \in 2^{\Omega} \end{cases}$$

Si l'on applique la Transformée Pignistique, nous obtenons :

$$\begin{cases} \mathrm{BetP}\left(\mathcal{C}^1\right) = m_{\textcircled{\cap}}\left(\mathcal{C}^1\right) + \dfrac{m_{\textcircled{\cap}}\left(\left\{\mathcal{C}^1, \mathcal{C}^2\right\}\right)}{2} + \cdots + \dfrac{m_{\textcircled{\cap}}\left(\left\{\mathcal{C}^1, \ldots, \mathcal{C}^i\right\}\right)}{i} + \cdots + \dfrac{m_{\textcircled{\cap}}(\Omega)}{N} \\ \mathrm{BetP}\left(\mathcal{C}^2\right) = \dfrac{m_{\textcircled{\cap}}\left(\left\{\mathcal{C}^1, \mathcal{C}^2\right\}\right)}{2} + \cdots + \dfrac{m_{\textcircled{\cap}}\left(\left\{\mathcal{C}^1, \ldots, \mathcal{C}^i\right\}\right)}{i} + \cdots + \dfrac{m_{\textcircled{\cap}}(\Omega)}{N} \\ \vdots \\ \mathrm{BetP}\left(\mathcal{C}^i\right) = \dfrac{m_{\textcircled{\cap}}\left(\left\{\mathcal{C}^1, \ldots, \mathcal{C}^i\right\}\right)}{i} + \cdots + \dfrac{m_{\textcircled{\cap}}(\Omega)}{N} \\ \vdots \\ \mathrm{BetP}\left(\mathcal{C}^N\right) = \dfrac{m_{\textcircled{\cap}}(\Omega)}{N} \end{cases}$$

Comme BetP($\mathcal{C}^i$) = P($\mathcal{C}^i$) $\forall$ i, nous avons,

$$m_i(\Omega) = 1 - i \cdot \frac{\mathrm{P}(\mathcal{C}^i) - \mathrm{P}(\mathcal{C}^{i+1})}{\prod_{k=1}^{i-1} m_k(\Omega)}$$

qui peuvent être calculées de manière itérative selon les i croissants, et qui ne dépendent que des résultats des soustractions des valeurs consécutives des probabilités subjectives P(C^i)–P(C^{i+1}).

Ainsi, nous définissons de manière formelle une transformée crédale inverse de la Transformée Pignistique (mais qui ne correspond pas aux nombreuses Transformées Pignistiques inverses que l'on trouve dans la littérature). Notons $\boldsymbol{M}_p(.)$ le résultat de cette transformée appliquée à une fonction de probabilité p définie sur Ω, pour laquelle la probabilité en un élément C de Ω est notée P(C). $\boldsymbol{M}_p(.)$ est donc définie de la manière suivante :

$$\boldsymbol{M}_p = m_1 \textcircled{\cap} m_2 \textcircled{\cap} \ldots \textcircled{\cap} m_{N-1}$$

avec

$$m_i(\Omega) = 1 - i \cdot \frac{\mathrm{P}(\mathcal{C}^i) - \mathrm{P}(\mathcal{C}^{i+1})}{\prod_{k=1}^{i-1} m_k(\Omega)}$$

$$m_i\left(\left\{\mathcal{C}^1, \ldots, \mathcal{C}^i\right\}\right) = i \cdot \frac{\mathrm{P}(\mathcal{C}^i) - \mathrm{P}(\mathcal{C}^{i+1})}{\prod_{k=1}^{i-1} m_k(\Omega)}$$

$$m_i(A) = 0 \quad \forall \text{ autre } A \in 2^{\Omega}$$

Par construction, $\boldsymbol{M}_p(.)$ est une fonction de croyance consonante.

Bien sûr, si une autre transformée est utilisée pour convertir les fonctions de croyance en probabilité (par exemple la transformée de plausibilité), il est possible de définir une autre transformée crédale. En revanche, il n'est pas garanti que celle-ci ait une expression analytique telle que celle que nous venons de donner.

V.4 Conclusion du chapitre

Dans ce chapitre, nous avons abordés trois points. Dans la première section, nous avons traité de la reconnaissance de la Position. Nous avons brièvement résumé les différentes étapes détaillées aux chapitres précédents permettant de la reconnaître par un algorithme naïf. Son évaluation a permis de quantifier les points forts et faibles de la méthode proposée.

La reconnaissance de la Configuration représente la principale contribution de ce chapitre. Nous proposons une méthode permettant de réduire la variabilité des formes de main en supprimant le poignet. Ensuite, nous proposons d'utiliser les DFM comme attributs de classification, car il ressort de nos évaluations que :

– Ils sont les plus performants.

– Ils sont stables par rapport au paramétrage des SVM.

– Ils ont une bonne capacité de généralisation mono-codeur comme multi-codeurs.

La méthode de classification proposée est la Combinaison Evidentielle d'un banc de C-SVM dont la fonction noyau est une sigmoïde.

En parallèle de cela, la preuve de l'efficacité de la Combinaison Evidentielle est donnée, ainsi que son intérêt pour la combinaison de classifieurs hétérogènes. Ensuite, dans la troisième section, son utilisation est généralisée à :

– des classifieurs binaires non crédaux pour lesquels il n'est pas possible de définir une marge de doute ;

– des classifieurs unaires ;

– des classifieurs probabilistes.

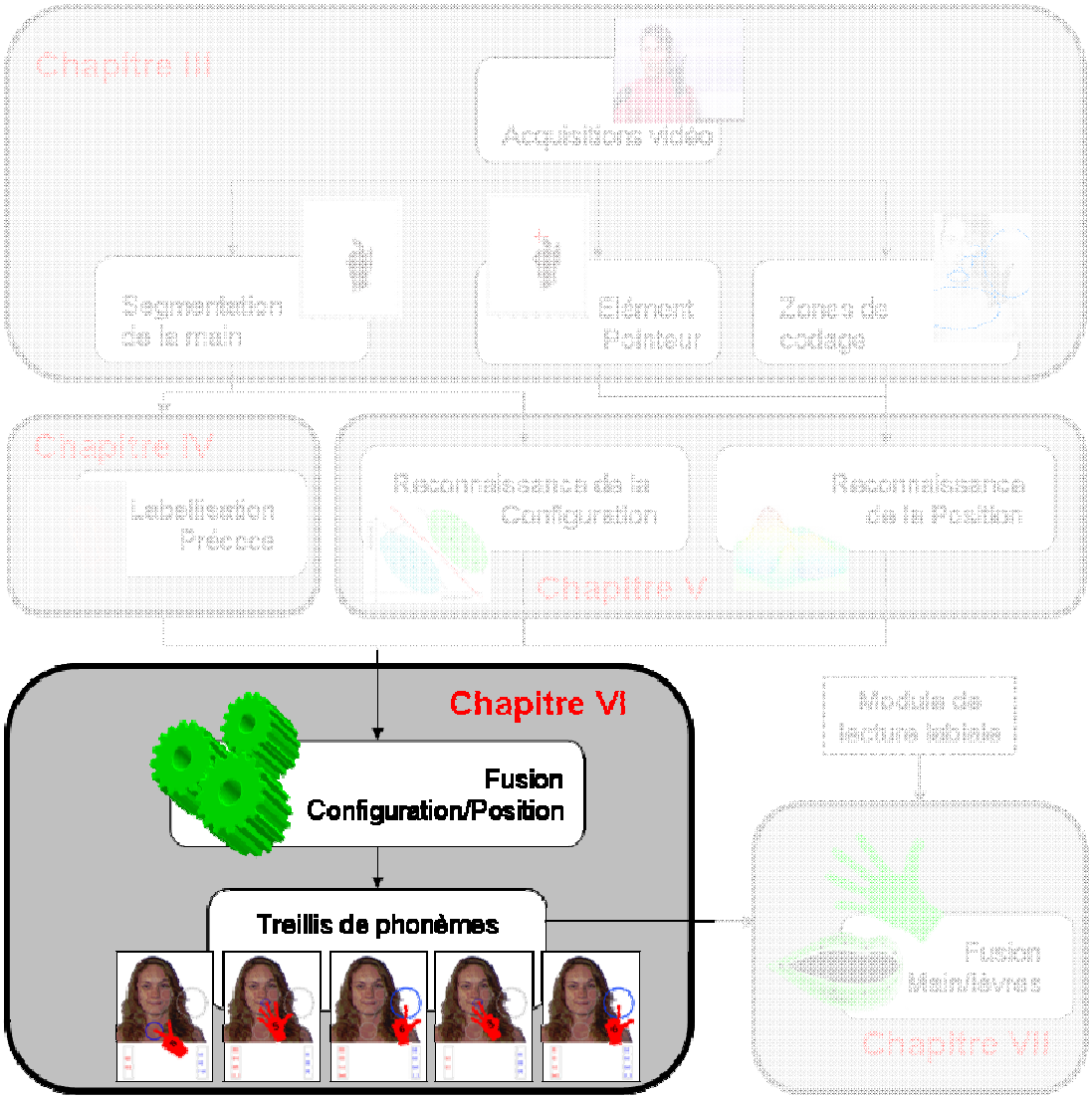
Ce dernier cas n'est possible que par la définition de transformées crédales permettant de convertir un tableau de scores et éventuellement une probabilité (même subjective) en une fonction de croyance.

D'une manière générale, l'utilisation de fonctions de croyance pour combiner les SVM permet de nouveaux développements. Il serait intéressant d'utiliser les SVRDM afin de mettre en place de véritables classes de rejet, mais aussi, de les utiliser de manière comparable à [106], afin d'augmenter la qualité de la classification.

Il serait aussi intéressant d'appliquer cette méthode à des schémas de type ECC, et de voir dans quelles mesures cela permet une amélioration. A ce titre, l'utilisation d'une métrique du type de celle déduite de la fonction d'appartenance que nous utilisons permettrait de raffiner la distance de Hamming classiquement utilisée (il ne s'agirait donc plus de compter le nombre de bits avec une erreur, mais de faire la somme de la valeur absolue des erreurs, évaluée entre 0 et 1). Même si dans les codes correcteurs d'erreurs classiques, cela n'a aucun intérêt, dans le cas d'un problème de classification, cela peut prendre tout son sens.

Enfin, il serait intéressant d'utiliser une sigmoïde comme fonction d'appartenance dans le cas de la Combinaison Evidentielle et de définir celle-ci conjointement à l'apprentissage, comme c'est déjà le cas pour la définition de probabilités *a posteriori*. Dès lors, il serait possible de comparer ces deux méthodes en toute objectivité.

CHAPITRE VI
INTERPRETATION PHONEMIQUE



Jusqu'à présent, nous avons extrait les informations pertinentes de chaque image (chapitre III p. 58), et nous avons mis en place une méthode permettant de découper une séquence vidéo à partir d'informations cinématiques de bas niveau, afin de faire ressortir des zones de stabilité et des zones de transition pour chacune des deux composantes du geste manuel du LPC (chapitre IV p. 101). Par la suite, (chapitre V p. 123), nous avons mis en place des algorithmes de reconnaissance efficaces permettant d'extraire les informations de Configuration et de Position d'une image cible. Maintenant, il reste à reconstituer le geste original complet, afin de l'interpréter en termes phonémiques.

Comme chaque geste est composé d'une Position et d'une Configuration, il est nécessaire d'effectuer un couplage entre les deux flux que nous avons traités séparément. En effet, cela seul permet de considérer un doublet Position/Configuration comme une entité gestuelle complète. Une fois ce couplage réalisé pour toute la séquence, il est possible de reconstituer la succession des gestes réalisés par le codeur (cf. Figure VI-1). Cependant, cette tâche est difficile, en raison des problèmes de synchronisation entre les deux flux (section II.4 p. 48), ce qui nous a obligés à utiliser des images cibles différentes pour la Position et pour la Configuration au chapitre IV (p. 101).

Figure VI-1 : illustration du problème à l'issu de la reconnaissance image à image (première et deuxième lignes), et de la labellisation précoce (troisième ligne). A partir de la définition des zones de stabilité désynchronisées entre les deux flux, il faut reconstruire la succession gestuelle (quatrième ligne).

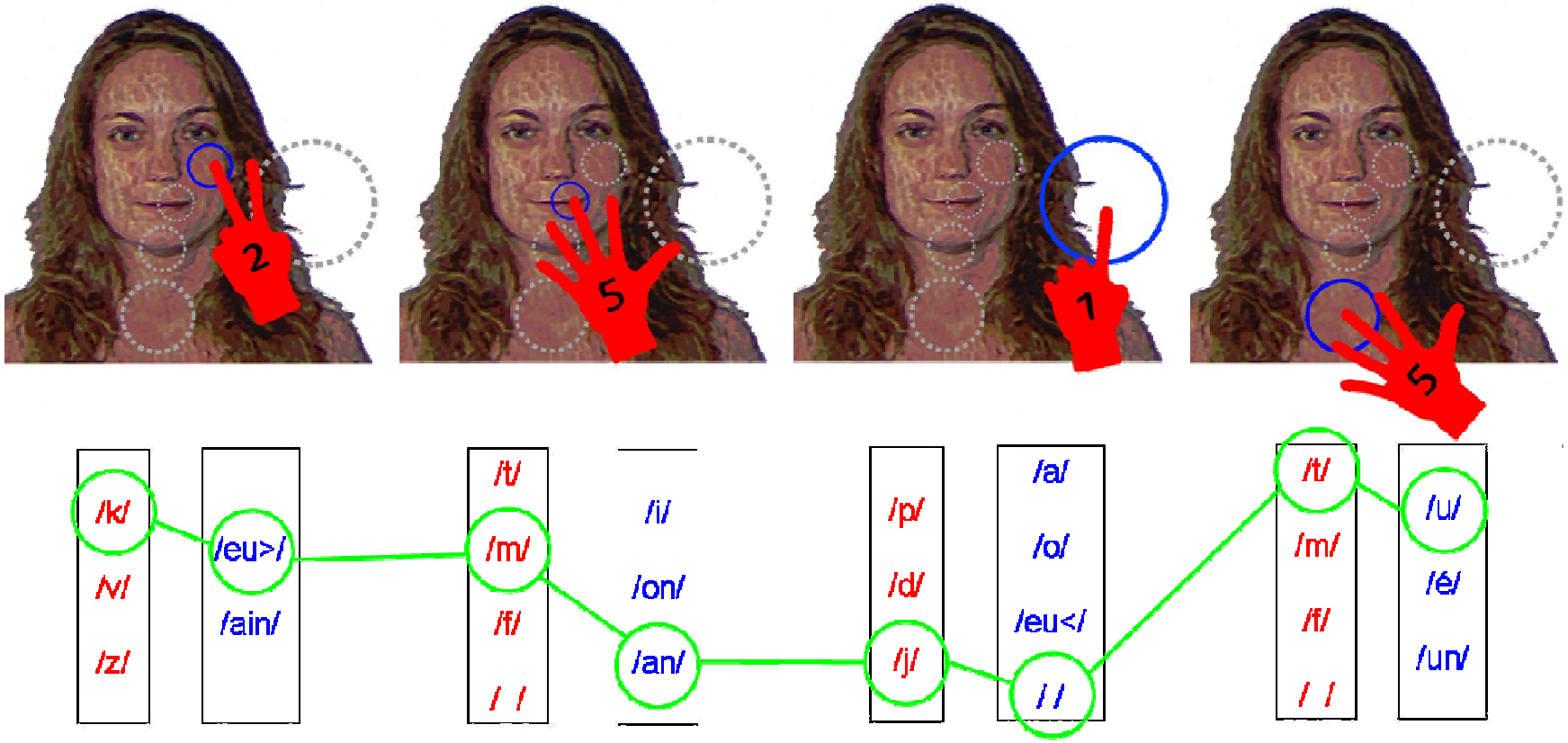


Figure VI-2 : illustration d'une séquence de gestes reconnus. Pour chacun des 4 gestes réalisés, une IC représentant à la fois la Configuration et la Position est présentée ainsi que l'ensemble de consonnes correspondant à la Configuration et l'ensemble des voyelles correspondant à la Position. Par exemple, pour la première IC, la Configuration 2 peut être utilisée par le codeur pour articuler /k/, /v/ ou /z/, chacun de ces phonèmes étant différent aux lèvres. De même la Position Pommette permet de coder /eu>/ (le /eu/ fermé de "feu", différent du /eu/ ouvert de "beurre", que nous avons noté /eu</, plutôt que d'utiliser l'alphabet phonétique). Il en va de même pour toutes les IC. Ainsi, l'ensemble des combinaisons de phonèmes possibles est représentable par un treillis. Celui-ci contient le message codé, qui peut être retrouvé en fonction du mouvement labial. Dans notre exemple, il s'agit de la question "Que manges-tu ?".

Une fois le couplage Position/Configuration réalisé, la succession de gestes du codeur devient une trajectoire articulatoire à part entière, et il est possible de l'interpréter en termes d'une succession de phonèmes. En revanche, il est encore trop tôt pour associer un unique phonème à chaque geste. Rappelons que le LPC est composé de gestes manuels et de mouvements labiaux similaires à la parole orale. Les gestes manuels seuls sont aussi ambigus que le mouvement labial seul. Ainsi, l'interprétation phonémique possible à ce niveau n'est pas le transcodage complet de la succession des phonèmes qui constituent le message. Tout au plus, il est possible de reconstituer le treillis de phonèmes qui contient le message complet. Ce treillis est constitué d'autant de couches qu'il y a de gestes. Chacun des gestes peut coder plusieurs phonèmes qui correspondent aux éléments de la couche du treillis associé à ce geste. L'ensemble des couches du treillis permet de considérer l'ensemble des combinaisons de phonèmes que peut encoder la séquence gestuelle. Parmi toutes ces combinaisons, une seule correspond au mouvement des lèvres, et cette combinaison correspond au message codé (cf. Figure VI-2).

Une autre manière de voir cela est de considérer que le geste manuel permet d'obtenir un premier treillis, et que le mouvement labial permet d'en obtenir un second. C'est l'intersection des deux treillis qui permet de trouver la chaîne phonémique du message (cf. Figure VI-3).

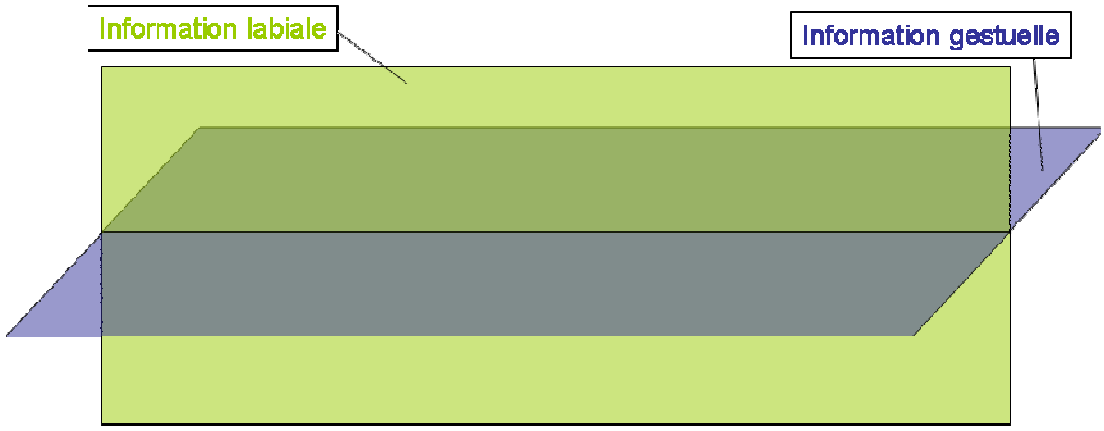


Figure VI-3 : intersection des treillis manuel et labial pour la reconstitution complète du message codé.

Finalement, il apparaît que la récupération du message original correspond à la fusion de deux informations labiale et manuelle, et que nous ne nous intéressons qu'à cette seconde information. De même il apparaît que le mouvement manuel est lui aussi le résultat de la fusion de deux canaux d'information. Or, il apparaît dans [7] que les difficultés concernant la fusion des informations labiale et manuelle sont les mêmes que pour la fusion des informations de Configuration et de Position. Ces difficultés sont de deux niveaux différents :

– Le niveau **multimodal**, puisqu'il s'agit de fusionner deux modalités différentes (la succession des Configurations et des Positions dans un cas, et la succession des gestes manuels et des mouvements labiaux dans l'autre) afin d'obtenir un seul message.

– Le niveau **temporel**, puisqu'il apparaît dans [7] que la main et les lèvres sont fortement désynchronisées, et puisque nous avons montré dans la section II.4 (p. 48) qu'il en est de même pour les Configurations et les Positions.

En théorie, ces deux difficultés sont quasiment inexistantes dans le cas du LPC. En effet, dans un codage théorique parfait, les différentes modalités ne sont pas désynchronisées (Position, Configuration et mouvement labial sont réalisés en même temps pour chaque phonème), et leur réalisation est non ambigüe. Ainsi, la fusion des modalités se résume au calcul de l'intersection des ensembles d'éléments du code, eux-mêmes codés par chaque modalité à chaque instant. Une telle opération d'intersection est très simple, d'autant que par construction, le LPC garantit l'existence et l'unicité de cette intersection.

En pratique, cela n'est évidemment pas le cas :

– la réalisation des gestes manuels (ou labiaux) peut être imparfaite, ou sa reconnaissance ambigüe, de sorte qu'il faille s'appuyer sur le contexte pour arriver à calculer l'intersection des flux des différentes modalités. Ainsi, leur combinaison devient non triviale.

– La synchronisation parfaite correspond à un académisme beaucoup trop strict pour qu'il soit raisonnable de se restreindre à ce type de code.

Dans ce travail, nous ne nous intéressons qu'à sa résolution pour le cas de la fusion des Configuration et des Positions, et la dynamique labiale est laissée de côté. Dans ce chapitre plus particulièrement, nous traitons la fusion des informations de Configuration et de Position de manière simplifiée.

Les difficultés de la fusion des flux de Positions et de Configurations sont décrites à deux niveaux (le niveau temporel, et le niveau multimodal). En général, ces deux aspects sont indissociables, chacun des deux aspects rendant l'autre plus difficile. Ainsi, la fusion de plusieurs modalités non corrélées et désynchronisées est un problème très complexe encore non résolu [137]. C'est pourquoi, dans la littérature, la plupart des solutions proposées ne permettent de traiter que des situations simplifiées, dans lesquelles l'une des difficultés a été supprimée ou fortement réduite, par l'imposition de contraintes fortes sur les données du problème. Par exemple, dans le cas de l'**ASL** (American Sign Language, ou langue des signes américaine), dont la reconnaissance automatique est un problème largement abordé, il n'existe aucun travail traitant de ces deux aspects simultanément : peu d'études s'intéressent au codage continu (problématique d'intégration temporelle [138], [153]), ou aux modalités non manuelles de ce langage [139], [140] (problématique de la combinaison multimodale). La plupart ne portent que sur la reconnaissance de gestes isolés

[131], [132], [133], [134], [135], et aucune ne traite ces deux aspects. Ainsi, la reconnaissance automatique de l'ASL lors d'un codage continu et multimodal est trop compliquée, et les recherches ne portent que sur des situations permettant de contraindre un des aspects des difficultés que cela implique.

Ainsi, nous envisageons de faire de même, et nous considérons dans ce chapitre que, parmi les deux difficultés du problème (combinaison multimodale et intégration temporelle), la complexité de l'une d'elle est négligeable devant l'autre. Pour cela, nous ne nous intéressons qu'à des situations où la composante de l'une des difficultés n'apparaît pas de manière significative. Cela nous simplifie la tâche, car l'espace des sémantiques à inférer est beaucoup plus restreint. Ainsi, nous considérons successivement les deux situations plus simples suivantes :

– dans un cas, nous considérons que les **désynchronisations sont fortes**, mais que la **fusion des modalités est triviale** ; par conséquent, le problème est principalement celui de l'intégration temporelle des résultats. En pratique, cela correspond à une situation où (1) le codage LPC est de bonne qualité sans être parfait (il y a des désynchronisations propres au codage réel), (2) la séquence est acquise dans de bonnes conditions (afin que les traitements image à image ne soient pas ambigus et que la reconnaissance soit fiable). En conséquence, la difficulté principale est de resynchroniser les flux de Positions et de Configurations, c'est-à-dire réussir à mettre en correspondance les ICP et les ICC. Une fois cela effectué, la combinaison des deux informations est très simple puisque les résultats de la reconnaissance sur chacun des flux sont fiables et précis. En pratique les séquences de nos corpus pour lesquelles le codage est réalisé par un codeur professionnel correspondent à ces critères. En revanche, dès que l'on utilise des vidéos pour lesquelles le codage est de moindre qualité, la reconnaissance de chacun des flux est moins fiable et leur combinaison n'est plus évidente. Ainsi, pour de telles vidéos, nous sortons du cadre de cette hypothèse simplificatrice.

– Dans l'autre cas, les **informations de Configuration et de Position sont présentées en même temps** (comme dans un codage théorique parfait), mais leur **combinaison multimodale est difficile** (parce que la reconnaissance des composantes du geste sur les ICX est ambigüe). Cela correspond soit à des vidéos de mauvaise qualité (conditions d'acquisition dégradées), représentant du LPC où le codage est parfaitement synchronisé (mais cela n'existe pas), soit à des vidéos ne correspondant qu'à un seul geste LPC (il ne peut donc pas y avoir de désynchronisation), soit à des vidéos ne présentant qu'un seul signe multimodal de l'ASL.

Chacun de ces deux points de vue simplificateurs est traité dans une section de ce chapitre. Dans la première section, nous traitons le cas de l'intégration temporelle des modalités, en considérant qu'il s'agit de l'unique problème, alors que dans la seconde section, nous ne traitons que de la combinaison multimodale.

Comme ces travaux ont en partie été réalisés en collaboration avec l'université Boğaziçi d'Istanbul (lors d'un "PhD exchange" rendu possible par le réseau d'excellence européen Similar), nous n'avons pas uniquement travaillé sur le LPC, mais aussi sur l'ASL. Par conséquent, nous illustrons ces méthodes de fusion soit sur le LPC soit sur l'ASL en fonction de ce à quoi elles se prêtent le plus. Ainsi, la première hypothèse selon laquelle l'intégration temporelle est prépondérante, est traitée sur des séquences de LPC appartenant aux corpus déjà utilisés ; la seconde hypothèse, selon laquelle l'aspect multimodal est plus important, est illustrée sur des séquences d'ASL (en effet, l'étude de séquences vidéos ne représentant qu'un seul geste LPC isolé n'est pas d'un grand intérêt).

VI.1 Intégration Temporelle

VI.1.1 Spécification du problème

Dans la section II.4 (p. 48), nous avons introduit qualitativement les différents types de désynchronisation intervenant dans la production du code LPC par des humains, et nous avons expliqué en quoi ces désynchronisations rendaient le décodage automatique plus difficile. C'est entre autre pour cette raison que nous avons dû définir deux types d'IC distinctes, les ICC et les ICP au chapitre IV (p. 101). Lors de l'évaluation de la pertinence des ICX trouvées automatiquement, nous avons été confrontés au problème de la définition de la vérité terrain. En effet, dans certains cas, plusieurs images correspondent aux critères d'une ICX, alors qu'une seule est labellisée comme telle. Or ce choix implique aussi des considérations d'ordre temporel : si parmi toutes les images pouvant être choisies comme ICP, la labellisation précoce en choisit une qui est proche de l'ICC correspondant au même geste, alors, la fusion des deux informations s'en trouve facilité. En revanche, si une ICC relativement éloignée est choisie, la fusion est rendue plus difficile. Ainsi, il y a deux phénomènes pouvant se renforcer ou au contraire s'annuler, et qui ont une influence sur la fusion des informations de Configuration et de Position :

– Il y a la **désynchronisation par rapport au codage théorique** : si l'on considère l'existence d'un codage théorique parfaitement synchronisé (tel que le ferait un clone virtuel), nous parlons de désynchronisation pour tous les défauts d'ordonnancement (décrits au chapitre II) d'un code réel (produit par un humain) par rapport à ce code théorique. Dans l'idéal, il serait intéressant d'inférer ce codage théorique à partir du code réel, et ainsi de supprimer toutes les désynchronisations.

– Il y a les **décalages induits par la labellisation précoce** : une fois les modalités traitées séparément, les labellisations produites sur chacune d'elles ne sont pas forcément concordantes, puisque rien n'a été fait pour cela. Il peut donc apparaître des décalages supplémentaires.

Le problème est qu'il est impossible de connaître la part de désynchronisation et la part de décalage dans les deux flux d'informations que l'on cherche à fusionner. Ainsi, la seule chose à faire est de tenter de faire concorder les deux

flux, sans savoir si cette opération consiste en un recalage ou une resynchronisation. Il est néanmoins possible de se douter que si la désynchronisation est trop forte, il ne sera pas possible d'inférer le codage théorique correspondant, et donc, il ne sera pas possible de déchiffrer le message. C'est pour cela que dans le chapitre II, nous avons décidé de ne pas considérer des codages de trop mauvaise qualité. A l'inverse, il est raisonnable d'espérer corriger entièrement les décalages issus des autres traitements.

Dans cette section, nous mettons donc en place des algorithmes dont le but est de corriger les légères désynchronisations et les décalages issus de la labellisation précoce. Nous pouvons résumer les informations à notre disposition de la manière suivante :

– labellisation des images en IC et IT ;

– labellisation des images en fonction de leur appartenance à une des plages de stabilité entourant les ICX ;

– reconnaissance de la Position sur les ICP et les ITPM ;

– reconnaissance de la Configuration sur les ICC et les ITCM ;

– réunion éventuelle de plages de stabilité correspondant à un même geste, grâce à l'étude des ITXM.

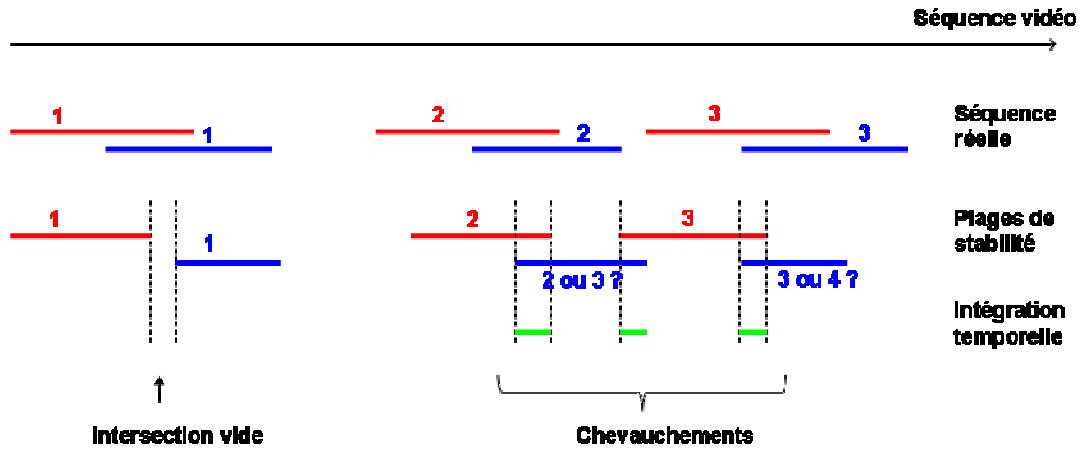


Figure VI-4 : illustration schématique du problème de décalage entre 2 flux avec la Configuration au dessus et la Position en dessous. Alors qu'en réalité les gestes sont réalisés avec une synchronisation suffisante pour qu'un humain les décode, le processus complet de décodage automatique amène une définition biaisée des plages de stabilité. Il peut en résulter une intersection vide ou un chevauchement.

A ce stade, il n'y a aucune continuité ni aucune cohérence dans la succession des informations extraites. En effet, nous avons une succession d'ICP et une succession d'ICC, toutes deux délimitées par des transitions. Or, à cause de la désynchronisation et du décalage qui apparaissent entre les ICC et les ICP, la connaissance de ces deux séries de cibles n'est pas équivalente à la série de

gestes. Pour resynchroniser/recaler le tout, il faut retrouver le couplage réel (produit par le codeur) qu'il y a entre chaque Configuration du flux de Configurations et chaque Position du flux de Positions. Cela n'est pas immédiat, comme le montre la Figure VI-1. La difficulté vient du fait que dans certains cas, les zones de stabilité de Configuration et de Position n'ont pas d'intersection commune, alors que dans d'autres cas, elles se chevauchent toutes mutuellement (par exemple une zone de stabilité en Configuration a une intersection commune avec deux zones de stabilité en Position - cf. Figure VI-4). Finalement, les cas où chaque zone de stabilité n'a qu'une seule intersection avec une seule autre zone de stabilité sont les seuls cas où la désynchronisation n'est pas difficile à compenser.

VI.1.2 Méthode proposée

Afin de resynchroniser/recaler les modalités et de faire apparaître des gestes complets, nous proposons les traitements suivants :

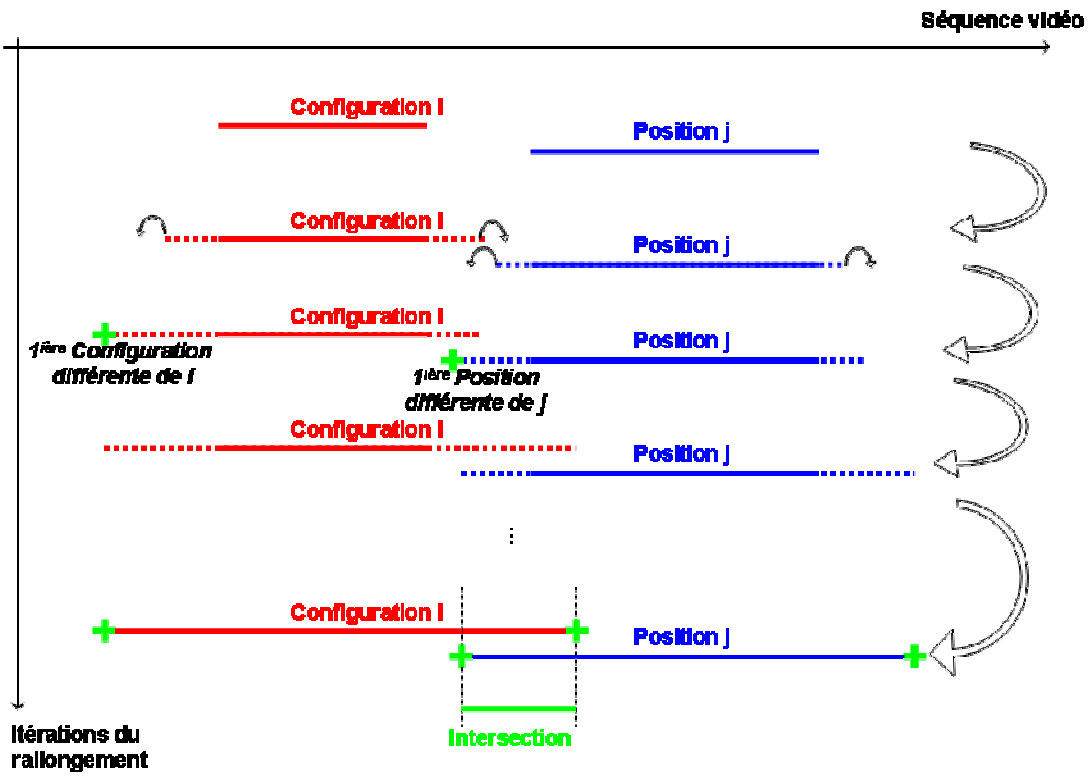


Figure VI-5 : processus de rallongement des plages de stabilité.

Etape 1 - Reconnaissance de la Position et de la Configuration sur toutes les images n'appartenant pas aux zones de stabilité (c'est-à-dire toutes les IT). Il est à noter que les classifieurs proposés au chapitre V ne sont pas *a priori* destinés à traiter de telles images de transition (leurs apprentissages sont réalisés sur des IC), mais cela n'a pas d'importance, comme nous le verrons plus loin.

Etape 2 - Chaque zone de stabilité de Configuration (resp. de Position) est ensuite étendue au maximum par le processus itératif suivant qui consiste à agréger les images aux frontières de la zone de stabilité : soit *C* (resp. *P*) la Configuration (resp. la Position) de l'ICC (resp. de l'ICP) de la zone de stabilité de Configuration (resp. de Position) considérée. Pour chaque image de transition immédiatement avant et immédiatement après la zone de stabilité, on compare *C* (resp. *P*) au résultat de la reconnaissance effectuée à l'étape 1. Si, pour une de ces images de transition, les deux Configurations (resp. Positions) correspondent, l'image est ajoutée dans la zone de stabilité. Ce processus s'effectue de manière itérative tant que l'on agrandit une zone de stabilité. Dès qu'une image présente une différence, le processus s'arrête (cf. Figure VI-5). Ainsi, malgré la sélectivité des seuils S^{immo} et S^{motion}, l'ensemble des images consécutives ayant une composante du geste commune à une IC appartient à la zone de stabilité de l'IC en question.

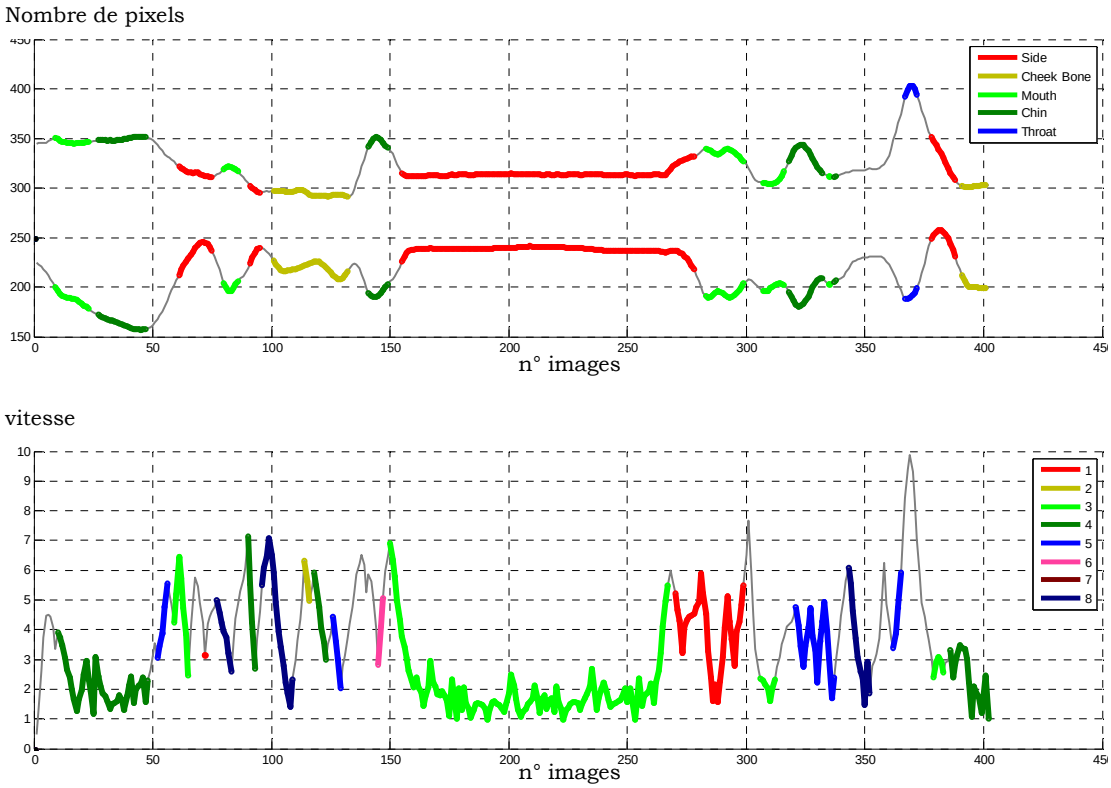


Figure VI-6 : représentation graphique du code obtenu en sortie de l'étape 3, superposé à la trajectoire sur laquelle la labellisation précoce est réalisée (cf. chapitre IV). En haut le codage de la Position et en bas celui de la Configuration.

Etape 3 - Toutes les images n'appartenant pas aux zones de stabilité allongées sont définitivement considérées comme des IT. A ce niveau-là, la séquence peut être représentée par deux tableaux de nombres d'une longueur correspondant au nombre d'images de la séquence. Le premier tableau permet de coder de 1 à 5 l'ensemble des Positions et le second, de 0 à 8 l'ensemble des Configurations. Nous utilisons la valeur 10 pour chacun des tableaux pour indiquer une transition de Position dans le premier et de

Configuration dans le second. Cela est illustré sur le schéma de la Figure VI-6, où la trajectoire permettant la labellisation précoce est représentée et surlignée dans une couleur propre à chaque valeur des tableaux en question. L'absence de surlignage correspond aux valeurs 0 et 10.

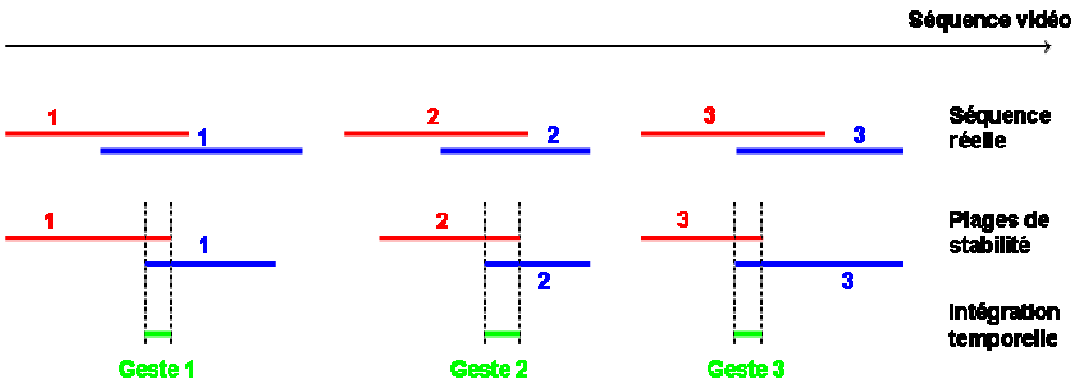


Figure VI-7 : après l'allongement des plages de stabilité, il est possible de trouver des images pour lesquelles le geste est entièrement défini.

Etape 4 - Jusqu'à présent, très peu d'information a été ajoutée : nous avons seulement rallongé au maximum les plages de stabilité. L'intérêt est de maximiser la possibilité que les deux plages de stabilité de Configuration et de Position d'un même geste aient une intersection non vide, et ce, malgré la désynchronisation/décalage entre Position et Configuration. Ainsi, sur l'intersection de ces deux plages de stabilité, il est possible de définir un geste complet (cf. Figure VI-7). La succession de ces derniers (cf. Figure VI-8) ainsi obtenue permet de définir le treillis de phonèmes contenant le message codé.

Notons que nous ne proposons à ce stade aucune méthode permettant de supprimer les chevauchements. La raison est la suivante : un chevauchement non supprimé rajoute un geste ; cela n'enlève aucune information, et le geste redondant peut éventuellement être supprimé par la suite par un traitement de plus haut niveau. En revanche la suppression par erreur d'un geste qui a été pris pour un chevauchement consiste en une suppression d'information qui n'est pas réparable par la suite. Nous préférons donc ne pas chercher à supprimer ce type d'erreur pour l'instant, et éventuellement proposer une solution plus tard, une fois que les situations entraînant ce type de chevauchement seront mieux connues.

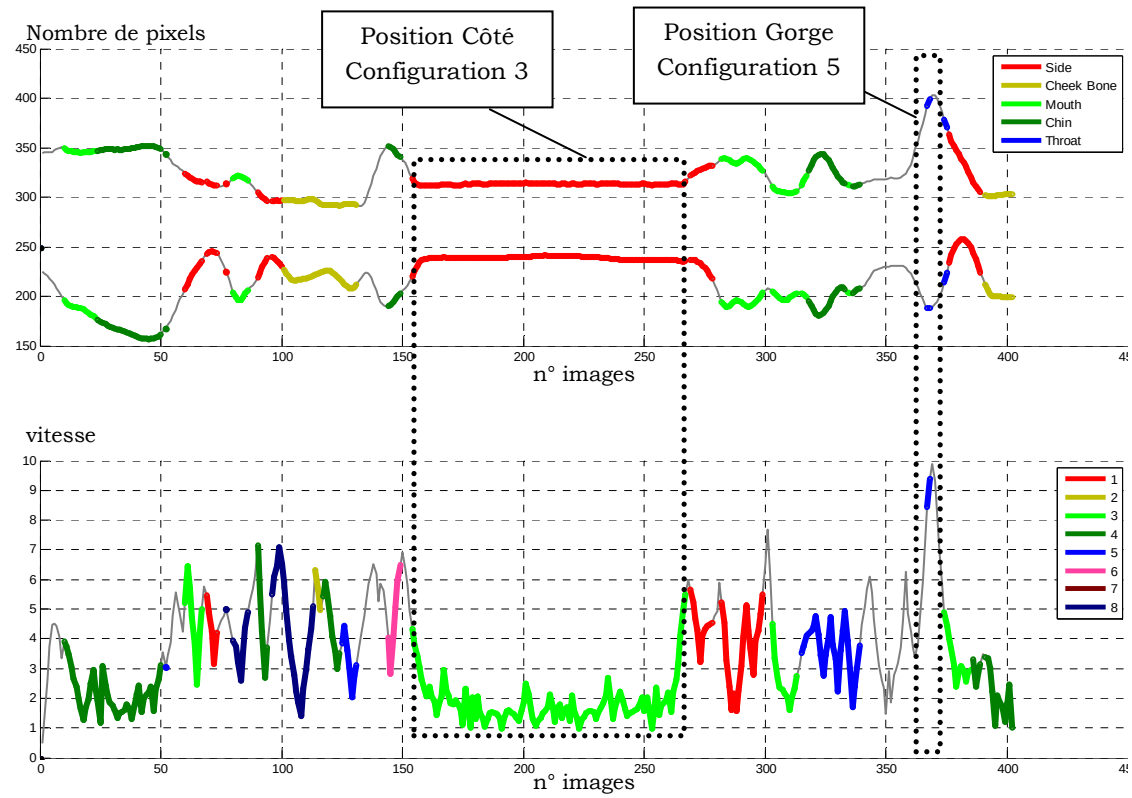


Figure VI-8 : après l'opération de rallongement/intersection de l'étape 4, la Configuration et la Position sont définies pour chaque geste.

VI.1.3 Evaluation

Avant l'évaluation proprement dite de notre méthode, nous proposons de justifier le choix de celle-ci. En effet, d'après les difficultés que nous devons résoudre, la manière la plus classique de procéder est de modéliser le problème sur un graphe où l'on cherche à trouver le couplage de poids maximum :

Soit G = ({ICP, ICC}, E), un graphe biparti où une classe de sommets représente les ICP et l'autre, les ICC, et où E est un ensemble d'arêtes pondérées de telle sorte que la pondération soit positivement corrélée à la synchronisation qu'il y a entre les plages de stabilité associées à une ICC et une ICP. Il s'agit alors de trouver le sous-graphe C de G qui représente le couplage de poids maximum de G. Ce sous-graphe a les propriétés suivantes :

– C = ({ICP, ICC}, M), avec $M \subset E$ tel que deux arêtes de M ne sont pas adjacentes.

– Tout autre couplage C' = ({ICP, ICC}, M') de G est tel que la somme des poids de M' est inférieure à la somme des poids de M.

Trouver C revient à associer les Configurations et les Positions de telle sorte que la synchronisation est maximisée. Cette méthode est séduisante, mais sa mise en pratique fait face à plusieurs difficultés :

– Tout d'abord, il n'est pas évident de trouver une mesure de pondération de E qui soit adaptée.

– Ensuite, il y a des situations où seule l'une des composantes change, ce qui en pratique revient à devoir coupler une ICC avec 2 ICP, ou le contraire. Cela n'est *a priori* pas possible avec les algorithmes classiques, puisque l'on empêche l'adjacence de deux arêtes. Il est donc nécessaire de mettre en place une heuristique adaptée.

– Il n'y a aucune preuve que le couplage de poids maximum corresponde au couplage réel réalisé par le codeur. Il y a donc à vérifier cette hypothèse en premier lieu (même si celle-ci est tout à fait raisonnable).

C'est au regard de ces difficultés, que nous avons choisi la méthode décrite précédemment. Intéressons-nous maintenant à l'évaluation de celle-ci.

Même si les traitements que nous proposons sont relativement élémentaires, ils permettent d'interpréter les résultats à un niveau d'abstraction beaucoup plus élevé que précédemment. En effet, il est désormais possible de mesurer la similarité entre le geste réel et le geste reconnu, et de considérer le geste comme un articulateur permettant de produire le message au même titre que le mouvement labial. Afin de focaliser le test sur la pertinence avec laquelle ce changement de niveau d'interprétation est réalisé, nous proposons le protocole expérimental suivant :

Choix d'un corpus de test. Il s'agit de récupérer un ensemble de séquences complètes mettant en scène un codeur professionnel dont le codage répond aux exigences que nous avons émises : il est nécessaire d'avoir un code suffisamment bien réalisé pour qu'il n'y ait aucune ambiguïté sur la reconnaissance des flux de Position et de Configuration. De plus comme nous cherchons à connaître les capacités du système en situation de codage continu, nous évitons les séquences trop courtes (de moins de 10 syllabes), car pour ces dernières les difficultés de désynchronisation n'apparaissent pas. Dans le même ordre d'idée, nous ne pouvons pas considérer les séquences où trop d'erreurs de codage ou trop d'hésitations viennent perturber la dynamique de codage, car cela modifie l'ordonnancement des flux de Position et de Configuration. Ainsi, la sélection du corpus de test se révèle être très délicate : c'est avec difficulté que nous extrayons 20 séquences du corpus ETTRAN N (environ 6000 images). A la section III.1 (p. 59), nous expliquons qu'aucun apprentissage n'est fait au niveau du contenu des séquences, et que par conséquent le pouvoir de généralisation à leur égard est complet. Afin de vérifier cela, nous ajoutons dans le corpus la séquence 158, que nous avons utilisée à maintes reprises dans nos illustrations, et qui a souvent servie à des tests préliminaires, ou à l'affinage de paramètres. Ainsi, il s'agit de la séquence qui est susceptible d'avoir le plus influencé les apprentissages. Il s'agira donc de vérifier que nos algorithmes ne permettent pas une meilleure interprétation de cette séquence que d'autres. Enfin, nous ajoutons d'autres séquences dont le codage est moins académique (reconnaissance plus difficile), afin d'avoir une idée de comment se

généralise notre procédure à des situations plus complexes. Ainsi, nous considérons une séquence du corpus ETTRAN BF (même codeuse que ETTRAN N, mais gant différent), une séquence de MAGOZ R (codeuse professionnelle), une séquence de MAGOZ B (codeuse malentendante ayant un code de bonne qualité) et une séquence de MAGOZ J, dont le code est plus difficilement interprétable qu'un codage professionnel. Dans tous les cas, les éventuels apprentissages (pour la reconnaissance de la Configuration) sont réalisés pour la reconnaissance mono-codeur sur ETTRAN N. Pour les autres séquences, la morphologie de la main ou du gant est donc inconnue.

Définition de la vérité terrain. Comme toujours, il est très délicat de déterminer une vérité terrain. Afin de déterminer la séquence de gestes réalisés, il est aussi possible de s'aider des scripts du corpus. Cependant, cela n'est pas toujours très fiable. En effet, (1) le texte n'est pas toujours parfaitement respecté, et celui-ci ne tient pas compte des pauses, des hésitations ou de petites erreurs invisibles au premier abord ; (2) de même qu'il existe des accents et des patois dans le français oral, il y a de nombreuses manières de coder une même phrase. Finalement, même en supposant qu'il est possible de déterminer sans erreur la séquence gestuelle en termes de contenu sémantique, il n'est pas garanti que cette vérité terrain soit la meilleure. En effet, si d'un point de vue cinématique, le mouvement réalisé est différent de celui codant le message, lequel des deux doit être reconnu ? Il est plus intéressant de retrouver directement le message réel, cependant, comme aucun traitement n'est effectué au niveau sémantique, cela revient à accepter que la correction de la différence entre le geste cinématique et le message réel relève de la chance. Ainsi, s'intéresser au mouvement cinématique semble plus honnête. Cependant sa définition n'est pas plus aisée. En effet, il est difficile de déterminer la séquence des mouvements réalisés quand on ne décode pas couramment le LPC. En passant la vidéo à vitesse normale, un œil non exercé n'est pas capable de repérer tous les gestes. Au ralenti, les phases de transition ont tendance à sembler plus stables qu'elles ne le sont réellement. Finalement, nous prenons le parti de visionner simultanément sur un écran partagé la séquence vidéo et un clone présentant la succession des gestes statiques (une image choisie parmi les 41 gestes à considérer, et une image noire durant les transitions), comme cela est illustré sur la Figure VI-9. Il s'agit alors de comparer les deux codes et de vérifier qu'ils sont similaires (aussi bien durant les transitions que durant les zones de stabilité). Cela n'est malgré tout pas parfaitement fiable, car quand une forte désynchronisation par rapport au codage théorique apparaît, il est difficile pour l'opérateur de déterminer la vérité terrain.

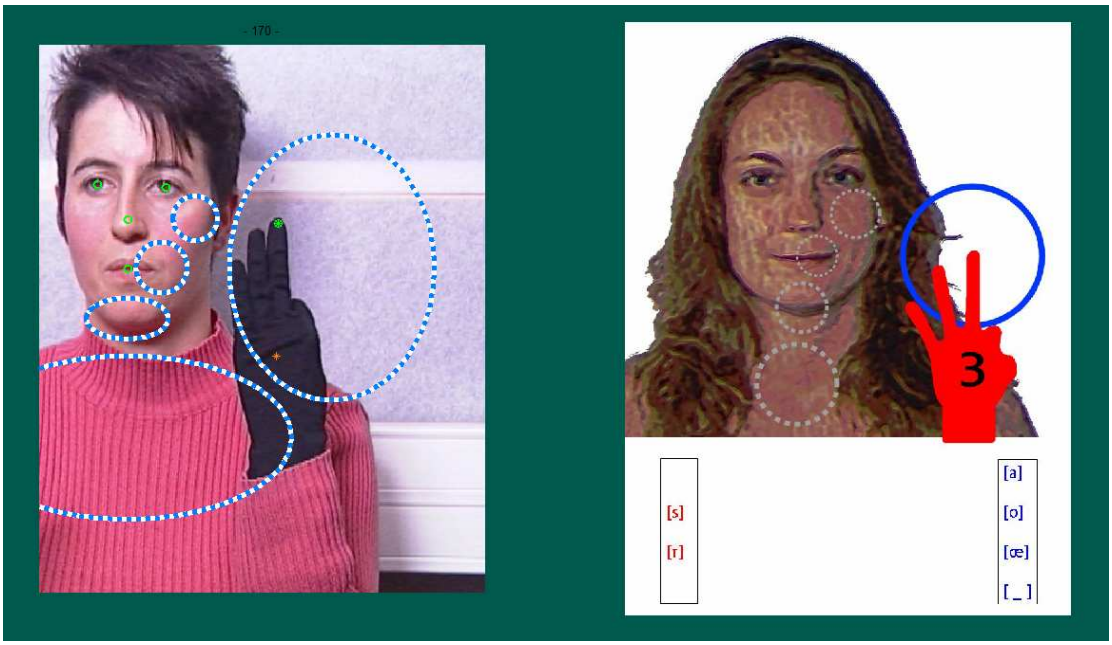


Figure VI-9 : illustration de l'interface permettant la validation du test.

Types d'erreurs remarquables. En raison de la difficulté de la définition de la vérité terrain, nous ne définissons pas le nombre total de gestes à reconnaître. Nous prenons comme référence le nombre de **gestes parfaitement reconnus**, c'est-à-dire pour lesquels la reconnaissance est correcte, mais aussi pour lesquels la segmentation temporelle en série de gestes est correcte. Ainsi, une image de transition prise par erreur pour une cible n'est pas comptabilisée comme correcte, même si une Position et une Configuration apparaissent par hasard et peuvent être reconnues dessus. Ensuite, nous comptabilisons le nombre de **zones de stabilité correctement repérées**, mais pour lesquelles la reconnaissance est erronée (par exemple une erreur de segmentation implique une confusion sur la Configuration ou sur la Position). Si l'on cherche à évaluer les méthodes de ce chapitre exclusivement, ces erreurs ne doivent pas être comptabilisées. En revanche, elles sont intéressantes pour l'évaluation du système global. Ensuite, nous comptabilisons les **petites erreurs de définition des zones de stabilité des gestes**. Il s'agit d'une catégorie d'erreur un peu "fourre tout", dans la mesure où il n'est pas toujours possible de déterminer l'origine de l'erreur. La plupart du temps, il s'agit d'une image qui est considérée comme stable au milieu d'une transition, en raison d'un problème de chevauchement des zones de stabilité. De temps en temps, il s'agit d'une erreur de reconnaissance qui a une influence sur la définition des zones de stabilité ou sur la fusion de deux d'entre elles après avoir mal reconnu une ITXM. Parmi les erreurs qui sont comptabilisés ici, un grand nombre sont dues à l'ensemble des algorithmes que nous avons mis en place, mais une proportion conséquente est aussi due au codage hésitant ou trop précipité. A chaque fois qu'une incohérence est apparue, nous la comptabilisons comme erreur parce que nous partons du principe qu'il est délicat de déterminer si elle vient du codeur ou de l'algorithme. Il s'agit donc d'une évaluation sévère donnant une borne supérieure de la quantité d'erreur du système. Enfin, nous comptons les **erreurs**

importantes. Il s'agit de gestes non équivoques qui n'ont pas du tout été repérés, soit par un défaut flagrant de la labellisation précoce, soit par une erreur à un autre niveau, mais ayant eu des conséquences sur la labellisation précoce.

Précaution de tests. En raison de la subjectivité du test, nous l'avons effectué une seconde fois sur un quart du corpus (5 séquences de ETTRAN N). Entre les deux évaluations des mêmes séquences, il n'y a que 92% de correspondance sur les erreurs, ce qui signifie que les résultats que nous donnons par la suite sont à interpréter avec précaution. Enfin notons que :

– Les quatre séquences issues des autres corpus (ETTRAN BF, MAGOZ R, B et J) sont comptabilisées séparément. Dans un premier temps, nous ne considérons que les 21 séquences de ETTRAN N (en incluant la séquence 158).

– La Configuration 0 en Position Neutre (ou Côté) n'est pas évaluée. En effet, sur l'ensemble des séquences, l'instruction de commencer et de finir la phrase par ce geste n'est pas toujours respectée (il n'est pas facile de contraindre à ce point un codage que la personne a automatisé depuis de nombreuses années). Ainsi, nous l'avons exclu des évaluations.

– Des tests sur des séquences acquises à des cadences plus faibles n'ont pas été possibles : les deux filtres de Kalman utilisés dans l'ensemble du traitement décrochent car le mouvement est trop important d'une image à l'autre.

Résultats quantitatifs. Sur l'ensemble des 20 séquences d'ETTRAN N, il y a 309 gestes parfaitement reconnus, 40 gestes bien repérés dans le temps mais mal reconnus, 61 "petites" erreurs de labellisation (dues à un mauvais codage ou à l'algorithme, sans faire la différence) et 15 "grosses" erreurs. Parmi ces dernières, les 2/3 ont une origine extérieure à la labellisation précoce : écartement des doigts trop important en Configuration 8, ce qui ne permet pas de repérer la composante de Position, ou flexion du poignet en Position Gorge empêchant de reconnaître la Configuration. Tout cela est résumé dans le Tableau VI-1. La séquence 158 ne donne pas des résultats différents : il y a 20 gestes parfaitement repérés, 2 erreurs de reconnaissance sur un geste bien repéré, 4 petites erreurs, et 1 grosse erreur (encore une fois due à un codage en Position Gorge). Ainsi, cela confirme qu'il n'y a pas d'influence des apprentissages au niveau du geste complet. Les 4 autres séquences avec gants/codeurs différents impliquent un même nombre d'erreurs dues à un mauvais fonctionnement de la labellisation précoce (ce qui indique un bon pouvoir de généralisation) mais un plus grand nombre d'erreurs dues à une mauvaise reconnaissance (la reconnaissance est plus faible face à plusieurs codeurs inconnus) et à un mauvais codage (celui-ci est de moins bonne qualité) : il y a 53 gestes parfaitement reconnus, 9 erreurs de reconnaissance, 8 petites erreurs et 5 grosses erreurs de labellisation.

Evaluation qualitative. L'évaluation qualitative des erreurs est assez caractéristique. L'augmentation de la taille des plages amplifie certains défauts que l'on avait déjà rencontrés sur la reconnaissance de la Configuration et de la

Position. Au niveau de la reconnaissance de la Configuration, les erreurs types étaient les suivantes : (1) rotation du poignet qui masque les doigts (2) flexion du poignet en Position Gorge qui déforme la main, (3) mélange de Configurations donnant une forme de main difficilement reconnaissable. Pour la reconnaissance de la Position, les erreurs types que l'on retrouve sont : (1) le biais dû à l'écartement des doigts dans la Configuration 8 et la Position Côté, (2) l'erreur de définition du doigt pointeur en Position Gorge, (3) les imprécisions cumulées du codage, de la définition des zones de pointage et du doigt pointeur.

Tableau VI-1 : évaluation de l'interprétation gestuelle sur les 20 séquences du corpus ETTRAN N.

Types de gestes repérés	Nombre et commentaire	Pourcentage par rapport au nombre de gestes parfaitement reconnus
Gestes parfaitement reconnus	309. C'est la référence du nombre de gestes	100%
Erreur de reconnaissance seulement	40. Ok par rapport à la labellisation des zones de stabilités	12.9%
Petites erreurs	61. Causes variables	19.7%
Grosses erreurs	15. dont 10 pour des raisons autres que la labellisation des zones de stabilité	4.9%

Ainsi, dans certains cas, l'intégration temporelle met en valeur les défauts déjà remarqués. Elle peut donc sembler avoir un effet néfaste. La solution la plus simple serait de proposer un traitement particulier pour ces cas, cependant cela n'est pas très élégant ; en effet, il est difficile de prévoir la généralisation à un grand nombre de codeur de la levée de telles exceptions.

Il apparaît aussi de nouvelles erreurs, intrinsèques au principe relativement simple selon lequel nous cherchons à effectuer cette intégration temporelle :

– En début et en fin de phrases, la désynchronisation entre Configuration et Position est tellement importante qu'il arrive qu'il n'y ait pas de recouvrement. (cf. section II.4 p. 48).

– Durant une transition assez lente entre deux Positions proches (Menton et Bouche par exemple), il arrive que le doigt pointeur soit instable et passe de l'une à l'autre à plusieurs reprises. Normalement, cela est filtré par la labellisation précoce, mais si ce phénomène est assez lent (phénomène de dérive), il laisse apparaître une succession de cibles. Un système de lissage supplémentaire serait donc nécessaire. Ce type de phénomène n'a pas

d'équivalent du point de vue de la Configuration (ou alors le codage est trop hésitant pour pouvoir espérer être décodé).

– Les Positions Côté ou Poitrine peuvent être codées tout en maintenant un certain mouvement, de par la faible précision du pointage qu'elles requièrent. Il ressort que certains mouvements de transition sur la Configuration commencent alors que la Position (soit Côté soit Poitrine) semble stable. Ainsi, au lieu d'un enchaînement du type {Côté, 2}, {Pommette, 1} il arrive qu'un enchaînement du type {Côté, 2}, {Côté, 1}, {Pommette, 1} soit reconnu. Ce problème est une illustration parfaite des cas où les plages de stabilité se chevauchent mutuellement (cf. Figure VI-4).

Comme ces chevauchements ne sont pas traités, il est naturel que ce type d'erreur apparaisse. Ainsi, la quasi-totalité des petites erreurs comptabilisées correspondent à des situations de chevauchement. Cependant, il est très difficile de connaître la proportion exacte de ces chevauchements. En effet, dans bien des cas, il est très difficile pour l'opérateur faisant les tests de déterminer s'il s'agit d'un chevauchement à supprimer ou au contraire, s'il s'agit d'un geste particulièrement transitif (codé de manière trop rapide et sans atteinte de cible). C'est principalement pour cette raison que, dans l'évaluation, nous ne faisons pas la différence entre les erreurs provenant du codage et les erreurs provenant des algorithmes utilisés.

Ces tests n'ont pu être réalisés que sur un faible nombre de codeurs. Comme ils ont trait à un niveau sémantique plus élevé que dans le reste de ces travaux, nous pensons que le nombre de codeurs, suffisant jusqu'à maintenant, ne l'est plus. Ainsi, de nouvelles acquisitions seraient nécessaires.

Discussion. Il existe des problématiques de fusion de l'information au cours du temps relativement similaires dans le cas des langues des signes. Ces dernières ont une approche radicalement différente du LPC. La forme des mains, la position de chaque articulation, leurs mouvements et la dynamique associée, les pantomimes, le positionnement spatial, etc. sont utilisés. Le message est transmis non seulement par des gestes manuels, mais aussi par des gestes non manuels et par leur enchaînement en un ensemble syntaxique. Ce sont donc des langages complètement dynamiques et multimodaux, pour lesquels les problèmes de fusion traités ici sont particulièrement ardus.

La reconnaissance de gestes isolés dans le cas de l'ASL est un sujet largement abordé dans la littérature, mais néanmoins non encore résolu. La plupart des méthodes ayant des résultats efficaces dans l'état de l'art sont des améliorations de la trame suivante, que l'on retrouve dans le traitement de la parole : pour chaque type de geste, un HMM est généré à partir d'une base d'exemples. Quand un signe à reconnaître se présente, on calcule pour chacun des gestes de la base un score corrélé à la probabilité que le HMM correspondant génère la séquence d'observations du signe inconnu. Enfin, on reconnaît le geste comme celui dont le HMM a fourni le plus haut score.

Quand il s'agit de s'intéresser à des phrases complètes, le travail est beaucoup plus difficile : la méthode précédemment décrite nécessite de connaître le début et la fin de chaque geste. En plus de cela, l'enchaînement de ces derniers n'est pas simplement séquentiel dans le temps. Il peut y avoir des combinaisons de gestes, des superpositions, etc. Finalement, il s'agit d'une véritable intégration temporelle, et non d'une "simple" segmentation temporelle. Concernant la parole orale, des méthodes existent pour cela ; en revanche, pour ce qui est de la langue des signes, cela est beaucoup plus compliqué, comme cela est expliqué dans [153]. Les raisons sont les suivantes : (1) il s'agit de traiter un signal vidéo, et la complexité calculatoire augmente beaucoup trop vite ; (2) il y a plusieurs modalités non synchronisées à prendre en compte ; (3) une décomposition en unités de base (comme dans le cas du traitement de la parole continue) est beaucoup plus difficile ; (4) un équivalent de la décomposition en éléments de base ne peut pas aboutir d'un point de vue calculatoire.

Ainsi, une "simple" segmentation temporelle en une série de séquences ayant une certaine unité permettrait déjà de réduire très fortement la complexité du problème. Ces séquences unitaires ne peuvent en aucun cas prétendre à être des éléments de base comparables aux phonèmes dans le cas de la parole. Néanmoins, leur segmentation permettrait de simplifier en partie le problème, de manière similaire à ce qui est proposé dans [153]. Nous avons cherché à mettre en place une telle segmentation en utilisant la méthode précédemment décrite, (notamment le Filtre Rétinien Dédié), qui fonctionne dans le cas du LPC. Les résultats n'ont pas été concluants. Selon nous, la principale cause de notre échec est la qualité des vidéos sur lesquelles nous avons travaillé. Comme cela est illustré sur la Figure VI-10, la langue des signes fait intervenir une gestuelle beaucoup plus complexe, que l'on ne peut aborder entièrement sur une acquisition, même stéréo. Les vidéos utilisées, adaptées à la mise en place de modèles génératifs, ne sont pas de qualité suffisante pour un FRD. Après les traitements nécessaires, celles-ci fournissent un signal tellement dégradé qu'un opérateur n'arrive pas à déterminer de vérité terrain quand à la segmentation attendue.

Malgré cet échec, nous pensons encore qu'il est possible d'utiliser cette méthode dans le cas de la langue des signes. De même que pour les travaux présentés sur le LPC, ce type d'études préliminaires nécessite dans un premier temps des bases de données parfaitement adaptées aux problèmes afin de vérifier le bien-fondé des hypothèses. La mise en place de tels corpus de données, ainsi que la poursuite de ces travaux est une piste de premier intérêt, tant sur la nouveauté que cela amène dans l'étude de l'ASL que pour la confirmation des algorithmes que nous avons utilisés dans le cas du LPC.

VI.2 Combinaison multimodale

Concernant le problème de la combinaison multimodale d'informations, nous nous sommes surtout concentrés sur son application au cas de l'ASL. Comme nous l'avons dit plus haut, cette combinaison est particulièrement riche dans les

langues des signes. Nous nous sommes donc intéressés à la fusion de gestes manuels et non-manuels. Nous reprenons ici les principaux résultats obtenus qui ont été publiés et détaillés dans [J1], [J2], [C1]. Au chapitre suivant, nous discutons de l'application de ces méthodes au cas du LPC.

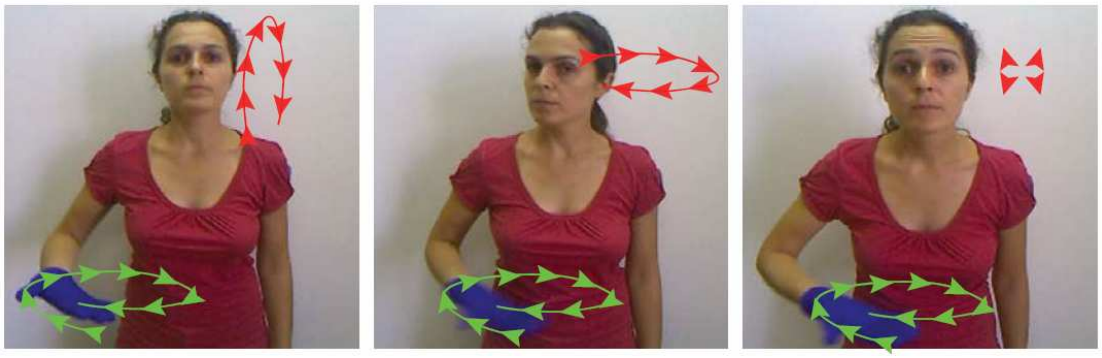


Figure VI-10 : illustration de la manière dont un mouvement de tête peut changer le sens du signe. A gauche, "Here", au centre "Not here" et à droite "Is here ?". Issus de [155].

Les signes non-manuels ne sont l'objet que de peu d'études car il s'agit d'un sujet encore très récent. La plupart de celles-ci s'intéressent à l'information non-manuelle, sans prendre en compte le contexte de l'information manuelle. Certains travaux se concentrent sur les expressions faciales [163], d'autres sur les mouvements de tête [155], mais toujours sans interaction avec le geste manuel. Cela est d'autant plus dommage que le geste non manuel a souvent comme objet d'apporter une variation, une précision ou une nuance par rapport au geste manuel (cf. Figure VI-10).

La combinaison des informations manuelles et non-manuelles dans le cas d'un codage continu revêt une importante dimension temporelle que nous ne traitons pas ici. Ainsi, nous nous concentrons sur des signes isolés, où le signe manuel et le signe non-manuel coïncident approximativement malgré une corrélation relativement basse (c'est-à-dire que la connaissance d'une modalité n'est pas informative par rapport aux autres modalités).

Cette faible corrélation entre les deux flux d'informations rend leur combinaison difficile et incertaine. En conséquence de quoi, nous avons dû développer des méthodes de décision adaptées à ce problème. Dans le paragraphe suivant, nous résumons les aspects théoriques nécessaires. De plus amples détails sont donnés en appendice B (p. 277). Dans les paragraphes d'après, nous présentons la manière d'utiliser ces nouveaux outils. Ensuite, viennent le protocole expérimental, puis les résultats obtenus.

VI.2.1 Transformée Pignistique Partielle (PPT)

Quand une décision doit être prise dans un environnement incertain, il y a deux manières de se comporter. Il est possible soit d'attendre que toutes les informations nécessaires soient disponibles pour décider, soit de faire un pari

sur la décision la plus raisonnable en fonction des informations déjà à disposition et du risque que le manque d'information implique.

La première méthode correspond au **mode de décision orienté preuve**. Le processus de décision consiste surtout à rassembler toutes les informations possibles et à les combiner. S'il manque des informations pour prendre une décision complète et précise, celle-ci ne sera pas prise. On se contentera donc d'une décision partielle ou incomplète par rapport au problème initial, mais qui sera la plus "déterminée" possible par rapport aux informations disponibles. La prise de décision partielle est un problème complexe dans le formalisme probabiliste car il n'est pas adapté à ce mode de décision. En revanche, le formalisme évidentiel prend naturellement en compte cette incertitude ou cette incomplétude.

Le second mode de décision est le **mode orienté pari**. Comme il est impossible de prendre une décision partielle, l'idée est de prendre la décision complète la plus raisonnable, en faisant le pari qu'elle est la plus probable ; cela revient à dire que statistiquement, prendre toujours cette décision consiste en une stratégie gagnante. Ce mode de décision est particulièrement bien pris en compte dans un modèle probabiliste.

Il y a bien des cas où le pari est la seule alternative en cas de manque d'information. Ainsi, quand un robot autonome doit éviter un obstacle par une décision entre "tourner à gauche" ou "tourner à droite", il est indispensable de prendre une décision (quitte à la remettre en cause plus tard), car rester dans l'indécision et attendre de plus amples informations est le meilleur moyen d'entrer en collision avec l'obstacle. A l'inverse, il y a des cas où il est impossible de raisonner en termes de stratégie gagnante et où la complétude de la décision est moins importante que le risque qu'elle implique. Par exemple, juridiquement, quand le nombre de preuves à charge n'est pas suffisant, il n'est pas possible de maintenir un chef d'accusation, et ce, même si tout le monde est près à prendre un pari sur les éventuelles culpabilités.

D'un côté, il y a le formalisme évidentiel qui permet des prises de décision, mêmes incomplètes, par la preuve, et de l'autre côté, il y a le formalisme probabiliste, orienté sur le pari. Il y a bien le **Modèle de Croyance Transférable** (ou **Transferable Belief Model** – **TBM**), une interprétation particulière des FC qui permet de modéliser un pari dans le formalisme évidentiel au travers de l'utilisation de la Transformée Pignistique (déjà introduite au V.2.3.3 p. 156). Pour autant, aucun modèle ne permet de prendre une décision basée sur un modèle mixte. Un tel modèle fonctionnerait sur le principe suivant : la décision peut être à la fois incomplète et relever du pari. Cela permet d'hésiter entre un certain nombre d'hypothèses pour lesquelles une discrimination plus précise n'est pas possible, tout en effectuant un pari en rejetant un certain nombre d'hypothèses moins crédibles. En déterminant un niveau d'incertitude maximum autorisé, il serait possible d'effectuer un pari afin d'éliminer un certain nombre d'hypothèses parmi les choix possibles, mais néanmoins conserver un certain doute acceptable. Afin que le pari reste un vrai pari, il faut

néanmoins être sûr que la décision ne va pas porter sur un nombre d'hypothèses qui est toujours maximal dans l'intervalle autorisé ; si la décision permet un niveau de certitude plus élevé, celui-ci doit automatiquement être atteint.

Afin de pallier ce besoin, nous proposons d'étendre la PT à de tels types de décision. L'ensemble des justifications est fournie en appendice B (p. 277), et dans [J3]. Ici, nous nous contentons de donner une définition, une illustration sur un exemple, et une implantation algorithmique.

Soit γ un seuil d'incertitude et S^γ l'ensemble de toutes les parties du cadre telles que leur cardinal est compris entre 0 et γ. Cela revient à tronquer le powerset aux éléments dont le cardinal est inférieur à un certain seuil. Nous appelons S^γ le γ^ème **cadre de décision**.

$$S^\gamma = \left\{ A \in 2^\Omega \;\middle|\; |A| \in \lfloor 0, \gamma \rfloor \right\}$$

où $|.|$ est la fonction cardinal. Le résultat $M_\gamma(.)$ de la Transformée Pignistique Partielle d'ordre γ (γ^ème-PPT) de $m(.)$ est défini sur 2^Ω de la manière suivante :

$$M_\gamma(A) = \begin{cases} m(A) & \text{si} \quad A = \varnothing \\ m(A) + \sum \left(\dfrac{m(B) \cdot |A|}{\sum_{k=1}^{\gamma} \binom{|B|}{k} . k} \;\middle|\; \begin{matrix} B \supseteq A \\ B \notin S^\gamma \end{matrix} \right) & \text{si} \quad A \subseteq S^\gamma \\ 0 & \text{sinon} \end{cases}$$

Ensuite, la décision est prise par la simple sélection de l'élément du γ^ème cadre de décision pour lequel $M_\gamma(.)$ est maximum :

$$D^* = \underset{2^\Omega}{\operatorname{argmax}} \left(M_\gamma \right)$$

Afin d'illustrer le comportement de la PPT, analysons les résultats pour différentes valeurs de γ. Soit $\Omega = \{h^A, h^B, h^C, h^D, h^E\}$ et m une fonction de croyance sur Ω. Les différents éléments du powerset sont représentés dans la partie gauche du Tableau VI-2. Chaque ligne représente un élément du powerset au moyen d'un codage binaire des éléments de Ω qu'il contient. m est représentée dans la première colonne de la partie droite du tableau. Cette colonne représente aussi la 5ème-PPT puisque celle-ci est équivalente à la fonction de croyance : $|\Omega| = 5$. Les résultats $M_4 \ldots M_1$ des γ^ème-PPT pour γ allant de 4 à 1 sont donnés dans les colonnes suivantes. Les cellules grisées indiquent le maximum de chaque colonne.

M_4 est en réalité très proche de m. La raison est que seul un élément focal a un cardinal trop grand pour être pris en compte, et donc que seul celui-ci est redistribué. L'élément focal qui rassemble la plus grande croyance est $\{h^A, h^B, h^C\}$. Celui-ci est d'un cardinal strictement inférieur à la valeur de γ. Ainsi, plutôt que de considérer $\{h^A, h^B, h^C\}$, il pourrait être tentant de considérer $\{h^A, h^B, h^C, h^D\}$ ou $\{h^A, h^B, h^C, h^E\}$, puisque eux aussi sont dans le 4ème cadre de décision. Il est en

effet naturel de faire un tel choix puisque à l'évidence, en prenant une décision partielle mais néanmoins autorisée, le risque est diminué. Cependant, nous ne désirons pas d'un tel comportement. En effet, nous souhaitons que dans un tel cas, si l'information disponible permet de faire un choix plus restrictif, que cela se fasse automatiquement. Dès lors, le choix de $\{h^A, h^B, h^C\}$ est celui que nous attendons, et celui-ci est parfaitement naturel dans le sens où il correspond à ce qu'un humain aurait fait, c'est-à-dire effectuer un compromis entre preuve et pari, tout en étant capable d'adapter l'équilibre de ce compromis à la distribution de la croyance. Ainsi, dans le cas de M_3, la décision reste la même que pour M_4.

Tableau VI-2 : exemple du fonctionnement des PPT.

h^A	h^B	h^C	h^D	h^E	m	M_4	M_3	M_2	M_1
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0.1	0.102	0.1027	0.1671	0.3133
0	1	0	0	0	0	0.002	0.0027	0.0782	0.2967
1	1	0	0	0	0	0.004	0.0055	0.1342	0
0	0	1	0	0	0	0.002	0.0027	0.0782	0.2717
1	0	1	0	0	0	0.004	0.0055	0.1342	0
0	1	1	0	0	0.05	0.054	0.0555	0.2064	0
1	1	1	0	0	0.55	0.556	0.5582	0	0
0	0	0	1	0	0	0.002	0.0027	0.006	0.03
1	0	0	1	0	0	0.004	0.0055	0.012	0
0	1	0	1	0	0	0.004	0.0055	0.012	0
1	1	0	1	0	0	0.006	0.0082	0	0
0	0	1	1	0	0	0.004	0.0055	0.012	0
1	0	1	1	0	0	0.006	0.0082	0	0
0	1	1	1	0	0	0.006	0.0082	0	0
1	1	1	1	0	0	0.008	0	0	0
0	0	0	0	1	0	0.002	0.0027	0.0171	0.0883
1	0	0	0	1	0	0.004	0.0055	0.012	0
0	1	0	0	1	0.05	0.054	0.0555	0.0842	0
1	1	0	0	1	0	0.006	0.0082	0	0
0	0	1	0	1	0	0.004	0.0055	0.0342	0
1	0	1	0	1	0	0.006	0.0082	0	0
0	1	1	0	1	0.1	0.106	0.1082	0	0
1	1	1	0	1	0	0.008	0	0	0
0	0	0	1	1	0	0.004	0.0055	0.012	0
1	0	0	1	1	0	0.006	0.0082	0	0
0	1	0	1	1	0	0.006	0.0082	0	0
1	1	0	1	1	0	0.008	0	0	0
0	0	1	1	1	0	0.006	0.0082	0	0
1	0	1	1	1	0	0.008	0	0	0
0	1	1	1	1	0	0.008	0	0	0
1	1	1	1	1	0.15	0	0	0	0

La décision sur le 2ème cadre de décision est plus difficile à prendre. En effet, malgré la forte indécision entre les hypothèses h^A, h^B et h^C, seulement deux parmi les trois peuvent être retenues. Parmi toutes les paires potentielles, nous

proposons de sélectionner $\{h^B, h^C\}$, parce qu'elle rassemble le plus de croyance. Finalement, le cas du premier cadre de décision correspond à la Transformée Pignistique classique, et préconise de choisir h^A. Cela ne va pas être discuté ici, puisqu'il ne s'agit pas de revenir sur le bienfondé de la Transformée Pignistique. Néanmoins, on peut se poser la question de l'adéquation entre la concordance des résultats entre les décisions sur les 1er et 2ème cadres de décision : l'un préconise $\{h^B, h^C\}$, et l'autre h^A, et ces deux décisions ne concordent pas. Cela signifie simplement que pourvu qu'une certaine incertitude soit autorisée dans la décision, il vaut mieux choisir deux hypothèses qui individuellement ne sont pas intéressantes, mais qui vont bien ensemble en tant qu'union d'hypothèses, qu'une seule hypothèse individuellement plus probable mais incompatible avec les autres. Bien que cela semble d'un premier abord discutable, cela ne fait aucun doute si l'on raisonne en termes de stratégie gagnante.

D'un point de vue algorithmique, la PPT est relativement simple à implanter. Il s'agit simplement de distinguer les éléments du cadre de décision, auxquels on va ajouter un héritage de la croyance en des hypothèses plus vastes, à savoir des éléments n'appartenant pas au cadre de décision dont on va annuler la croyance et la redistribuer aux premiers. Il s'agit donc seulement de calculer à chaque fois la part de l'héritage de chacun des éléments du cadre de décision. La seule difficulté est de connaître le schéma d'inclusion des différents éléments focaux. Ceci peut facilement être stocké dans un tableau que l'on appelle SubSetInvolved. Voici l'algorithme C correspondant à la PPT. Fact(.) représente la fonction de calcul d'une factorielle.

```
void PartPigTransfunction(
   double *BeliefFunction,int *SubSetInvolved, unsigned int PowerSet,
   unsigned int NbOfClasses, int gamma, double *BeliefFunctionResult){

int    i,j,k,l,m,tempVal,NbImpliedHyp,Card,CardImplied,isCompletlyIncluded;
double MassToShare;

for(i=0;i<PowerSet;i++){
 Card = 0;
 for(j=0;j<NbOfClasses;j++)Card =+ (int) SubSetInvolved[i*NbOfClasses+j];
 if(Card>gamma){
  // the belief is to be shared
  MassToShare = BeliefFunction[i];
  NbImpliedHyp = 0;
  // Cmpt of the size of the hyp which receive a part
  for(k=1;k<gamma+1;k++){
   tempVal = (Fact(Card-k)*Fact(k-1)));
   NbImpliedHyp =+ (int)(Fact(Card)/ tempVal;}
  // Cmpt of the number of part for the sharing
  for(l=1;l<PowerSet;l++){
   CardImplied = 0;
   for(m=0;m<NbOfClasses;m++)
             CardImplied=+(int)SubSetInvolved[l*NbOfClasses+m];
   isCompletlyIncluded = 0;
   // test for the hyp to receive a part of belief
   for(m=0;m<NbOfClasses;m++){
    if (((int)SubSetInvolved[i*NbOfClasses+m]==1) &&
        ((int)SubSetInvolved[l*NbOfClasses+m] == 1) ){
     isCompletlyIncluded++;}}
   // attribution itself
   if ((CardImplied <= gamma) && (isCompletlyIncluded == CardImplied))
    BeliefFunctionResult[l] =+ MassToShare*CardImplied/NbImpliedHyp;}
  BeliefFunctionResult[i] = 0;
  }else{
   BeliefFunctionResult[i] = BeliefFunctionResult[i]+BeliefFunction[i];}}}
```

VI.2.2 eNTERFACE'06 ASL database

La base de données que nous utilisons est celle définie lors du workshop eNTERFACE'06 [136]. Elle est constituée de 8 signes de base déclinés en un ensemble de 19 signes. La différence entre les diverses déclinaisons d'un signe de base est constituée soit de gestes non-manuels, soit de variations mineures dans la dynamique manuelle du geste. Le contenu de la base de données, en termes de signes de base et de signes dérivés est résumé dans le Tableau VI-3. Pour plus de détails concernant la mise en place et la structure du corpus, cf. [135], [J2], [C1]. Ici, nous nous contentons de mentionner quelles sont les informations qui permettent de distinguer les déclinaisons d'un même signe de base :

Tableau VI-3 : descriptions des signes Les 19 signes regroupés en 8 groupes de signes de base.

Les 19 signes et leurs 8 groupes de signes de bases	Description de la composante non-manuelle	Description de la composante manuelle
Clean	-	La paume de la main droite vient balayer d'un coup la paume de la main gauche, qui est placée horizontalement paume vers le ciel
Very Clean	Lèvres serrées. La tête fait un petit quart de tour sec	
Afraid	-	Les mains partent des côtés pour se rejoindre au milieu
Very Afraid	Bouche ouverte et yeux écarquillés	
Fast	-	Les mains placées devant la poitrine sont ramenées avec les doigts fermés à l'exception des pouces
Very Fast	Bouche ouverte et yeux écarquillés	
To drink	Hochement de tête	Mime de l'action de boire
Drink (le nom)	-	Mime répétitif de l'action de boire
To open	-	Mime d'ouverture de porte
Door	-	Mime répétitif d'ouverture de porte
Here	Hochement de tête	Mouvements circulaires et parallèle au sol de la main droite
Is here?	Sourcils levés et avancée de la tête	
not here	Négation de la tête	
Study	-	La main droite fait l'action de pianoter légèrement au dessus de la paume de la main gauche placée à l'horizontal
Study continuously	La tête fait un mouvement circulaire	Signe similaire à "Study" mais le mouvement des doigts est limité. A la place, un mouvement de va et vient vers la paume de la main gauche
Study regularly	La tête fait un mouvement d'aller-retour	
Look at	-	Les mains partent des yeux vers l'avant avec les doigts en "V"
Look at continuously	La tête suit le mouvement des mains	Similaire à "Look at" mais c'est un mouvement d'aller-retour
Look at regularly		Similaire à "Look at" mais c'est un mouvement d'aller-retour

– "**Clean**"/"**Very clean**", "**Fast**"/"**Very fast**" et "**Afraid**"/"**Very afraid**", sont différentiables seulement à partir d'informations non manuelles. L'emphase de "very" ne se signe qu'avec un mouvement du visage et un changement d'expression faciale.

– "**Here**"/"**Is here?**"/"**Not here**" sont différentiables seulement à partir d'informations non-manuelles.

– "**Door**"/"**To open**" ne se différencient qu'au niveau du geste manuel, mais la différence est relativement légère, et il y a peu de chance que ces deux gestes soient séparables.

– "**Drink**"/"**To drink**" et "**Look at**"/"**Look at regularly**"/"**Look at continuously**" peuvent se distinguer à la fois par les gestes manuels et par les gestes non-manuels. Cependant, la main étant devant le visage, les mouvements de têtes (non manuels) sont occultés.

– "**Study continuously**"/"**Study regularly**" ne sont différentiables que par la composante non manuelle, alors que "**Study**" est différentiable des deux précédents par la composante manuelle.

VI.2.3 Application de la PPT à la prise de décision

A terme, nous avons pour objectif de réaliser la fusion multimodale des gestes de l'ASL [J2], [J1], [C1]. Cependant, nous voulons d'abord tester la PPT pour la prise de décision. Il s'agit donc d'utiliser le formalisme crédal, et de convertir la connaissance en une FC par application d'un des outils que nous avons développés et décrits dans la section V.3 (p. 176) : Combinaison Evidentielle de classifieurs binaire non crédaux, Combinaison Evidentielle de classifieurs unaires ou Transformées Crédales. Ensuite, il est possible d'appliquer la PPT et de prendre une décision.

Pour réaliser cela, nous considérons le problème de la reconnaissance de gestes monomodaux de l'ASL. Notons que nous ne nous intéressons qu'à la reconnaissance proprement dite, à partir d'informations issues de traitements vidéo. Nous n'abordons pas ces traitements, mais nous renvoyons le lecteur intéressé à [J2], [C1]. Ensuite, nous procédons de manière classique : chaque signe est modélisé par un modèle génératif (un HMM) qui peut fournir un score de vraisemblance.

Pour ce qui est de la classification, le choix du **maximum de vraisemblance** (ou **ML** pour **Maximum Likelihood**) ne nous convient pas : il fournit toujours une décision, même dans les cas incertains, où par exemple plusieurs modèles ont un pouvoir explicatif à peu près équivalent. Ainsi, des signes ne se différenciant que par des gestes non manuels peu différents, pour lesquels la discrimination est subtile, risquent d'être source d'erreur. Dans de tels cas, il est préférable de ne pas prendre une décision complète, par exemple en précisant le geste manuel, mais en s'autorisant plusieurs possibilités sur la variation du geste non manuel.

Ainsi, au prix d'une certaine indécision, il peut être possible de diminuer le nombre d'erreurs de reconnaissance. Par la suite, ces indécisions peuvent être levées soit par le contexte, soit par un second niveau de classification.

Nous proposons de considérer le tableau de scores associés aux calculs des vraisemblances pour un banc de HMM. Ensuite, il s'agit de convertir ce tableau de scores en une FC grâce à une des méthodes présentées à la section V.3 (p. 176). Comme les classifieurs utilisés sont des modèles génératifs, nous utilisons la méthode **Combinaison Evidentielle de classifieurs unaires** (paragraphe V.3.2 p. 177). Ensuite, il s'agit de déterminer un seuil d'incertitude et d'appliquer la PPT correspondante sur la FC. Puis, l'élément du cadre de décision qui possède le plus haut score est déterminé. Celui-ci correspond à un HMM, qui lui-même est génératif d'un modèle de signe. Le signe inconnu est alors simplement reconnu comme étant ce signe.

L'intérêt de la PPT est de permettre des comportements adaptés à des situations où il n'est pas possible de prendre une décision complète. Pour créer une telle situation, nous utilisons les signes multimodaux de base eNTERFACE'06 [136], en considérant les différentes modalités en un seul vecteur, ce qui revient à considérer que les gestes ne contiennent qu'une modalité ne permettant pas de faire une distinction complète entre les signes. En effet, les composantes non-manuelles sont sous-représentées, de sorte qu'elles ne permettent pas une distinction complète entre les déclinaisons d'un signe de base.

Afin de vérifier le bon fonctionnement de la PPT, nous comparons la méthode décrite à une décision classique par ML. Il s'agit alors de comparer les résultats que donnent les deux processus de décision.

Un seul banc de 19 HMM correspondant aux 19 signes de la base eNTERFACE'06 est entraîné (cf. [J2] pour le détail de l'apprentissage), et un seul corpus de test (de 228 éléments) est utilisé afin de fournir un seul tableau de scores de vraisemblance. Ce dernier est utilisé pour tester les deux méthodes de classification.

La première consiste en l'utilisation du maximum de vraisemblance. Cette stratégie est bien sur équivalente à l'utilisation de la transformée crédale et de la Transformée Pignistique (ou la 1ère-PPT). Cela permet d'obtenir un taux de bonne classification de 75.88% sur le corpus de test.

La seconde consiste en l'utilisation de la transformée crédale et de la PPT. Le paramètre d'incertitude est réglé sur 2 ou 3 ; il apparaît en effet que les applications de la 2ème-PPT et de la 3ème-PPT fournissent des résultats équivalents. Sur les 228 items du corpus de test, il y en a 189 pour lesquels une décision complète est possible. Sur ceux-ci, il y a 79.4% de bonne classification. Cela signifie que la PPT reste dans le doute dans des cas correspondant plutôt à des cas facteurs d'erreurs, et qu'au contraire, elle permet de prendre une décision complète plutôt sur les éléments ne prêtant pas à confusion. Cela justifie donc l'hypothèse de départ.

Pour ce qui est des cas indécis, le degré d'incertitude varie en fonction du paramètre γ. En revanche, quelque soit sa valeur, l'incertitude ne monte jamais au-delà d'un ordre 3 (pour seulement 4 signes sur 228). Pour les autres, elle reste d'ordre 2. Dans le cas où les réponses incertaines contenant la bonne réponse sont considérées comme justes, le taux de bonne reconnaissance est de 82.08%. Tout cela est résumé dans le Tableau VI-4.

Tableau VI-4 : évaluation de l'intérêt de la PPT.

Stratégie	Classification effectuée sur[7]	Résultat sur corpus non rejeté
Maximum de vraisemblance	100%	75.9%
PPT avec décision complète	82.9%	79.4%
PPT (γ=2 ou 3)	100%	82.08%

VI.2.4 Classification Multimodale à deux étages

A ce niveau, nous avons présenté un système permettant de donner des décisions partielles et par là même nous avons justifié expérimentalement la transformation crédale et la PPT ; en aucun cas, nous n'avons mis en place un système de fusion multimodale. Pour ce faire, nous proposons de compléter le système du paragraphe précédent de la manière suivante :

– Nous faisons l'hypothèse qu'il est possible de focaliser l'indécision au sein d'un groupe de signes dérivant du même signe de base, en paramétrant correctement γ.

– Nous levons l'indécision du processus que nous venons de décrire au paragraphe précédent en ajoutant un second niveau de classification.

Ainsi, nous proposons d'effectuer la fusion multimodale par la méthode suivante :

Etape 1 - Définition automatique de clusters correspondant à des groupes de gestes au sein desquels il peut y avoir facilement des confusions.

Etape 2 - Application d'une première étape de classification. Cette première classification correspond à ce que nous venons de décrire : conversion des vraisemblances des HMM en une fonction de croyance et application de la PPT, avec un paramètre γ adapté permettant de focaliser l'indécision au sein d'un signe de base. Si la décision fournie ne prête pas à hésitation, alors le processus est terminé. Sinon, on passe à l'étape 3.

[7] Dans la mesure où il est possible de ne pas se prononcer dans tous les cas (il est autorisé de ne pas prendre de décision), il est nécessaire de préciser sur quel proportion du corpus une décision est prise.

Etape 3 - Trouver le cluster contenant les signes entre lesquels il y a hésitation et procéder à une seconde étape de classification restreinte à ce cluster. Comme en pratique les clusters sont très proches des signes de base (mais pas identiques), et que les informations non-manuelles ont tendance à être masquées par la prépondérance des informations manuelles, la seconde étape de classification ne se base que sur des HMM modélisant les informations non-manuelles (à l'exclusion des informations manuelles). Comme cette seconde classification doit forcément fournir un résultat, celui-ci est donné par un maximum de vraisemblance.

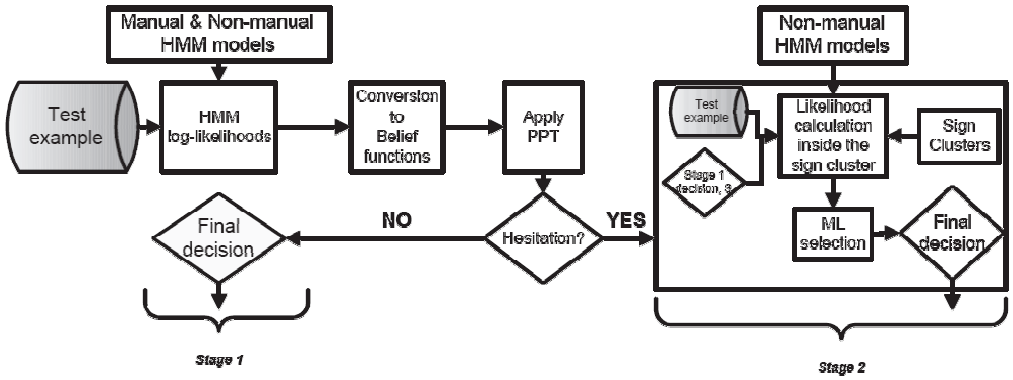


Figure VI-11 : diagramme de fonctionnement du système de fusion multimodale, issu de [J2].

Finalement, ce système est basé sur deux bancs de HMM, l'un utilisant la totalité des informations manuelles et non-manuelles ($HMM_{M\&N}$) et l'autre exclusivement les informations non-manuelles (HMM_N). L'ensemble de l'architecture est représenté sur la Figure VI-11 (cf. [J2], [C1] pour le détail des apprentissages). Il ne reste que deux choses à préciser : le niveau d'incertitude de la PPT lors du premier étage de la classification, et le processus permettant de définir les clusters[8].

Le niveau d'incertitude γ de la PPT est déterminé en fonction de la taille des clusters. Nous préconisons que γ soit du même ordre de grandeur que les clusters. Cela peut se calculer automatiquement, par exemple en prenant l'entier le plus proche de la moyenne de la taille des clusters.

Nous proposons de définir les clusters au moyen d'un corpus de validation (ou en simulant un tel corpus par des validations croisées). Pour cela, nous appliquons le premier étage de classification, et nous ne conservons que les réponses qui sont hésitantes afin d'étudier les signes à rassembler. Ainsi, toutes les réponses uniques, qu'elles soient justes ou fausses ne sont pas considérées.

[8] Le terme anglais de "cluster" désigne dans la terminologie de la classification, soit un groupement de classes, soit une classe déterminée de manière non-supervisée. Ce mot n'ayant pas d'équivalent français, il est couramment utilisé sans être traduit. Pour faciliter la compréhension de la lecture, nous avons donc gardé ce terme.

En effet, les réponses hésitantes reflètent particulièrement bien les types de confusions potentielles. Ainsi, que la reconnaissance soit juste ou non, cela n'a pas d'importance : si un signe *i* est reconnu comme un signe *i* ou au contraire si une erreur apparaît et qu'il est un signe *j*, cela n'apporte pas d'information fiable sur le type d'hésitation. En revanche quand un signe *i* est reconnu comme soit un signe *i*, soit un signe *j*, et même quand un signe *k* est reconnu comme soit un signe *i*, soit un signe *j*, cela est beaucoup plus informatif sur la confusion qu'il peut y avoir entre les signes *i* et *j*. De plus, cela a comme principal avantage de ne pas avoir besoin d'étiqueter le corpus de validation. Toutes les informations de confusion sont ensuite résumées dans une matrice, que l'on rend réflexive et pour laquelle on ferme la transitivité[9]. Afin de faire apparaître des hésitations lors de la prise de décision permettant de définir les clusters, nous utilisons la PPT. Pour celle-ci, le seuil d'incertitude n'a que très peu d'importance, puisque l'on garantit la complétude des clusters par la fermeture transitive et réflexive de la matrice de confusion. En pratique, une valeur de 3 permet de faire apparaître jusqu'à deux dérivations transitives, ce qui garantit d'agrandir suffisamment l'exploration de l'espace des signes à chaque itération de la fermeture de la matrice. Une valeur de 3 pour le paramètre d'incertitude est donc suffisamment faible pour ne pas perdre en généralité tout en permettant de créer des clusters de taille arbitraire.

VI.2.5 Evaluation de la qualité des clusters

Afin d'évaluer la qualité des clusters ainsi obtenus, nous les comparons avec une méthode classique de définition des clusters, avec pour vérité terrain, les clusters associés aux signes de base.

La manière classique de définir ce genre de clusters est d'utiliser un ensemble de validation étiqueté dont on peut obtenir après classification la matrice de confusion. Les clusters se déduisent alors des confusions possibles qui apparaissent dans la matrice de confusion.

La Figure VI-12 montre les deux types de clusters que l'on obtient avec ces deux méthodes. Il ressort que la méthode que nous proposons a deux avantages :

– Tout d'abord, cette méthode est non supervisée, puisqu'elle permet de définir les clusters sur une base de données non étiquetées.

– De plus, elle est beaucoup plus robuste. Ainsi le cluster de "look at continuously" est constitué des signes "look at continuously", look at regularly",

[9] Nous empruntons ici la terminologie de la théorie des langages, et plus particulièrement des langages dotés d'une grammaire hors-contexte, que l'on peut explorer à l'aide d'une machine de Turing. En pratique, il s'agit simplement de définir les opérations sur la matrice d'hésitation qui permettent de faire apparaître une régle d'équivalence à partir de la notion d'hésitation. Les clusters sont simplement les classes d'équivalences ainsi définies. Du point de vue matriciel, il faut faire appaître que si la reconnaissance de I est source de confusion entre I et J, et que la reconnaissance de J entraine de la confusion sur J et K, alors, la reconnaissance de K implique de la confusion sur I, J et K.

"fast" et "very fast". Un tel cluster n'est pas très intéressant. En effet, les signes du signe de base "look at..." et du signe de base "fast" n'ont aucune raison d'être confondus. Cependant, comme pour une raison inconnue (comme une erreur de prétraitement par exemple), une telle erreur est survenue une fois dans le corpus de validation, il est considéré à tord que ces deux signes sont proches. En revanche, une telle erreur n'est pas considérée dans la méthode que nous proposons, puisque la vérité terrain n'est pas connue. Ainsi, l'erreur apparue dans le corpus de validation n'a pas la même conséquence : si cette erreur est réalisée avec certitude (le mauvais "fast" ou "very fast" est annoncé à la place de "look at continuously" sans aucune hésitation), celle-ci ne participe pas à l'élaboration des clusters. En revanche, si celle-ci est annoncée avec hésitation, cette hésitation a beaucoup plus de chance d'avoir lieu entre deux signes proches (par exemple "fast" et "very fast") et par conséquent, elle va venir renforcer le cluster en question, sans perturber celui contenant la vérité terrain. C'est pourquoi, l'ensemble des clusters que nous trouvons est inclus dans les signes de base. Malgré tout, ils ne sont pas rigoureusement égaux. Ainsi les clusters des signes déclinés des signes de base "look at ..." et "study" sont plus restrictifs. La raison est simplement que ces différentes déclinaisons comportent des variations au niveau manuel, et ainsi, le premier banc de HMM est suffisant pour prendre une décision. Ainsi les clusters que nous proposons automatiquement pour le signe de base "Study" sont beaucoup plus cohérents que ceux correspondant aux signes de base eux-mêmes (cf. p. 206).

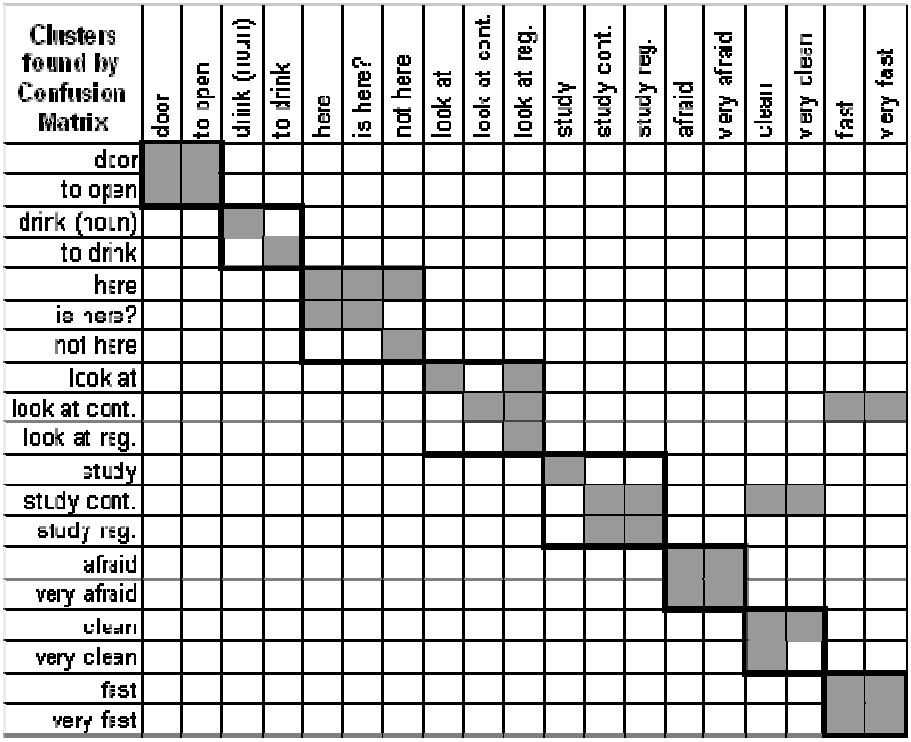
Finalement, nous pensons qu'il est même plus intéressant d'utiliser les clusters définis automatiquement par la méthode que nous proposons, plutôt que d'utiliser ceux dérivés de la définition des signes de bases. Il y a 2 raisons à cela :

– Les signes de base sont définis par un opérateur en fonction d'une interprétation qui ne peut s'affranchir totalement du sens syntaxique que l'on retrouve derrière les signes. Celle-ci peut éventuellement biaiser les parentés à des niveaux d'observation plus bas. C'est pour cette raison que l'on a commencé par grouper les signes dérivés du signe de base "study" (ils ont des significations proches), alors qu'il n'y a aucune raison pour le faire (les informations manuelles sont suffisantes).

– La principale raison est cependant la possibilité d'étendre le vocabulaire : si les clusters sont définis de manière automatique plutôt que manuellement, il est possible de rajouter de nouveaux signes au moyen d'un ensemble de séquences d'apprentissage et de relancer le processus de définition des clusters à souhait, sans l'intervention d'un opérateur. Ainsi, la méthode est beaucoup plus générale.

En conclusion, nous pensons que cette nouvelle manière de définir des clusters à partir des résultats de classification sur un corpus de validation non étiqueté est plus robuste et permet une meilleure automatisation du processus.

(a)



(b)

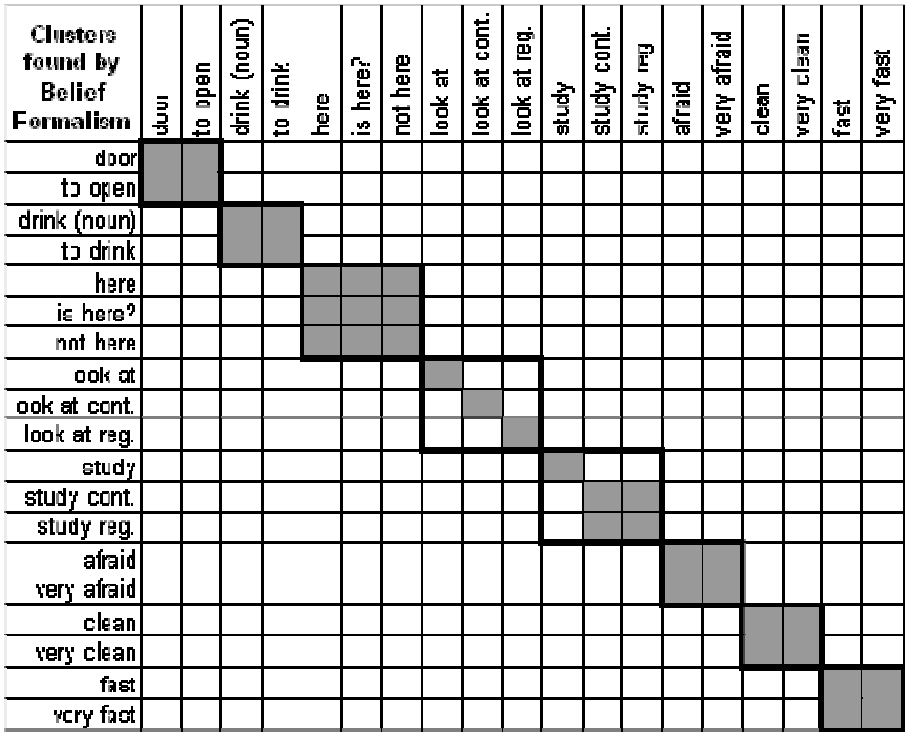


Figure VI-12 : clusters définis par (a) la méthode classique, et par (b) la méthode proposée. Pour chaque signe, le cluster est constitué de l'ensemble des cases grisées sur la ligne. Les cases entourées en gras représentent les signes de base.

VI.2.6 Evaluation de la reconnaissance globale

Afin d'évaluer notre méthode de fusion de classification de signes multimodaux de l'ASL, nous procédons à plusieurs expériences :

– L'utilisation de $HMM_{M\&N}$ seuls en une seule étape de classification. Il s'agit donc de la même méthode que celle qui a servi de témoin pour la mise en place du premier étage seul au IV.2.3. Le taux de reconnaissance est donc toujours de 75.9%.

– La sommation des vraisemblances de $HMM_{M\&N}$ et HMM_N afin de procéder à la fusion des deux informations avant une unique prise de décision. Le taux de reconnaissance atteint 78.1%.

– Une reconnaissance faite de deux étages de classification par le maximum de vraisemblance sur $HMM_{M\&N}$ puis HMM_N ; le second étage n'est utilisé qu'au sein du cluster contenant le signe reconnu au premier étage de classification. Les clusters sont définis par les signes de base. Le taux de reconnaissance n'est que de 73.7%. La raison d'un si faible score est la suivante : il y a de nombreux cas où le premier étage donne un résultat correct, mais comme le second étage est utilisé systématiquement (à défaut d'avoir un critère permettant de ne pas le faire), il arrive que celui-ci remette en cause des résultats justes (les informations manuelles ne sont plus utilisées).

– Même méthode que précédemment, mais les clusters sont définis par la méthode classique d'étude des confusions. Le taux de reconnaissance n'est là aussi que de 75%, et la raison est la même que pour l'expérience précédente.

– Enfin, la méthode que nous proposons (avec l'utilisation de la PPT et un second étage de classification optionnel) avec des clusters définis à partir des hésitations plutôt que des confusions. Le taux de reconnaissance est de 81.6%.

Tableau VI-5 : évaluation de la reconnaissance globale.

Stratégie	Commentaire	Résultat
$HMM_{M\&N}$	Une seule classification	75.9%
$HMM_{M\&N}$ + HMM_N	Une seule classification Sommation des vraisemblances	78.1%
$HMM_{M\&N} \rightarrow HMM_N$	Classification séquentielle Clusters = signes de base	73.7%
$HMM_{M\&N} \rightarrow HMM_N$	Classification séquentielle Clusters définis sur matrice de confusion	75%
$HMM_{M\&N} \rightarrow HMM_N$	**Classification séquentielle optionnelle Clusters définis sur matrice d'hésitation**	**81.6%**

L'ensemble de ces résultats (résumés dans le Tableau VI-5) prouve clairement d'un point de vue expérimental que la PPT est un outil puissant et utile. Il nous permet de définir des clusters avec plus de précision, et d'effectuer des décisions de type crédal, plus adaptées à des cadres où l'information est relativement peu fiable. Enfin, ces expériences confirment et illustrent la transformation d'un tableau de scores en une fonction de croyance dont nous avons développé le concept dans la section V.3 (p. 176).

VI.3 Conclusion du chapitre

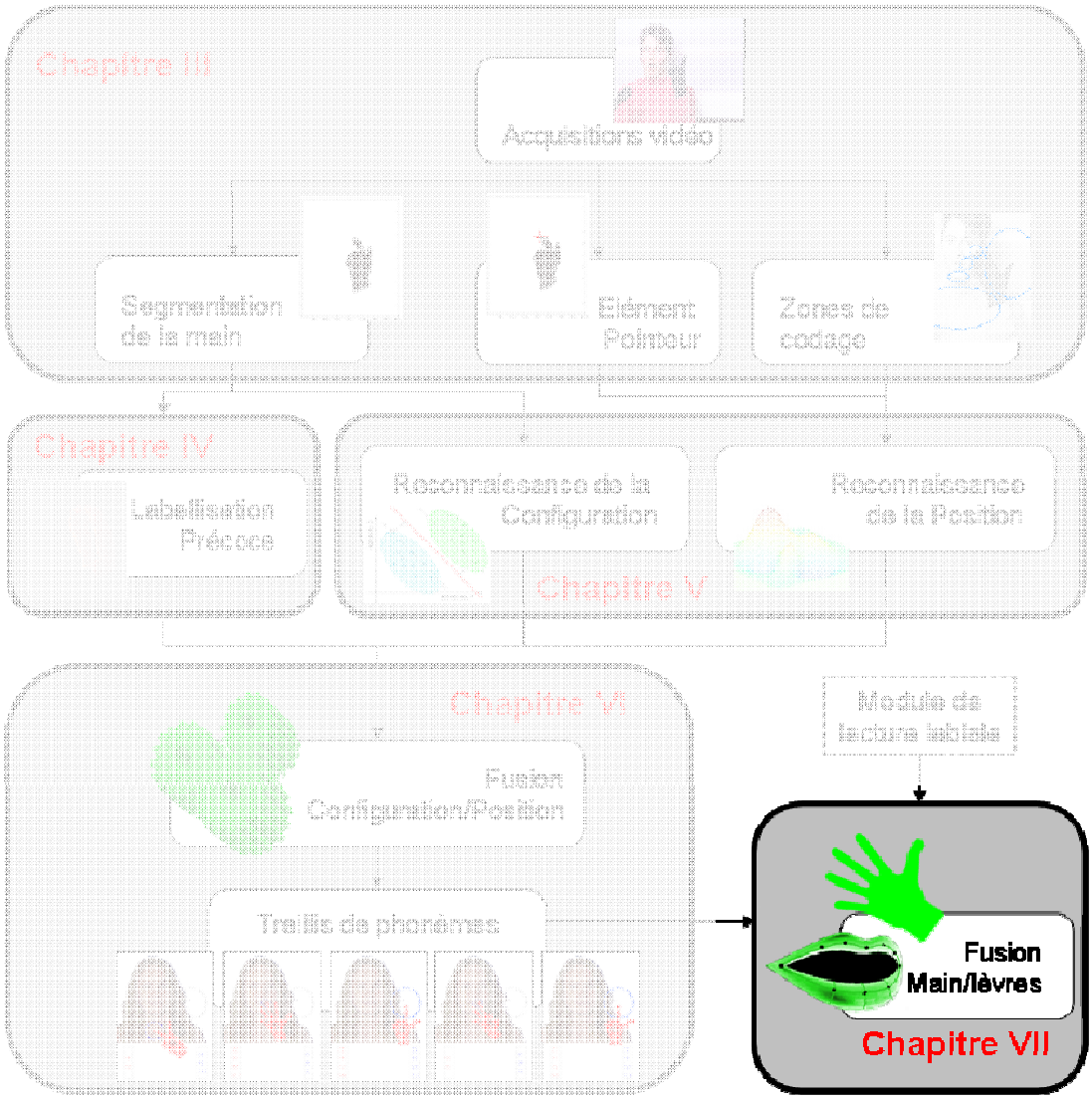
L'objectif de ce chapitre est la fusion temporelle et multimodale des flux de Configurations et de Positions du LPC. Nous pensons qu'il s'agit d'un problème complexe et que cette complexité est similaire à celle du cas de la fusion main/lèvre. Par conséquent, nous proposons de traiter ces deux cas de fusion en une seule étape, et donc de la repousser momentanément. En revanche, nous proposons l'étude de deux cas simplifiés dans lesquels seule une des deux difficultés de la fusion est traitée : soit l'intégration temporelle, soit la combinaison multimodale. Cela permet d'obtenir un interpréteur complet de geste, et de valider l'ensemble de nos travaux.

Le cas de l'intégration temporelle est illustré sur des vidéos de LPC : l'interprétation en termes de gestes est suffisante pour valider de manière globale l'ensemble de ces travaux. De plus, cela valide les hypothèses que nous avons émises à propos de la nature du geste LPC ou de nos algorithmes. Enfin, les résultats sont suffisants pour des applications où la reconnaissance de gestes n'a pas besoin d'être aussi précise que pour le LPC. En revanche, pour l'interprétation phonémique du LPC proprement dite, les erreurs sont encore trop nombreuses. En s'inspirant de la manière dont les problèmes de couplage en recherche opérationnelle sont résolus, il devrait être possible d'améliorer ces résultats. Il apparaît cependant comme nécessaire de mettre en place des outils de fusion traitant à la fois la désynchronisation et la multimodalité.

Le cas de la combinaison multimodale est illustré sur des séquences d'ASL. C'est à cette occasion que la PPT est élaborée et testée. C'est aussi l'occasion de tester l'intérêt de la Combinaison Evidentielle de classifieurs binaires non crédaux, présentées au chapitre V, mais non évaluées à ce moment-là. Notons que dans le cas de l'ASL, il s'agit des premiers travaux traitant à la fois de la reconnaissance de signes manuels, de signes non-manuels et de leur fusion.

Le chapitre suivant propose des pistes d'exploration concernant la suite de ce projet, à savoir la fusion Position/Configuration/lèvres. Les pistes que nous proposons sont bien sûr dans la continuité de celles proposées dans ce chapitre. C'est l'occasion d'appliquer les méthodes et les résultats de la combinaison multimodale des signes de l'ASL au cas du LPC. Tout cela constitue de nombreuses pistes à explorer pour le problème encore ouvert de la combinaison multimodale.

CHAPITRE VII
VERS LA FUSION MAIN/LEVRES



Le décodage du LPC implique la reconnaissance du mouvement manuel (Configuration et Position de chaque geste) mais aussi la lecture du mouvement labial du codeur. Dans ce document, nous nous sommes concentrés sur le geste manuel. En parallèle, divers travaux ont été menés sur le mouvement labial au sein du projet TELMA. La suite logique consiste en l'utilisation conjointe de nos travaux et de ceux concernant la lecture labiale, dans le but d'un décodage complet du LPC. Dans ce dernier chapitre, nous proposons des pistes pour atteindre cet objectif. Pour cela, nous nous basons sur les travaux réalisés dans le cadre de TELMA pour la reconnaissance des flux de Configurations, de Positions et de formes labiales, puis nous proposons un schéma de fusion de ces informations. Les difficultés liées à l'étude de la **multimodalité** (Position, Configuration, forme labiale) sont traitées de manière similaire à ce que nous avons fait dans le cas de l'ASL, alors que celles liées aux problèmes de désynchronisation/décalage sont traitées par une méthode d'estimation de ces désynchronisations/décalages par rapport au code théorique (défini à la section II.4, p. 48).

Dans une première section, nous présentons tour à tour les différents travaux ayant trait à l'étude du mouvement labial réalisés par d'autres équipes du projet TELMA. Ensuite, nous passons en revue les différentes difficultés à surmonter : nous insistons sur les difficultés de l'étude du mouvement labial, puisque celles intrinsèques au geste manuel ont déjà été longuement discutées dans ce document. Enfin, dans la dernière section, nous émettons en hypothèse les grandes lignes d'une méthode de décodage du geste complet.

VII.1 Le mouvement labial dans TELMA

De nombreux travaux ont déjà été menés sur l'analyse du mouvement des lèvres, au sein ou en marge du projet TELMA. Tout d'abord, l'expertise du DCP et du DIS de GIPSA-Lab est déterminante dans le domaine, avec les travaux de Virginie Attina [7], de Guillaume Gibert [8], dans le contexte de l'étude du LPC, et de Nicolas Eveno [10] et de Pierre Gacon [9] de manière plus générale. Plus en lien avec TELMA, mais néanmoins directement basés sur ces premiers travaux, il y a eu ensuite les travaux de DEA de Thomas Burger [C6], [A1], ainsi que les deux travaux de thèse (en cours) de Noureddine Aboutabit[10] [2], [3], [4], [5], [6], et de Sébastien Stillittano [13]. Nous présentons ici les principaux résultats actuels en lien avec (1) les études sur le synchronisme main/lèvres, (2) la reconnaissance des mouvements labiaux, et (3) la segmentation des contours des lèvres.

VII.1.1 Synchronisme main/lèvres

Dans cette section, nous reprenons les principaux résultats de travaux menés au GIPSA-Lab/DPC et publiés dans [7] et [3]. Ils traitent de la coordination entre

[10] La thèse de Noureddine Aboutabit était en phase terminale au moment de l'écriture de ce rapport, de sorte que celle-ci est peut-être maintenant disponible.

le mouvement de la main, le mouvement des lèvres, et le son de la parole orale. Ils s'intéressent à deux cas : (1) le cas de production de code par des codeurs LPC professionnels, et (2) le cas de la réception et du décodage du message par des enfants ou jeunes adultes sourds. Ces études sont donc davantage axées sur un plan cognitif/linguistique que sur un plan algorithmique. Elles sont cependant indispensables à une compréhension réelle de la manière dont le code théorique s'est vu approprié par la population des utilisateurs.

De ces travaux, il ressort principalement un schéma temporel des syllabes CV (Consonne-Voyelle) assez structuré entre les 3 flux en question (main dans son ensemble, lèvre et son de parole). Les instants suivants sont définis :

– **M1** : début du mouvement transitoire du geste de la main précédent vers la cible manuelle[11] suivante ;

– **M2** : fin de la transition et atteinte de la Configuration et de la Position (atteinte de la cible manuelle) ;

– **M3** : fin de tenue de la cible manuelle et début du mouvement transitoire vers le geste suivant (qui correspond donc au M1 du geste précédent) ;

– **A1** : début de la réalisation acoustique de la consonne de la syllabe CV ;

– **L2** : cible aux lèvres correspondant à la pleine réalisation de la voyelle de la syllabe CV.

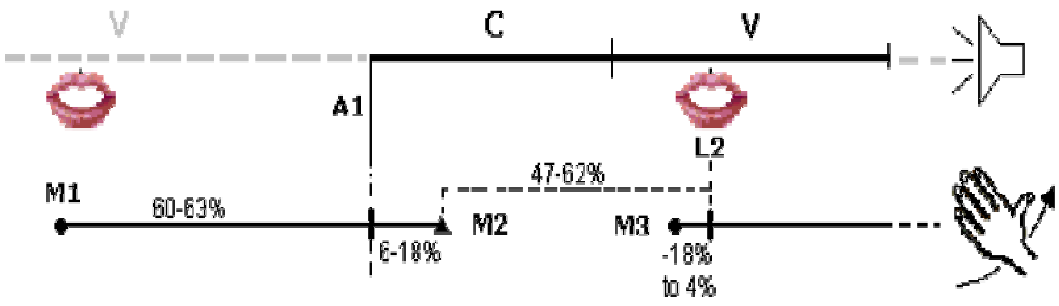


Figure VII-1: schéma général de l'ordonnancement du LPC, issus de [7].

Le schéma d'ordonnancement est proposé en fonction de ce qui a été observé, aussi bien en termes de production, qu'en termes de confort de décodage (cf. Figure VII-1 tirée de [7]). Les pourcentages associés aux plages de temps indiquent un intervalle de confiance pour la durée de chacune d'elles en proportion du temps total de la réalisation de la syllabe CV.

[11] En opposition avec ce que nous proposons, ces travaux ne considèrent qu'un seul type de cible pour le mouvement complet de la main, et ne font pas de distinction entre le mouvement de changement de Positions et celui de Configurations.

La première observation de l'auteur est que **la main est en avance sur les lèvres**, et que visiblement, ce sont les lèvres qui désambiguïsent le mouvement de la main plutôt que le contraire.

Ensuite, la variabilité des temps d'enchaînement est beaucoup trop importante pour pouvoir être exploitée facilement dans un système de reconnaissance automatisée du LPC[12]. Malgré tout, ce schéma d'ordonnancement contient de l'information qui peut être utilisée afin de contraindre les paramètres d'algorithmes de synchronisation. Nous proposons un tel algorithme dans la troisième section.

VII.1.2 Reconnaissance de trajectoires labiales

A l'heure actuelle, les méthodes mises au point au DPC permettent de reconnaître les mouvements labiaux associés à l'articulation de syllabes CV en contexte prosodique mono-codeur. Cet algorithme est fondé sur l'utilisation d'un banc de HMM au sein duquel chaque HMM représente le mouvement labial d'un type de syllabe CV. Son objectif principal est de définir des modèles adaptés pour la reconnaissance des formes labiales. Il est donc essentiel que les données recueillies pour ce faire soient les plus précises possibles. Dans cette optique, et afin d'éviter l'accumulation d'erreurs, des artifices ont été utilisés lors de l'acquisition des vidéos dédiées à l'apprentissage des mouvements labiaux. Par ailleurs le message audio a aussi été enregistré. Ainsi, avant de détailler le fonctionnement de cet algorithme, nous présentons un résumé des prétraitements nécessaires devant être appliqués en amont de la reconnaissance [6] :

– **Segmentation des lèvres** : les vidéos traitées représentent un codeur dont les lèvres sont maquillées en bleu saturé, et dont la tête est maintenue attachée afin d'éviter toute invariance par similitude dans l'image (cf. Figure VII-2). Cela permet de segmenter de manière précise et rapide les contours labiaux internes et externes, tout en s'affranchissant des défauts d'une segmentation en condition naturelle. Comme la couleur "bleu saturée" n'est pas présente dans

[12] Ceci est illustré par [5] qui propose un algorithme de synchronisation naïf basé sur ce décalage. Bien que ce modèle linguistique soit à la fois innovant et pertinent du point de vue explicatif dans le contexte de la linguistique, il n'est pas conçu dans l'optique d'avoir une forte capacité de généralisation : le système permet 77.6% de reconnaissance sur des voyelles (main et lèvres) à partir d'un système de capture avec artifices et dont les données sont prétraitées et synchronisées par un système expert. Comme un seul codeur est considéré, le système expert est paramétré et testé sur des données différentes, mais issues de la même personne, et dans les mêmes conditions. Il en est de même pour le système de reconnaissance. Ainsi, le pouvoir de généralisation des résultats n'est pas encore démontré. C'est cela, ainsi que le schéma d'ordonnancement d'Attina qui nous amènent à considérer que la synchronisation de modalités impliquées dans le LPC ne peut être reconstituée simplement et directement. Encore une fois, cette limite du pouvoir de généralisation des travaux de [5] ne doit pas être perçue comme une remise en cause la pertinence du modèle linguistique, ni de la rigueur expérimentale et scientifique de la démarche : c'est la mise au point de tels modèles linguistiques qui permet par la suite l'élaboration de méthodes calculatoirement plus complexes telles que celle que nous envisageons dans ce chapitre.

l'image à exception des artifices d'acquisition, la segmentation du contour des lèvres est réalisée automatiquement par seuillage.

– **Segmentation des syllabes** : conjointement à l'enregistrement vidéo, l'enregistrement audio du corpus est réalisé. En effet, la codeuse est une professionnelle bien-entendante capable d'oraliser parfaitement. Comme à l'heure actuelle, il est difficile de déterminer automatiquement le début et la fin des syllabes à partir du mouvement labial, l'enregistrement audio du corpus est utilisé pour labelliser les phonèmes. Cela permet de déterminer les vérités terrains pour les apprentissages et les tests, mais aussi de segmenter chaque syllabe pour ensuite les reconnaître individuellement.

Une fois ces deux pré-traitements réalisés, il est possible d'appliquer le processus de reconnaissance proprement dit à chacune des trajectoires labiales. Voici un résumé de ce processus :

Figure VII-2 : conditions d'acquisition pour la reconnaissance labiale. Issus de [6].

– Les **classes** sont les syllabes CV du français. Cependant, comme seule la forme labiale est considérée et que la discrimination des sosies labiaux n'a pas d'importance (c'est l'information manuelle qui permet la distinction), les phonèmes composant les syllabes sont regroupés en 3 clusters pour les voyelles (hautes et mi-hautes arrondies : /ɔ̃/, /y/, /o/, /ø/, /u/ ; basses et mi-basses arrondies : /a/,/ɛ̃/, /i/, /œ/, /e/, /ɛ/ ; et non arrondies : /ã/, /ɔ/, /œ/) et en 5 clusters pour les consonnes (bilabiales : /p/, /b/, /m/ ; labiodentales : /f/, /v/ ; dentales : /t/, /d/, /s/, /z/, /n/, /l/ ; palatales /ʃ/, /ʒ/, /ɲ/ ; et vélaires : /k/, /g/, /R/). Cependant, en raison de la structure du LPC et de la proximité de certaines formes labiales, les clusters des consonnes issus de la littérature linguistique sont légèrement modifiés. Dans, [6] il est conseillé d'utiliser [/p/, /b/, /m/], [/f/, /v/], [/t/, /d/, /s/, /z/, /n/, / ɲ /], [/ʃ/, /ʒ/] et [/k/, /g/, /R/, /l/] (cf. à la Figure I-2 pour l'alphabet phonétique).

– Les **attributs** sont définis comme suit : à partir du contour des lèvres, les mesures des ouvertures labiales interne et externe (ou apertures, désignées

respectivement par **A** et **A'**), des étirements labiaux interne et externe (désignés par **B** et **B'**), des surfaces interolabiales internes et externes (désignées par **S** et **S'**), et des épaisseurs labiales supérieure et inférieure (désignées par **Bsup** et **Binf**) sont récupérées (cf. Figure VII-3). Ces 8 mesures sont converties en centimètres (grâce à un étalonnage réalisé au début de la campagne d'acquisition), afin d'être ensuite utilisées comme attributs de classification. Notons que le mouvement de protrusion des lèvres n'est pas mesurable sur des vidéos monocaméra où le codeur est présenté de face.

– La **classification** proprement dite est réalisée par la mise en compétition des modèles génératifs de toutes les combinaisons de CV : des HMM à trois états sont entraînés sur une première répétition de la série de phrases ETTRAN, et les tests sont réalisés sur une seconde répétition. En revanche, les deux corpus représentent le code d'une seule et même personne. Comme les syllabes sont produites lors d'un codage continu, cela permet de reconnaître des éléments de base CV au sein de leur contexte prosodique. Cependant, les CV sont isolées de leur contexte avant d'être traitées. Cela revient donc à s'intéresser à la reconnaissance des éléments unitaires du mouvement labial, la reconnaissance d'un codage continu non étiqueté étant encore un problème ouvert.

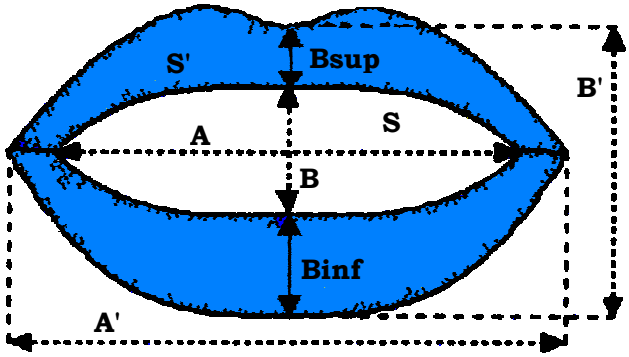


Figure VII-3 : les 8 attributs de classification utilisés pour la reconnaissance de formes labiales, issus de [6].

– La **prise de décision** se fait par maximum de vraisemblance (ML). Sur le corpus de test, le taux de reconnaissance est à l'heure actuelle de 80.3% pour 15 classes (correspondant au produit des 3 clusters de voyelles et 5 clusters de consonnes).

D'une manière générale, ces travaux envisagent la reconnaissance du mouvement labial d'une manière très similaire à celle avec laquelle le problème de la reconnaissance de la parole est abordé dans l'état de l'art. Cela est justifié par le fait que le signal à reconnaître est très proche : dans tous les cas, le message contient le même enchaînement de phonèmes. Au chapitre II (p. 35) nous avons adopté une démarche relevant d'une modélisation différente en émettant l'hypothèse d'un code statique, par nature différent du continuum oral. Malgré nos réticences à envisager le problème sous l'angle du traitement de la

parole, il faut reconnaître que les résultats préliminaires de [2], [3], [4], [5] et [6] montrent que cette approche est aussi efficace que celle que nous avons choisie.

En conclusion, il est possible à l'heure actuelle de reconnaître les différentes formes labiales de syllabes CV en situation mono-codeur (un seul codeur et même codeur à l'apprentissage et durant l'évaluation) sur des vidéos avec artifices (maquillage bleu et casque de maintien). Cependant, ces travaux sont en cours de développement, la reconnaissance du mouvement des lèvres dans le cas d'un codage continu et sans artifice étant l'objectif à terme.

VII.1.3 Segmentation du contour des lèvres

Les ergonomes de France Telecom R&D sont formels : il n'est pas réaliste de proposer un terminal TELMA pour lequel il serait nécessaire de se maquiller les lèvres en bleu. De ce fait, en parallèle des modèles de reconnaissance que nous venons de résumer, d'autres travaux ont pour objectif de réaliser la segmentation des lèvres sans maquillage spécifique. Ce point là est le plus délicat à traiter en termes de traitement d'images. En effet, c'est celui qui nécessite la plus grande précision, et la plus haute résolution d'image lors de l'acquisition. Ce point-ci a déjà été abordé lors de la description de protocoles d'acquisition, pour la simple raison que ces contraintes ont une influence sur toutes les autres briques de TELMA.

Figure VII-4 : exemples de segmentation des lèvres d'après [13].

Les travaux de Stillittano [13] ont pour objet l'extraction automatique du contour interne des lèvres. Pour cela, l'algorithme suppose qu'un détecteur permet de localiser le visage du codeur, et que le contour externe des lèvres a déjà été extrait selon l'algorithme proposé dans [10]. L'ensemble des contours intérieur et extérieur est retourné sous la forme de 26 coefficients permettant de spécifier un modèle paramétrique. Des exemples de résultats obtenus sont présentés sur la Figure VII-4.

VII.2 Les difficultés de l'analyse labiale

Avant de considérer le problème de la reconnaissance conjointe des mouvements des lèvres et de la main, il convient de s'intéresser à des aspects de la reconnaissance du mouvement labial qui n'ont pas encore été évoqués. En effet, la reconnaissance d'une trajectoire labiale est un problème beaucoup plus complexe que la reconnaissance d'un geste manuel, et de nombreux verrous restent encore à lever dans ce domaine. On y retrouve les mêmes difficultés que pour l'analyse manuelle, en plus d'un nombre important de difficultés plus proches de celles rencontrées dans l'analyse de la parole. Il y a aussi un certain nombre de difficultés intrinsèques. Voici un aperçu de toutes ces difficultés :

– **Normalisation des lèvres** : il n'est à terme pas possible de maintenir la tête du codeur attachée. Ainsi, il va falloir trouver une méthode permettant de rendre les attributs de classification des formes labiales {**A**, **B**, **S**, **A'**, **B'**, **S'**, **Binf**, **Bsup**} robuste aux similitudes.

– **Variabilité inter-locuteur** : celle-ci intervient à plusieurs niveaux. Au niveau morphologique, comme pour la main, mais aussi au niveau dynamique, pour lequel cette variabilité est beaucoup plus forte. Enfin, il y a le niveau articulatoire, absent pour la main.

– **Perte de l'information non labiale** : il y a une grande quantité d'information que l'on ne peut récupérer (ou que très difficilement) à partir du seul contour labial, comme par exemple, les contacts linguo-dentaux, les mouvements de la gorge et de la pomme d'Adam, des joues, la modification de la texture labiale, etc. Ainsi, une importante quantité d'information normalement nécessaire à la lecture labiale n'est pas accessible.

– **Coarticulation** : la notion de cible aux lèvres est beaucoup moins évidente que celles de cibles manuelles pour la simple raison que la coarticulation est beaucoup plus forte. Pour les consonnes, c'est encore pire, puisqu'elles n'ont quasiment pas de représentation propre. Il y en a même certaines qui sont invisibles, telles que les consonnes vélaires /k/, /g/ et /R/. Les classes sont moins bien définies, et il risque de falloir avoir recours à une classification hiérarchique, ou à des représentations en dendrogramme (inclusion arborescente de clusters) [2].

En conséquence de tout cela, certains points ne sont pas encore résolus et sont autant d'obstacles à l'étude conjointe des mouvements manuel et labial. Nous avons relevé deux cas précis, et nous proposons une ébauche de stratégie pour chacun d'eux. Le premier cas concerne l'utilisation d'algorithme de classification de formes labiales sur des données segmentées automatiquement. Le second cas concerne la reconnaissance de phrases continues.

VII.2.1 Classification et segmentation automatique

Actuellement, la segmentation du contour des lèvres est étudiée de manière indépendante, et n'est pas évaluée au regard des possibilités de classification à des fins de lecture labiale. De même, les méthodes de classification des formes labiales actuelles sont élaborées sur des données issues d'acquisition avec artifices (maquillage bleu des lèvres et casque de maintien) beaucoup plus précises, et non sur des lèvres segmentées automatiquement en conditions non contrôlées. Ainsi, le premier travail important est d'évaluer les résultats des méthodes de segmentation/classification conjointes, en couplant les algorithmes développés jusqu'ici séparément. Une utilisation sur des données de moins bonne qualité a de grande chance de conduire à des résultats moins robustes. Pour pallier ces éventuelles dégradations de performances, nous proposons les pistes suivantes :

– Utiliser les 26 coefficients du modèle de segmentation des formes de lèvres comme attributs de classification.

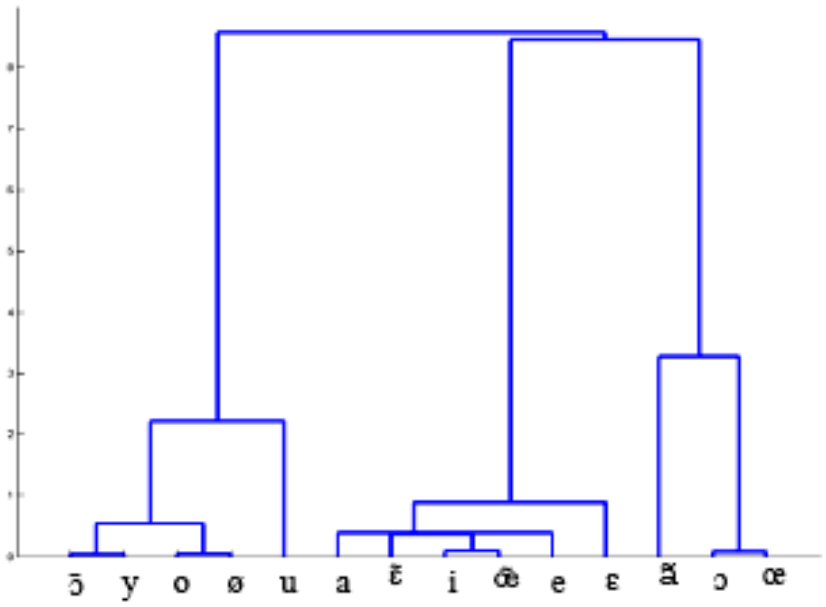


Figure VII-5 : dendrogramme des formes labiales associées aux voyelles du français. issu de [2].

– En raison de la suppression du casque de maintien, il faut proposer une solution au problème de normalisation des lèvres. Nous proposons de normaliser les formes labiales en fonction de la distance interoculaire du codeur. Une alternative est d'utiliser des attributs invariants aux similitudes tels que les DFM. Cependant, comme certaines formes labiales se différencient principalement par la taille de l'ouverture buccale (par exemple le /u/ et le /a/), nous ne pensons pas qu'il s'agisse d'une bonne solution. Ainsi, la normalisation des 26 paramètres de formes labiales semble plus appropriée.

– En pratique, il n'y a aucune raison d'être confronté à une plus grande difficulté liée à la malédiction de la dimension (cf. appendice C.5, p. 301) avec 26 ou 33 paramètres qu'avec 8. Si des problèmes de sur-apprentissage arrivent malgré tout, c'est que les corpus d'apprentissage sont trop petits. Pour compenser cela, nous proposons de ne plus imposer un modèle gaussien aux classes et d'utiliser des méthodes de classification non paramétriques, moins sensibles à la malédiction. Nous choisirions les SVM et la Combinaison Evidentielle (V.2.3.1, p. 148) parce que nous possédons une expertise à ce sujet, mais d'autres méthodes peuvent aussi être efficaces. Néanmoins, comme il est proposé dans [2] de représenter les formes labiales par un dendrogramme, nous pensons qu'une méthode de classification crédale serait particulièrement adaptée. En effet, elle permettrait par la suite, l'application de la PPT, dont la gestion automatique de l'incertitude correspond au type de décision à prendre sur une arborescence de clusters. Nous illustrons cela dans [J3], en reprenant le dendrogramme de [2]. Celui-ci représente les formes de lèvres associées aux voyelles du français (cf. Figure VII-5).

Ce dendrogramme de 14 classes laisse clairement voir 3 clusters, que les auteurs associent aux classes canoniques des voyelles en linguistique : voyelles

antérieures non arrondies, voyelles hautes et mi-haute arrondies et voyelles basses et mi-basses arrondies. La taille des clusters étant respectivement de 5, 6 et 3, la 6ème-PPT serait un outil adapté à une telle prise de décision : il est possible d'hésiter entre 6 classes ou moins, mais aussi de focaliser sa décision si la quantité d'information le permet.

VII.2.2 Etude du codage continu

Comme nous l'avons dit, l'approche utilisée dans TELMA pour la reconnaissance du mouvement labial est très proche des méthodes de reconnaissance de la parole orale. A l'heure actuelle, ces travaux considèrent le cas de syllabes CV isolées, étape préliminaire à la reconnaissance d'un codage continu. Cependant, le passage de la reconnaissance d'éléments unitaires (les CV) à des phrases continues constitue toujours une véritable difficulté. Comme nous l'avons mentionné au chapitre VI (p. 182), il existe des méthodes efficaces dans le cas de la parole orale, mais dans le cas de langages gestuels complexes tels que l'ASL, il s'agit encore d'un problème ouvert. Le cas du LPC n'ayant pas encore été traité, personne ne sait si ce signal gestuel (tout comme l'ASL) mais néanmoins proche du signal de parole pourra être traité avec succès par les méthodes classiques de traitement de la parole, ou si, au contraire, les mêmes difficultés que dans le cas de l'ASL rendront ces méthodes inefficaces.

Nous ne présupposons rien des résultats que cela pourrait donner. Cependant, nous proposons de répondre à ce problème par une méthode similaire à celles utilisées jusqu'à présent. Sur le principe, il s'agit de **rechercher des Images Cibles aux Lèvres** (des ICL, correspondant à des instants particuliers du mouvement labial), et de **proposer une première segmentation temporelle** du codage continu à partir de celles-ci. L'objectif est de pouvoir **isoler des morceaux de séquences correspondant à des mouvements labiaux de type CV**, afin de les reconnaître via le banc de HMM déjà existant.

Finalement, il s'agit de mettre en place une labellisation précoce du mouvement labial. En raison de la coarticulation très importante, cette labellisation du mouvement labial sera plus difficile celle de la main. Cependant, de manière similaire à la main, nous pouvons remarquer que :

– Lors de la prononciation de certains phonèmes, la déformation du contour des lèvres augmente jusqu'à atteindre un maximum (ou un minimum), puis s'inverse. Cette inversion du mouvement s'accompagnant d'une diminution de la vitesse, elle est caractéristique du type d'indices sur lesquels est basé la labellisation précoce.

– Le mouvement du contour des lèvres est un mouvement de déformation, que l'on peut comparer au mouvement de changement de Configuration.

Ainsi, nous pensons qu'il est possible d'utiliser le FRD afin d'appréhender le mouvement de déformation de lèvres, de détecter au moins certains instants-clés de la trajectoire labiale et ainsi de labelliser :

– les zones d'étirements maximum à faible aperture tels que les /i/,

– les instants de faible étirement et de faible aperture, tels que les /u/ ("ou") et les /y/ ("u"),

– etc.

Cependant, certains visèmes ne s'accompagnent pas d'une déformation du contour labial (tels que les consonnes vélaires /k/, /g/ et /R/). Ainsi, contrairement à la main, il n'est pas du tout garanti qu'il soit possible d'extraire toutes les ICL. En revanche, en fonction de notre expertise sur le mouvement des lèvres ([C6], [A1]), et de la main, de nos travaux préliminaires sur l'ASL (section VI.1 p. 188) et des performances du modèle de rétine que nous avons utilisé pour mettre en place le FRD [155], nous avons la conviction qu'au moins les ICL les plus marquées (telles que certaines voyelles, ou telles que les consonnes occlusives /p/, /b/ et /m/) pourront être labellisées. Cela n'est en aucun cas suffisant pour résoudre le complexe problème du traitement du codage continu. Néanmoins, cela peut participer à la réduction de sa complexité, en fournissant une première segmentation temporelle du signal.

Dans tous les cas, cette méthode a pour principal avantage d'être facilement compatible avec la méthode de reconnaissance gestuelle présentée dans ce document, puisqu'elle se base sur les mêmes outils théoriques.

VII.3 Proposition d'une stratégie complète de reconnaissance du LPC

Dans cette dernière section, nous présentons une stratégie de fusion des informations de Configuration, de Position et de lèvres qu'il nous semblerait intéressant de tester. Il s'agit d'une piste de travail, basée sur notre expertise du traitement automatique du LPC. Nous sommes convaincus que cette manière de procéder peut donner de bons résultats, sans pour autant que ce soit ni la seule, ni la meilleure manière de procéder. Elle correspond simplement à l'esprit, à la logique de ce travail.

VII.3.1 Principe de la méthode et motivations

Dans les grandes lignes, le principe de la méthode que nous imaginons est le suivant :

(1) Tout d'abord, il s'agit de reconnaître les informations contenues dans les trois flux de Configurations, et Positions et labial. Pour pouvoir gérer efficacement la multimodalité de l'ensemble des composantes, nous proposons de ne pas forcer de décision dans un contexte incertain, et d'utiliser des outils tels que la PPT.

(2) Ensuite, il s'agit de mettre en place un système permettant d'inférer la désynchronisation des différentes composantes du LPC par rapport à un codage

théorique parfait, ainsi que les éventuels décalages introduits par les divers traitements, et notamment, la labellisation précoce. Dès lors, il est possible de déterminer quelles sont les corrections à apporter aux signaux observés pour qu'ils soient synchronisés avec le code théorique. Une fois que les corrections nécessaires sont connues, il suffit de resynchroniser en conséquence les différents flux (succession des formes de lèvres reconnues, succession des Configurations reconnues, et succession des Positions reconnues). Pour inférer les différents décalages/désynchronisations, nous proposons de faire appel à l'algorithme de Viterbi, mais de l'utiliser dans une optique de filtrage, c'est-à-dire d'une manière différente de ce qui est habituellement fait en vision par ordinateur.

(3) A partir de là, chaque phonème articulé dans l'espace de codage du LPC peut être inféré par fusion des différentes modalités. Afin de traiter les difficultés de synchronisation en même temps que les ambiguïtés de décodage ayant pour origine la multimodalité du LPC, nous proposons de faire appel aux outils mis en place sur l'ASL, dont nous avons montré l'intérêt pour la fusion de modalités gestuelles.

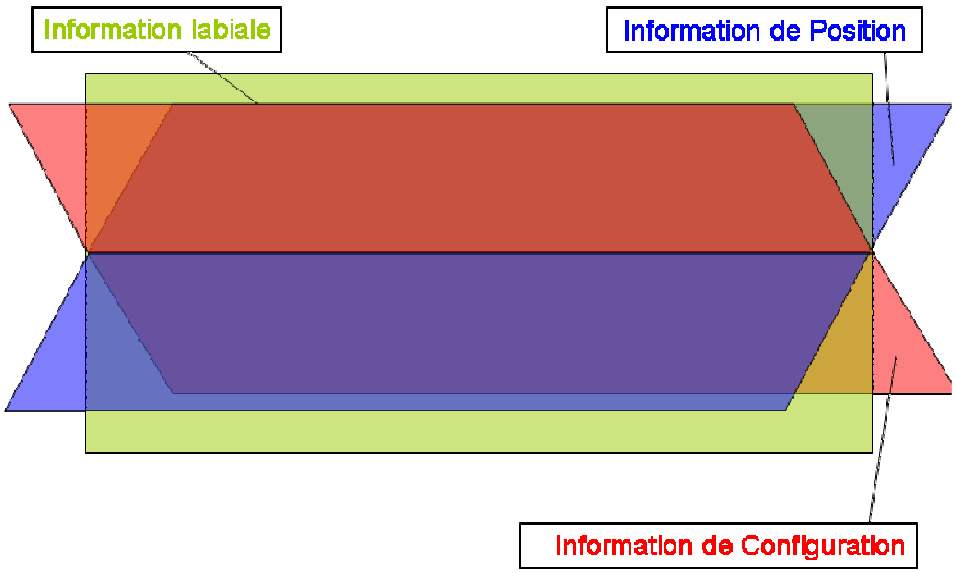


Figure VII-6 : intersection des treillis de Configuration, de Position et de mouvement labial pour la reconstitution complète du message codé.

Voici les différentes considérations qui nous ont poussées à choisir une telle méthode :

– à notre avis, traiter le problème de la fusion de la Position et de la Configuration pour obtenir l'information fournie par la main dans son ensemble, puis utiliser un second étage pour procéder à une fusion similaire entre la main et les lèvres, n'est pas la meilleure solution. Même si celle-ci semble la plus naturelle quand cette tâche est réalisée par un humain (cf. le modèle d'Attina),

elle peut être inadaptée dans le cas d'une fusion automatique. En effet, cette méthode oblige à prendre une décision intermédiaire qui peut ne pas avoir beaucoup de sens, et empêche de la remettre en cause par la suite. En conséquence, cela multiplie le nombre d'erreurs possibles. Ainsi, nous pensons qu'il est plus judicieux de résoudre ce problème en une fois, en considérant directement trois flux d'informations à combiner (les flux labial, de Position et de Configuration), comme illustré sur la Figure VII-6.

– de même, nous ne cherchons pas à généraliser à trois flux nos résultats sur la fusion de deux flux (cf. la fusion de Configuration/Position au chapitre VI). En effet, les méthodes expérimentées dans ce cas-là ne peuvent pas être réutilisées car nous avons restreint la difficulté à sa seule composante temporelle.

– enfin, l'autre méthode que nous avons mentionnée (basée sur la résolution d'un couplage dans un graphe) est difficile à généraliser au cas de trois flux. En effet, cela signifie que les arêtes du graphe doivent relier 3 sommets, ce qui nous amène à considérer le problème dans le formalisme des hypergraphes, beaucoup plus complexe.

Dans les paragraphes suivants, nous détaillons les étapes de la méthode que nous venons d'introduire. Tout d'abord, nous expliquons comment nous proposons d'améliorer la reconnaissance des flux de Configurations, de Position et labial par application de la PPT. Ensuite, nous rappelons quels sont les résultats préalablement nécessaires à la fusion pour chacune des modalités. Dans un troisième temps, nous détaillons la méthode de resynchronisation des différentes modalités par l'utilisation de l'algorithme de Viterbi. Cette section étant relativement longue, elle est découpée en plusieurs parties. Enfin, dans un dernier paragraphe, nous présentons quelques considérations pour la fusion des modalités resynchronisées, en insistant sur les difficultés que l'on peut y rencontrer.

VII.3.2 Application de la PPT au cas du LPC

Il serait intéressant d'évaluer l'apport de l'utilisation de méthodes proches de celles utilisées pour l'ASL et décrites dans la section VI.2 (p. 200). Nous pensons en effet que cela peut largement améliorer le résultat de la reconnaissance de chacun des flux du LPC (Configuration, Position, lèvres), pris de manière indépendante. Cela permet de se baser sur des informations plus fiables pour aborder la fusion de ces trois composantes. Si malgré tout, il reste des cas où l'information n'est pas fiable, nous avons montré qu'il est possible de modéliser cela et de le prendre en compte dans le processus de fusion des différentes modalités. Voici deux points via lesquels nous pensons qu'il est possible d'améliorer le résultat de la reconnaissance complète en appliquant les méthodes de la section VI.2 :

– La PPT permet de maintenir une certaine hésitation lors de la décision inhérente à un processus de classification. Ceci peut être appliqué au processus de reconnaissance du LPC de manière à gérer plus facilement les ambiguïtés

d'un codage peu académique. Dans le cas de la Configuration, cette adaptation peut se faire assez simplement : il suffit d'utiliser la PPT sur la FCF en sortie des SVM, lors du processus de classification. Nous justifions le choix d'un paramètre d'incertitude de 2, car cela permet une hésitation adaptée sur une transition (représentant un mélange des Configurations précédente et suivante). Nous avons expliqué à la section précédente comment cela pourrait être appliqué à la reconnaissance de formes labiales (application de la 6ème-PPT sur le dendrogramme des voyelles). Dans le cas de la reconnaissance de la Position, cela n'est pas aussi immédiat, dans la mesure où le classifieur utilisé n'est pas crédal. Cependant, cela peut faire partie des améliorations à apporter. Dans le cas de la Position, cela serait particulièrement intéressant pour traiter les cas où la main dérive trop lentement d'une Position à une autre pour qu'une décision complète soit fournie (ce qui est généralement la marque d'une hésitation/bégaiement de la part du codeur).

– Afin de pouvoir spécifier le comportement du module de reconnaissance complet du LPC, il pourrait être intéressant d'apprendre pour un utilisateur particulier quelles sont les Configurations (resp. les Positions, resp. les formes de lèvres) qu'il code (resp. qu'il articule) de telle sorte qu'elles sont souvent confondues. Avec les méthodes classiques, cela nécessite la mise en place d'un corpus d'apprentissage sur lequel les confusions vont être étudiées. En pratique, cela signifie qu'une phase de "rodage" de l'algorithme est nécessaire : l'utilisateur doit enregistrer un corpus donné avant que la reconnaissance puisse fonctionner. Grâce à la méthode que nous proposons pour définir une matrice d'hésitation sur un corpus non étiqueté, il est possible d'effectuer cela automatiquement et de manière non supervisée. Ainsi le système peut être spécialisé au cours du temps sans intervention extérieure.

VII.3.3 Pré-requis sur les traitements des flux séparés de Positions, de Configurations et labiaux

La Configuration et la Position doivent être traitées conformément à ce que nous avons décrit dans ce document. Ainsi, pour chaque image de la séquence vidéo à traiter, il faut fournir les informations suivantes :

– Quelle Configuration ou absence de Configuration (dans le cas d'une transition repérée par la labellisation précoce) contient l'image ;

– Quelle Position ou absence de Position contient l'image ;

– Quels labels lui a attribué la labellisation précoce. L'image est-t-elle est une ICX ? Appartient-elle à une zone de stabilité ? Est-elle une ITX ? Est-elle une ITXM ?

Ces informations peuvent éventuellement être plus riches. Par exemple, il pourrait être intéressant d'avoir également les informations suivantes :

– Une FC sur l'espace des Configurations, afin de conserver une éventuelle information conflictuelle, ou un éventuel doute, conformément à ce que nous proposons dans le paragraphe précédent.

– De même sur l'espace des Positions.

– Au lieu de fournir le label de la labellisation précoce, il est possible de fournir un vecteur contenant la quantité de mouvement (sortie du FRD ou de l'étude de la trajectoire du CG en fonction de la composante considérée), sur les 5 images précédentes, afin de fournir une information plus riche quant à la variation de la quantité de mouvement pour arriver sur cette image.

Concernant la trajectoire labiale, nous n'effectuons aucune hypothèse quand à la manière dont celle-ci est reconnue. Tout au plus nous avons émis précédemment quelques suggestions. Les seules informations qui nous semblent requises sont similaires à celles concernant la Position ou la Configuration, à savoir :

– Un résultat de classification pour chaque image de la séquence. La classe ainsi attribuée pouvant soit être l'un des visèmes du français, soit une forme labiale particulièrement reconnaissable.

– Une mesure indiquant la quantité de déformation de la bouche d'un type similaire à celle que produirait le FRD sur le contour labial, éventuellement assorti d'un label indiquant si l'image en question est une ICL.

VII.3.4 Inférence des désynchronisations/décalages

Nous avons pour objectif de déterminer les décalages/désynchronisations des différentes composantes du LPC (Position, Configuration, Lèvres) par rapport à un codage théorique. Pour cela, nous faisons l'hypothèse que l'évolution de l'état de désynchronisation suit un processus de Markov à temps continu. Quand ce processus est caché et qu'il s'agit de l'implanter sur une machine à temps discret, l'outil généralement utilisé est le HMM. Dans notre cas, nous pensons qu'il est inadapté, et nous proposons de le remplacer par un Segmental-HMM. C'est sur celui-ci que nous proposons de réaliser l'inférence proprement dite des décalages/désynchronisations. Pour cela, nous proposons l'utilisation de l'algorithme de Viterbi.

Le plan de ce paragraphe est le suivant. Dans un premier temps, nous partons du modèle d'Attina sur la désynchronisation main/lèvres pour discuter de la nature des apparitions de décalage, et nous émettons l'hypothèse que l'organisation temporelle des flux Position/Configurations/lèvres suit des lois identiques. Cela permet d'aboutir dans un deuxième temps à la modélisation par processus de Markov à temps continu (aussi appelé processus de Markov à sauts et abrégé en PMS) et nous expliquons pourquoi son implantation classique via un HMM ne nous convient pas. Dans un troisième temps, nous expliquons

les principes du Segmental-HMM et enfin, nous proposons une implantation possible d'un Segmental-HMM pour notre problème.

VII.3.4.1 Discussion à partir du modèle d'Attina

Dans ce paragraphe, nous discutons du type de connaissance *a priori* que permet d'apporter un modèle d'ordonnancement tel que celui d'Attina, afin de les injecter dans notre système de resynchronisation.

Comme nous l'avons déjà mentionné plus haut, ce schéma d'ordonnancement contient trop de variabilité pour qu'il puisse être utilisé pour prédire les décalages entre les différentes modalités. En revanche, il peut servir à paramétrer un système d'inférence en lui injectant de la connaissance *a priori* à propos des lois d'apparition de désynchronisations/décalages. Dès lors, il s'agit d'extraire cette information.

La plupart des méthodes d'inférence efficaces sont basées sur des méthodes probabilistes, parce qu'elles sont un compromis entre complexité de calcul et efficacité. Le problème est que la seule distribution continue avec laquelle il est possible de faire de l'inférence facilement et de manière exacte est la distribution gaussienne. En effet, une gaussienne multipliée par un scalaire reste une gaussienne (loi de composition externe), et le produit de deux gaussiennes est une gaussienne (loi de composition interne). Cette structure vectorielle permet de propager facilement de la connaissance dans une structure inférentielle, même très complexe, sous la forme d'un doublet {moyenne, variance}.

Cependant, nous pensons qu'il n'est pas possible d'extraire de ce schéma d'ordonnancement des relations probabilistes de synchronisation basées sur la loi normale. Il y a deux raisons à cela :

– Tout d'abord, l'hypothèse gaussienne implique une certaine symétrie que le décalage temporel ne permet évidemment pas dans ce cas.

– Ensuite, l'hypothèse gaussienne n'est fondée que sur l'hypothèse que la loi des grands nombres fera converger toutes statistiques d'observations vers la loi normale [175]. Cette hypothèse est souvent utilisée dans un cadre subjectiviste, mais ce cadre correspond mal à l'objectivité des mesures d'Attina.

Dans notre cas, il s'agit de caractériser une séquence sur laquelle une légère désynchronisation apparaît. Cette désynchronisation est caractérisable par deux instants :

– Celui où, pour la première fois, les deux flux ne sont plus en phase. C'est le début de la désynchronisation main/lèvres.

– Celui où après la désynchronisation, pour la première fois, les deux flux sont de nouveau en phase.

Au sein du continuum temporel sur lequel ces deux flux sont analysés, ces instants sont de probabilité nulle, et par conséquent, il serait plus judicieux de les considérer comme des phénomènes rares (un phénomène rare correspond au comportement d'une variable aléatoire de comptage de succès d'un nombre infini de tirages binomiaux de paramètre de succès infinitésimal). Or justement, ce type de phénomènes rares ne se modélise pas bien par des gaussiennes, mais plutôt par des lois de Poisson : étant donné un intervalle de temps de durée D, le nombre de phénomènes rares qu'il contient suit une loi de Poisson $\mathcal{P}(\lambda)$ dont le paramètre λ est proportionnel à D.

Nous pensons que le schéma d'ordonnancement d'Attina est à la fois intéressant et utile à la mise en place d'un système de synchronisation des modalités du LPC. Cependant, nous pensons qu'un tel système sera encore plus efficace si cette modélisation de Poisson est respectée. Ainsi, les données qui ont servi à la mise en place du schéma d'ordonnancement pourraient être réutilisées afin d'estimer les paramètres des diverses lois de Poisson qui gouvernent l'ordonnancement temporel de la main par rapport aux lèvres.

VII.3.4.2 Justification de la modélisation par PMS

Un Processus de Markov à sauts est la généralisation d'une chaîne de Markov dans le cas où le temps est continu. Il s'agit d'un processus bidimensionnel. Une des dimensions correspond à une chaîne de Markov classique, pour laquelle la diagonale de la matrice de transition est nulle. L'autre dimension est un processus de Poisson, c'est-à-dire un processus de comptage pour lequel les temps inter-occurrences suivent des lois exponentielles.

Commençons par montrer que l'état de synchronisation/désynchronisation des composantes Configuration/Position/lèvres du LPC peut-être modélisé par un **Processus de Markov à temps continu** (ou **Processus de Markov à saut**) :

– Le processus de Poisson correspond à la succession (ou au comptage) des phénomènes rares que sont les instants de resynchronisation/désynchronisation. Par définition, le temps entre deux sauts du processus de Poisson (ou comptage) suit une loi exponentielle.

– L'état du système indique parmi les 3 flux, lesquels sont synchronisés par rapport au codage théorique et lesquels ne le sont pas (dans notre cas de 3 flux, il y a donc $2^3 = 8$ états). Le changement d'état du système suivant une chaîne de Markov permet quand à lui de modéliser la succession des synchronisation/désynchronisation des flux les uns par rapport aux autres.

Ensuite, nous proposons d'appliquer l'algorithme de Viterbi sur ce processus afin de déterminer quand le système se trouve dans des états désynchronisés et quand le système se trouve dans un état synchronisé.

L'algorithme de Viterbi [128] permet de trouver par une méthode de programmation dynamique un chemin de poids maximal dans un treillis (un

graphe à plusieurs couches où chaque couche est un graphe biparti), dans lequel les arêtes, mais aussi les sommets ont un poids. D'une manière générale, les sommets représentent les états du système étudié, et leur poids, la probabilité que cet état corresponde aux observations ; les arêtes représentent quant à elles les transitions entre états, elles aussi assorties d'une probabilité. Dans la communauté de la vision par ordinateur, cet algorithme est souvent utilisé face à des problèmes d'apprentissage afin de mettre en place une méthode de classification basée sur des HMM [142], [143], [144]. Cependant, cet algorithme est aussi très populaire pour le séquençage de l'ADN [141], ou en théorie des codes pour le décodage d'un code convolutif par blocs après son passage dans un canal de transmission sans mémoire. Dans ce cas-ci, il s'agit en pratique de répondre à une problématique de décodage, ou encore de filtrage, d'une manière similaire au filtre de Kalman. Nous proposons de considérer le traitement de la désynchronisation et du décalage des modalités du LPC comme un problème de filtrage : le codage théorique parfait est le signal débruité que l'on cherche à obtenir à partir du code désynchronisé (qui représente le signal bruité). L'algorithme de Viterbi est donc utilisé pour rechercher dans le treillis des désynchronisations possibles celle qui maximise la probabilité d'avoir été produit par le code théorique parfait.

Afin de pouvoir appliquer cet algorithme sur le PMS qui modélise l'état de désynchronisation, nous devons l'implanter sur une machine (pour laquelle le temps est discret). Comme nous venons de le dire, cela se fait classiquement à l'aide d'une chaîne de Markov. En effet, dans une chaîne de Markov, la probabilité que le système passe un temps t dans un même état S_i est $a_{ii}^{t-1}(1-a_{ii})$, où a_{ii} désigne la probabilité de self-transition[13] pour l'état S_i. En pratique, la loi de probabilité du temps passé dans un même état est donc très proche de la loi exponentielle des temps inter-occurrences considérée dans une modélisation continue du temps avec un PMS. Cependant, nous n'avons pas accès à ce processus : celui-ci est caché, et nous devons l'inférer en fonction d'une série d'observations. Nous devons donc considérer un **PMS caché** sur une machine à temps discret, c'est-à-dire un HMM.

Malheureusement, en raison de notre utilisation particulière de l'algorithme de Viterbi, nous pensons que la modélisation de notre PMS caché par un HMM est inadaptée. En effet, dans le cas d'un HMM classique, la probabilité $a_{ii}^{t-1}(1-a_{ii})$ dépend directement de a_{ii}, qui est une valeur de la matrice de transition. Ainsi, la valeur du temps passé dans un même état d'une part, et les probabilités de quitter cet état pour chacun des autres états d'autre part, sont définies de manière conjointe. Or dans un modèle à temps continu, les deux processus sont indépendants. Quand les HMM sont utilisés comme générateur de modèles de classe dans un problème de classification, cette dépendance n'a pas beaucoup de conséquence. En effet, tous les exemples utilisés à l'apprentissage d'une

[13] Nous désignons par le le néologisme "self-transition", issu de l'anglais, la transition d'un état vers lui-même. En pratique quand le système emprunte une telle transition, il reste dans le même état.

classe doivent avoir le même comportement pour que la classe en question soit bien définie. Cependant, dans notre cas, c'est la désynchronisation qui nous intéresse et que nous voulons faire ressortir. Ainsi, il est nécessaire de pouvoir modifier les constantes de temps des lois en fonction du rythme auquel le code LPC à resynchroniser est produit.

Pour cela, nous pensons que la méthode la plus adaptée est d'utiliser un HMM avec **Densités Explicites de Probabilités de Rester dans un Etat** [128], ou bien un **Segmental-HMM** [145], [146]. Ces deux outils étant très proches dans leur conception, nous nous focalisons sur le second.

VII.3.4.3 Segmental-HMM

Les Segmental-HMM [145] ne sont théoriquement pas des HMM à proprement parler, mais ils permettent de résoudre certains problèmes avec beaucoup plus de précision. Nous présentons ici brièvement cet outil, et nous montrons par la suite qu'il est adapté à des problèmes de synchronisation.

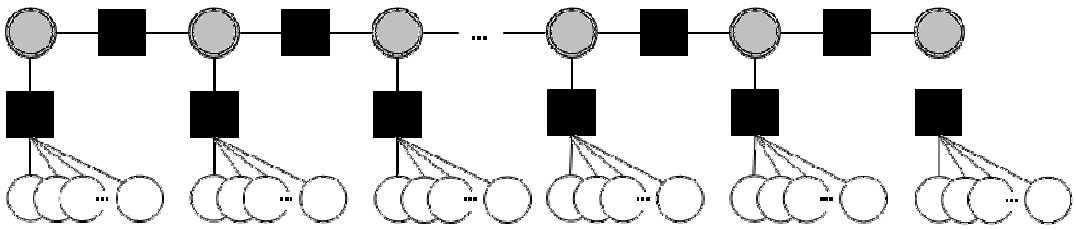


Figure VII-7 : représentation de type Factor Graph d'un Segmental-HMM : à chaque changement d'état, le système peut émettre une observation de longueur variable. Du point de vue de la séquence d'observations, c'est comme si le temps pendant lequel le système reste dans le même état était déterminé de manière indépendante.

Sur le principe, un Segmental-HMM permet de considérer que la chaîne de Markov sous-jacente peut émettre des séquences d'observations de longueur variable dans le temps, en fonction de lois de probabilité préalablement définies (cf. Figure VII-7). En théorie, il ne s'agit plus complètement d'une chaîne de Markov cachée, pour la simple raison que du point de vue de la séquence d'observations générée, deux processus complètement indépendants sont en concurrence. Tout d'abord, il y a celui qui détermine le temps pendant lequel le système reste dans le même état, puis une fois que celui-ci permet un changement d'état, le nouvel état est déterminé. En pratique, cet outil s'utilise cependant exactement de la même manière qu'un HMM, puisque l'ensemble des problèmes se résout par différentes versions de l'algorithme du forward-backward. La seule différence est que le treillis associé ne possède pas deux dimensions (le temps et l'espace d'état), mais trois (le temps par rapport aux changements d'état, le temps par rapport à la durée de la séquence d'observation et l'espace d'état). Le seul inconvénient est que le coût calculatoire d'un tel système est beaucoup plus important, puisque le système possède un

degré de liberté en plus.Pour qu'un Segmental-HMM permette d'implanter un **PMS caché**, il est nécessaire que :

– La loi déterminant la longueur des observations soit une loi exponentielle. Ainsi, le processus de comptage des changements d'état du système suit une loi de Poisson.

– Que la machine à état ne possède pas de self-transitions, afin d'empêcher que le système reste dans le même état quand le processus de Poisson indique un changement d'état.

En résumé, nous pensons donc que l'évolution de la synchronisation du code LPC au cours du temps peut être modélisée par un processus de Markov à saut caché, et que le meilleur moyen d'implanter un tel processus sur une machine à temps discret à des fins de filtrage est d'utiliser un Segmental-HMM ayant des probabilités de saut suivant des lois exponentielles, et dont la machine à état ne possède pas de self-transition.

VII.3.4.4 Esquisse de l'implantation

Au regard de toutes ces considérations, voici la trame générale de résolution que nous proposons pour la resynchronisation des trois modalités du LPC :

Etape 1 - Définition d'une machine à état modélisant la synchronisation de chacun des flux par rapport au code LPC théorique. Un Segmental-HMM est plaqué sur celle-ci. Cela signifie que (1) on associe à chaque arc de la machine à état une loi de probabilité de transition, et à chaque état une loi de probabilité d'émission d'une observation. L'espace des observations est l'espace correspondant aux informations issues de la reconnaissance des différentes modalités (classe et quantité de mouvement). De plus, on associe à chaque état une loi de probabilité de rester dans cet état, ou d'en sortir. Cette dernière loi est impérativement exponentielle, mais de paramètre encore indéterminé. Tout cela est illustré sur la Figure VII-8.

Etape 2 - Définition des lois de probabilités proprement dites. Cela peut être réalisé au moyen d'un schéma d'ordonnancement du type de celui d'Attina, mais prenant en compte les 3 flux de Configurations, de Positions et de Lèvres. Une alternative plus classique est de réaliser un apprentissage.

Etape 3 - Application à la resynchronisation : à partir d'une séquence d'observations tirée d'une séquence vidéo, utiliser l'algorithme de Viterbi pour inférer le schéma de désynchronisation de la séquence par rapport au codage théorique.

Notre utilisation du Segmental-HMM est plus proche du filtrage, ou du décodage, que de la classique utilisation permettant de mettre en place des modèles génératifs et des classifieurs unaires. Ainsi, nous proposons d'utiliser un seul modèle quelque soit la séquence à traiter (et non pas un modèle par séquence à

reconnaître), ce qui devrait permettre de compenser le coût calculatoire supérieur du Segmental-HMM.

Une fois que l'on sait quelles sont les modalités qui sont désynchronisées, il est facile de les corriger une à une. Dès lors, on se retrouve dans la situation de devoir fusionner les différentes modalités du LPC, sans avoir à considérer les problèmes d'ordonnancement temporel, de manière similaire à nos travaux sur l'ASL. Cette fusion constitue la dernière étape de la méthode de reconnaissance complète du LPC que nous proposons. Nous lui consacrons quelques mots dans le paragraphe suivant.

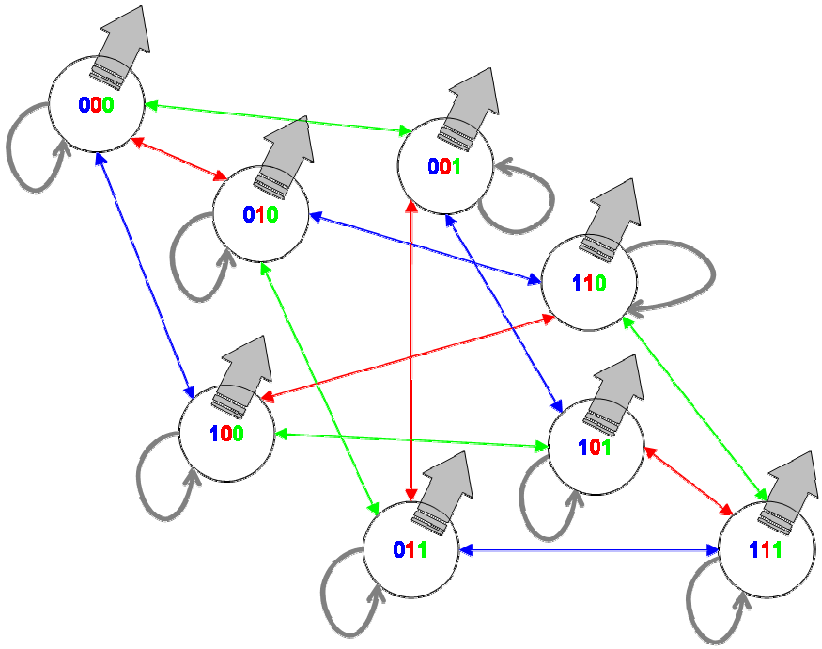


Figure VII-8 : représentation de la machine à état que nous proposons d'utiliser. Chaque état est codé sur 3 chiffres binaires, indiquant la synchronisation de chacune des modalités (Configuration en bleu, Position en rouge et lèvres en vert) par rapport au code théorique (0 = désynchronisé et 1 = synchronisé). Les doubles flèches indiquent les arcs de transition entre les différents états. Suivant que l'on interprète le schéma comme celui d'un processus de Markov à temps continu ou comme un Segmental-HMM, nous avons : (1) les flèches épaisses indiquent qu'une unique observation est émise, et les self-transitions sont indépendantes du processus de Markov, et sont régies par une loi de Poisson ; (2) les flèches épaisses indiquent l'émission d'une série d'observations de longueur variable et les self-transitions ne doivent pas être considérées dans notre cas particulier (mais peuvent exister dans le cas général ; il s'agit alors de transitions du même type que celles reliant deux états).

VII.3.5 Quelques considérations sur la reconnaissance finale

Il est difficile de déterminer longtemps à l'avance quelle devra être la puissance des algorithmes utilisés pour la reconnaissance finale. En effet, si l'ensemble des processus décrit jusqu'à maintenant fonctionne parfaitement, il y a fort à parier que la reconnaissance sera immédiate. Cependant, même dans un tel cas, il y

aura des moments où le code sera réalisé de manière peu académique, et des difficultés ou des incertitudes vont apparaître. Dès lors, il sera nécessaire de faire appel à des techniques de fusion plus sophistiquées, telles que celles que nous avons mises en place pour l'ASL.

Pour cela, il faut avant tout pouvoir faire le lien entre les méthodes crédales que nous avons massivement utilisées, et les méthodes markoviennes que nous n'avons utilisés que dans le cas de l'ASL. Pour cela, les transformées crédales sont adaptées. Elles permettent en effet de faire le lien entre ces deux formalismes.

Lors de la mise en place initiale de notre stratégie de reconnaissance du geste manuel, nous avons écarté les méthodes de l'état de l'art utilisant des modélisations markoviennes pour plusieurs raisons : (1) nous n'avons pas la possibilité d'obtenir un suffisamment grand nombre de données pour espérer des méthodes généralisables en dehors du contexte de nos expériences ; (2) une connaissance approfondie et une étude minutieuse des particularités du LPC montrent qu'il est possible de simplifier le problème ; (3) appliquées directement aux différents cas de reconnaissance gestuelle, ces méthodes donnent de bons résultats sur des gestes isolés, mais se généralisent mal au cas de codages continus.

Maintenant que nous avons (1) la certitude que le problème du décodage du LPC est réalisable, (2) suffisamment simplifié le problème au regard de ce que permet de faire la structure particulière du LPC, (3) et mis en place des méthodes de segmentation des gestes au cours du temps lors d'un codage continu, les méthodes basées sur des hypothèses markoviennes valent de nouveau la peine d'être explorées. Ainsi, il pourrait être intéressant d'utiliser des bancs de HMM semblables à ceux mis en place pour l'ASL, afin de décoder certains gestes déjà segmentés du reste du codage continu dans lequel ils sont immergés.

De plus, quelque soit le niveau de difficulté de la reconnaissance finale, il y a des situations qui resteront délicates. Pour ces situations, il sera nécessaire de mettre en place des algorithmes supplémentaires de traitement. A titre d'exemple, il est difficile de :

– faire la distinction entre un geste codé une seule fois, et un même geste répété deux fois de manière identique pour le codage de deux syllabes où la répétition du geste n'est accompagnée d'aucun indice cinématique repérable ;

– traiter les cas de fortes désynchronisations en début et fin de phrase ;

– traiter les cas où le mouvement de changement de Configuration commence alors que le mouvement de changement de Position est trop faible pour que le doigt pointeur sorte de la zone correspondant à la Position Côté ou Gorge (problème que nous avons abordé plusieurs fois) ;

– lisser les phénomènes de dérive de la Position ;

– Reconnaître un geste complet quand, au milieu de la zone de stabilité, une image cible n'est pas reconnue (comme quand la Configuration devient indistincte en Position Gorge à cause de la flexion du poignet).

Dans tous ces cas, les modélisations markoviennes sont particulièrement adaptées, même sous leur forme la plus commune, à savoir le HMM. Dès lors, il est intéressant de les utiliser sur ces situations bien ciblées.

Enfin, ces modélisations sont très proches des méthodes de traitement de la parole. Ainsi, cela permettrait d'avoir une méthodologie plus proche de celle concernant le traitement des lèvres, et donc de pouvoir unifier tous ces traitements plus facilement.

VII.4 Conclusion du chapitre

Dans ce chapitre, nous proposons une stratégie de décodage complet du LPC, basée d'une part sur la reconnaissance du contenu des trois flux de Positions, de Configurations et labial, et d'autre part, sur un algorithme permettant de resynchroniser ces trois flux.

Pour cela, nous partons des résultats de ce document et des différents travaux menés par ailleurs dan le cadre de TELMA sur la reconnaissance du mouvement labial. Comme cette dernière problématique est très complexe, certains de ces aspects sont encore ouverts. Ainsi, nous proposons quelques pistes d'exploration inspirées des méthodes que nous avons utilisées dans ce document.

Une fois la problématique de reconnaissance de chacun des flux traitée, nous proposons une méthode dont le but est de resynchroniser les différentes modalités du LPC. Nous nous basons sur l'étude de l'ordonnancement temporel du LPC d'Attina pour justifier une modélisation par processus de Markov à sauts caché, et nous émettons l'hypothèse qu'il est possible de résoudre cette désynchronisation en utilisant l'algorithme de Viterbi à des fins de filtrage. Enfin, nous montrons comment implanter sur une machine un tel processus, et nous donnons un exemple de la machine à état sous-jacente.

Toutes ces considérations sont purement théoriques, dans la mesure où elles dépassent largement le cadre de travail que nous nous sommes fixés, et que par conséquent, elles n'ont pas pu être testées sur des données expérimentales. Néanmoins, leur application à des données LPC réelles est une suite à la fois logique et intéressante de nos travaux.

CONCLUSION GENERALE & PERSPECTIVES

Dans ce rapport, nous avons présenté une méthode de reconnaissance automatique du mouvement manuel du LPC sur des vidéos mono-caméras avec acquisition face au codeur. Ce travail participe à la mise en place d'un module complet dont l'objectif est de permettre la reconnaissance complète du code LPC, c'est-à-dire la reconnaissance du geste manuel ici présenté et de la lecture labiale complémentaire au geste manuel. Il s'inscrit aussi dans un projet plus vaste de mise en place d'éléments algorithmiques et matériels permettant l'accession de la téléphonie aux malentendants.

Pour cela, nous avons développé une méthode particulière s'appuyant sur des outils théoriques préexistants dans les domaines du traitement d'images, de l'interprétation de geste, de la classification, de l'aide à la décision, et sur des outils nouveaux développés directement à cette fin précise, mais qui nous l'espérons, trouveront d'autres applications.

Les spécificités du LPC ont nécessité le développement d'une méthode adaptée. En effet :

– Le codage du LPC correspond à un mouvement constitué de gestes statiques rendus dynamiques par la coarticulation.

– Il est possible de s'affranchir à bas niveau de cet aspect dynamique, en étudiant la cinématique du codage.

– Cela permet de récupérer les primitives du codage que sont la Position et la Configuration sur des images uniques appelées ICP et ICC.

– Le LPC est théoriquement facile à décoder, mais c'est sa production humaine et la variabilité qui en découle qui introduit la difficulté. L'une des raisons de cette difficulté est la désynchronisation du mouvement de codage de la Configuration par rapport à celui de la Position. Ces travaux ont permis de l'identifier et de la mettre en valeur.

La méthode proposée consiste à reconnaître les éléments déterminants du code LPC sur chaque image de manière indépendante, sans se soucier dans un

premier temps de la difficulté de la prise en compte de l'aspect dynamique du codage. En parallèle, un algorithme basé sur l'étude cinématique du mouvement de la main permet de définir les ICP et les ICC. Cet algorithme fonctionne à partir de la trajectoire du centre de gravité de la main et d'un filtre inspiré du fonctionnement de la rétine des vertébrés. Ensuite, les informations issues des images prises indépendamment sont classées, afin de reconnaître les primitives gestuelles. A partir de ces primitives, nous proposons diverses méthodes permettant d'obtenir le geste complet sous réserve de restrictions sur la qualité du codage.

En plus de cette méthodologie, nous proposons quelques éléments de discussion et d'amélioration de systèmes traitant de problématiques proches de la nôtre. Ainsi, nous avons travaillé sur la fusion de modalités dans la langue des signes américaine, et nous proposons d'appliquer ces résultats à la reconnaissance du geste LPC, mais aussi à la problématique de fusion du mouvement des lèvres et du mouvement de la main, qui est indispensable à un décodage complet du LPC.

Les différents outils que nous avons développés à cet égard sont : (1) le filtre rétinien dédié, (2) la Combinaison Evidentielle de classifieurs (amélioration de la combinaison de SVM, combinaisons de classifieurs hétérogènes, qu'ils soient unaires, binaires ou probabilistes, et définition automatique de clusters), (3) les Transformées Crédales, et (4) la Transformée Pignistique Partielle.

L'ensemble de ces travaux (méthode de reconnaissance du geste manuel, applications à des domaines proches, et développement de nouveaux outils) ont tous été évalués ; tout au long de ce document, les résultats de ces évaluations suivent directement la présentation des méthodes. Il ressort de ces dernières que certaines méthodes sont plus robustes que d'autres, mais pour chacune, des limites ont été posées. Ces limites sont évidemment autant de nouveaux problèmes et de nouveaux axes de recherche ; elles ont vocation à être repoussées.

Au niveau de la reconnaissance du geste proprement dite, des améliorations sont possibles à plusieurs niveaux. D'un point de vue général, l'utilisation de méthodes d'apprentissage semi-supervisé permettant de s'adapter au codeur n'ont pas du tout été envisagées. Ces méthodes sont pourtant la garantie d'une reconnaissance plus robuste, et à terme, cela ouvre la possibilité de décoder un code plus "humain" et moins académique.

De manière plus précise, la méthode de segmentation que nous proposons n'est pas suffisamment robuste pour permettre des acquisitions en conditions non contrôlées. De même, le port d'un gant est nécessaire. La segmentation de la main dans un flux vidéo étant un sujet de recherche à part entière, il est possible d'espérer pouvoir à terme remplacer ce module par un autre plus performant.

La reconnaissance des différentes Positions de codage est un autre point faible de notre méthode. C'est aussi celui qui bénéficierait le plus d'une méthode

d'apprentissage semi-supervisé. En revanche, une première amélioration sera possible une fois ces algorithmes couplés avec ceux traitant de la reconnaissance des lèvres. La localisation précise des lèvres permettra de définir avec beaucoup plus de précision les Positions Menton et Bouche.

La labellisation des images cibles est effectuée de manière relativement sommaire. Si malgré tout, les résultats sont particulièrement robustes, c'est en grande partie grâce à l'utilisation du FRD. Ainsi, son application à la reconnaissance des cibles de Position et des cibles labiales est un axe de recherche prometteur. Au cours de diverses collaborations, nous avons aussi cherché à l'appliquer à l'étude d'autres types de trajectoires. Ces travaux n'ont pas été poussés suffisamment loin, mais c'est indéniablement une perspective intéressante. Enfin, la limitation du FRD la plus importante est la nécessité de paramétrer manuellement le filtre IPL. Il serait intéressant d'utiliser les propriétés auto-adaptatives du système visuel pour pouvoir supprimer ce paramétrage manuel.

En termes de classification, nous avons de nombreuses perspectives. D'un point de vue applicatif, il est nécessaire de tester nos méthodes dans des situations où le nombre de codeurs est beaucoup plus élevé. D'un point de vue plus théorique, il serait intéressant :

– De généraliser les méthodes codes correcteurs d'erreurs à la classification évidentielle.

– D'apprendre les fonctions d'appartenance associées aux SVM de manière conjointe à l'hyperplan séparateur.

– D'utiliser des SVRDM pour délimiter le champ de validité de chaque classifieur.

– D'utiliser les transformées crédales pour pouvoir fusionner des classifieurs non crédaux entre eux.

– D'utiliser la méthode que nous proposons pour déterminer des clusters de manière non supervisée à d'autres problèmes.

– De partir de cette méthode pour essayer de mettre en place un système de classification non supervisée complet.

– De tester les Transformées Crédales et la Transformée Pignistique Partielle dans d'autres situations.

– D'appliquer la PPT à des situations de décision/fusion de données mixtes, telles que les Modèles de Markov Cachés (classiques ou crédaux). Il serait intéressant d'en profiter pour continuer à explorer les liens entre fonctions de croyance et probabilités.

Enfin, au sujet de la fusion de modalités non synchronisées, nous n'avons abordé que quelques aspects d'un immense problème dont nous n'avons même pas pu considérer toutes les implications. Parmi les premières pistes que nous envisageons, il y a la possibilité de revenir sur les méthodes d'apprentissage bayesien ou markovien que nous avons dans un premier temps délaissées aussi bien pour des raisons matérielles que par désir de chercher des solutions alternatives, et de voir comment elles s'appliquent à notre problème maintenant qu'il est partiellement résolu. Notamment, les problèmes de segmentation temporelle, qui constituait un verrou majeur à leur application, ont en grande partie été traités.

Ainsi, la mise en place de Segmental-HMM dans le but de modéliser des processus de Markov à sauts, à des fins de filtrages plutôt que de classification, est une voie peu explorée, originale, intéressante et potentiellement source de nombreuses avancées concernant la problématique de la reconnaissance automatique du LPC.

REFERENCES

Langue française Parlée Complétée et TELMA

[1] R.O. Cornett. "Cued Speech". *American Annals of the Deaf* 112, pp. 3-13, 1967.

[2] N. Aboutabit, D. Beautemps and L. Besacier. "Vowel classification from lips: the Cued Speech production case", *ISSP*, Ubatuba, Brazil, 2007.

[3] N. Aboutabit, D. Beautemps and L. Besacier. "Hand and Lips desynchronization analysis in French Cued Speech: Automatic segmentation of Hand flow". *Proceedings of ICASSP'06*, Toulouse, France, 2006.

[4] N. Aboutabit, D. Beautemps and L. Besacier. "Characterization of Cued Speech vowels from the inner lip contour". *In Proceedings of ICSLP'06*, Pittsburg, USA, 2006.

[5] N. Aboutabit, D. Beautemps, L. Besacier. "Automatic identification of vowels in the Cued Speech context". *Proceedings of AVSP (International Conference on Auditory-Visual Speech Processing)*, Hilvarenbeek, The Netherlands, September 2007.

[6] N. Aboutabit, D. Beautemps, J. Clarke, L. Besacier. "A HMM recognition of consonant-vowel syllables from lip contours: the Cued Speech case". *Proceedings of INTERSPEECH'07*, Antwerp, Belgium, August 2007.

[7] V. Attina. "La Langue française Parlée Complétée : production et perception". Thèse de Doctorat en Sciences Cognitives, Institut National Polytechnique de Grenoble, France, 2005.

[8] G. Guibert. "Conception et évaluation d'un système de synthèse 3D de Langue française Parlée Complétée (LPC) à partir du texte". Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 2006.

[9] P. Gacon. "Analyse d'Images et Modèles de Forme pour la Détection et la Reconnaissance. Application au Visage en Multimédia". Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 2006.

[10] N. Eveno. "Segmentation des lèvres par un modèle déformable analytique". Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 2003.

[11] B. Rivet. "The bimodality of speech as a help to source separation". Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 2006.

[12] A. Caplier, L. Bonnaud, S. Malassiotis, M. G. Strintzis. "Comparison of 2D and 3D Analysis For Automated Cued Speech Gesture Recognition", *SPECOM*, St Petersburg, Russia, 2004.

[13] S. Stillitanno, A. Caplier. "Segmentation du contour intérieur des lèvres en combinant contours actifs et modèles paramètriques". *12ième journées d'études et d'échange Compression et Représentation des Signaux Audiovisuels*, Montpellier, 8-9 novembre 2007.

[14] M. Tribout, S. Vidal, D. Chêne, D. Beautemps. "Etude de l'interaction distante en mode LPC". Soumis à *ASSISTH*'07, 2007.

[15] Dossier de soumission du projet RNTS "Téléphonie à l'usage des malentendants – TELMA", 2005.

[16] ADIDA 38 - 28 chemin des Ecoutoux - Cedex 412 - 38330 Saint Nazaire Les Eymes. http://www.adida38.fr/

[17] J. Robert-Ribes. "Modèles d'intégration audiovisuelle de signaux linguistiques : de la perception humaine à la reconnaissance automatique des voyelles". Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 1991.

[18] Réseaux d'Excellence Européen Similar : www.similar.cc

[19] http://www.lis.inpg.fr/pages_perso/caplier/french/geste.html.fr/geste1.html.fr.html

[20] http://www.icp.inpg.fr/~savariaux

Segmentations, description et traitements d'images

[21] V. Pavlovic, R. Sharma, T. S. Huang. "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, n. 7, p. 677-695, 1997.

[22] K. G. Derpanis. *A review on vision-based hand gestures*, internal report, 2004. http://cvr.yorku.ca/members/gradstudents/kosta/publications/file_Gesture_review.pdf.

[23] S. Jayaram, S. Schmugge, M. C. Shin, L. V. Tsap. "Effect of Color space Transformation, the Illuminance Component, and Color Modeling on Skin Detection". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Washington, USA, vol 2, pp. 813-818, 2004.

[24] Y. Wu, T. S. Huang. "Hand modeling, analysis, and recognition for vision based human computer interaction". *IEEE Signal Processing Magazine*, 21, 51-60, 2001.

[25] D. Zhang, G. Lu. "Evaluation of MPEG-7 shape descriptors against other shape descriptors". *Multimedia Systems*, vol. 9, issue 1, 2003.

[26] F. Mokhtarian, A. K. Mackworth. "A theory of multiscale, curvature-based shape representation for planar curves". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 789–805, 1992.

[27] J. Martin. "Reconnaissance de Gestes en Vision par Ordinateur". Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 2000.

[28] S. Adam, J.-M. Ogier, C. Cariou, R. Mullot, J. Gardes, Y. Lecourtier. "Utilisation de la transformée de Fourier-Mellin pour la reconnaissance de formes multi-orientées et multi-échelles : application à l'analyse de documents techniques". *Traitement du Signal*, Vol. 18-1, 2001.

[29] J. Gardes, C. Cariou, J. Iviglia, J.-M. Ogier. Pattern recognition process. Brevet US 6694054.

[30] M. A. Rodrigues (Eds). *Invariants for pattern recognition and classification*. Series in machine perception and artificial intelligence, vol. 42, 2000.

[31] D. Blostein, Y.-B. Kwon (Eds). *Graphics recognition. Proceedings of the 4th International Workshop GREC 2001*, 2001.

[32] M.-K. Hu. "Visual pattern recognition by moment invariants". *IRE Transaction on Information Theory*, IT-8:pp.179-187, 1962.

[33] C. Schmid. "Image matching and retrieval based on local greyvalue invariants". Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 1996.

[34] J. L. Mundy, A. Zisserman. *Geometric Invariance in Computer Vision*. MIT press, 1992.

[35] P. Gros, L. Quan. *Projective Invariants for Vision*. Rapport Technique 90 IMAG – 15 LIFIA, Grenoble, 1992.

[36] C. Chesnaud. "Techniques statistiques de segmentation par contour actif et mise en œuvre rapide". Thèse de Doctorat, Université de droit, d'économie et des sciences d'Aix-Marseille, France, 2000.

[37] T. Morris, O. S. Elshehry. "Hand segmentation from live video". *International Conference on Imaging Science, Systems, and Technology, UMIST*, Manchester, UK, 2002.

[38] L. Barron, D. J. Fleet, S. S. Beauchemin, T. A. Burkitt. “Performance of optical flow techniques”. *International Journal of Computer Vision*, 12(1):43-77, 1994.

[39] M. Irani B. Rousso, S. Peleg. “Computing Occluding and Transparent Motions”. *International Journal of Computer Vision*, 12:1, 5-16, 1994.

[40] S. Wang and al. "Simplest operator based edge detection of binary image" *International Computer Congress*, Logistical Engineering University, Popular République of China, 28 - 30 May 2004.

[41] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham. "Active Shape Models: Their Training and Application". *Computer Vision and Image Understanding*, volume: 61, No. 1, page(s): 38-59, 1995.

[42] C. Kervrann, F. Heitz. "Apprentissage non supervise et suivi de modèles déformables dans une séquence d'images". *10ième congrès AFCET, Reconnaissance des formes et intelligence artificielle*, Rennes, France, 1996.

[43] L. Brèthes, P. Menezes, F. Lerasle, M. Briot. "Segmentation couleur et condensation pour le suivi et la reconnaissance de gestes humains". *Proceedings of 14eme Congres Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'04)*, volume Vol.2, Toulouse, France, 2004.

[44] B. Dorner, E. Hagen. "Towards an American Sign Language Interface". *Artificial Intelligence Revue, 8*(2-3): 235-253, 1994.

[45] S. Askar, Y. Kondratyuk, K. Elazouzi, P. Kauf, O. Schreer. "Vision-based skin-colour segmentation of moving hands for real-time applications". *Proceedings of the first European Conference on Visual Média Production*, London, UK, 2004.

[46] P. Gejgus, J. Placek, M. Sperka."Skin color segmentation method based on mixture of Gaussians and its application in Learning System for Finger Alphabet". *Proceedings of International Conference on Computer Systems and Technologies - CompSysTech’04*, Rousse, Bulgaria, June 17-18 2004.

[47] J. Canny. "A Computational Approach To Edge Detection". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8:679-714, 1986.

[48] S. D. Zenzo. "A note on the gradient of a multi-image" *Computer Vision, Graphics, and Image Processing*, 33:116--125, 1986.

[49] C. Garcia, M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423, 2004.

[50] S. Roux, F. Mamalet, Christophe Garcia. "Embedded Convolutional Face Finder". *Proceedings* of *ICME IEEE International Conference on Multimedia and Expo*, Toronto, Canada, 2006.

[51] S. Duffner, C. Garcia, "A Hierarchical Approach for Precise Facial Feature Detection", *Compression et Représentation des Signaux Audiovisuel*s *(CORESA)*, Rennes, France, 2005.

[52] D. T. Bravo, S. R. M. Pellegrino. "2D images calibration to facial features extraction". *Proceedings of VISAPP'07*, Barcelona, Spain, 2007.

[53] E.-J. Holden, R. Owens. "Recognising Moving Hand Shapes". 1*2th International Conference on Image Analysis and Processing (ICIAP'03)*, p. 14, Mantoue, Italy, September 17-19, 2003.

[54] R.W. Conners, C.A. Harlow. “A Theoretical Comparison of Texture Algorithms”. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 2, pp.204-222, 1980.

[55] J. Bezy-Wendling, M. Kretowski, Y. Rolland, W. Le Bidon. "Toward a better understanding of texture in vascular CT scan simulated images", *IEEE Transaction on Biomedical Engineering,* 48(1):120-4, 2001.

[56] S. Rital, H. Cherifi. "Détection de contours d'images couleur par hypergraphe de voisinage spatio colorimétrique". *Compression et Représentation des Signaux Audiovisuel*s *(CORESA)*, Lille, France, 2004.

Classification, SVM et combinaison de classifieur

[57] B. Boser, I. Guyon, and V. Vapnik. “A training algorithm for optimal margin classifiers”. *Proceedings off the Fifth Annual Workshop on Computational Learning Theory*, pp.144-152, Pittsburg, USA, 1992.

[58] C. Cortes and V. Vapnik. “Support-vector network”. *Machine Learning* 20, 273–297, 1995.

[59] R. Rifkin and A. Klautau. “In defense of one-vs-all classification”. *Journal of Machine Learning Research*, Vol.5, pp. 101-141, 2004.

[60] C.-W. Hsu and C.-J. Lin. “A comparison of methods for multi-class support vector machines”. *IEEE Transactions on Neural Networks*, Vol. 13, pp. 415-425, 2002.

[61] C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*. 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[62] UCI machine learning database repository, available at: http://www.ics.uci.edu/~mlearn/.

[63] S. Grikschat, J. A. Costa, A. O. Hero III, O. Michel. "Dual Rooted-Diffusions for Clustering and Classification on Manifolds". *Proceedings of ICASSP'06*, Toulouse, France, 2006.

[64] R.O. Duda, P.E. Hart, D.G. Stork. *Pattern Classification*. New York: Wiley-Interscience, 2001.

[65] S. Adam, M. Rigamonti, E. Clavier, É. Trupin, J.-M. Ogier, K. Tombre, J. Gardes. "DocMining: A Document Analysis System Builder". *Document Analysis Systems* pp. 472-483, 2004.

[66] J. Herault, C. Jutten. *Réseaux neuronnaux et traitement du signal*, 1994.

[67] J. Dréo, A. Pétrowski, P. Siarry, E. Taillard. *Métaheuristiques pour l'optimisation difficile*. Eyrolle, 2003.

[68] H. Lei, V. Govindaraju. "Half-against-half multi-class support vector machines". *Proceeding of the 6th International Workshop on Multiple Classifier*, Seaside, USA, 2005.

[69] M. Seeger. *Learning with labeled and unlabeled data*. Technical Report. University of Edinburgh, 2001

[70] A. Cornuéjols. "Une introduction aux SVM". *Bulletin #51 de l'AFIA (Association Française d'Intelligence Artificielle)*, 2002.

[71] J. Mercer. "Functions of positive and negative type and their connection with the theory of integral equations". *Philosophical Transaction of the Royal Society*, London, 1909.

[72] D. Casasent C. Yuan. "Face recognition with pose and illumination variations using new SVRDM support-vector machine". *Optical engineering*, vol. 43 - 8, pp. 1804-1813, 2004.

[73] G. Fumera, F. Roli. "Support Vector Machines with Embedded Reject Option". *SVM*, pp. 68-82, 2002.

[74] L. Xu, A. Kryzak, C. Y. Suen. "Method of combining multiple classifiers and their application to handwriting recognition". *IEEE Transactions on Systems, Man & Cybernetics*, 22(3), pp 418-435, 1992.

[75] J. Milgram, M. Cheriet, R. Sabourin. " “One Against One” or “One Against All”: Which One is Better for Handwriting Recognition with SVMs?". *Proceedings of Tenth International Workshop on Frontiers in Handwriting Recognition*, LA Baule, France, October 23-26 2006.

[76] T. Kikuchi, S. Abe. "Error Correcting Output Codes vs. Fuzzy Suport Vector Machines". *Proceedings of Artificial Neural Networks in Pattern Recognition, ANNPR*, pp. 192–196, Florencia, Italy, September 12-13, 2003.

[77] H. Laanaya, A. Martin, D. Aboutajdine, A. Khenchaf. " Classification des sédiments marins par fusion de classifieurs". *CMM’06 – Caractérisation du milieu marin*, Brest, France, October 16-19, 2006.

[78] Z. Xie, Q. Hu, D. Yu. "Fuzzy Output Support Vector Machines for Classification". *Lecture Notes in Computer Science*, Vol. 3612, 2005.

[79] C.-W. Hsu and C.-J. Lin. "A comparison of methods for multi-class support vector machines". *Technical report, IEEE Transaction on Neural Network*, vol. 13-2, 2002.

[80] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. "Gradient-Based Learning Applied to Document Recognition". *Proceedings of the IEEE*, vol. 86, #11, pp. 2278-2324, 1998.

[81] J. Kittler, M. Hatef, R. P. W. Duin, J. Matas. "On combining classifiers". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20-3, 1998.

[82] Y. Freund, R. E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of Computer and System Sciences* #55, 1997.

[83] I. Naseem. "Combining classifieurs using the Dempster-Shafer theory of evidence". Mémoire de Master of science, 2005.

[84] J. Franke and E. Mandler. "A comparison of two approaches for combining the votes of cooperating classifiers". *Proceedings of the 11th International Conference on Pattern Recognition*, vol. 2, pp. 611-614, The Hague, The Netherlands, 1992.

Fonctions de croyance

[85] A. P. Dempster. "A generalization of Bayesian inference". *Journal of the Royal Statistical Society, Series B*, 30(2) 205–247, 1968.

[86] G. Shafer. *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

[87] G. Shafer. "Comments on "Constructing a logic of plausible inference: a guide to Cox's Theorem", by Kevin S. Van Horn". *International Journal of Approximate Reasoning*, 2004.

[88] G. Shafer and P. P. Shenoy. Local Computation on hypertrees. Working paper #201, School of Business, University of Kansas, 1988. https://kuscholarworks.ku.edu/dspace/bitstream/1808/143/1/WP201.pdf.

[89] P.P. Shenoy and G. Shafer. "Axioms for Probability and Belief Function Propagation". *Uncertainty in Artificial Intelligence 4,* R.D. Shacter. T.S. Levitt, L.N. Kanal, and J.F. Lemmer, eds. North Holland: Elsevier Science, 1990.

[90] B. R. Cobb and P. Shenoy. "On the plausibility transformation method for translating belief function models to probability models". *International Journal of Approximate Reasoning*, vol. 41, pp. 314-330, 2006.

[91] B. R. Cobb, and P.P. Shenoy. "A Comparison of Methods for Transforming Belief Functions Models to Probability Models". *Lecture Notes in Artificial Intelligence 2711*, pp. 255 – 266, 2003.

[92] P. Smets and R. Kennes. "The transferable belief model", *Artificial Intelligence*, 66(2): 191–234, 1994.

[93] P. Smets. "Decision Making in the TBM: the Necessity of the Pignistic Transformation". *International Journal of Approximate Reasoning*, 38, 133-147, 2005.

[94] P. Smets. "Decision making in a context where uncertainty is represented by belief functions". *Belief Functions in Business Decisions*, R.P.Srivastava, Ed. Physica-Verlag, 2002.

[95] P. Smets. "No Dutch book can be built against the TBM even though update is not obtained by bayes rule of conditioning". *Workshop on probabilistic expert systems*, pp. 181–204, Societa Italiana di Statistica, Rome, Italy, 1993.

[96] P. Smets. "Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem". *International Journal of Approximate Reasoning*, 1991.

[97] P. Vannoorenberghe and P. Smets. "Partially Supervised Learning by a Credal EM Approach". *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, 2005.

[98] B. Ristic and P. Smets. "Kalman filters for tracking and classification and the transferable belief model". *IF04-0046, Proceedings of the International Conference on Information Fusion,* 2004.

[99] H. Xu, P. Smets. "Reasoning in Evidential Networks with Conditional Belief Functions". *International Journal of Approximate Reasoning*, 14, 155-185, 1996.

[100] T. Denoeux and A. Ben Yaghlane. "Approximating the Combination of Belief Functions using the Fast Moebius Transform in a coarsened frame". *International Journal of Approximate Reasoning*, Vol. 31, No. 1-2, 77-101, 2002.

[101] T. Denoeux. "Analysis of evidence-theoretic decision rules for pattern classification". *Pattern Recognition*, 30(7): 1095–1107, 1997.

[102] T. Denoeux. "A k-nearest neighbour classification rule based on Dempster-Shafer theory". *IEEE Transactions on Systems, Man and Cybernetics*, 25(5): 804–813, 1995.

[103] T. Denoeux. "Construction of predictive belief functions using a frequentist approach". *Proceedings of IPMU'2006*, Vol II, pp. 1412-1419, France, 2006.

[104] T. Denoeux. "A neural network classifier based on Dempster-Shafer theory". *IEEE Transactions on Systems, Man and Cybernetics A*, 30(2) :131–150, 2000.

[105] T. Denoeux. "Modeling vague beliefs using fuzzy-valued belief structures". *Fuzzy Sets and Systems*, 116(2):167-199, 2000.

[106] B. Quost. "Combinaison de classifieurs binaires dans le cadre de la théorie des fonctions de croyance". Thèse de doctorat, Université de Technologie de Compiègne, 2006.

[107] J. Kohlas and P. A. Monney. "A mathematical theory of hints: An approach to Dempster-Shafer theory of evidence". *Lecture Notes in Economics and Mathematical Systems*, No. 425, 1995.

[108] R. Haenni. "Uncover Dempster's Rule Where It Is Hidden". *Proceedings of the International Conference on Information Fusion*, Florence, Italy. 2006.

[109] D. Schmeidler. "Subjective probability and expected utility without additivity". *Econometrica*, 57, 571–587, 1989.

[110] R. Jeffery. "Conditioning, kinematics, and exchangeability". *Causation, Chance, and Credence*, Skyrms B. and Harper W.L. eds., Reidel, Dordrecht,,vol.1, 221-255, 1988.

[111] P. Teller. "Conditionalization and Observation". *Synthesis* 26:218-258, 1973.

[112] F. Janez, A. Appriou. "Theory of evidence and non-exhaustive frames of discernment: Plausibilities correction methods". *International Journal of Approximate Reasoning*, 18(1-2):1–19, 1998.

[113] L. S. Shapley. "A value for n-person games". *Contributions to the Theory of Games*, vol. 2, eds. H. Kuhn and A.W. Tucker. Princeton University Press, pp. 307-317, 1953.

[114] M. Daniel. "Probabilistic Transformations of Belief Functions". *Proceedings of ECSQARU,* pp.539-551, Barcelona, Spain, 2005.

[115] J. J. Sudano. "Pignistic Probability Transforms for Mixes of Low- and High-Probability Events", *Proceedings of the International Conference on Information Fusion*, Montreal, Canada, August 7-10, 2001.

[116] J. J. Sudano. "Inverse Pignistic Probability Transforms". *in Proceedings of the 5th International Conference on Information Fusion*, Annapolis, MD, USA, pp. 763-768, July 8-11, 2002.

[117] E. Ramasso, M. Rombaut and D. Pellerin. "Forward-Backward-Viterbi procedures in the Transferable Belief Model for state sequence analysis using belief functions". *ECSQARU, Lecture Notes in Computer Science*, Hammamet, Tunisia, 2007. Accepted.

[118] J.M. Nigro, M. Rombaut. "Idres: a rule-based system for driving situation recognition with uncertainty management". *Proceedings of the International Conference on Information Fusion* Vol. 4, december 2003.

[119] A.-S. Capelle. "Segmentation des images IRM multis-échos tridimensionnelles pour la détection des tumeurs cérébrales par la théorie de l'évidence". Thèse de doctorat, Université de Poitiers, 2003.

[120] A.-S. Capelle, O. Colot, C. Fernandez-Maloigne. "3D Segmentation of MR Brain Images into White Matter, Gray Matter and Cerebro-spinal Fluid by Means of Evidence Theory", *Lecture Notes of Computer Science Series, Artificial Intelligence in Medicine, pages* 112–116, M. Dojat, E. Keravnou, P. Barahona (Eds.), Springer-Verlag, 2003.

[121] A.-S. Capelle, O. Colot, C. Fernandez-Maloigne. "Evidential Clustering Algorithm for Color Quantization". *Accepted in the proceedings of IS&T/ CSIST 2005 Beijing International Conference on Imaging*, Beijing, China, 23-26 may 2005.

[122] G. Choquet. "Forme abstraite du théorème de capacitabilité". *Annales de l'institut Fourier*, 9, p. 83-89, 1959.

[123] http://www.poli.usp.br/p/fabio.cozman/Research/CredalSetsTutorial/Introduction/node7.html

[124] C. ´O Conaire, N. E. O'Connor, E. Cooke, A. F. Smeaton. "Multispectral Object Segmentation and Retrieval in Surveillance Video". *In proceedings of IEEE International Conference on Image Processing*, USA, 2006.

[125] R. T. Cox, "Probability, Frequency, and Reasonable Expectation". *American Journal Physique*, 14, 1-13, (1946).

[126] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

[127] H. B. Prosper. *Probabilistic and Statistical Inference*. Lecture in Department of Physics, Florida State University, Tallahassee, Florida 32306, USA

Inférence Graphique, HMM et langue des signes

[128] L. R. Rabiner. "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proc IEEE*, vol.77, pp. 257 – 285, 1989.

[129] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. "Factor graphs and the sum-product algorithm". *IEEE Transaction Information Theory*, 2001.

[130] S. M. Aji and R. J. McEliece, "The generalized distributive law". *IEEE Transaction Information Theory*, vol. 46, pp. 325-343, 2000.

[131] N. Liu, B. C. Lovell, P. J. Kootsookos, R. I. A. Davis. "Model structure selection and training algorithms for an HMM gesture recognition system". *Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR'04)*, IEEE Computer Society, Washington DC, USA, 2004, pp. 100-105.

[132] C. Vogler, D. Metaxas. "Parallel hidden Markov models for American sign language recognition". *Proceedings of the International Conference on Computer Vision*, Kerkyra, Greece, Vol. 1, 1999, pp. 116-122.

[133] O. Aran, C. Keskin, L. Akarun. "Sign Language Tutoring Tool". *Proceedings of EUSIPCO'05*, Antalya, Turkey, 2005.

[134] O. Aran, L. Akarun. "Recognizing Two Handed Gestures with Generative, Discriminative and Ensemble Methods via Fisher Kernels". *Proceedings of Lecture Notes in Computer Science: Multimedia Content Representation, Classification and Security International Workshop (MRCS'06)*, Istanbul, Turkey, September 11-13, 2006.

[135] O. Aran, I. Ari, A. Benoit, A. H. Carrillo, F. Fanard, P. Campr, L. Akarun, A. Caplier, M. Rombaut, & B. Sankur. "Sign Language Tutoring Tool". eNTERFACE 2006, The Summer Workshop on Multimodal Interfaces, Dubrovnik, Croatia, 2006.

[136] eNTERFACE06 ASL Database.
http://www.enterface.net/enterface06/docs/results/databases/eNTERFACE06_ASL.zip

[137] S. C. W. Ong, S. Ranganath. "Automatic Sign Language Analysis: A survey and the Future beyond Lexical Meaning". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 27, 6, 873-891, 2005.

[138] Q. Munib, M. Habeeba, B. Takruria, H. A. Al-Malika. "American sign language (ASL) recognition based on Hough transform and neural networks". *Expert Systems with Applications*. 32, 24-37, 2007.

[139] K. W. Ming, S. Ranganath. "Representations for Facial Expressions". *Proceedings of International Conference on Control Automation, Robotics and Vision*, vol. 2, pp. 716-721, 2002.

[140] U. M. Erdem, S. Sclaroff. "Automatic Detection of Relevant Head Gestures in American Sign Language Communication". *Proceedings of the International Conference on Pattern Recognition* vol. 1, pp. 460-463, 2002.

[141] S. Francke, L. Weysans. "Etude sur les modèles de Markov cachés et les applications à la bioinformatique." ENSTA, 2002.

[142] H. Yashwanth, H. Mahendrakar, and S. David, "Automatic speech recognition using audio visual cue". *Proceedings of the IEEE INDICON 2004*, First, pp. 166–169, 2004.

[143] X. Liu, Y. Zhao, X. Pi, L. Liang, and A. Nefian, "Audio-visual continuous speech recognition using a coupled hidden Markov model". *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pp. 213–216, 2002.

[144] T. Kristjansson, B. Frey, and T. Huang, "Event-coupled hidden Markov models". *Multimedia and Expo, IEEE International Conference*, vol. 1, 2000.

[145] M. Russell. "A segmental HMM for speech pattern modeling". *ICASSP'93*, pp. II-499–II-502, 1993.

[146] T. Artières, S. Marukatat, P. Gallinari. "Online Handwritten Shape Recognition Using Segmental Hidden Markov Models". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(2): 205-217, 2007.

[147] F. Perronnin. "Probabilistic model of face mapping applied to person recognition". Thèse de Doctorat, Ecole Polytechnique Fédérale de Lauzanne, Suisse, 2004.

[148] H. Guiol. Processus Aléatoires. Cours de l'ENSIMAG, 2005.

[149] R. E. Kalman, R. S. Bucy. "New Results in Linear Filtering and Prediction Theory". *Transaction ASME, Journal of Basic Engineering*, Series 83D (Mar. 1961), 95-108

[150] E. A. Wan, R. van der Merwe. "The Unscented Kalman Filter". *Kalman Filtering and Neural network*, 2001.

[151] A. Taylan Cemgil. "Tutorial on graphical Models and Monte-Carlo". http://www-sigproc.eng.cam.ac.uk/%7Eatc27/

[152] J. Bilmes, "What HMMs can do", Technical Report UWEETR-2002-2003, University of Washington, Dep. Of EE, 2002.

[153] G. Fang, W. Gao and D. Zhao. "Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models". *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 37, no. 1, pp. 1-9, 2007.

Divers

[154] L. Perrinet. "Comment déchiffrer le code impulsionnel de la Vision ? Etude du flux parallèle, asynchrone et épars dans le traitement visuel ultra-rapide". Thèse de Doctorat, Université Paul Sabatier, France, 2003.

[155] A. Benoit. "Le système visuel humain au secours de la vision par oridnateur". Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 2007.

[156] V. Vonikakis et al. "Adaptative document binarization: a human vision approach". *Proceedings of VISAPP*, Spain, 2006.

[157] J. Herault. "Retine et cortex visuel : formalisation et application au traitement des images". *Cerveaux et Machines*, Vincent Bloch Ed. Hermes, Paris, 1999.

[158] B. Russell. *Science et religion*. Gallimard. 1971.

[159] J. Dezert, F. Smarandache and M. Daniel. "The Generalized Pignistic Transformation", *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, 2004.

[160] J. Dezert and F. Smarandache. "An Introduction to the DSm Theory for the Combination of Paradoxical, Uncertain, and Imprecise Sources of Information", *Information & Security International Journal*, 2006.

[161] D. Dubois and H. Prade, *Possibility Theory: An Approach to the Computerized Processing of Uncertainty*. New York: Plenum Press, 1988.

[162] F. Cozman. "Credal Networks". *Artificial Intelligence* 120:199-233, 2000.

[163] Z. Hammal. "Facial Features Segmentation, Analysis and Recognition of Facial Expressions". Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 2006.

[164] C. C. Norkin, P. K. Levangie. *Joint structure and function.* (2nd ed.). Philadelphia: F.A. Davis, 1992.

[165] http://www-sante.ujf-grenoble.fr/SANTE/hand

[166] M. Royo. http://www.maxroyo.com/dessinmain01.htm

[167] http://alize.finances.gouv.fr/criph/dgi/08_grammaire/dactylo.htm

[168] http://bbouillon.free.fr/univ/ling/Fichiers/phon/api.htm

[169] A. Urankar. "Développement et optimisation d'une application de traitement d'images à des fins de téléphonie pour malentendants". Mémoire de DEA, 2006.

[170] P. Lemaire. "Evaluation des performances d'une application de reconnaissance gestuelle pour malentendants". Mémoire de Projet de fin d'étude. 2007.

[171] http://www.bioid.com/downloads/facedb/index.php

[172] J.-M. Park, C. G. Looney, H.-C. Chen. "Fast Connected Component Labeling Algorithm, Using a Divide and Conquer Technique". University of Alabama and University of Nevada, USA. http://cs.ua.edu/TechnicalReports/TR-2000-04.pdf, 2000.

[173] P.F Felzenszwald and D.P Huttenlocher. "Distance Transforms of Sampled Functions". Univ of Chicago and Cornell Univ., http://people.cs.uchicago.edu/~pff/dt/.

[174] A. Rosenfeld, J.L Pfatz. "Sequential Operations in digital Processing". *JACM*, 13, 471-494, 1966.

[175] Propos d'un professeur de Probabilités, attribués à Lippmann : "*Everybody believes in the exponential law of errors* [i.e. la loi normale] *: the experimenters, because they think it can be proved by mathematics; and the mathematicians, because they believe it has been established by obsservation*".

[176] D. Hofsadter. *Gödel, Escher, Bach: an Eternal Golden Braid.* Basic Books, 1979. Traduit en français par Jacqueline Henry et Robert French sous le titre *Gödel, Escher, Bach : les Brins d'une Guirlande Eternelle.*

PUBLICATIONS

Chapitres de livre

[L1] O. Aran, **T. Burger**, L. Akarun, A. Caplier. "Gestural interfaces for hearing-impaired communication". *Similar Dreams book*, Submitted.

Journaux internationaux

[J1] O. Aran, **T. Burger**, A. Caplier, L. Akarun. "Sequential Belief-Based Fusion of Manual and Non-Manual Signs". *Lecture Notes of Computer Sciences series*, Submitted.[14]

[J2] O. Aran, **T. Burger**, A. Caplier, L. Akarun. "A Belief-Based Sequential Fusion Approach for Fusing Manual and Non-Manual Signs". *Pattern Recognition Letters*, Submitted.

[J3] **T. Burger**, A. Caplier. "Partial Pignistic Transform". *International Journal of Approximate Reasoning*, Submitted.

[J4] **T. Burger**, A. Caplier, P. Perret. "Cued Speech Gesture Recognition: a First Prototype Based on Early Reduction". *International Journal of Image and Video Processus, Special Issue on Image & Video Processing for Disability*. Accepted for publication in 2007.

Conférences internationales

[C1] O. Aran, **T. Burger**, A. Caplier, L. Akarun. "Sequential Belief-Based Fusion of Manual and Non-Manual Signs". *Gesture Workshop*, Lisbon, Portugal, May 2007.

[C2] **T. Burger**, O. Aran, A. Urankar, L. Akarun, A. Caplier. "Cued Speech Hand Shape Recognition". *Proceeding of VISAPP'07*, Barcelona, Spain, March 2007.

[C3] **T. Burger**, O. Aran, A. Caplier. "Modeling hesitation and conflict: a belief-based approach for multi-class problems". *Proceeding of ICMLA'06*, Orlando, USA, December 2006.

[C4] **T. Burger**, A. Benoit, A. Caplier. "Extracting static hand gestures in dynamic context". *Proceeding of ICIP'06*, Atlanta, USA, October 2006.

[C5] **T. Burger**, A. Caplier, S. Mancini. "Cued Speech hand gestures recognition tool". *Proceeding of EUSIPCO'05*, Antalya, Turkey, September 2005.

[14] Il s'agit d'une version "journal" de la présentation au congrés [C1].

[C6] D. Beautemps, **T. Burger**, L. Girin. "Characterizing and classifying Cued Speech vowels from labial parameters". *Proceeding of ICSLP'04*, Jeju, South Korea, October 2004.

[C7] D. Beautemps, L. Girin, N. Aboutabit, G. Bailly, L. Besacier, G. Breton, **T. Burger**, A. Caplier, M.-A. Cathiard, D. Chêne, J. Clarke, F. Elisei, O. Govokhina, V.-B. Le, M. Marthouret, S. Mancini, Y. Mathieu, P. Perret, B. Rivet, P. Sacher, C. Savariaux, S. Schmerber, J.-F. Sérignat, M. Tribout, S. Vidal. "TELMA: Téléphony for Hearing-Impaired People. From models to user tests". *ASSISTH 2007*, Toulouse, 19-21 november 2007.

Autres

[A1] **T. Burger**. "Caractérisation Labiale et Classification des phonèmes de la Langue Française Parlée Complétée". Mémoire de DEA, 2004.

[A2] D. Beautemps, L. Girin, N. Aboutabit, G. Bailly, L. Besacier, G. Breton, **T. Burger**, A. Caplier, M.-A. Cathiard, D. Chêne, J. Clarke, F. Elisei, O. Govokhina, V.-B. Le, M. Marthouret, S. Mancini, Y. Mathieu, P. Perret, B. Rivet, P. Sacher, C. Savariaux, S. Schmerber, J.-F. Sérignat, M. Tribout, S. Vidal. "TELMA : Téléphonie à l'Usage des Malentendants. Des modèles aux tests d'usage". *ASSISTH 2007*, Toulouse, 19-21 novembre 2007. [15]

[15] Il s'agit de la version fraînçaise de la communication [C7].

APPENDICE A
LES FONCTION DE CROYANCE

Dans cet appendice, nous présentons les fonctions de croyance. Il s'agit d'un outil de quantification et de manipulation de l'information. Il est souvent opposé aux probabilités, et de temps en temps présenté comme une généralisation de celles-ci. Il existe de très nombreuses visions des fonctions de croyance. Ces visions ont souvent été construites en fonction de considérations épistémologiques plus larges ou afin de permettre la résolution d'une certaine classe de problèmes bien particulière. Ainsi, il est naturel que ces différentes visions ne soient pas toujours compatibles.

Avertissement : en tant que jeune chercheur n'ayant pas encore de quoi étayer des convictions scientifiques, ces "guerres de chapelles" sont très précieuses : elles donnent envie de changer de point de vue, elles poussent à remettre en cause l'état de l'art et elles sont le moteur de la curiosité. Cependant, dans certains cas, elles peuvent se révéler être un fardeau, et c'est partiellement ce que nous avons vécu : les travaux présentés dans ce document ont une finalité applicative incontestable. Il s'agit avant tout de résoudre un problème réel, la reconnaissance des gestes du LPC, et de trouver les méthodes permettant de mettre en œuvre la meilleure solution possible. Pour cela, il peut être nécessaire d'avoir recours à des outils très différents, dont les théories respectives ne sont pas toujours compatibles et cela peut conduire à certaines difficultés : difficulté de discussion avec des spécialistes, difficulté de justification d'une méthode, et même parfois, difficulté de publication. Face à cela, il faut se forger sa propre vision des choses et s'en servir comme argument de discussion à propos de notre objectif final à vocation applicative.

De nombreuses personnes ont des théories différentes à propos des fonctions de croyance, mais chacune de ces théories est suffisamment cohérente pour convaincre des néophytes tels que nous. Cela laisse la désagréable impression d'être toujours de l'avis du dernier qui a parlé. Ensuite, à nous d'en tirer une vision qui permet de faire le lien avec des axiomatiques différentes que nous n'avons pas la compétence de nier.

Pour cette raison, la présentation des fonctions de croyance que nous proposons résulte d'une expérience et d'une interprétation personnelles et n'engage que son auteur. Nous avons décidé de les présenter telles que nous les avons perçues durant ces travaux, et telles qu'il nous a paru judicieux de les utiliser. Il ne s'agit en aucun cas d'une théorie se voulant plus générale. Nous cherchons seulement à expliquer comment nous avons fait pour accorder plus d'importance aux points qui rapprochent des systèmes différents plutôt qu'à ceux qui les séparent. En ce sens, cette présentation est très subjective. Elle a néanmoins l'avantage de correspondre à l'état d'esprit de nos travaux.

Cet appendice est structuré comme suit : tout d'abord dans une première section, les éléments de base permettant de comprendre les fonctions de croyance sont présentés et illustrés au travers d'un exemple. Cette partie traite de notions qui sont communes à toutes les théories sur les fonctions de croyance. Nous nous contentons donc de les reprendre sans autre interprétation. Dans la section suivante, nous présentons deux modèles dominants en

soulignant les points auxquels nous sommes sensibles. D'abord le Modèle de Croyance Transférable de Smets (ou TBM, qui est une interprétation dominante, subjectiviste et indépendante des probabilités) ; nous détaillons les quatre points qui, selon nous, permettent à ce modèle de se démarquer des autres tout en enrichissant la base théorique des fonctions de croyance. Ensuite, nous comparons des travaux d'origines différentes, afin de montrer la parenté de pensée que nous ressentons entre le courant bayesien de l'inférence probabiliste et le modèle évidentiel de Shafer and Shenoy. Nous n'y voyons pas une parenté totale, mais aussi surprenant que cela puisse paraître, le TBM (qui est indépendant des probabilités) nous permet de la compléter.

Enfin, dans la troisième section, nous continuons la discussion sur la parenté entre fonctions de croyance (subjectives) et probabilité (subjectives), en nous basant sur un article où Shafer discute de ce sujet. Nous expliquerons pourquoi les arguments qu'il donne afin de marquer des différences nous ont permis de mieux comprendre ce qui les rapproche.

A.1 Présentation des fonctions de croyance

A.1.1 Introduction

Les fonctions de croyances ont été initialement formalisées par G. Shafer dans *A Mathematical Theory of Evidence* [86], à partir de travaux antérieurs de Dempster [85] sur l'étude des bornes inférieure et supérieure d'une famille de probabilités. Son objet est la quantification d'informations incertaines ou contradictoires en vue de leur fusion et d'une prise de décision. Depuis, la formalisation des fonctions de croyance est souvent dénommée **Théorie de Dempster-Shafer** (**DST**) ou, aussi étrange que cela puisse paraître, **théorie de l'évidence**... "A Mathematical Theory of Evidence" aurait pu être traduit soit par "Théorie Mathématique du Témoignage", afin de mettre en relief l'aspect incertain ou conflictuel des sources d'information, soit par "Théorie Mathématique de la Preuve", afin d'opposer le raisonnement sous-jacent à celui du pari, inhérent aux probabilités. En effet, le mot "evidence" en anglais, qui peut être considéré comme un synonyme de "hint", de "clue", "testimony", de "indication" ou de "proof", est un faux ami, et ne veut pas dire "évidence" en français. En effet, "évidence" se traduit par "obviousness", mais comme "à l'évidence" peut aussi se traduire par "evidently", on comprend la source de confusion. Et confusion il y a eu, puisque dans la communauté francophone des fonctions de croyance, on parle désormais de la "Théorie de l'évidence"... Mais, pire, l'utilisation de cette théorie francisée dans des articles internationaux a donné naissance à "The Evidence Theory" par le fruit d'une seconde traduction. Il n'en reste pas moins que l'on parle désormais de "evidential" pour designer en anglais ce qui a trait aux fonctions de croyance, et qu'il serait logique de le traduire par "évidentiel" dans le jargon français, mais l'utilisation du mot "crédal" plus adapté pour les francophones, est de rude concurrence ; on retrouve aussi celui-ci en langue anglaise à travers le mot "credal".

Depuis son avènement, de nombreuses autres interprétations des fonctions de croyance ont vu le jour, telles que la **Theory of Hints** [107] ou le **Modèle de Croyance Transférable** (**TBM** - **Transferable Belief Model**) [92], si bien qu'il n'existe plus une seule et unique théorie des fonctions de croyance. Dans cette section, nous présentons les aspects généraux et fondamentaux communs à toutes ces théories. Ceux-ci sont donc principalement issus de [86].

Notons qu'afin de faciliter la lecture, nous utiliserons souvent la notation suivante :

$$\sum(\text{expression}|\,\text{condition})$$

plutôt que la notation classique

$$\sum_{\text{condition}} \text{expression}$$

qui est peu lisible quand la condition de sommation a une formulation relativement complexe.

A.1.2 Notions générales

Soit $X=\{x_1,\ldots x_M\}$ un ensemble de variables et $\Omega_X = \{h_1, \ldots, h_N\}$ un ensemble de N hypothèses exhaustives et exclusives que peut satisfaire la "multivariable" X. Ω_X est le **cadre de discernement** de X ou tout simplement, le **cadre** de X. Soit 2^{Ω_X} l'ensemble de toutes les parties A de Ω_X, y compris l'ensemble vide :

$$2^{\Omega_X} = \{A / A \subseteq \Omega_X\}$$

2^{Ω_X} est appelé **powerset** de Ω_X. Soit m une **masse de croyance,** ou une **fonction de croyance** (FC en abrégé) sur 2^{Ω_X} qui représente la croyance que l'on peut placer sur les hypothèses de Ω_X :

$$m : \begin{cases} 2^{\Omega_X} \to [0,1] \\ A \mapsto m(A) \end{cases} \quad \text{avec} \quad \begin{cases} \sum(m(A)|\, A \subseteq \Omega) = 1 \\ m(\varnothing) = 0 \end{cases}$$

$m(A)$ représente la croyance que l'on place exactement en A, et non en une partie plus grande ou plus petite du cadre. Un **élément focal** est un élément de 2^{Ω_X} (ou de manière équivalente une partie du cadre Ω_X) pour lequel la croyance est non nulle. Une **FC consonante** est une FC dont les éléments sont emboités, c'est à dire qu'ils sont ordonnés par rapport à l'opérateur d'inclusion ($\subseteq$). Dans le cas contraire, une FC est dite **non consonante**. Si, pour chaque paire d'éléments focaux l'intersection est vide, la FC non consonante est dite **dissonante**. Si les éléments focaux sont tous des singletons, la FC dissonante est dite **Bayesienne**.

Soient m une FC sur Ω_X et X et Y des variables telles que $X \subseteq Y$. L'**Extension Vide** de m à Y, noté $m^{\uparrow Y}$ est définie et notée de la manière suivante :

$$m^{\uparrow Y}\left(A \times \Omega_{Y \setminus X}\right) = m(A) \qquad \forall A \subseteq 2^{\Omega_X}$$

L'extension vide permet simplement de définir une FC sur un support plus vaste (le cadre Ω_Y) en agrandissant chaque élément focal avec les éléments de Ω_Y qui ne sont pas dans le cadre original Ω_X.

La **marginalisation** (qui correspond à l'opération consistant à focaliser l'information sur un ensemble plus petit de variables) d'une FC m (définie sur Ω_Y) de Y sur X est définie de la manière suivante :

$$m^{\downarrow X}(A) = \sum\left(m(B) \,\middle|\, B = A \times \Omega_{Y \setminus X}\right) \qquad \forall A \subseteq 2^{\Omega_X}$$

La combinaison de N FC provenant de N sources indépendantes se calcule en utilisant la **règle de combinaison de Dempster**. C'est un opérateur N-aire associatif, commutatif et symétrique, défini de la manière suivante :

$$(\cap) : \overbrace{\mathcal{B}^{\Omega_{X_1}} \times \mathcal{B}^{\Omega_{X_2}} \times \ldots \times \mathcal{B}^{\Omega_{X_N}}}^{N} \to \mathcal{B}^{\Omega_X}$$
$$m_1 \;(\cap)\; m_2 \;(\cap)\ldots(\cap)\; m_N \quad \mapsto m_{(\cap)}$$

avec $\mathcal{B}^{\Omega_{Xi}}$ l'ensemble des FC définies sur Ω_{Xi} et Ω_X le résultat du **produit cylindrique** des ensembles Ω_{Xi} :

$$\Omega_X = \Omega_{X_1} \times \left[\Omega_{X_2} \setminus \left(\Omega_{X_1} \cap \Omega_{X_2}\right)\right] \times \ldots \times \left[\Omega_{X_{N-1}} \setminus \left(\bigcap_{i=1}^{N-1} \Omega_{X_i}\right)\right]$$

L'expression analytique du résultat d'une combinaison de Dempster est :

$$m_{(\cap)}(A) = \frac{1}{1-\mathcal{K}} \cdot \sum\left(\prod_{n=1}^{N} m_n{}^{\uparrow X}(A_n) \,\middle|\, \bigcap_{n=1}^{N} A_n = A\right) \qquad \forall A \subseteq 2^{\Omega_X}$$

où la constante de normalisation,

$$\mathcal{K} = \sum\left(\prod_{n=1}^{N} m_n{}^{\uparrow X}(A_n) \,\middle|\, \bigcap_{n=1}^{N} A_n = \varnothing\right)$$

quantifie l'incohérence entre les FC ainsi combinées.

Il y a de nombreuses autres opérations définies dans le formalisme des fonctions de croyance, mais comme cela est explicitement démontré dans [108], la plupart d'entre elles sont des combinaisons de Dempster cachées entre la FC sur laquelle l'opération est effectuée et une autre FC modélisant une méta-information permettant de définir l'opération en question. L'opération de **raffinement** est une telle opération. Elle est définie de la manière suivante :

Soient deux cadres Ω_1 et Ω_2, et R une application du powerset de Ω_1 vers le powerset de Ω_2, appelée raffinement de Ω_1 vers Ω_2, telle que :

- l'ensemble $\left\{\mathrm{R}(\{h\}), h \in \Omega_1\right\} \subseteq 2^{\Omega_2}$
- l'ensemble $\left\{\mathrm{R}(\{h\}), h \in \Omega_1\right\}$ est une partition de Ω_2
- $\forall A_1 \subset \Omega_1, \mathrm{R}(\{A_1\}) = \bigcup\left(\mathrm{R}(\{h\}) | h \in A_1\right)$

Pour ce qui est de la prise de décision, il s'agit toujours de prendre celle qui est la moins risquée, quitte à ne pas prendre une décision complète. L'idée est donc de ne prendre une décision que si l'on a la preuve que cela est la bonne chose à faire. Dans le cas contraire, il est possible de rester dans l'indécision en choisissant de ne pas éliminer certaines hypothèses. En pratique, il y a plusieurs stratégies différentes pour réaliser cela, aucune n'ayant la prépondérance sur les autres. Le choix de l'une d'entre elles dépend de la prudence de la décision à prendre. Chacune de ces stratégies correspond plus ou moins à choisir l'hypothèse qui maximise une structure de croyance particulière. Ainsi, le choix de la stratégie est équivalent à celui de la structure de croyance. En plus de la masse de croyance, déjà mentionnée, il y a 3 autres structures possibles :

La **fonction de crédibilité**, notée *Cred*. *Cred*(*A*) correspond à la croyance en toutes les hypothèses qui impliquent *A* :

$$Cred(A) = \sum\left(m(B) \left| \begin{array}{l} B \subseteq A \\ \\ B \neq \varnothing \end{array} \right. \right) \qquad \forall A \subseteq \Omega_x$$

La **fonction de plausibilité**, notée *Pl*, représente la croyance en toutes les hypothèses qui ne peuvent pas réfuter l'hypothèse considérée :

$$Pl(A) = \sum\left(m(B) \,\middle|\, (B \cap A) \neq \varnothing\right) \qquad \forall A \subseteq \Omega_x$$

Par construction, nous avons la relation suivante entre les fonctions de crédibilité et de plausibilité :

$$Pl(A) = Cred\left(\Omega_x\right) - Cred\left(\bar{A}\right)$$

Notons aussi que :

– Les fonctions de crédibilité et de plausibilité sont monotones : *Cred*(A) ≤ *Cred*(B) et *Pl*(A) ≤ *Pl*(B) pour tout A⊆B.

– *Cred*(A) ≤ *Pl*(A), quelque soit A.

– *Cred*(A) = *Pl*(A) si et seulement si les éléments focaux sont des singletons. Dans ce cas, il s'agit d'une **masse de croyance bayesienne**.

Finalement, il y a la **fonction de communalité**, *Q* qui n'a malheureusement pas d'interprétation facile. Elle est définie de la manière suivante :

$$Q(A) = \sum\left(m(B) \,\middle|\, A \subseteq B\right) \qquad \forall A \subseteq \Omega_x$$

Ces structures sont complètement équivalentes et il est possible d'effectuer les conversions de l'une à l'autre en utilisant les transformées de Möbius [100] :

$$\begin{cases} m(A) = \sum\left((-1)^{|B|-|A|} Q(B) \,\middle|\, A \subseteq B\right) \\ Q(A) = \sum\left((-1)^{|B|+1} Pl(B) \,\middle|\, \begin{matrix} B \subseteq A \\ B \neq \varnothing \end{matrix}\right) \\ m(A) = \sum\left((-1)^{|A|-|B|} Cred(B) \,\middle|\, B \subseteq A\right) \end{cases}$$

$\forall A \subseteq \Omega_X$ et $|\,.\,|$ étant la fonction cardinal .

Enfin, notons qu'une fonction d'appartenance (au sens de la théorie des ensembles flous) ayant un support fini peut être considérée comme une FC. Dans un tel cas, cette FC a les propriétés décrites dans [105].

A.1.3 Exemple

Dans ce paragraphe, nous donnons un exemple du fonctionnement de la combinaison de Dempster.

Considérons le cas élémentaire où il n'y a que deux FC à combiner définies sur une seule et même variable pour laquelle le cadre ne contient que 3 hypothèses. Chacune des deux FC correspond à une source partielle d'information (venant d'un capteur de mauvaise qualité par exemple) concernant la couleur d'un objet (Rouge (R), Vert (G), ou Bleu (B)). La combinaison des deux FC s'écrit de la manière suivante :

$$\begin{aligned} m_{\cap}(A) &= m_1(A) \textcircled{\cap} m_2(A) \\ &= \sum\left(\left(\prod_{n=1}^{2} m_n(A_n)\right) \,\middle|\, A_1 \cap A_2 = A\right) \\ &= \sum\left(m_1(A_1) \cdot m_2(A_2) \,\middle|\, A_1 \cap A_2 = A\right) \end{aligned}$$

La combinaison de Dempster est une somme sur une condition d'intersection d'un produit de tous les éléments possibles du powerset. Il est possible de représenter ce produit

$$m_1(A_1) \cdot m_2(A_2)$$

sur un tableau à double entrée (cf. Figure A-1), chacune d'entre elles correspondant à une des deux FC à combiner. Dans chaque cellule du tableau se trouve la valeur du produit des deux entrées.

Dès lors, la somme $\sum\left(\ldots \middle| A_1 \cap A_2 = A\right)$ correspond simplement à la sommation des éléments du tableau selon le schéma de couleur appliqué aux cellules : une cellule est associée à l'élément du powerset correspondant à l'intersection des éléments des deux entrées considérées. Dans le cas général de N FC, le principe est le même.

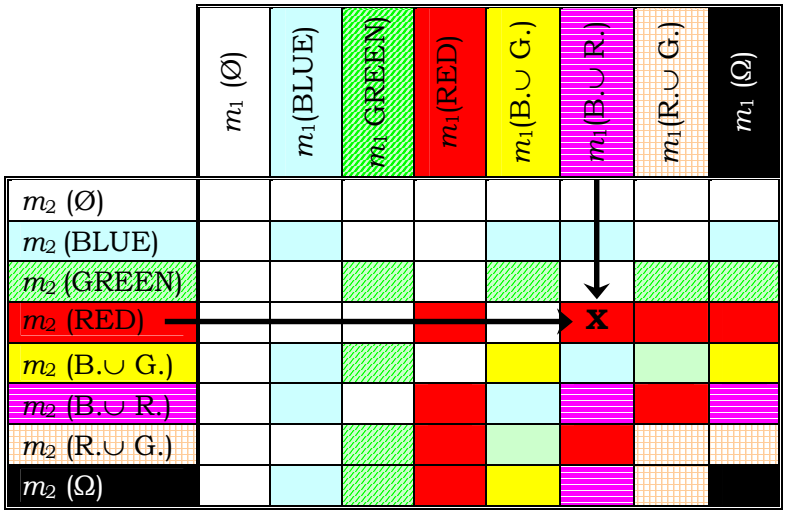


Figure A-1 : combinaison de Dempster de 2 sources.

A.2 Deux points de vue sur les fonctions de croyance

A.2.1 Le Modèle de Croyance Transférable (TBM)

Le TBM [92] de P. Smets est une des interprétations les plus populaires de la notion de croyance. Ce modèle possède un grand nombre d'avantages d'un point de vue applicatif. D'un point de vue théorique, les différences avec les autres modèles traitant des FC ne sont cependant pas suffisantes selon nous pour complètement détacher le TBM des théories sur les FC. Selon nous, parmi ces différences, quatre sont majeures :

Tout d'abord, Smets propose une définition des FC qui est complètement indépendante de l'existence d'un modèle probabiliste, alors qu'un tel modèle est habituellement présupposé. Ainsi, lorsqu'un objet de type probabiliste est nécessaire conjointement au TBM, celui-ci est souvent introduit d'une manière non-intuitive et artificielle, afin d'éviter tout lien entre probabilité et croyance [98]. Ceci est souvent une source de critique, mais cela a le gros avantage de donner une justification axiomatique à des objets qui n'ont pas d'origine stochastique ou statistique. Dans le même ordre d'idée, il s'agit d'un modèle subjectiviste, c'est-à-dire qu'une FC ne représente que la croyance de celui qui la définie. Notons qu'en pratique le TBM permet des modélisations objectivistes par rapport à des observations réelles [97], [98], [103]. A titre de comparaison, les modèles de Shafer ou de Shenoy sont des modèles eux aussi subjectifs, mais acceptant l'existence de probabilités.

Ensuite, Smets permet aux FC d'avoir une masse non nulle en l'élément vide. Cela signifie relâcher la contrainte $m(\varnothing)=0$ dans la définition des FC. De plus, il propose une interprétation particulière de cette masse de croyance : il s'agit de la croyance que l'on a que la bonne hypothèse ne soit pas dans le cadre. C'est

l'**Hypothèse du Monde Ouvert** (HMO), en opposition à l'**Hypothèse du Monde Fermé** (HMF) correspondant à la définition jusque là utilisée. Comme corollaire direct, il propose de dénormaliser la combinaison de Dempster et d'associer la constante de normalisation à $m_{(\cap)}(\varnothing)$, et baptise cette combinaison dénormalisée, **combinaison conjonctive**.

Troisièmement, il y a le **Theorème de Bayes Généralisé** (GBT), une généralisation aux FC du Théorème de Bayes en probabilités. Ce théorème n'est pas exclusif du TBM, dans le sens où il est applicable aux autres modèles de FC ; en effet, il est basé sur l'existence de deux autres concepts (la "**ballooning extension**" et la **combinaison disjonctive**, cf. [108], [96]) qui ne sont pas exclusifs du TBM, mais c'est au sein du TBM que celui-ci a été formalisé [96]. Le GBT stipule que :

$$Pl(Y=y \mid X=x) = 1 - \prod \left[1 - Pl\left(X=x \mid Y=y^i\right) \,\middle|\, y^i \in y \right]$$

où X et Y sont des multivariables, x et y sont des éléments des powersets correspondant à X et Y, et y^i est un élément du cadre de Y.

Finalement, le TBM est décomposé en deux niveaux ; le **Niveau Crédal** et le **Niveau Pignistique**. En pratique, le Niveau Crédal correspond au niveau de la définition et de la fusion des FC, d'une manière tout à fait classique par rapport aux autres modèles. Au contraire, le Niveau Pignistique est original. Il permet de prendre une décision en pariant sur l'hypothèse la plus probable, de manière similaire à ce qui est fait avec les probabilités, et de manière complètement contraire aux méthodes classiques de décision détaillées plus haut ("Pignistique" vient du latin "pignus" qui signifie "pari"). Au niveau Pignistique, on applique la **Transformée Pignistique** (PT) à la FC résultant du Niveau Crédal, afin de la convertir en une probabilité sur laquelle une décision est prise en choisissant l'hypothèse de probabilité maximum. La Transformée Pignistique est définie de la manière suivante :

$$\begin{array}{ll} \text{PT}: & \mathcal{B}^{\Omega} \rightarrow \mathcal{P}r^{\Omega} \\ & m(.) \mapsto \text{BetP}(.) \end{array} \quad \text{avec} \quad \text{BetP}(h) = \frac{1}{1-m(\varnothing)} \sum_{h \in A,\, A \subset \Omega} \frac{m(A)}{|A|} \quad \forall h \in \Omega$$

avec $\mathcal{P}r^{\Omega}$ désignant l'ensemble des fonctions de probabilité sur Ω, et $|A|$ désignant le cardinal de A. BetP est appelée **Probabilité Pignistique**. Comme la possibilité de prendre une décision en pariant sur l'hypothèse la plus crédible est d'un intérêt réel, de nombreuses autres propositions de transformées équivalentes ont vu le jour, telles que la **Transformée de Plausibilité** [90], la "**Proportional Probabilistic Transform**" [114], la "**Disjunctive Probabilistic Transform**" [114] ou la "**Cautious Probabilistic Transform**" [114], etc. Nous y ferons globalement référence en les dénommant **Transformées Probabilistes**. Notons tout de même que la Transformée de Plausibilité est apparue plus tôt de manière implicite dans [88], [89] sous la forme d'une prise de décision orientée pari, "**the most plausible configuration choice**".

En tout état de cause, il est démontré dans [88] que les considérations de normalisation ne sont que secondaires d'un point de vue calculatoire. De plus, il est montré dans [108] que l'HMO et l'HMF sont des considérations toutes théoriques et qu'il est possible de passer de l'une à l'autre par la prise en compte d'une FC particulière. En revanche, afin de simplifier les notations et de maintenir une certaine clarté, nous prenons pour convention d'utiliser dans le reste du document une notation simple mais sujette à discussion pour faire la différence entre l'HMO et l'HMF. Par rapport à l'HMO, dans l'HMF, l'hypothèse est faite que la vérité se trouve dans le cadre. Ainsi, une FC dans le monde fermé peut être vue comme la même information dans le monde ouvert conditionnée par Ω [119]. Ainsi, nous nous permettons de noter :

$$m_{HMO}\left(\,.\,\middle|\,\Omega\right) = m_{HMF}\left(\,.\,\right)$$

A.2.2 L'inférence graphique

Pour tout lecteur de *Factor Graphs and the Sum-Product Algorithm* [129], ou de *The Generalized Distributive Law* [130], les FC doivent sembler familières : d'une part les opérations définies sur les FC sont réellement proches de celles utilisées sur les **factor graphs**, et d'autre part, la distributivité de la combinaison de Dempster par rapport à la marginalisation (qui se déduit directement de la distributivité de la multiplication par rapport à l'addition) permet de définir un semi-anneau commutatif sur l'ensemble des FC, et ainsi de rester dans le cadre théorique de la distributivité généralisée.

Ces deux articles de référence sont des théories unificatrices sur l'inférence graphique (généralement centrée sur les probabilités et la théorie des codes), et en conséquence des remarques précédentes, il peut sembler naturel de mettre en place des méthodes d'inférence similaires sur les FC.

Chacune des manières de présenter l'inférence graphique dans ces deux articles ont leurs avantages : d'un point de vue pédagogique, les factor graphs de [129] sont plus intéressants, et la simplicité conceptuelle qui permet leur compréhension est à l'origine de leur popularité. Au contraire, les **Junction Trees** de [130] sont plus généraux, et sous ce formalisme, une plus grande classe de problèmes se résout de manière exacte (c'est-à-dire par un algorithme de calcul ayant la structure d'un graphe acyclique). Ceci ne se démontre pas de manière immédiate, mais aussi surprenant que cela puisse paraître, cela a été fait par Shafer et Shenoy dans [88] et dans [89], bien avant la publication du sum-product algorithm et de la distributivité généralisée.

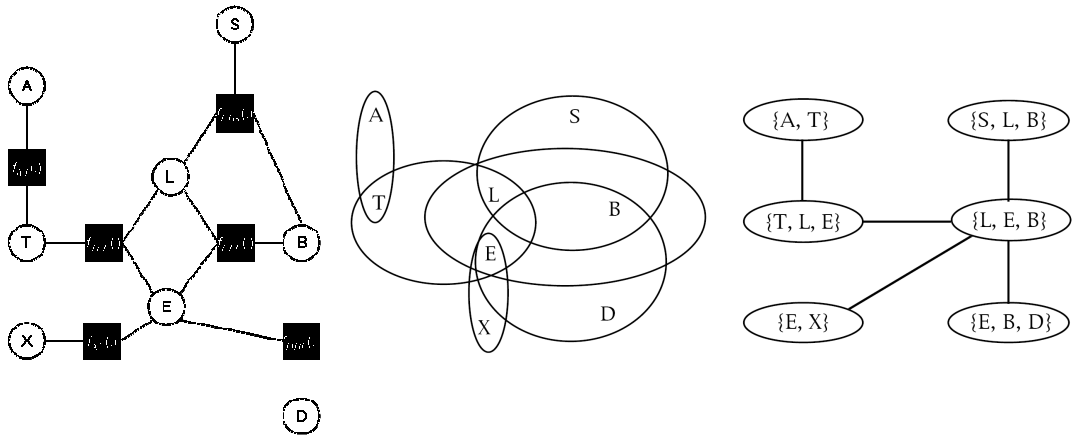
En effet, Shafer et Shenoy ont proposé en 1988 une axiomatique pour le calcul distribué sur des hyperarbres, afin de proposer des schémas d'inférence utilisant des FC ou des distributions de probabilité.

Les hypergraphes sont des objets un peu particuliers de la théorie des graphes pour lesquels les hyperarêtes peuvent connecter plus de deux hypersommets. Un hyperarbre est un hypergraphe pour lequel les hyperarêtes peuvent être ordonnées d'une manière particulière pour fournir une séquence de

construction que l'on pourrait définir d'acyclique. Une analogie est possible avec la factorisation d'une fonction multivariée en un produit de fonctions (de croyance ou de probabilité) multivariées permettant de calculer le plus simplement possible l'ensemble des marginales associées à la fonction. Dans cette analogie, les hyperarêtes correspondent à chacune des fonctions facteurs de la factorisation, et l'hyperarbre correspond à la fonction complète. Un hyperarbre est une séquence d'hyperarêtes ordonnée de telle sorte que chaque facteur multivarié associé puisse être calculé à partir des facteurs correspondant à des hyperarêtes placées en amont dans la séquence, pour permettre au final le calcul complet de la fonction. Dans une telle définition des hyperarbres, la notion d'hypersommet (que l'on peut comparer aux variables de la fonction multivariée à factoriser) est complètement insignifiante.

D'une manière générale, tout hypergraphe peut se ramener à un graphe biparti par la transformation suivante : les sommets de type 1 correspondent aux hypersommets, et les sommets de type 2 correspondent aux hyperarêtes. Les arêtes du graphe biparti permettent simplement d'indiquer quels sommets et quelles hyperarêtes sont adjacents. Bien sûr, le graphe biparti associé à un hypergraphe n'est pas forcément un arbre. En revanche, il se trouve aussi que le graphe biparti correspondant à un hyperarbre n'est pas non plus acyclique, ce qui au premier abord peut sembler gênant.

Il se trouve que les factor graphs sont des graphes bipartis qui représentent exactement le même type de relations entre variables et fonctions multivariées. En fait, les factor graphs et les hyperarbres de [88] et [89] sont deux représentations d'un même objet (Figure A-2). Les factor graphs sont cependant faciles à manipuler et à représenter puisqu'il s'agit de graphes classiques, et par conséquent, il est naturel d'effectuer des calculs sur leur structure. Mais ce n'est pas le cas des hyperarbres. En conséquence, Shafer et Shenoy proposent une transformation des hyperarbres en **arbre de Markov**. Ce qu'ils appellent un arbre de Markov n'est en fait rien de plus qu'un Junction Tree, d'un point de vue mathématique comme du point de vue de sa construction, comme cela est expliqué à la fois dans [88] et dans [89]. Ainsi, en résumé, un hyperarbre est équivalent à un Junction Tree (qui, comme son nom l'indique est un arbre), alors que la représentation d'un hyperarbre dans le formalisme des factor graph n'est pas nécessairement un factor tree (ou un factor graph couvrant et acyclique). D'un point de vue calculatoire, cette différence est mineure puisque plusieurs transformations de graphes permettent de ramener des problèmes cycliques à des problèmes acycliques. De plus ces transformations sont structurellement très proches du procédé permettant la conversion d'un hyperarbre en un Junction Tree. Cependant, d'un point de vue mathématique, le formalisme de la distributivité générale est plus puissant (puisqu'il permet de modéliser une plus grande classe de problème de manière acyclique) que celui des factor graphs (ce dernier formalisme compensant cela par une approche pédagogique beaucoup plus intéressante).



(a) Factor Graph (b) Hyperarbre (b) Junction Tree ou arbre de Markov

Figure A-2 : équivalence des représentations graphiques sur un exemple de [88].

Finalement, le travail de Shafer et Shenoy peut être vu comme très général par rapport à [129] et [130] au regard de son antériorité. D'autant plus que celui-ci a été d'abord effectué sur les FC avant d'être transposé aux probabilités (leur vision des FC est très liée aux probabilités, puisque selon eux, une FC est une distribution de probabilité sur l'ensemble des parties d'un cadre plutôt que sur le cadre lui-même). Cependant, ce travail n'est pas aussi général que [129] et [130] plus récents. En effet, ses conséquences sur les FC ne sont pas aussi importantes que leur contrepartie sur les probabilités, pour la simple raison qu'il manque pour les premières la notion de conditionnement naturellement utilisé avec les probabilités. Comme cela est discuté par Shafer [88] et par Smets [99], cette notion n'est pas indispensable d'un point de vue mathématique, mais elle est d'une aide précieuse d'un point de vue calculatoire car elle permet de réduire la complexité des problèmes. Finalement cette notion est amenée dans le cadre des fonctions de croyance par Smets dans [99], avec l'application du GBT (présenté précédemment) au cas des problèmes d'inférence évidentielle.

Par rapport à notre problématique d'étude des différentes théories des FC et de leur rapprochement avec les autres théories de manipulation de l'information, il est intéressant de noter que :

– la propagation de connaissance probabiliste sur un graphe a été initialement formalisée par Shafer et Shenoy.

– cette formalisation sur les probabilités découle de leurs travaux sur la propagation de FC.

– Leur travail, moins moderne que celui sur les factor graphs [129] et que celui sur la distributivité généralisée [130] est la clef de voûte de la passerelle qui existe entre ces deux travaux ([129] et [130]).

– Travailler sur les FC d'une manière similaire à celle des probabilités n'est pas une idée nouvelle.

– Le lien entre l'inférence probabiliste de [129], [130] et la vision probabiliste subjective des FC de Shenoy et Shafer n'est pas complet sans l'utilisation du GBT et donc sans l'existence du TBM, qui est une interprétation indépendante des probabilités.

Ainsi, nous voyons une forte complémentarité entre le TBM et d'autres modèles basés sur l'existence de probabilité. Notons que nous exploitons cette complémentarité à propos des méthodes de décision de Smets et de Shenoy à l'appendice B.2 (p. 280). Alors que celles-ci sont habituellement opposées dans la littérature, nous proposons une vision les rendant compatibles.

Evidemment, pour les modèles basés sur l'existence de fonctions de probabilité, la parenté entre FC et probabilité est naturelle. Cependant, celle-ci n'a *a priori* pas lieu d'être dans le cadre du TBM. Pourtant, les commentaires précédents nous permettent d'en ressentir une malgré tout. Nous proposons de décrire cette impression de parenté dans la section suivante.

A.3 Comparaison avec les Probabilités

La parenté entre les fonctions de croyance et les probabilités est intuitivement forte :

– Parenté historique. C'est par l'étude des probabilités que les fonctions de croyance sont apparues.

– Parenté de concept. Une masse de croyance bayesienne est équivalente (au support près de la fonction qui la définit) à une probabilité. La combinaison de Dempster de deux masses bayesiennes donne un résultat équivalent à la multiplication terme à terme de deux distributions discrètes de probabilité.

– Seconde parenté de concept. Certains modèles crédaux postulent de manière axiomatique l'existence de probabilités.

– Parenté d'utilisation. Les deux formalismes se manipulent de la même manière au travers de l'inférence graphique.

– Seconde parenté historique. L'inférence probabiliste de Shafer et Shenoy a pour origine l'inférence crédale qu'ils ont développée (au conditionnement près).

Cependant, cette parenté intuitive n'est pas suffisante pour avoir une vision dépourvue de contradiction. Nous proposons donc de nous baser sur une axiomatique préexistante (celle du théorème de Cox-Jaynes) afin de l'asseoir de manière plus convaincante. L'idée de voir ce théorème comme un moyen de relier nos deux rivages mathématiques vient de notre interprétation d'un article de Shafer [87] où celui-ci aborde plusieurs sujets : principalement, il propose de

clarifier un vocabulaire technique ambigu, et au passage, il enlève à Cox la primauté de son théorème en le replaçant dans un contexte historique fort de beaucoup de travaux européens similaires mais non traduits en anglais ; ensuite il argumente l'inintérêt du travail de Cox proprement dit.

Mais avant de nous plonger dans ce théorème, nous proposons un aperçu des dissensions que nous avons perçues entre les diverses théories probabilistes et crédales.

A.3.1 Introduction

De manière historique, une probabilité est envisagée par la notion de fréquence. D'un point de vue mathématique, il est pourtant difficile de dériver le formalisme probabiliste à partir de telles considérations : il faut définir une probabilité comme la limite vers quoi tend une suite d'événements. Ainsi, pour des raisons de simplicité, la théorie des probabilités est souvent construite d'une autre manière, en utilisant un point de vue plus algébrique dans lequel les variables aléatoires et les lois de probabilités sont définies. Bien sûr ces deux approches se rejoignent très fortement, et sont parfaitement compatibles par le biais de la **loi des grands nombres** et le **théorème central-limite**. Ainsi, quelque soit l'axiomatique de base qui est choisie, la théorie des probabilités est définie de manière unique. Du moins, c'était le cas au début. En effet, quand nous sommes entrés dans l'ère de la fusion de données, de la prise de décision automatique et de l'intelligence artificielle, les choses sont devenues moins claires :

– E.T. Jaynes a ainsi défini une axiomatique des méthodes d'apprentissage automatique dans [126]. Il se trouve que cette axiomatique était équivalente à celle de Cox [125], et correspondait à la définition d'objets mathématiques identiques à des probabilités, malgré un processus de construction qui n'a absolument rien à voir avec ceux que nous venons de décrire. Afin de tenir compte des remarques de Shafer dans [87], notons que de nombreux mathématiciens célèbres (tels que Poincaré, Borel, Bernstein, Kolmogorov, etc., cf. [87]) ont proposé des travaux similaires au début du XX^{ème} siècle un peu partout en Europe et en Russie. Cependant, ce que l'on appelle désormais le théorème de Cox-Jaynes reste le plus connu. Ce théorème a permis d'aboutir à une relation équivalente à celle du Théorème de Bayes, et c'est pour cela que ce dernier est maintenant la fondation d'une des théories majeures de l'apprentissage, à savoir l'**inférence bayesienne**. Dans un tel processus d'apprentissage, les objets de Cox-Jaynes permettent de définir une quantité de connaissance, et donc un degré d'incertitude. C'est pourquoi ces derniers sont souvent appelés **probabilités subjectives**, par opposition aux **probabilités objectives** qui correspondent à la définition classique des probabilités. Comme selon certains ce formalisme ne permet pas d'appréhender la totalité de l'absence de connaissance lors de la modélisation d'un problème (une absence de connaissance est souvent confondue à tort avec la connaissance d'une répartition équiprobable), des extensions de la théorie ont été proposées. Les

deux principaux courants se réclament tous les deux de l'appellation "crédale", mais selon leurs partisans, n'ont aucun point en commun.

– Le premier est le formalisme des fonctions de croyance que nous avons déjà présenté.

– Le second est souvent appelé inférence **quasi-bayesienne**. Cette théorie est basée sur les mêmes considérations que l'inférence bayesienne, mais les variables aléatoires y sont bornées par des lois de probabilité plutôt que d'être connues avec précision [162].

Finalement, notre bestiaire mathématique n'est plus aussi bien catégorisé qu'auparavant, et de nombreuses espèces hybrides semblent y cohabiter de manière incompatible, mais chacune a suffisamment de partisans pour avoir un intérêt : il y a d'abord les probabilités objectives telles que définies historiquement. Il y a ensuite les probabilités subjectives selon le théorème de Cox-Jaynes, dont l'intérêt croissant est lié à l'avènement de l'apprentissage automatique. Puis viennent les différentes interprétations des FC, que nous avons déjà cherchées à rassembler en les présentant de manière globale. Parmi celles-ci, certaines présupposent l'existence d'un modèle probabiliste subjectif, d'autres, l'existence d'un modèle non probabiliste subjectif, et encore d'autres ne sont pas gênées par le caractère objectif d'une connaissance. Enfin, il y a l'inférence quasi-bayesienne.

Toutes ces structures permettant de quantifier la connaissance peuvent bien sûr être interprétées mathématiquement par le concept de **capacité de Choquet** [122], mais néanmoins, leurs auteurs considèrent qu'elles ne correspondent pas pour autant d'un point de vue philosophique. Ainsi, dans [123], l'auteur qui se réclame de l'inférence quasi-bayesienne avertit ses lecteurs de la manière suivante : "*DST uses the mathematical structure [Choquet Capacity], but as far as I can see, the interpretation of the functions has nothing to do with probabilities nor decision theory. This is a matter of lively debate…*".

De même, Shafer critique dans [87] l'intérêt du travail de Cox. De son point de vue, les axiomes de Cox ne sont pas réellement axiomatiques, et sont même dérivés *a posteriori*. En outre, il explique qu'il ne faut pas faire de confusion sur le terme "axiome". Dans les mathématiques modernes, un axiome est une hypothèse que l'on pose, dans le but d'en explorer les conséquences. Cet axiome peut être intuitif ou au contraire complètement farfelu ; de même, il peut être vrai comme faux. Au contraire, le sens classique fait référence aux axiomes d'Euclide pour la géométrie. Il s'agit alors de propriétés indémontrables qui sont reconnues comme vraies. Ainsi, le fait que les FC ne remplissent pas les conditions du théorème de Cox-Jaynes ne suffit pas à discréditer l'intérêt des FC pour les problèmes classiquement modélisés par des probabilités subjectives, et particulièrement, l'inférence graphique.

Pendant ces 3 années de travaux, nous avons dû essayer d'avoir un point de vue moins sectaire sur ces différentes théories afin de pouvoir les utiliser toutes. Sur

le plan pratique, nous pensons que les analogies de la section précédente sont suffisantes. Sur le plan théorique, le travail reste à faire. Pour cela, nous proposons de nous désintéresser de la théorie quasi-bayesienne, et de nous focaliser sur les FC et les probabilités subjectives ou objectives. A ce propos, notons que des travaux existent déjà, tels que la définition de FC à partir d'échantillons statistiques [103] (en fort lien avec les probabilités objectives), ou à partir d'une généralisation de l'algorithme EM [97] indispensable en apprentissage (en fort lien avec les probabilités subjectives), de même que la description de réseaux de neurones évidentiels [104]. Pour notre part, nous nous focaliserons sur les liens qui peuvent être tirés du Théorème de Cox-Jaynes.

A.3.2 Le théorème de Cox-Jaynes

Commençons par adapter quelques définitions de [88] et de [89]. Etant donné un cadre Ω_x, nous appelons **tableau de scores** toute fonction à valeurs réelles dont le support est Ω_x. Si les valeurs du tableau sont non-négatives et toutes non-nulles, alors le tableau est appelé **potentiel**. Si l'intégrale (ou la somme) sur le potentiel vaut 1, alors le potentiel est dit **normé**. Ainsi, une FC (normalisée ou non) sur un cadre Ω_x peut être considérée comme un potentiel normé sur $2^{\Omega x}$.

Le **premier axiome** de Cox-Jaynes [127] suppose (ou stipule, suivant le sens que l'on donne au terme axiome) l'existence d'une **mesure de connaissance** $\boldsymbol{\pi}(.)$, qui modélise le crédit que l'on peut donner aux hypothèses du cadre d'une variable. Si une hypothèse X reçoit plus de crédit qu'une autre hypothèse Y dans un contexte particulier noté C, alors nous devons avoir $\boldsymbol{\pi}(X|C) > \boldsymbol{\pi}(Y|C)$, et l'image de la fonction $\boldsymbol{\pi}$ doit être complètement ordonnée. Ainsi, $\boldsymbol{\pi}$ est à valeurs dans un ensemble isomorphe à $\mathbb{R}$ ou à un sous-ensemble de $\mathbb{R}$, et $\boldsymbol{\pi}$ peut être considérée comme un tableau de scores sans perte de généralité.

Le **deuxième axiome** [127] suppose/stipule l'existence d'une relation fonctionnelle F entre les crédits donnés à une hypothèse et sa négation, ce qui s'exprime de la manière suivante : $\boldsymbol{\pi}(X|C) = F[\boldsymbol{\pi}(\overline{X}\,|C)]$.

Le **dernier axiome** [127] suppose/stipule que la connaissance de X et Y dépend de la connaissance de X et de la connaissance de Y sous la condition que X est vraie. Il existe donc une relation fonctionnelle G telle que :

$$\boldsymbol{\pi}(X,Y|C) = G[\boldsymbol{\pi}(X|C), \boldsymbol{\pi}(Y|X,C)].$$

Comme il n'y a pas de contrainte sur la positivité de $\boldsymbol{\pi}(.)$, ni de valeur de référence telles que 0 ou 1, si nous écartons le cas dégénéré d'un tableau à valeurs toutes nulles, alors il est clair que $\boldsymbol{\pi}(.)$ peut être considérée comme un potentiel sans perte de généralité : quelque soit la méthode pour rééchelonner $\boldsymbol{\pi}(.)$ sur des valeurs positives, cela est transparent du point de vue de la connaissance encodée dans la mesure de connaissance.

Enfin, si nous définissons $\mathbf{\Pi}(.) = k \cdot \boldsymbol{\pi}(.)$ avec $k \in \mathbb{R}^+$ de telle sorte que $\mathbf{\Pi}(.)$ soit un potentiel normé, et si la fonction G(.) est la fonction produit, alors nous avons $\mathbf{\Pi}(X,Y \mid C) = \mathbf{\Pi}(X \mid C) \cdot \mathbf{\Pi}(Y \mid X,C)$, ce qui conduit à :

$$\mathbf{\Pi}(X|Y,C) = \mathbf{\Pi}(Y|X,C) \cdot \frac{\mathbf{\Pi}(X|C)}{\mathbf{\Pi}(Y|C)}$$

qui se trouve être une relation équivalente au Théorème de Bayes quand $\mathbf{\Pi}(.)$ est une fonction de probabilité. Ainsi, une mesure de connaissance normée semble équivalente à une fonction de probabilité. C'est pour cela que la mesure de Cox-Jaynes est appelée **probabilité subjective**.

Soit *x* une variable à propos d'un phénomène réel que l'on modélise. Si de nombreuses observations du phénomène sont disponibles, il est alors naturel de définir notre connaissance (la distribution de $\boldsymbol{\pi}$ sur les différentes hypothèses X de *x*) à partir des observations. Dans le cas de l'apprentissage automatique, ce bon sens est même devenu un paradigme (celui-ci est d'ailleurs de temps en temps critiqué, comme dans [158], mais nous n'en viendrons pas là). Ainsi, l'utilisation d'outils statistiques pour inférer des distributions de probabilités est naturelle, et il est légitime d'utiliser cette distribution pour définir une mesure de connaissance, ou pour donner du crédit à une série d'hypothèses. Cela est même la meilleure méthode à utiliser quand il s'agit de tenir compte de cette connaissance pour définir une stratégie gagnante sur le long terme, comme en théorie des jeux. D'un point de vue théorique, cela signifie que naturellement, nous utilisons une distribution de probabilité objective pour définir et quantifier une mesure de connaissance subjective $\mathbf{\Pi}(.)$. Cela est parfaitement acceptable puisqu'une probabilité objective satisfait de manière triviale les axiomes de Cox-Jaynes (les probabilités objectives sont donc un cas particulier des probabilités subjectives : leur origine doit en plus être stochastique ou fréquentielle pour que celles-ci deviennent objectives).

Au contraire, quand il n'y a pas de connaissance statistique antérieure à la définition de la probabilité subjective, il est impossible de définir à partir de celle-ci une probabilité objective avec les capacités de généralisation que celle-ci est censée avoir une fois confrontée à la diversité de la réalité du problème qu'elle participe à modéliser. En pratique, cela signifie qu'un ensemble d'apprentissage trop petit n'est pas représentatif et donc qu'il n'est pas fiable. Bien sûr, rien n'interdit sa définition et son utilisation. Il est tout à fait autorisé d'utiliser des connaissances qui n'ont aucune application réelle pour résoudre un problème d'inférence, mais dès lors, la solution ne doit pas prétendre à modéliser le réel. Effectuer l'opération contraire équivaut à définir une probabilité objective à partir d'une connaissance purement subjective mais cela n'est pas rigoureux. Ce point est vivement critiqué dans [145], et pour éviter cela, il était courant, à l'époque des prémisses de l'apprentissage automatique, de prendre quelques précautions comme :

– Lister les principales variables cachées du modèle qui peuvent avoir une influence non contrôlée, et par là même diminuer la qualité de l'apprentissage ;

– Lister l'ensemble des informations ***a priori*** du modèle ;

– Vérifier la validité statistique des corpus (intervalles de confiance, tests d'hypothèse, test du χ^2, etc.) ;

Ici, le terme "*a priori*" est de grande importance, puisqu'il indique clairement la nature subjective et sujette à discussion de l'information. Comme actuellement l'efficacité des méthodes d'apprentissage automatique est montrée en permanence dans l'état de l'art, ces précautions sont parfois oubliées et l'utilisation de connaissances subjectives apparaît dans des systèmes bayesiens prétendant fournir une information probabiliste au sens statistique du terme. L'utilisation de probabilités subjectives ne peut pas être critiquée d'un point de vue théorique, mais d'un point de vue expérimental, les résultats obtenus n'ont pas de valeur par rapport à la réalité des phénomènes observés. En effet, faire un tel raccourci est équivalent à réaliser un grand nombre d'expériences de pensées et à imaginer vers quoi converge la répartition de ces expériences fantômes.

De surcroît, la confusion entre probabilités objectives et subjectives peut avoir une conséquence encore plus déconcertante qui sert d'argument aux tenants des FC : lorsqu'aucune information n'est connue sur un processus pour lequel il faut prendre une décision, autant prendre cette décision au hasard. La justification de cela est que dans un tel cas, toute permutation au sein de l'ensemble des hypothèses est transparente du point de vue du preneur de décision (mais exclusivement de son point de vue). Ainsi, quelque soit la stratégie de choix, celle-ci a autant de chance d'être efficace que les autres. Il est naturel d'associer une équiprobabilité objective à l'ensemble des stratégies de choix. Il est aussi possible d'associer une probabilité subjective équiprobable à l'ensemble des hypothèses du problème puisque la finesse du processus de décision n'en sera pas altérée, même si cette équiprobabilité n'est pas vraie objectivement. En revanche, il est complètement faux de dire directement que l'absence de connaissance est objectivement équivalente à l'équiprobabilité des hypothèses, ce qui est pourtant très régulièrement le cas. Cela est souvent repéré par les partisans des FC qui en déduisent que les probabilités ne permettent pas une modélisation fine de l'incapacité à prendre une décision : qu'il s'agisse d'une équiprobabilité ou d'une absence de connaissance, cela est modélisé de la même manière, alors que les FC permettent de faire la différence. D'après nous, il n'en est rien, cependant, il est vrai que la confusion a plus facilement lieu dans le formalisme probabiliste. Ainsi, [98] décrit une utilisation du filtre de Kalman dont la décision n'est pas adaptée et propose une alternative dans le cadre du TBM. Cette alternative rétablit une juste décision. Les auteurs assurent que cela est dû au TBM et que seul celui-ci permet de prendre une décision juste : cet exemple sert en priorité d'argument face aux méthodes probabilistes. Nous pensons qu'une telle décision "juste" est aussi possible dans

un cadre probabiliste. L'intérêt du TBM est seulement de forcer ou d'imposer une séparation entre les points de vue subjectif et objectif par la séparation des niveaux crédal et pignistique. Dans la formulation probabiliste du problème donnée par les auteurs, celle-ci est en effet inexistante, et cela est à la source du résultat erroné, qu'ils corrigent ensuite avec l'utilisation du TBM.

En conclusion, nous ne rejetons en aucun cas le paradigme bayesien et l'apprentissage automatique, et nous le considérons comme parfaitement efficace même quand les limites invoquées plus haut ne sont pas respectées, mais nous maintenons que ne pas avoir une compréhension approfondie de l'origine de la connaissance manipulée est le meilleur moyen de commettre des erreurs d'interprétation. De plus une telle erreur d'interprétation peut servir d'argument à un adepte des FC afin de critiquer les méthodes bayesiennes.

Maintenant, nous allons nous intéresser aux points communs entre FC subjectives et probabilités subjectives. Mais avant cela, nous nous permettons d'émettre une mise en garde :

Il ne faut pas considérer ces notions comme plus générales qu'elles ne le sont. Ainsi, les définitions que nous avons utilisées ne doivent pas revêtir un caractère axiomatique à nos yeux, même si cela est très tentant. Il est en effet difficile de faire la part des choses quand il s'agit de la quantification de la connaissance, car pour cela, nous sommes forcés de manipuler des structures d'un niveau supérieur de connaissance. Concrètement, il est possible de définir l'opération de "normation" des potentiels d'une manière complètement différente. Par exemple, si nous fixons à 1 le crédit donné à une hypothèse certaine et la valeur 0 à une hypothèse impossible, alors, nous quittons le formalisme des probabilités pour celui des **possibilités** [161]. Ce formalisme existe bel et bien, même si nous ne le connaissons pas en détail. C'est pour cela qu'il se trouve en dehors du champ de cette discussion. Malgré tout, nous sommes persuadés qu'il est possible de l'utiliser pour répondre à des problèmes pour lesquels nous penserions au prime abord que les formalismes probabiliste ou crédal sont uniquement valables. Cet exemple permet d'illustrer dans quelle mesure la définition d'axiomes pour la manipulation de connaissance est délicate car elle semble toujours guidée par des pré-requis de plus haut niveau. Cet argument est très proche de celui qu'utilise Shafer face à l'édifice de Cox : il ne faut pas se tromper sur le sens que l'on donne au mot axiome ; il peut s'agir soit d'une hypothèse dont on explore les conséquences, soit d'une primitive indémontrable. Dans le cas de la manipulation de la connaissance, la séparation devient obscure, pour la simple raison que la définition de la connaissance est elle-même une connaissance. C'est ce que Hofstadter appelle **autoréférence**, un phénomène qui se trouve au cœur de la plupart des processus cognitifs. Dans [176], il propose de l'expliquer en termes mathématiques en se basant sur le théorème d'incomplétude de Gödel.

A.3.3 Théorème de Cox-Jaynes et FC

Intéressons-nous à la possibilité que les FC satisfassent les axiomes de Cox-Jaynes, même si cela est clairement rejeté par [87].

Le premier axiome suppose/stipule que (1) une mesure de connaissance est une fonction à valeurs dans un espace strictement ordonné et que (2) cet ordre est positivement corrélé au crédit que l'on peut placer dans un ensemble d'hypothèses. C'est le cas par définition pour les FC, prises comme une application du powerset dans [0, 1]. Cependant, si l'on interprète une fonction de croyance comme une structure complète et globale qui permet d'attribuer une borne supérieure (la plausibilité) et une borne inférieure (la masse de croyance) à une hypothèse, ce n'est plus le cas. Ainsi, quand on se rapproche de la vision de Dempster, le premier axiome n'est pas satisfait. Dès lors, si l'on accepte cette seconde interprétation, il ne sert à rien de vérifier les deux autres axiomes. En revanche, si l'on conserve la première, cela a du sens.

Le deuxième axiome suppose/stipule l'existence d'une relation entre le crédit placé en X et celui placé en $\overline{X}$. Une telle relation existe dans le cas des FC :

$$Pl(A) = Cred(\Omega_x) - Cred(\bar{A}) \quad \forall A \subseteq \Omega_x$$

Cette expression n'est pas directement une relation fonctionnelle entre une hypothèse et sa contraposée pour deux raisons :

– Il est fait référence à la crédibilité et à la plausibilité et non exclusivement à la masse de croyance.

– Il est fait référence à la connaissance que l'on a du cadre.

Le premier point n'est pas gênant pour la simple raison que ces structures sont en correspondance directe et qu'il est possible de passer de la crédibilité à la plausibilité par les transformées de Möbius. Concernant le second point, il est possible de le réfuter de trois manières : (1) $Cred(\Omega_x) = 1$ dans l'HMF et nous avons vu que le passage l'HMF à l'HMO n'est finalement pas déterminant ; (2) $Cred(\Omega_x)$ est avant tout une constante de normalisation, et nous avons déjà discuté de la possibilité de repousser cette étape de normalisation à la fin du processus calculatoire ; (3) il est possible de voir une analogie forte entre la notion de contexte introduite par Ω_x et celle introduite par le contexte C nécessaire à la définition d'une probabilité subjective $\boldsymbol{\pi}(.\,|\,C)$.

Le troisième axiome suppose/stipule l'existence d'une relation fonctionnelle du type de celle du Théorème de Bayes. Bien sûr le GBT décrit précédemment est là dans cette optique. Mais il y a quelque chose d'encore plus intéressant dans le GBT. Si les éléments focaux de la fonction de croyance et de la croyance *a priori* sont tous deux des singletons (il s'agit alors de masses bayesiennes), alors le GBT est équivalent à :

$$Cred(Y \mid X) = \frac{Cred(X \mid Y) \cdot Cred_{\text{Prior}}(Y)}{\sum_i Cred(X \mid Y = Y^i) \cdot Cred_{\text{Prior}}(Y = Y^i)} = Cred(X \mid Y) \cdot \frac{Cred_{\text{Prior}}(Y)}{Cred(X)}$$

où l'indice Prior indique qu'il s'agit de connaissance *a priori*. Cette relation correspond au Théorème de Bayes (avec en plus une décomposition par rapport aux différentes hypothèses de la croyance *a priori* au dénominateur). Cela signifie qu'en plus de satisfaire le troisième axiome de Cox-Jaynes, le GBT généralise aussi le Théorème de Bayes (ce qui peut sembler évident d'après le nom que lui a donné son auteur, mais qui donne aussi un sens à la continuité de pensée que nous pouvons voir entre FC et probabilités).

A.3.4 Conclusion

Finalement, la question n'est pas de savoir si les fonctions de croyance satisfont le Théorème de Cox-Jaynes : Shafer a déjà conclu que non, et nous ne nous permettons pas de le contredire. Cependant, à notre manière, nous nous sommes arrangés pour que les FC concordent avec lui. La question est donc de savoir où nous avons fait une erreur. Celle-ci se cache dans l'interprétation de la connaissance que nous manipulons : afin de montrer que les FC satisfont le théorème, nous nous sommes complètement affranchis de la signification d'une information ou d'une connaissance que peut contenir une FC pour ne la considérer que sous son aspect "fonction mathématique". Finalement, nous avons montré qu'en termes calculatoires, les FC s'utilisent comme des probabilités. Ainsi une machine qui manipule des probabilités peut théoriquement aussi manipuler des FC. Cela n'est pas bien intéressant, et nous imaginions déjà ce résultat à partir des considérations que nous avons établies sur l'inférence graphique.

Formellement, les probabilités sont définies sur un cadre et les FC sur le powerset de ce cadre : les supports de ces deux types de fonction sont différents. Cependant, comme un cadre est un sous-ensemble de son propre powerset, il est possible d'immerger celui-là dans celui-ci et de définir une probabilité subjective sur le powerset de son cadre en attribuant simplement une valeur nulle aux éléments du powerset qui n'appartiennent pas au cadre (et par là même justifier l'appellation de masse de croyance bayesienne). C'est seulement ainsi que d'un point de vue calculatoire, il n'y a plus de problème à considérer les probabilités subjectives comme un cas particulier des FC (comme cela est d'ailleurs indiqué dans [88] et [89]). La combinaison de Dempster et la marginalisation se trouvent alors simplement correspondre à une multiplication terme à terme et à une sommation.

Dès lors, quel peut bien être l'intérêt de toutes les transformées probabilistes de la littérature et des transformées crédales que nous avons introduites à la section V.3 (p. 176) ? En effet, si fonctions de probabilité et fonctions de croyance ne sont qu'un même objet comment se fait-il qu'il y ait besoin d'outils pour effectuer des conversions entre les deux ? En fait, d'un point de vue calculatoire, les deux sont identiques, mais il faut se rappeler que pour pouvoir les rapprocher, nous leur avons ôté leur signification pour ne plus considérer que leur aspect "numérique".

Ainsi, quand une probabilité subjective est énoncée, celle-ci représente la conversion d'une certaine connaissance avec un certain mode de pensée impliquant qu'il n'est entre autre pas possible de donner du crédit à des unions d'hypothèses. Ainsi, il est une erreur de vouloir par la suite les manipuler comme si cela avait été autorisé dans la modélisation de départ. Les transformées crédales et probabilistes sont un moyen d'adapter cette modélisation de manière à ne pas faire l'amalgame (et ainsi éviter des fautes de raisonnement similaires à celles que l'on fait en prenant une probabilité subjective pour objective).

Si nous rendons leur sens (leur contenu informationnel) à ces objets, alors, les FC ne satisfont plus les axiomes de Cox-Jaynes. Dès lors que penser ? Que les axiomes sont justes et que la théorie de l'évidence n'a pas de raison d'être, ou que les axiomes sont faux ? Shafer lève la contradiction en expliquant que cela dépend du sens que l'on donne au mot axiome. Nous pensons qu'il est possible d'aller plus loin. Dans la mesure où l'on traite de la connaissance, c'est-à-dire d'un processus auto-référant, ce choix à faire sur le sens du mot "axiome" n'est plus possible : il y a ambiguïté entre les deux sens. Ainsi, nous imaginons qu'il n'est pas possible d'axiomatiser la quantification de la connaissance. Pour reprendre les termes de Hofstadter [176] à propos de Gödel, l'affirmation d'un tel axiome est une proposition indécidable.

Dès lors nous pensons que la meilleure chose à faire est de distinguer la manipulation de l'information de son contenu, telle que nous l'avons fait pour vérifier la pertinence des axiomes de Cox-Jaynes auprès des FC. En faisant cela, nous acceptons deux choses :

– Il n'est pas possible d'émettre des axiomes sans se détacher de l'information contenue dans la mesure de la connaissance. En conséquence, l'information elle-même n'est pas axiomatisée, et il est nécessaire de vérifier à chaque fois la pertinence des présupposés que nous manipulons. C'est ce que nous avons illustré sur la confusion entre probabilités subjectives et objectives dans les cas d'absence de connaissance suffisante.

– Une fois détachée de leur sens, les structures de manipulation de l'information sont plus ou moins équivalentes. Il est possible de les axiomatiser de différentes manières, toutes étant satisfaisantes : il y a les axiomes de Cox-Jaynes, ceux de Shenoy-Shafer, ceux utilisés dans les factor graphs, ceux basés sur la structure d'un semi-anneau commutatif, ou encore ceux de Smets, ...).

Finalement, pour quelqu'un qui est obligé de manipuler des connaissances au travers de différents systèmes théoriques, nous conseillerions de détacher la manipulation des structures codant l'information de l'information elle-même. Bien sur, cela oblige à régulièrement justifier toute manipulation. Cependant, cela permet d'éviter de nombreuses erreurs, tout en permettant d'effectuer avec rigueur les conversions nécessaires au moment venu, et par là, permettre de répondre à d'éventuelles critiques théoriques.

APPENDICE B
DEMONSTRATIONS A PROPOS DE LA PPT

Dans ce chapitre, nous apportons des compléments à la PPT et nous la justifions de manière théorique, afin d'apporter du crédit à la justification expérimentale que nous proposons à la section VI.2 (p. 207). Une connaissance élémentaire des fonctions de croyance est nécessaire à sa compréhension (cf. appendice A p. 255). Dans une première section, nous justifions que selon nous, aucune solution satisfaisante n'existe dans l'état de l'art. Ensuite, nous justifions le fait que la transformée Pignistique est un bon départ à l'élaboration d'une solution satisfaisante. Dans la 3ème section, nous précisons le schéma de partage de la croyance que nous avons donné au paragraphe VI.2.1 (p. 201). Cela revient à justifier l'expression de la PPT. Ensuite, nous effectuons une comparaison avec la transformée qui est structurellement la plus proche dans l'état de l'art : celle permettant de calculer la valeur de Shapley. La section suivante permet d'aborder les aspects calculatoires de la PPT. Enfin, nous discutons des futurs développements possibles autour de la PPT.

B.1 Etat de l'art

Notre objectif est d'aider le preneur de décision à trouver un équilibre entre le risque d'un pari et l'incertitude d'une décision entièrement fondée sur la preuve. Si un certain degré d'hésitation est autorisé, cela permet de faire un choix parmi des unions d'hypothèses crédibles, sans pour autant être forcé d'en choisir une seule si l'information ne le permet pas. Cependant, il ne s'agit pas de faire un pari multiple. En effet, ce n'est pas parce qu'une certaine quantité d'incertitude est autorisée que celle-ci doit être maintenue. En effet, si quelque soit la certitude que l'on a en une hypothèse, nous nous contentons de choisir un sous-ensemble contenant cette hypothèse mais d'un cardinal maximum autorisé, cela signifie simplement que l'on parie sur une union d'hypothèses. Or il n'est toujours satisfaisant de maintenir une hésitation en pariant sur le plus grand nombre possible d'hypothèses s'il est possible de focaliser plus sa décision. Il doit pouvoir être possible d'obtenir un comportement proche de celui d'un humain, qui est capable de prendre une décision complète si cela a du sens, mais qui peut aussi rester dans l'incertitude dans le cas contraire. C'est seulement en étant capable d'effectuer cela que l'on peut dire que la décision est à la fois orientée preuve et orientée pari. A notre connaissance, aucun modèle mathématique ne permet cela.

Dans [95], Smets dérive la PT au cas de paris non singletons. Comme cela est expliqué, il se trouve que l'expression correspond à celle de la valeur de Shapley [113] :

$$\mathrm{BetP}(B) = \frac{1}{1-m(\emptyset)} \cdot \sum \left(\frac{m(A) \cdot |A \cap B|}{|A|} \;\middle|\; A \subseteq \Omega \right) \quad \forall B \subseteq \Omega$$

Dans le cas où B est une hypothèse singleton, nous nous retrouvons la PT classique. Sinon, la valeur associée à B est la somme de (1) $m(B)$, (2) de la masse de croyance en toutes unions d'hypothèses de cardinal plus petit que B, pondérée par la taille de l'intersection entre **B** et ces unions d'hypothèses, (3) un "héritage" issu des unions d'hypothèses de cardinal plus grand que le cardinal de B, d'une manière classique à ce que préconise la PT, mais aussi

proportionnellement à la taille de B. Le premier et le troisème élément cités pour cette somme sont parfaitement naturels pour la réalisation de paris partiels, mais le deuxième est problématique. Il implique en effet que l'on a toujours intérêt à choisir une union d'hypothèses ayant le cardinal le plus grand possible, ce qui correspond simplement à un pari multiple et non à un pari partiel. Ainsi, cette solution ne convient pas.

Il serait aussi possible d'utiliser d'autres structures comme celle de la crédibilité ou de la plausibilité pour prendre la décision. Nous pensons que cela n'est pas une bonne solution car ces méthodes favorisent toujours les choix de cardinal maximum.

Un autre outil classique est l'utilisation d'une fonction de coût (telle que le risque empirique) un peu à la manière dont sont réalisés les tests d'hypothèses pour la prise de décision en probabilité objective et en statistique. Cela ne peut évidemment pas correspondre au cas du TBM (de nature fortement subjective) qui axiomatiquement rejette le lien entre probabilité objective et FC.

Finalement, la manière la plus directe serait d'utiliser des fonctions de pondération permettant de relativiser l'intérêt d'une solution en fonction du cardinal du choix qu'elle préconise, et ainsi favoriser les choix focalisés sur un faible nombre d'hypothèses. En associant un poids inférieur à 1 aux éléments du powerset dont le cardinal est trop important, ceux-ci ne seront choisis que si leur intérêt est vraiment prépondérant. Cela peut convenir sur le principe mais nous doutons de l'intérêt d'une telle méthode pour la simple raison qu'elle ne tient pas compte de l'intégralité de la distribution de la masse de croyance. Illustrons cela sur un exemple. Considérons une FC dont le cadre contient 5 hypothèses : $\{h^1, h^2, h^3, h^4, h^5\}$. Nous voulons utiliser une fonction de pondération permettant de ne choisir que des solutions hésitant au plus entre 2 hypothèses. Il se trouve que les croyances en $\{h^1\}$ et $\{h^2, h^3\}$ sont relativement fortes, et qu'elles sont à peu près équivalentes suite à l'application de la fonction de pondération qui diminue la croyance en $\{h^2, h^3\}$ afin de favoriser $\{h^1\}$. Cela signifie qu'originellement, $m(\{h^2, h^3\}) > m(\{h^1\})$, mais que suite à la pondération, ces solutions sont comparablement intéressantes. Il se trouve aussi que les autres solutions de cardinal inférieur ou égal à 2 sont insignifiantes, mais qu'antérieurement à la pondération il existait une union d'hypothèses de cardinal 3 pour laquelle la masse de croyance était aussi très élevée. Cette croyance représente une quantité d'information qu'il est indispensable de ne pas négliger. Si cette solution est $\{h^1, h^4, h^5\}$ par exemple, il est naturel que le choix se porte sur $\{h^1\}$, alors que si cette solution est $\{h^2, h^3, h^4\}$, il est plus logique que le choix se porte sur $\{h^2, h^3\}$. Or, il n'est pas possible de faire une telle différenciation de manière automatique en utilisant une fonction de pondération. En revanche, c'est le propre d'une transformée probabiliste. De la même manière que ces dernières sont plus efficaces que la troncation du powerset au cadre pour émettre une décision, nous pensons que des paris partiels efficaces ne peuvent pas être pris via des fonctions de pondération. Cette solution ne répond donc toujours pas à notre besoin.

Comme aucune des méthodes existantes ne remplit nos objectifs, nous proposons de développer une méthode originale. De plus, nous pensons que ceci n'est possible qu'en généralisant le procédé d'une transformée probabiliste. Il nous reste à déterminer laquelle.

B.2 Justification du choix de la PT comme point de départ

Parmi les nombreuses transformées probabilistes qui existent dans la littérature, les deux dont la justification est la plus solide sont la **Transformée Pignistique** (PT) [93], [94], et la **Transformée de Plausibilité** (PlT) [90]. La transformée de Plausibilité est définie de la manière suivante :

$$\text{PlT}: \quad \begin{array}{l} \mathfrak{B}^{\Omega} \to \mathfrak{Pl}^{\Omega} \\ m(.) \mapsto \text{BetPl}(.) \end{array} \quad \text{avec} \quad \text{BetPl}(h) = \frac{1}{\sum \left(Pl(x) \mid x \in \Omega \right)} \cdot Pl(h) \qquad \forall h \in \Omega$$

Les justifications de la PT sont données dans [93], [94] et [95]. Les arguments de rationalité sur lesquels la PT est basée sont les suivants (à l'exclusion du Principe de la raison suffisante, contrairement à ce qui est dit dans [114]) [93] :

– **Linéarité** : "*The linearity requirement corresponds to the requirement that the two derivations (combining the pignistic probability induced by each belief function or taking the pignistic probability induced by the combined beliefs) lead to the same pignistic probabilities* [93]". Cet argument est dit être obligatoire dès que l'on suppose que "**the expected utility theory**" doit être respectée dans les méthodes d'aide à la décision [109].

– **Projectivité** : la PT d'une FC Bayesienne est équivalente à la FC elle-même, au changement de support près.

– **Efficacité** : un pari sur le cadre entier est vainqueur avec une probabilité de 1.

– **Anonymat** : le résultat de la PT est insensible aux permutations des éléments de Ω.

– **Evénement Faux** : un événement faux a une Probabilité Pignistique nulle.

De plus, dans [95], Smets réfute un **Pari de Dupe** (Dutch Book) qui a été proposé contre la PT [110], [111]. Un Pari de Dupe est un scénario de jeu dans lequel une série de paris (avec les mises et les gains correspondants) garantit un gain ou une perte minimum quelque soit le résultat faisant l'objet du pari. L'existence d'un tel Pari de Dupe (heureusement impossible dans le cadre probabiliste) serait pour le TBM la preuve de son incohérence à l'égard de la prise de décision dans le cas d'un pari. Bien que ce Pari de Dupe ait été réfuté par Smets, celui-ci admet de manière explicite que cela ne constitue pour autant pas une preuve de la résistance du TBM et de la PT en particulier à tous les Livres Hollandais [95].

Dans [94], l'intérêt de la PT par rapport à la PlT est donné à partir d'exemples et de contre-exemples, mais aucune preuve n'est fournie.

D'un autre côté, nous pouvons trouver une justification à la PlT dans [90], [91], [112], et dans leurs références. Cobb et Shenoy indiquent qu'une transformée probabiliste doit vérifier plusieurs propriétés :

– Invariance par rapport à la combinaison de Dempster.

– Idempotence.

– Deux autres propriétés asymptotiques sur l'éventualité d'une hypothèse unique de plausibilité maximum. Ces deux propriétés étant plus compliquées, nous ne les détaillons pas ici.

Le principal argument de la PlT contre la PT est qu'elle n'est pas invariante par rapport à la combinaison de Dempster, et que selon Cobb et Shenoy, cela est beaucoup plus important que la linéarité. Nous proposons un argument supplémentaire au sujet de l'intérêt de la PlT, mais celui-ci ne doit être interprété comme un argument contre la PT.

Intuitivement, utiliser la PlT équivaut à prendre une décision sur la structure de plausibilité, conditionnellement au fait que l'on considère qu'une seule hypothèse est juste. Il est possible de pousser ce raisonnement intuitif par une preuve, en utilisant la méthode de Haenni pour faire apparaître une combinaison de Dempster dans certaines manipulations des FC [108]. En pratique, cela nous amène à démontrer que la combinaison d'une FC avec une autre FC modélisant la méta-connaissance "une seule hypothèse du cadre est juste", suivie d'une marginalisation sur le cadre, est un processus équivalent à la PlT.

Preuve :

Soit k le cardinal de $\Omega = \{ X_1, X_2, \ldots, X_k \}$ et m une FC définie sur Ω. Soit m^{Unique} une autre FC de la multivariable $REL = (REL_1, REL_2, \ldots, REL_k)$ qui modélise notre croyance dans le fait que chacune des hypothèses de Ω soit vraie. Ainsi, $\forall\ i$, $\Omega_{RELi} = \{T_i, F_i\}$ (T et F signifient True et False). Notre méta-information étant ce qu'elle est, nous avons évidemment :

$$m^{\text{Unique}}\left(\left\{(T_1, F_{2\ldots k}), (F_1, T_2, F_{3\ldots k}), \ldots, (F_{1\ldots i-1}, T_i, F_{i+1\ldots k}), \ldots, (F_{1\ldots k-1}, T_k)\right\}\right) = 1$$

avec la convention que $F_{i\ldots j}$ représente le vecteur $(F_i \ldots F_j)$. Cela signifie simplement que nous croyons avec certitude qu'une et une seule hypothèse de Ω est vraie. La **ballooning extension** [108] $m^{\text{Ballooning}}$ de m définie sur $\Omega \times \Omega_X$ est :

$$m^{\text{Ballooning}}\left(\left\{\bigcup (X_i, F_{1\ldots i-1}, T_i, F_{i+1\ldots k} \mid X_i \in A)\right\}\right) = m(A) \quad \forall A \subseteq \Omega$$

le résultat m^{Comb} de la combinaison de $m^{\text{Ballooning}}$ avec m^{unique} est :

$$\begin{cases} m^{\text{Comb}}(X_i, F_{1\ldots i-1}, T_i, F_{i+1\ldots k}) = \mathcal{K}' \cdot \sum \left(m^{\text{Ballooning}}(A) \cdot m^{\text{Unique}}(A) \;\middle|\; \begin{matrix} F_{1\ldots i-1}, T_i, F_{i+1\ldots k} = projection^{\Omega_{REL}}(A) \\ X_i = projection^{\Omega}(A) \end{matrix} \right) \\ m^{\text{Comb}}(.) = 0 \quad \text{sinon} \end{cases}$$

où COORD = projection$^{\text{ESPACE}}$(VECTEUR), indique que COORD est la projection de VECTEUR (les coordonnées excédantes de VECTEUR sont abandonnées) sur ESPACE et où $\mathcal{K}'$ est une constante de normalisation permettant de garantir que la FC a pour somme 1. Comme m^{Unique} ne prend que les valeurs 0 ou 1, il apparaît que :

$$\begin{cases} m^{\text{Comb}}(X_i, F_{1\ldots i-1}, T_i, F_{i+1\ldots k}) = \mathcal{K}' \cdot \sum \left(m^{\text{Ballooning}}(A) \middle| X_i = projection^{\Omega}(A) \right) \\ m^{\text{Comb}}(.) = 0 \quad \text{sinon} \end{cases}$$

ce qui par définition est équivalent à :

$$\begin{cases} m^{\text{Comb}}(X_i, F_{1\ldots i-1}, T_i, F_{i+1\ldots k}) = \mathcal{K}' \cdot \sum \left(m(A) \middle| X_i \in A \right) \\ m^{\text{Comb}}(.) = 0 \quad \text{sinon} \end{cases}$$

On applique ensuite une marginalisation (nous projetons sur les coordonnées qui nous intéressent) sur Ω :

$$\begin{cases} m^{\text{Comb}}(X_i) = \mathcal{K}' \cdot \sum \left(m(A) \middle| X_i \in A \right) \\ m^{\text{Comb}}(.) = 0 \quad \text{sinon} \end{cases}$$

Comme les éléments focaux de m^{Comb} sont maintenant des singletons, il est équivalent d'écrire :

$$\begin{cases} m^{\text{Comb}}(X_i) = \mathcal{K}' \cdot \sum \left(m(A) \middle| X_i \cap A \neq \varnothing \right) \\ m^{\text{Comb}}(.) = 0 \quad \text{sinon} \end{cases}$$

qui représente une FC bayesienne pour laquelle tous les éléments focaux ont les mêmes valeurs que la probabilité issue de la PlT appliquée à m. Ainsi, les résultats de ces deux opérations sont équivalents, à un changement de support près.

En conséquence, l'utilisation de la méta-information "une et une seule hypothèse du cadre est juste" est équivalente à l'utilisation de la PlT.

Fin de la preuve.

Cela prouve indéniablement l'intérêt de la PlT. Cependant, cela ne remet pas en cause l'intérêt de la PT. En effet, permettons-nous de citer un exemple célèbre pour lequel nous pensons que les deux décisions (celle issue de la PT et celle issue de la PlT) bien que différentes, sont toutes les deux justifiées en fonction du contexte de la prise de décision. Ainsi, la méta-information utilisée ci-dessus, peut très bien se révéler vraie ou fausse. L'utilisation d'une transformée plutôt que d'une autre ne doit pas être guidée par l'appartenance à un courant de pensée, mais plutôt par un type particulier de décision. L'exemple en question est celui de la Saga de Paul, Peter et Marie [95], [90] :

Un parrain de la pègre cherche à éliminer un concurrent. Il a le choix entre trois assassins : Peter, Paul et Marie. De ce que sait la police, le parrain commence par choisir à pile ou face (avec une pièce équilibrée) s'il désigne un homme (Paul ou Peter) ou au contraire, s'il désigne une femme (Marie). Dans le cas où il désigne un homme, la police ne connaît pas le processus qui permet de décider s'il s'agit de Peter ou Paul.

Ainsi, la croyance que l'on a dans le fait que l'assassin est Marie est de 0.5 et la croyance que l'on a dans le fait que Paul ou Peter est l'assassin est aussi de 0.5. Comment doit-on répondre à la question de qui est l'assassin ? La PlT nous amène à une équiprobabilité (1/3, 1/3, 1/3) sur les 3 suspects, alors que l'utilisation de la PT nous amène à une probabilité de 0.5 pour Marie, et de 0.25 pour Paul et autant pour Peter. Les distributions sont évidemment différentes et ne mènent pas au même type de décision.

Supposons que la décision doive être prise par le juge d'instruction. Il doit décider de qui va être inculpé en fonction des éléments qu'il possède, et qui sont incomplets. De ce qu'il sait, Marie est un peu plus probable, mais il n'y a aucune preuve de sa culpabilité. Elle ne peut pas être plus ou moins suspectée que les autres. Ainsi, la PlT correspond parfaitement à la manière dont le juge d'instruction doit raisonner. En l'occurrence, il ne peut pas prendre de décision.

Supposons maintenant que la Saga de Paul, Peter et Marie soit un jeu de rôle proposé par un croupier dans un casino, que ce jeu soit à somme nulle (si l'on ne trouve pas le coupable on perd sa mise, et dans le cas contraire, on la récupère trois fois puisqu'il peut y avoir jusqu'à 3 joueurs pariant chacun sur un des suspects), et qu'il est possible de jouer autant de fois que le client du casino le souhaite. Dès lors, parier sur Marie est une stratégie gagnante (et même la seule), car statistiquement, elle permet de gagner dans la moitié des cas une mise de 3 pour une perte de 1 mise dans l'autre moitié des cas. Cette décision est celle issue de la PT.

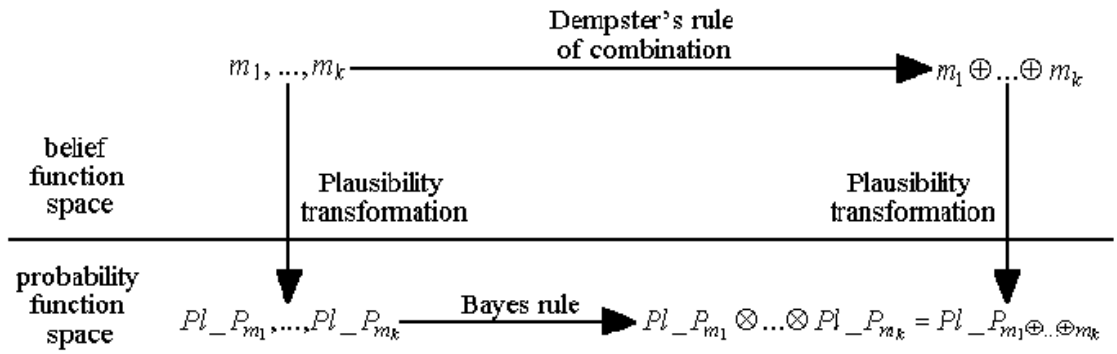


Figure B-1 : la PlT est le lien entre une modélisation crédale et probabiliste. Schéma issus de [90] .

Ainsi, il y a d'une part une décision qui est orientée sur la preuve. En effet, d'une manière générale, la PlT ne permet que de convertir une information codée

par une FC en cette même information codée par une probabilité. D'autre part, il y a une décision qui relève du pari et qui ne relève plus directement de la fusion d'information et de l'attente d'une preuve. Ainsi, cette conversion n'est pas censée être compatible avec la règle de Dempster. C'est pourquoi, si nous cherchons à faire un pari, la PT est la bonne méthode, alors que si nous cherchons à effectuer une conversion de structure évidentielle en une structure bayesienne de telle sorte que cela laisse l'information inchangée (comme expliqué sur le schéma de [90], p. 11, et que nous reprenons sur la Figure B-1), alors la PlT est préférable.

L'objectif initial de cette discussion est de déterminer laquelle des transformées probabilistes doit être généralisée pour servir de base à une transformée permettant de réaliser des paris partiels. Notre conclusion est que la PT est plus adaptée que la PlT à cette tâche.

Nous proposons donc de travailler sur la base de la PT et de la généraliser au cas des paris partiels. Une généralisation de la PT existe déjà. Il s'agit de la **Generalized Pignistic Transformation** [159]. Même si cette transformée consiste en une conversion d'une connaissance en une structure probabiliste (exactement de la même manière que la PT) dans un cadre complètement différent des FC (**Dezert-Smarandache Theory of Plausible and Paradoxical Reasoning** [160]) et que ce travail n'a aucun point commun avec le nôtre, nous proposons d'appeler notre généralisation d'une autre façon afin d'éviter ainsi toute confusion. Nous proposons donc l'appellation **Transformée Pignistique Partielle**, ou PPT en abrégé.

B.3 Justification formelle

Dans cette section, nous justifions la formulation de la PPT que nous proposons dans le paragraphe VI.2.1 (p. 201). Cela est fait une première fois expérimentalement par les travaux que nous avons menés sur l'ASL. Ici, nous la justifions de manière théorique. Pour mémoire, γ est un seuil d'incertitude et la γ[ème]-PPT est définie par :

$$M_\gamma(A) = \begin{cases} m(A) & \text{si} \quad A = \varnothing \\ m(A) + \sum \left(\dfrac{m(B) \cdot |A|}{\sum_{k=1}^{\gamma} \binom{|B|}{k} . k} \;\middle|\; \begin{matrix} B \supseteq A \\ B \notin S^\gamma \end{matrix} \right) & \text{si} \quad A \subseteq S^\gamma \\ 0 & \text{sinon} \end{cases}$$

Comme nous l'avons indiqué dans la section précédente, il nous semble logique que la PPT ait pour origine la PT. En conséquence, elle doit avoir le même comportement, être justifiée de la même manière, mais aussi elle peut être soumise aux mêmes critiques. Ainsi, nous nous contenterons de justifier la PPT par rapport à la PT, et nous ne prétendons pas répondre aux critiques de la PPT qui sont issues de celles normalement adressées à la PT.

Considérons H_i, une union d'hypothèses (ou une hypothèse composite) dont le cardinal est plus grand que γ, le seuil d'incertitude associée à la PPT. Le nombre de choix h^*_j possibles (correspondant à des unions d'hypothèses) dans le γ^{ème} cadre de décision et qui sont inclus dans H_i (cf. Figure B-2) est de :

$$N = \sum_{k=1}^{\gamma} \binom{|H_i|}{k}$$

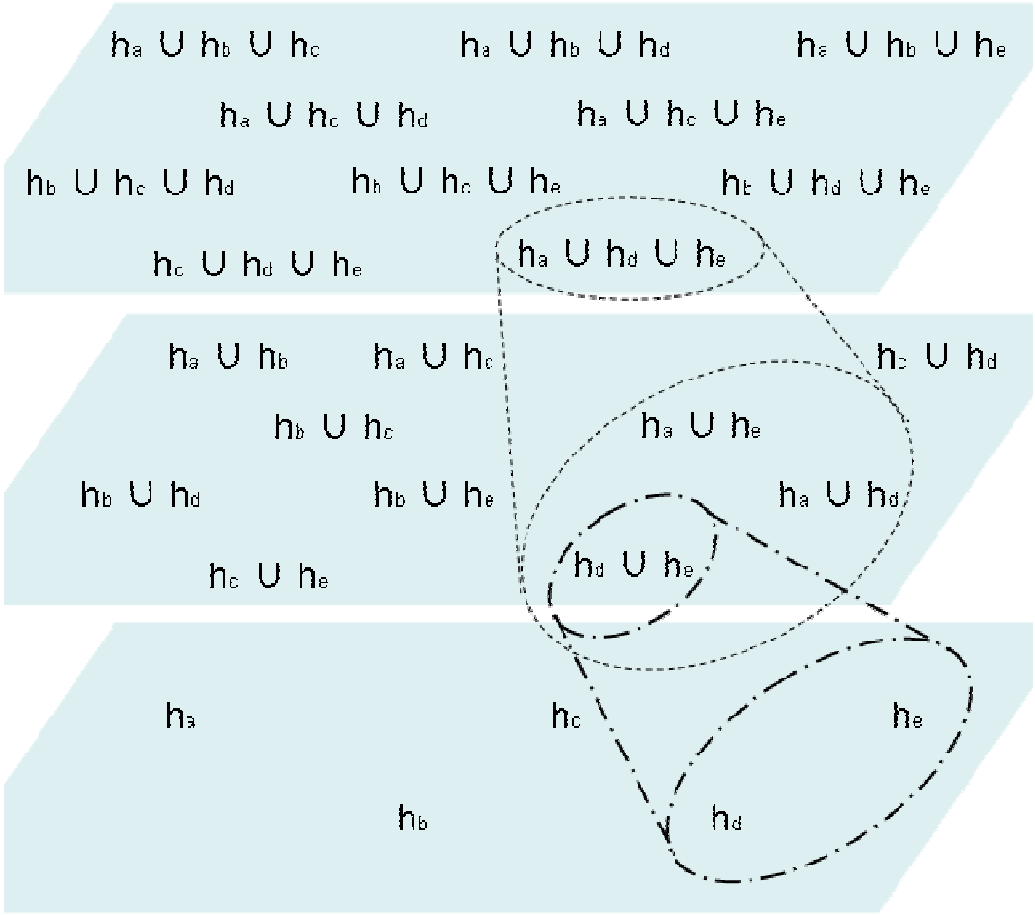


Figure B-2 : Représentation des hypothèses h^*_j impliquées par H_i dans le cas de l'utilisation de la 3ème-PPT.

Conformément à la PT, notre objectif est de partager la croyance entre les h^*_j. Nous attendons de la PPT qu'elle soit :

(1) **linéaire** : la croyance est distribuée linéairement selon la taille des h^*_j.

De plus, comme nous supposons que la PPT est pignistique, elle doit correspondre à la logique du raisonnement probabiliste. Ainsi, chaque choix h^*_j doit être créditable d'une quantité de croyance qui est positivement corrélée au nombre d'hypothèses singletons que chaque choix contient en commun avec les

hypothèses singletons de la connaissance *a priori*. Une seconde manière de considérer cela est de faire le lien avec l'appendice A : cette corrélation permet le respect des axiomes de Cox-Jaynes. Enfin, cela peut être envisagé comme le respect de deux **autres axiomes de la PT :**

(2) **Efficacité**.

(3) **Evénement Faux**.

Ainsi, nous proposons une transformée basée sur le même principe que la PT, où la croyance associée à des éléments du powerset dont le cardinal est trop grand est redistribuée linéairement selon la taille des éléments du cadre de décision et est corrélée positivement à la taille de ces éléments. Ce qui en pratique revient à considérer les N choix possibles calculés précédemment et à leur attribuer une part proportionnelle à leur taille. Cela se fait simplement de la manière suivante :

$$h_j^* \leftarrow \frac{|h_j^*|}{\tilde{N}} \cdot H_i \quad \text{avec} \quad \tilde{N} = \sum_{k=1}^{\gamma} \binom{|H_i|}{k} . k$$

Ensuite, il s'agit de sommer pour tous les éléments du cadre de décision, toutes les croyances ainsi héritées des hypothèses dont la croyance a été redistribuée. Cette manière de redistribuer la croyance semble curieuse, mais les axiomes de Smets l'imposent. Nous pouvons cependant l'illustrer de la manière suivante : si nous considérons h^*_1 dont le cardinal est 1 (h_d par exemple) et h^*_2 dont le cardinal est 3 (par exemple $\{h_a, h_d, h_e\}$), il est clair que ces choix ne sont pas impliqués de manière équivalente par $H_i = \{h_a, h_b, h_c, h_d, h_e\}$. Plus le cardinal de h^*_j est grand, plus h^*_j est proche de H_i et plus l'implication de h^*_j par H_i est importante.

Notons que dans le cas où γ=1, nous avons $|h_i|=1$ et $\tilde{N}=|H_i|$. Ainsi, dans l'HMF, c'est équivalent à la PT classique :

$$BetP(\cdot) = M_1(\ .\ )$$

En revanche, dans l'HMO une normalisation de la FC est nécessaire pour retrouver la PT classique :

$$BetP(\cdot) = M_1(\ .\ |\ \Omega\ )$$

B.4 Comparaison avec la valeur de Shapley

Revenons sur la valeur de Shapley. Il ne s'agit de la discuter ni en tant que telle, ni comme généralisation de la PT. En effet, dans [95], Smets généralise la PT au cas des paris non singletons, mais dans la discussion, rien ne laisse apparaître que la signification à lui donner dans le cadre du TBM est celle d'un pari partiel. De plus, nous avons déjà expliqué en quoi il s'agit d'un pari multiple. Ainsi, il n'y a aucune contradiction entre la valeur de Shapley, la généralisation de la PT aux paris multiples de la PT qui correspond à la valeur de Shapley, et la PPT que nous proposons. En conséquence, nous proposons seulement de les comparer

d'un point de vue structurel, de manière à faire apparaitre les points communs et les différences dans le calcul qu'elles impliquent. La valeur de Shapley est définie de la manière suivante :

$$v(B) = \sum \left(\frac{m(A) \cdot |A \cap B|}{|A|} \;\middle|\; A \subseteq \Omega \right) \quad \forall B \subseteq \Omega$$

La première différence, que nous avons utilisée pour rejeter l'intérêt de la valeur de Shapley pour les paris partiels, a déjà été mentionnée. Il s'agit du fait que l'on somme l'ensemble des croyances en des choix plus restrictifs. Ainsi, un ensemble emboîté d'éléments du powerset a une valeur de Shapley croissante, chose que la PPT évite précisément.

Une autre différence concerne la croyance héritée d'éléments du powerset. Dans le cas de la valeur de Shapley, cet héritage est proportionnel à la taille de l'intersection entre les éléments en jeu. Dans notre cas, il en va de même à la différence que l'élément focal qui reçoit la croyance doit être entièrement inclus dans celui qui la distribue. Cela est justifié par la signification de la masse de croyance : elle représente la croyance associée exactement à un élément focal, et non à un élément de taille inférieure ou supérieure. Ainsi si deux éléments A et B ont une intersection non-vide mais que celle-ci n'est pas égale à B, alors la croyance en A ne peut pas être redistribuée à B, à l'inverse de la croyance en $\{A, B\}$.

La dernière différence se trouve dans l'importance que l'on donne à la croyance redistribuée. Nous avons fait le choix que le résultat de la PPT était une distribution sur le powerset et que sa somme valait 1. Nous considérons cela comme un "principe de conservation" qui permet d'éviter que la croyance redistribuée ne soit comptée plusieurs fois. C'est ce qui arrive dans le cas de la valeur de Shapley. Pour nous, les principaux intérêts sont au nombre de deux. Tout d'abord, cela permet de garantir que la redistribution de la croyance n'a pas une influence trop importante par rapport aux croyances originales. Cela n'est pas nécessaire dans le cas de la valeur de Shapley pour la simple raison qu'il n'y a pas ensuite de comparaison entre des choix de cardinal différent. Ensuite, la manipulation de distributions normalisées est plus aisée.

B.5 Complexité et structure itérative

Il est intéressant d'avoir une connaissance au moins approximative de la complexité de la PPT : pour chaque élément du powerset 2^{Ω} dont le cardinal est plus grand que γ, il est nécessaire de parcourir un tableau de taille $2^{|\Omega|}$ pour pouvoir considérer la redistribution de la masse de croyance qui lui est associé. Le reste des calculs étant de faible importance par rapport à ces parcours, la complexité de la PPT est de l'ordre de $\mathcal{O}(2^{|\Omega|} \times 2^{|\Omega|}) = \mathcal{O}(4^{|\Omega|})$. En pratique, cette complexité est faible par rapport à la combinaison de Dempster, et celle-ci n'est donc pas une limitation à son utilisation.

En pratique, il y a des cas où il peut être intéressant de calculer le résultat de la PPT pour plusieurs valeurs de γ. Dans un tel cas, il est possible de le faire pour pratiquement le même coût calculatoire que pour le calcul direct de la 1ère-PPT. Pour cela, il suffit de calculer en premier la PPT avec $\gamma = \gamma_{max}$, γ_{max} étant la plus haute valeur de γ pour laquelle le résultat de la PPT nous intéresse, puis de calculer itérativement les PPT pour les valeurs γ_{max-1}, γ_{max-2}, et ainsi de suite, chaque calcul étant basé sur le résultat du précédent. γ_{max-i} représente la $i^{ème}$ valeur la plus grande dans l'ensemble des valeurs de γ que l'on considère. En pratique, γ_{max-i} est souvent égale à $\gamma_{max}-i$, mais ce n'est pas obligatoire. Cela est possible parce que :

$$\gamma_1^{ème}\text{-PPT}(.) = \gamma_1^{ème}\text{-PPT}\left(\gamma_2^{ème}\text{-PPT}(.)\right) \quad \forall \gamma_1 < \gamma_2$$

Preuve :

Soit *m(.)* une masse de croyance sur Ω et *A* un élément de 2^{Ω}. L'égalité suivante est évidente pour la simple raison que les deux expressions qu'elles impliquent ont une valeur nulle :

$$M_{\gamma_1}(A) = M_{\gamma_1\gamma_2}(A) \quad \forall A \setminus |A| < \gamma_1$$

où $M_{\gamma 1\gamma 2}$ est le résultat de $\gamma_1^{ème}$-PPT($\gamma_2^{ème}$-PPT(.)).

Considérons maintenant le cas moins évident où *A* a un cardinal inférieur ou égal à γ_1. Dans les deux cas $\gamma_1^{ème}$-PPT(.) et $\gamma_1^{ème}$-PPT($\gamma_2^{ème}$-PPT(.)), et quelque soit *A*, la croyance en *A* *après* la PPT vaut la croyance en *A* *avant* la PPT plus une part de la croyance héritée d'autres hypothèses de cardinal plus grand. Il suffit donc de démontrer que ces héritages sont équivalents pour $M_{\gamma 1}(.)$ et $M_{\gamma 1\gamma 2}(.)$. De plus, la somme des distributions pour $M_{\gamma 1}(.)$ et $M_{\gamma 1\gamma 2}(.)$ est de 1 :

$$\sum_A M_{\gamma_1}(A) = \sum_A M_{\gamma_1\gamma_2}(A) = \sum_A m(A) = 1$$

Dès lors, il suffit de montrer que pour tout couple (A_1, A_2) de cardinal inférieur ou égal à γ_1, si la part de croyance reçue par A_1 via $M_{\gamma 1}(.)$ est plus grande que celle reçue par A_2, il en sera de même par $M_{\gamma 1\gamma 2}(.)$:

$$\left[M'_{\gamma_1}(A_1) \geq M'_{\gamma_1}(A_2)\right] \Rightarrow \left[M'_{\gamma_1\gamma_2}(A_1) \geq M'_{\gamma_1\gamma_2}(A_2)\right] \quad \forall A_1, A_2 \setminus |A_1| < \gamma_1, |A_2| < \gamma_1$$
$$\text{avec } M'_{\gamma_1}(A) = M_{\gamma_1}(A) - m(A)$$

Or cela est vrai, puisque la linéarité de la PPT est un pré-requis à sa définition.

Fin de preuve.

B.6 Discussion

Il y a plusieurs points qu'il est intéressant de discuter à propos de la PPT :

– Tout d'abord, considérons l'équilibre entre indécision et amélioration de la décision. En effet, en autorisant un doute complet tout le temps, on ne fait jamais d'erreur. Idéalement, il ne faut ajouter de l'incertitude que dans les cas

où une erreur aurait été produite. L'inverse étant d'introduire une quantité importante de décisions incertaines sans pour autant réduire le nombre d'erreurs. Pour l'instant, nous ne savons pas comment discuter ce point d'une manière générale et théorique. Cela dépend de la distribution des données sur lesquelles des décisions sont prises.

– Ensuite, mentionnons un point faible de la PPT qui nécessite le choix d'une valeur pour le paramètre γ. Bien que la PPT soit robuste aux petites variations de γ, son choix précis est généralement délicat, car il nécessite d'avoir une vision précise du processus de décision. Il serait intéressant de mettre au point une méthode de calcul automatique de ce paramètre en fonction d'indices sur la distribution de croyance. Cela est inspiré du **Probability Information Content** (PIC) [115], qui permet de quantifier la présence de l'information nécessaire à la prise d'une décision probabiliste.

– Enfin, discutons de la nature du résultat de la PPT. En tout état de cause, il s'agit d'une distribution sur le powerset des hypothèses considérées. Est-ce pour autant une FC ? Si c'est le cas, cela signifie que sa combinaison avec une autre FC par la règle de Dempster a un sens. Or, rien ne prouve que c'est le cas, d'autant que la PPT est dérivée de la PT qui elle n'est pas compatible avec la règle de Dempster. Ainsi, son utilisation dans des modèles mixtes de prise de décision et de combinaison d'informations, tels que le choix d'un chemin dans un treillis selon l'algorithme de Viterbi [128], [117], doit être explorée avec beaucoup de précautions. Il s'agit néanmoins un problème digne d'intérêt.

APPENDICE C
COMPLEMENTS ALGORITHMIQUES

C.1 Le CNN, le CFF et le C3F

Un **Convolutional Neural Network** (CNN) [80] est un type particulier de réseau de neurones dérivé du perceptron multicouche (PMC) initialement proposé pour la reconnaissance de caractères. Son principe de fonctionnement est le suivant : une image est proposée en entrée du CNN et celui-ci calcule une série de descripteurs de l'image à partir d'une succession de convolutions, puis effectue une classification en fonction de ces descripteurs. Au premier abord, il est possible de décomposer l'architecture du CNN en deux parties. Celle où les descripteurs sont calculés et celle où la classification est effectuée. Dans cette dernière partie, les neurones sont conformes à ceux du classique PMC. En revanche, dans la première partie, les neurones sont différents pour plusieurs raisons :

– Le nombre de coefficients synaptiques par neurone est beaucoup plus important, et des neurones correspondant à des pixels voisins sur l'image d'entrée partagent certains coefficients. En pratique, c'est ce partage de certains coefficients qui permet de calculer une convolution à l'aide d'une couche de neurones. Le noyau d'une telle convolution se trouve donc être déterminé par les valeurs des coefficients synaptiques.

– Suite au calcul de la convolution, l'image est redimensionnée par un processus de sous-échantillonnage amélioré où chaque groupement de 4 pixels est combiné en fonction d'une autre série de coefficients synaptiques.

– En pratique, chaque couche de neurones possède plusieurs sorties et plusieurs convolutions/sous-échantillonnages sont effectués en parallèle.

Finalement, une image présentée en entrée d'un CNN va subir une série plusieurs convolutions/sous-échantillonnages en parallèle, via la première couche du réseau. Ensuite, le résultat de chacune de ces convolutions/sous-échantillonnages est transmis à une autre couche, et ainsi de suite, jusqu'à produire la série "d'imagettes" finale. Cette série d'imagettes sert de descripteurs ; elle est transmise à la seconde partie du CNN pour la classification.

Le principe du **Convolutional Face Finder** (CFF) [49] est d'utiliser un CNN couplé à un algorithme de parcours multi-échelle pour détecter des visages. Le CNN utilisé comporte deux couches de convolution/sous-échantillonnage (son architecture est présentée sur la Figure C-1 d'après [51]). Il est capable de classer des images en deux catégories : celles représentant un visage entier, et les autres. L'algorithme de parcours multi-échelle permet d'appliquer le CNN à toute sous-image (de taille quelconque) de l'image dont les visages doivent être détectés. Celles-ci sont présentées tour à tour à l'entrée du CNN qui est de taille fixe pour des raisons d'implantation. Pour résumer, cela permet de déterminer la présence de tous les visages que l'image originale contient.

Le **Convolutional Face and Features Finder** (C3F) fonctionne sur le même principe que le CFF. Il permet de trouver les 4 points correspondant aux positions les plus probables des centres des yeux, du nez et de la bouche pour l'image de visage proposée en entrée. Pour cela, un PMC entièrement connecté est utilisé pour la partie du CNN dédiée à la classification. Cela permet de proposer un jeu de 4 densités de probabilité de présence dans l'image de chacun des 4 centres. Une décision par maximum de probabilité sur chacune des 4 distributions permet de déterminer la constellation des 4 points recherchés.

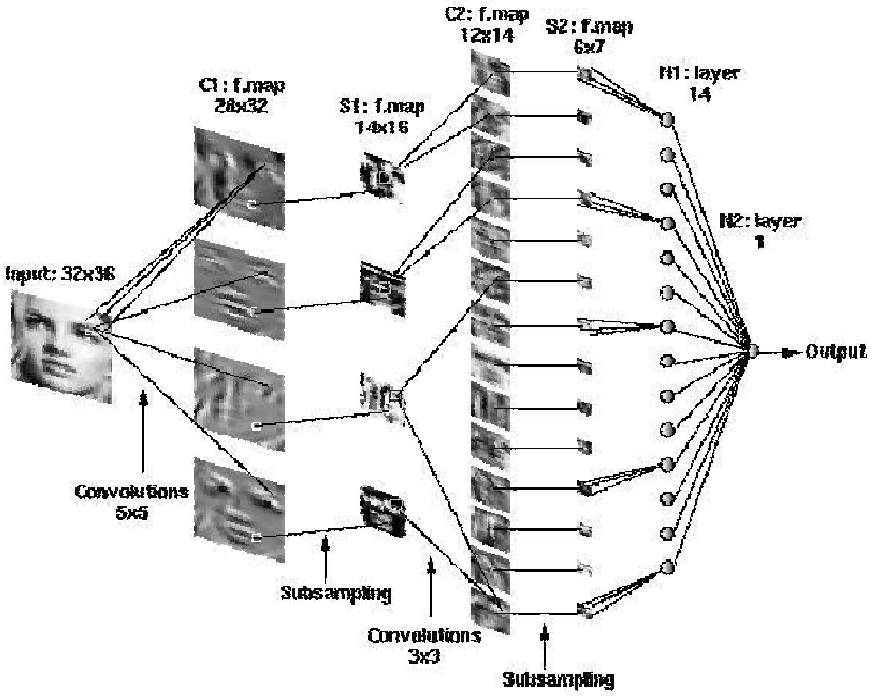


Figure C-1 : architecture du CFF, d'après [49].

La difficulté de la méthode est de mettre en place un CNN qui soit adapté à la classification désirée. Ainsi, les descripteurs qui sont pertinents pour la détection de visages n'ont pas de raison de l'être pour la reconnaissance de caractères. Ceux-ci sont donc appris au travers du jeu de coefficients synaptiques de manière conjointe avec ceux permettant la classification. Comme l'ensemble de l'apprentissage est effectué par un algorithme unique et que la rétro-propagation permet la spécification des attributs de classification et les séparations entre les classes en un seul apprentissage, il n'est pas possible de considérer ces deux étapes comme distinctes, comme nous l'avons présenté dans l'introduction du chapitre V (p. 123) : ainsi, il vaut mieux voir l'ensemble comme un unique algorithme ne rentrant pas dans la méthodologie que nous avons présentée.

Cependant, il est intéressant de se demander ce que valent les différentes sorties du CNN du CFF en tant que descripteur de visage, prises indépendamment de la seconde partie du CNN qui permet la classification. Il est impossible de répondre à cette question, mais on peut malgré tout conjecturer qu'elles sont assez

efficaces. La première raison est simplement le taux de réussite de l'algorithme final. Pour la seconde raison nous proposons de regarder le CNN différemment :

D'une manière générale, les descripteurs de formes utilisés en traitement d'images sont issus d'une transformée mathématique dont l'expression est du type :

$$\int_{x_1}\int_{x_2} K(x_1,x_2)\cdot f\left(\rho(x_1,x_2)\right)dx_2dx_1$$

c'est-à-dire, l'intégrale sur les deux coordonnées de l'image (x_1,x_2) de la fonction ρ associée à l'image par la valeur que portent les pixels, à laquelle est éventuellement appliquée une fonction f, mais impérativement multipliée par une fonction K, représentant le noyau de la transformée.

En pratique, la transformée de noyau K est appliquée à une image de taille finie et binarisée. En conséquence, K n'est généralement pas une fonction pathologique et l'approximation par troncation de sa décomposition dans une base orthonormée (ondelettes, bases polynomiales, fonctions trigonométriques, développement de Taylor, etc.), n'est pas problématique. Cependant, il n'existe pas toujours une transformée permettant de calculer formellement les descripteurs idéaux au regard du problème à résoudre. Cela revient à dire que K n'a pas de forme analytique. En revanche, un tel noyau peut très bien être approché par son élément le plus proche dans l'espace vectoriel associé à une base de fonctions.

L'objectif de l'apprentissage du CNN est de définir la manière de calculer les bons descripteurs en fonction du problème à résoudre. Comme une grande partie des traitements d'images couramment utilisés peut s'exprimer sous la forme de convolutions, l'approximation de K par une succession de convolutions est digne d'intérêt, même s'il n'est pas prouvé que l'espace des convolutions soit générateur de l'espace des noyaux qui nous intéresse.

Ainsi, une manière de comprendre l'efficacité du CFF est de voir le CNN comme un outil permettant, de manière itérative, de déterminer une forme calculatoire à base de convolution du noyau d'une transformée inconnue correspondant à la définition de descripteurs adaptés au problème.

C.2 Le filtre de Kalman

Dans cette section, nous présentons le principe du filtrage de Kalman [149] utilisé dans le section III.4 (p. 93) pour régulariser les trajectoires des traits permanents (centres des yeux, du nez et de la bouche) définies par la succession de leurs positions trouvées par le C3F.

Considérons x, le vecteur d'état (dont les composantes sont à valeurs dans $\mathbb{R}$) d'un système S qui évolue au cours d'un temps échantillonné (indexé par la variable t), en fonction de l'état précédent : x_t dépend de x_{t-1}. L'état du système n'est pas directement observable, mais il génère une observation z qui est une

fonction de l'état courant : z_t dépend de x_t. La récupération de l'état du système au cours du temps est un problème très vaste qui ne possède pas de solution analytique dans de nombreux cas. Le filtre de Kalman propose une solution optimale dans le cas où les deux dépendances que l'on vient de mettre en avant sont des dépendances linéaires, éventuellement perturbées de manière gaussienne.

Quelque soit le point de vue, on résume souvent les contraintes mentionnées plus haut par le système d'équations suivant :

$$S : \begin{cases} x_{t+1} = A_t.x_t + B_t.u_t + C_t.v_t \\ z_t = O_t.x_t + P_t.w_t \end{cases}$$

A chaque nouvel instant t :

– Une nouvelle observation z_t est disponible. Celle-ci est générée en fonction de l'état courant et d'une perturbation stochastique. O_t représente la "matrice d'observations en fonction de l'état" et $O_t.x_t$ correspond à la contribution déterministe de l'état courant dans l'observation courante. P_t est la "matrice de bruit d'observation" and $P_t.w_t$ représente le bruit de la mesure qui permet l'observation.

– Un nouvel état x_{t+1} est généré pour le temps $t+1$ sous l'hypothèse d'une évolution linéaire gaussienne : A_t est une "matrice de transition", et $A_t.x_t$ représente la contribution déterministe du présent pour déterminer le futur. B_t est une "matrice d'entrée" et u_t une perturbation connue qui modélise les phénomènes extérieurs au système. $B_t.u_t$ représente la contribution connue de l'extérieur sur la détermination de l'état futur. Enfin, v_t est un bruit gaussien interne au système et C_t est une "matrice de bruit en fonction de l'état" : $C_t.v_t$ représente la partie non déterministe de l'évolution du système.

Implanter un filtre de Kalman c'est :

– Accepter comme vraies les hypothèses de modélisation qui correspondent à la mise en place du système d'équations ci-dessus. Celles-ci peuvent être vraies dans l'absolu, ou bien constituer une modélisation acceptable du problème.

– Déterminer les matrices A_t, B_t, C_t, O_t et P_t, les vecteurs u_t, et les lois des vecteurs aléatoires v_t et w_t. Comme ces deux dernières sont des lois gaussiennes, seules leur moyenne et leur covariance doivent être définies. De plus, comme $C_t.v_t$ et $P_t.w_t$ sont aussi gaussiens, on peut simplifier le système d'équations S : $v^*_t = C_t.v_t$ et $w^*_t = P_t.w_t$ avec $v^* \sim \mathbf{N}(V_t, Q_t)$ et $w^* \sim \mathbf{N}(W_t, R_t)$.

– Résoudre de manière itérative le système S, en déterminant $X=\{x_1,\ldots,x_N\}$ ainsi que sa covariance σ, sachant $Z=\{z_1,\ldots,z_N\}$ et x_0. σ_t représente l'estimation de σ au temps t. Pour l'initialisation, on a $\sigma_{t=0} = O_{t=0}^{-1}.R.{}^TO_{t=0}^{-1}$, (où ${}^TO_{t=0}$ est la matrice transposée de $O_{t=0}$).

La solution optimale, proposée par Kalman [149], se calcule en deux temps : la prédiction de l'état futur en fonction du présent, et la mise à jour de l'état en fonction de l'observation qu'il génère. Voici la solution d'une telle itération pour x_t :

– Prédiction :

$$\begin{cases} x_t = A_{t-1}.x_{t-1} + B_{t-1}.u_{t-1} \\ \sigma_t = (A_{t-1}).(\sigma_{t-1}).({}^T A_{t-1}) + Q_{t-1} \end{cases}$$

– Mise à jour :

$$\begin{cases} K_t = \sigma_t.({}^T O_t).\left[(O_t).(\sigma_t).({}^T O_t) + R_t\right]^{-1} \\ x_t = x_t + K_t.(z_t - O_t.x_t) \\ \sigma_t = \sigma_t - K_t.O_t.\sigma_t \end{cases}$$

L'existence d'une solution analytique est possible dans la mesure où (1) l'on sait résoudre analytiquement une équation linéaire, et (2) une combinaison linéaire de lois gaussiennes est une loi gaussienne. Du moment que ces conditions sont respectées, il est possible de définir d'autres généralisations du filtre de Kalman :

– Le vecteur d'observations au cours du temps est disponible en entier (on connaît le futur). Il est alors possible d'effectuer une double itération (temps croissant et temps décroissant), afin de modéliser une interaction bidirectionnelle, ou afin d'obtenir un meilleur lissage [129].

– Utiliser un modèle non temporel : le graphe des états est un arbre non dégénéré (les graphes cycliques ne permettent pas de terminer l'itération) [129].

– Utiliser des interactions non linéaires, en les linéarisant localement, ou en utilisant la transformée "Unscent" [150].

– Modéliser des dépendances d'ordre 2 (l'état présent et l'état passé déterminent le futur). Cela se fait généralement en étendant le vecteur d'état.

Dans le cas de distributions de probabilité trop compliquées (la probabilité *a posteriori* est donc difficile à évaluer), on peut aussi utiliser les méthodes variationnelles [151], ou les méthodes d'évaluation de Monte-Carlo [151]. On parle alors dans le dernier cas de filtres particulaires, et la solution n'est plus optimale, mais l'algorithme "Condensation" [43] est un bel exemple de leur efficacité.

C.3 Recherche de composantes connexes

Nous présentons dans cette section le principe général de l'implantation de l'algorithme d'étiquetage en composantes connexes que nous utilisons [169].

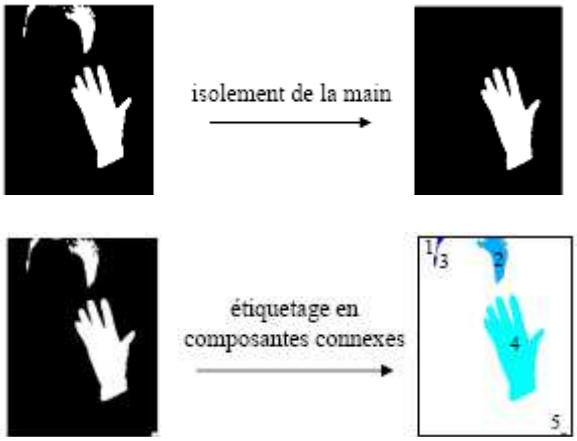


Figure C-2 : objectif de l'extraction d'une composante connexe (ici la main), et étiquetage des composantes connexes.

Un algorithme d'étiquetage et de recherche de composantes connexes a pour objectif d'identifier les différents objets présents dans l'image (Figure C-2). Pour cela, un label chiffré doit être assigné à chaque pixel qui n'appartient pas au fond, de telle sorte qu'un label soit l'identifiant d'un objet.

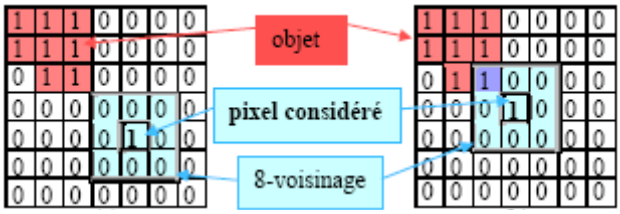


Figure C-3 : le pixel considéré et son 8-voisinage. Celui-ci intersecte l'objet rose sur l'exemple de droite (il appartient donc à cet objet) ; ce n'est pas le cas à gauche.

Deux objets sont considérés comme différents si aucun pixel de l'un n'est dans le voisinage d'un pixel de l'autre. Il est donc important de bien définir la notion de voisinage. En l'occurrence l'algorithme suivant est défini pour un 8-voisinage (en bleu sur la Figure C-3), mais l'on pourrait très bien le redéfinir pour un 4-voisinage (en ne considérant que les 4 voisins verticaux et horizontaux, et en ne prenant plus en compte les 4 voisins diagonaux).

Les algorithmes d'étiquetage en composantes connexes sont assez gourmands en temps de calcul. Ils sont en général construits sur le schéma suivant :

Etape 1 : Labellisation initiale. Assignation d'un label à chaque pixel. Dans l'exemple de la Figure C-4, avec un parcours de l'image de haut en bas et de droite à gauche, un label différent (ici, une couleur) est attribué à chaque objet. Un problème apparaît cependant : le dernier pixel réunit deux objets que l'on considérait ici comme distincts. Il faut donc résoudre cette équivalence, c'est-à-dire réunifier en un même objet les différents morceaux.

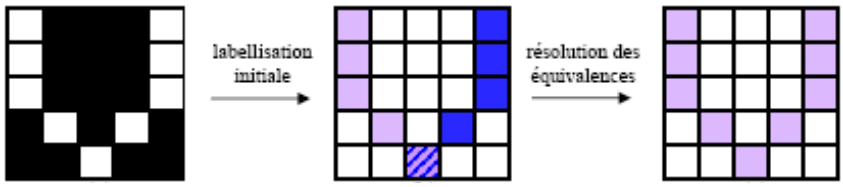


Figure C-4 : labellisation des objets et résolution des équivalences.

Etape 2 : Mémorisation des équivalences. Pour résoudre les équivalences, nous devons stocker dans une **matrice d'équivalence** les labels équivalents. Cela se traduit dans notre exemple par la mémorisation du fait que les couleurs bleu et violet sont équivalentes.

Etape 3 : Résolution des équivalences. Modification des labels initiaux par le label de leur classe d'équivalence. Dans l'exemple de la Figure C-4, il s'agit d'attribuer aux deux objets le même label (la même couleur) car ils ne font en réalité qu'un.

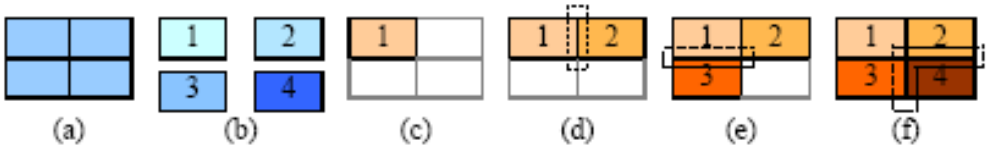


Figure C-5 : (a) Image initiale. (b) Division de l'image initiale en 4 imagettes. (c) Labellisation de l'imagette 1. (d) Labellisation de l'imagette 2, fusion avec l'imagette 1 et résolution des équivalences aux frontières. (e) Labellisation de l'imagette 3, fusion avec les imagettes 1 et 2 et résolution des équivalences aux frontières. (d) Labellisation de l'imagette 4, fusion avec les imagettes 1, 2 et 3 et résolution des équivalences aux frontières.

L'étape 1 (labellisation initiale) produit beaucoup trop de labels et par conséquent beaucoup d'équivalences qu'il faut résoudre. Or, la résolution d'équivalences a un coût exponentiel par rapport au nombre de labels. Pour diminuer ce dernier, Park et al. [172] proposent de diviser l'image en $N \times M$ imagettes, ce qui réduit le coût de la résolution d'équivalences. Cette méthode accélère l'algorithme de Rosenfeld et Pfatz [174] en diminuant sa complexité. Il a l'avantage d'être rapide et indépendant du hardware utilisé (il n'utilise pas de parallélisme machine et peut être implanté dans un langage de haut niveau). Le principe est simple :

– Division de l'image en N×M imagettes.

– Labellisation des imagettes, mémorisation des équivalences, puis résolution des équivalences pour les imagettes.

– Fusion au fur et à mesure des imagettes labellisées et résolution des équivalences aux frontières. Sur l'exemple de la Figure C-5, nous avons mis en pointillés les frontières sur lesquelles les équivalences doivent être résolues.

L'algorithme présenté implique N.M.(N.M+1)/2 parcours d'imagettes pour résoudre les équivalences aux frontières (cf. Figure C-5). Pour diminuer ce nombre de parcours, nous proposons l'amélioration suivante : les équivalences aux frontières ne sont résolues qu'après avoir labellisé toutes les imagettes. L'algorithme se découpe alors de la manière suivante (cf. Figure C-6) :

– Division de l'image en N×M imagettes.

– Labellisation de toutes les imagettes, avec résolution des équivalences dans les imagettes.

– Fusion des imagettes labellisées et résolution des équivalences dans les imagettes. Donc contrairement à ce qui précède, la fusion n'est pas faite au fur et à mesure, mais seulement après avoir traité toutes les imagettes. Cette méthode permet une diminution de 20% à 45% du temps de calcul, en fonction du cas.

Figure C-6 : (a) Image initiale. (b) Division de l'image initiale en 4 imagettes. (c) Labellisation des imagettes. (d) Fusion des imagettes et résolution des équivalences aux frontières.

Extraction de la main. Une fois l'image binarisée et les composantes connexes étiquetées, il faut isoler la main des autres objets. Pour cela, nous cherchons le label correspondant au centre de gravité de la main dans l'image précédente. Nous faisons donc l'hypothèse qu'il varie peu et que nous pouvons utiliser ses coordonnées pour localiser la main. Pour la première image, nous prenons le label de coordonnées égales à l'intersection des diagonales du rectangle correspondant à la zone d'apprentissage.

C.4 Transformée de distance

Dans cette section, nous donnons le principe du calcul de la transformée de distance, d'après [169]. Celle-ci est définie comme :

$$D(\mathrm{p}) = \min_{q \in fond} d(p,q)$$

où d est une mesure de distance, q un pixel du fond et p un pixel de l'objet. Autrement dit, chaque point de l'objet est étiqueté par la distance la plus courte le séparant du fond, et chaque point du fond prend la valeur zéro. L'algorithme mis en œuvre de complexité linéaire [173] est basé sur la généralisation de la transformée de distance d'une image binaire à une transformée de distance pour des fonctions échantillonnées (fonctions définies sur une grille).

Soit G une grille régulière et $f : G \to \mathbb{R}$ une fonction de cette grille. La transformée de distance de f est définie comme :

$$D_f(p) = \min_{q \in G} \left(d(p,q) + f(q) \right)$$

où $d(p,q)$ est la mesure de distance entre p et q. Ainsi, si l'image binaire est remplacée par une fonction échantillonnée définie par :

$$f(q) = \begin{cases} O & si\ q \in fond \\ \infty & si\ q \in objet \end{cases}$$

alors

$$\begin{aligned} D_f(p) &= \begin{cases} \min d(p,q) & si\ q \in fond \\ \min \left(d(p,q) + \infty \right) & si\ q \in objet \end{cases} \\ &= \begin{cases} \min d(p,q) & si\ q \in fond \\ \infty & si\ q \in objet \end{cases} \\ &= \min_{q \in fond} d(p,q) \end{aligned}$$

Nous retrouvons bien la définition classique de la transformée de distance d'une image binaire. Montrons que si l'on utilise ici la norme euclidienne, nous pouvons appliquer récursivement la transformée d'une fonction échantillonnée sur chaque dimension :

Soit G une grille et $f : G \to \mathbb{R}$ une fonction de cette grille

$$\begin{aligned} D_f\left(p_{x_1}, \ldots, p_{x_n}\right) &= \min_{q_{x_1}, \ldots, q_{x_n}} \left(\sum_{x_i=1}^{n} \left(p_{x_i} - q_{x_i} \right)^2 + f\left(q_{x_1}, \ldots, q_{x_n}\right) \right) \\ &= \min_{q_{x_2}, \ldots, q_{x_n}} \left(\sum_{x_i=2}^{n} \left(p_{x_i} - q_{x_i} \right)^2 + \min_{q_{x_1}} \left(\left(p_{x_1} - q_{x_1} \right)^2 + f\left(q_{x_2}, \ldots, q_{x_n}\right) \right) \right) \\ &= \min_{q_{x_2}, \ldots, q_{x_n}} \left(\sum_{x_i=2}^{n} \left(p_{x_i} - q_{x_i} \right)^2 + D_{f|q_{x_2}, \ldots, q_{x_n}}\left(p_{x_2}, \ldots, p_{x_n}\right) \right) \end{aligned}$$

où $D_{f|q_{x_2}, \ldots, q_{x_n}}\left(p_{x_2}, \ldots, p_{x_n}\right)$ est la transformée de distance à une dimension de f selon les dimensions x_2, …, x_n et indexée par x_2, …, x_n.

D'un point de vue algorithmique, cette propriété est très intéressante pour obtenir la transformée de distance d'une image binaire. En effet, il est possible d'appliquer la transformée de distance euclidienne quadratique à une dimension sur chaque colonne en utilisant la fonction échantillonnée définie plus haut. Nous obtenons alors une fonction f' elle aussi échantillonnée (qui peut être vue comme une image en niveaux de gris) sur laquelle nous appliquons la transformée de distance euclidienne quadratique à une dimension sur chaque ligne (cf. Figure C-7).

Par ailleurs, prendre le minimum d'une distance quadratique d'une ligne ou d'une colonne revient à prendre l'enveloppe convexe d'une série de paraboles (une par élément de la ligne ou de la colonne) ; cela permet de calculer facilement la transformée de distance à une dimension (cf. Figure C-8).

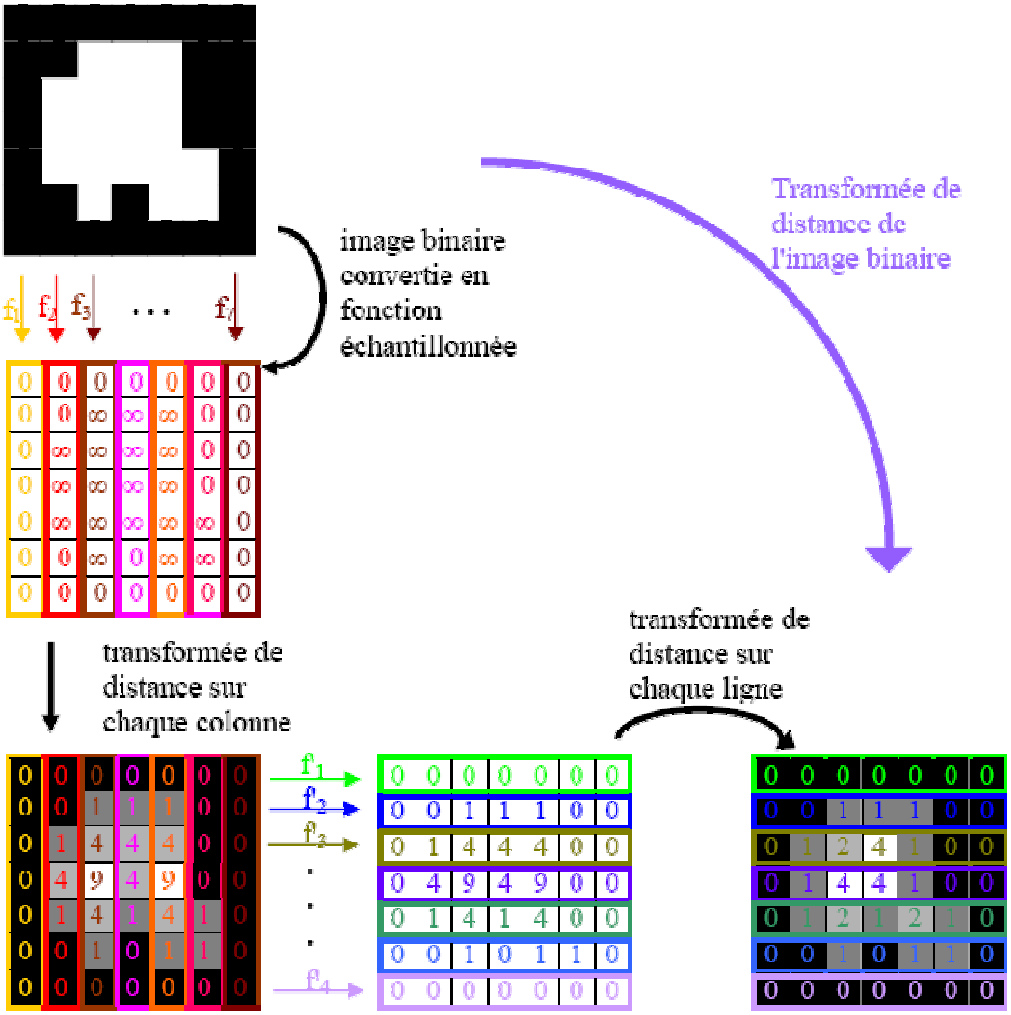


Figure C-7 : calcul de la transformée de distance par récursivité.

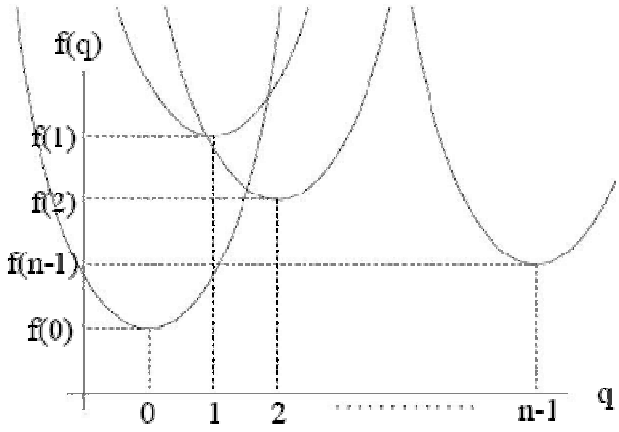


Figure C-8 : la transformée de distance est vue comme la plus basse enveloppe délimitée par les n paraboles.

C.5 Processus d'optimisation combinatoire d'un C-SVM

Dans le V.2.3.2 (p. 150), nous avons détaillé le fonctionnement d'un C-SVM, mais nous n'avons pas abordé la manière de résoudre l'optimisation combinatoire qui permet de définir l'hyperplan séparateur. Nous abordons cela maintenant, à partir de [58], [61] :

Soit $X = \{x_1, \ldots, x_i, \ldots, x_l\}$ un ensemble de l vecteurs d'apprentissage définis dans l'espace des attributs et $Y = \{y_1, \ldots, y_i, \ldots, y_l\}$ l'ensemble des labels à valeur dans $[\![-1 ; 1]\!]$ canoniquement associé à l'ensemble d'apprentissage. En fonction des points $W = \{w_1, \ldots, w_i, \ldots, w_l\}$ associés à chaque élément de X, de la valeur du coefficient C, et des variables ressorts $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_i, \ldots, \mathcal{E}_l\}$, il s'agit de définir l'équation de l'hyperplan séparateur en résolvant l'optimisation suivante :

$$\min_{W,b,\varepsilon}\left(\frac{1}{2}W^{T}W + C\cdot\sum_{i=1}^{l}\varepsilon_i\right)$$

sous les contraintes que $\forall i \in \lfloor 1, l \rfloor$:

$$y_i\cdot\left(W^{T}\varphi(x_i)+b\right)\geq 1-\varepsilon_i$$

$$\varepsilon_i \geq 0$$

Ce qui est un problème convexe. En conséquent, sa résolution est équivalente à la résolution du problème dual, dont la formulation est la suivante :

$$\min_{\alpha}\left(\alpha^{T}Q\alpha + e^{T}\alpha\right)$$

sous les contraintes que $\forall i \in \lfloor 1, l \rfloor$:

$$y^{T}\alpha = 0$$

$$C \geq \alpha_i \geq 0$$

où e est un vecteur unité, Q est une matrice semi-définie positive telle que $Q_{ij} = y_i y_j K(x_i, x_j)$, et où K est la fonction noyau. Une fois la valeur α déterminée, la règle de décision permettant d'effectuer la classification est :

$$\operatorname{sgn}\left(\sum_{i=1}^{l} y_i\cdot\alpha_i\cdot K(x_i, x) + b\right)$$

Ce qui correspond à regarder le signe de la distance algébrique de l'item à classer à l'hyperplan séparateur. Ainsi, dans le cas d'un problème de multi-classification où la combinaison des SVM s'effectue au moyen de la Combinaison Evidentielle, il suffit de récupérer la valeur

$$\sum_{i=1}^{l} y_i\cdot\alpha_i\cdot K(x_i, x) + b$$

qui correspond à la distance entre l'item et l'hyperplan, et de lui associer une masse de croyance dont la répartition dépend de la fonction d'appartenance utilisée : plus cette distance est positive, plus la fonction de croyance est focalisée sur l'élément singleton du powerset correspondant à la classe +1, et plus cette distance est négative, plus la masse de croyance est focalisée sur

l'élément singleton du powerset correspondant à la classe -1. Dans le cas où la distance est proche de 0, la masse de croyance est focalisée sur le doute.

Remarquons que la fonction φ, permettant d'augmenter la dimension de l'espace des attributs afin de trouver un hyperplan séparateur n'apparaît à aucun moment dans le calcul. Ainsi, l'augmentation de dimension n'est jamais réalisée explicitement. Le seul endroit où φ intervient est dans la multiplication par la matrice Q, permettant de pondérer la distance des éléments à l'hyperplan en fonction du produit scalaire K, lui-même déduit de φ. Ces pondérations permettent de modifier les distances, comme si l'espace des attributs était immergé dans un espace de dimension plus grande par une application non-linéaire. Par cette astuce combinatoire, il est possible de rester dans un espace de dimension raisonnable, et donc d'éviter la **malédiction de la dimension**.

On appelle ainsi la cause des problèmes qui sont inhérents au travail dans un espace de dimension trop grande. En effet, la densité d'une population diminue de manière exponentielle de la dimension de l'espace dans lequel elle est considérée. Ce phénomène étrange peut être expliqué de manière assez intuitive. Considérons un hypervolume d'un espace de dimension quelconque d, et regardons la proximité des points qu'il contient par rapport au centre de cet hypervolume. Par souci de simplicité, prenons comme hypervolume l'hypercube de côté 1, et regardons la quantité de l'espace à l'intérieur de l'hypercube qui est à une distante inférieure ou égale à 0.5 du centre de l'hypercube. Cela revient à comparer les volumes de la boule (en fait "l'hyperboule") de rayon 0.5 et de l'hypercube de côté 1. En dimension 1, ils sont de même taille. En dimension 2, la différence vaut 1-π/4. En dimension 3, 1- π/6, et en dimension d,

$$1 - \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)} \cdot \frac{1}{2^d}$$

La fonction Γ ayant le comportement asymptotique d'une factorielle, l'ensemble de cette expression converge vers 1, ce qui veut dire que plus l'espace à une grande dimension, plus la proportion de l'hypercube qui est près du centre est petite, pour finalement être nulle en dimension infinie. Cela illustre la baisse de densité exponentielle en fonction de la dimension dont nous parlions initialement.

Ainsi, quand il s'agit de mettre en place un classifieur selon une méthode d'apprentissage, il faut veiller à ce que le nombre d'attributs ne soit pas trop élevé. En effet, si celui-ci est trop grand, il devient difficile de fournir un corpus d'apprentissage suffisamment important. Il y a trop peu de données disponibles à l'apprentissage par rapport au nombre d'attributs, et l'espace n'est pas peuplé de manière suffisamment dense pour avoir un bon pouvoir de généralisation (les données fournies spécialise alors trop le classifieur et il y a sur-apprentissage). D'une manière générale, les classifieurs paramétriques sont beaucoup plus sensibles à la malédiction de la dimension.

C.6 Descriptif du matériel d'acquisition des corpus

L'ensemble de cette section est directement extraite de [20] :

Le GIPSA-Lab/DCP possède un banc d'acquisition et de stockage d'images vidéo.

Caméras : L'acquisition des images est réalisée par l'intermédiaire de 2 caméras 3CCD de marque JVC (Ref. JVC KY 15E). Les CCD (Charge Coupled Devide) sont des dispositifs d'analyse photosensibles de type matriciel dont la densité des photo-éléments détermine leur résolution. L'image est obtenue après intégration (de durée variable) des charges sur les 3 matrices (RGB). Les CCD présentent une grande dynamique et d'excellentes précisions géométriques (pas de déformation d'image). Ils sont dépourvus de rémanence et sont insensibles aux champs magnétiques et électrostatiques. Ces caméras sont munies d'obturateurs synchronisés sur chaque trame dont le temps de pose est réglable jusqu'à la milliseconde. Leur résolution est de l'ordre de 638x582 pixels et le rapport S/B de l'ordre de 55 dB.

Betacam : Le stockage des images est réalisé via des magnétoscopes Betacam SP de marque Sony (Ref. UVW 1400, UVW 1600 et UVW 1800). Ces magnétoscopes possèdent des entrées/sorties de 3 types : RGB/composante (résolution de 800 pts/ligne), Y/C (600 pts/ligne) et Composite (260pts/ligne).

Mixage : Une table de mixage permet de mixer la sortie des 2 caméras et d'obtenir ainsi sur une même bande des images de face et de profil.

Numérisation des images : La numérisation des images vidéo ainsi stockées peut se faire selon 2 procédures. En effet le GIPSA-Lab/DCP possède 2 types de carte d'acquisition vidéo :

– une carte d'acquisition avec compression (miroVideo DC 50) fabriquée par la société Pinnacle. Le taux de compression minimale de la DC 50 est de 2,85 pour un débit de transfert de 7400Kb/s.

– une carte d'acquisition sans compression (Meteor I) fabriquée par la société Matrox. Elle permet une acquisition des images au format RGB sur 24 bits sans compression. Un logiciel pilotant cette carte, (nommé **Capture**) a été développé à l'ICP par Marc Audouy.

Description de Capture : L'application Capture a été réalisée avec l'aide de Marc Audouy (Ingénieur ENSIMAG). Elle permet de numériser des images à partir d'une bande vidéo au format Betacam et de les stocker sur disque dur. Elle permet également de récupérer de façon synchronisée la bande audio associée. La numérisation se fait en donnant les time-codes de début et de fin : l'application pilote alors automatiquement le magnétoscope Betacam SP (cf. Acquisition vidéo), via une liaison RS 422, pour aller chercher les images. Capture fonctionne uniquement sous Windows NT.

Cette application est constituée de deux parties : l'interface graphique écrite en Visual Basic et un composant COM écrit en C++ qui effectue l'essentiel du travail. C'est une application dédiée, elle nécessite une carte Matrox Meteor I pour l'acquisition vidéo et une carte AEC (Ref. PC-LTC de la compagnie Adrienne Electronic Corporation) qui permet la lecture du time code sur le Betacam et réalise l'interface avec le magnétoscope via la RS 422. Capture est en revanche indépendante de la carte son, pourvu que celle-ci fonctionne correctement sous l'environnement de travail. De même elle ne peut fonctionner qu'avec des magnétoscopes possédant le même protocole de communication que le Sony Betacam SP. Les images stockées en sortie sont des bitmap codées en RVB (rouge, vert, bleu), elles sont acquises en 32 bits et sauvegardées en 24 bits, format le plus standard pour le BMP.

Il existe deux modes de fonctionnement :

– un fonctionnement en ligne par l'intermédiaire d'une interface graphique.

– un fonctionnement en mode batch qui permet d'effectuer une série de séquences de capture de manière automatique en fournissant les différents paramètres dans un fichier texte.

GLOSSAIRE

ASL : American Sign Language. C'est la langue des signes utilisée aux Etats-Unis.

Anacousie : perte auditive moyenne supérieure à 90dB. Quelqu'un atteint d'anacousie est sourd profond.

Boğaziçi : Université d'Istanbul avec laquelle un échange de thésard fut organise.

C-SVM : Il s'agit d'un algorithme particulier de résolution de l'optimisation combinatoire nécessaire pour l'utilisation des SVM dans le cas où les classes ne sont pas séparables.

Cued Speech : Version anglaise-américaine et originale du LPC.

CV : Syllabe constituée d'une consonne C suivi d'une voyelle V. Il s'agit de l'unité de codage de base du LPC.

dB : décibel.

DPC : Département Parole et Cognition du GIPSA-Lab.

Deixis : action de montrer (du grec *deiktikos*). Dans notre cas, il s'agit principalement d'une action de pointage.

DIS : Département des Images et Signaux du GIPSA-Lab.

ECOC ou **CCE** : Error Correcting Output Codes, ou Code Correcteur d'Erreur en français.

ETTRAN : série de phrases utilisée pendant la campagne préliminaire afin de générer les corpus ETTRAN N (avec un gant noir) et ETTRAN BF (avec un gant bleu foncé). ETTRAN signifie Etude TRANsition. Cette série de phrases représente l'ensemble des transitions possibles entre les phonèmes du français.

Factor graph : Formalisme graphique développé dans [129] permettant d'expliquer un grand nombre d'algorithmes d'inférence en termes de marginalisation de fonction.

FC : fonctions de croyance (cf. Appendice A, p. 255).

FRD : Filtre Rétinien Dédié. Il s'agit d'un algorithme d'évaluation de la quantité de mouvement déformable que nous avons mis au point en se basant sur le fonctionnement de la persistance rétinienne chez les vertébrés.

GIPSA-Lab : Laboratoire de Grenoble en Image, Parole, Signal et Automatique.

GMM : Gaussian Mixture Model, ou mixture de gaussiennes.

HMM : Hidden Markov Model. Formalisme permettant d'étudier un phénomène inconnu (ou caché) discret dans le temps obéissant à la propriété de Markov, à partir d'une série d'observations sur le comportement du phénomène.

Hypergraphe : objets mathématiques correspondant à la généralisation d'un graphe dans lequel les arêtes peuvent connecter plus de 2 sommets.

Hypoacousie : diminution pathologique de l'audition pour laquelle la perte moyenne est supérieure à 20dB.

IC / IT : Image Cible et Image de Transition. Une image cible est une image qui représente un geste statique noyé dans un flux dynamique. Les images de transition représentent la coarticulation consécutive à l'enchaînement des gestes statiques.

ICC / ITC : Image Cible de Configuration et Image de Transition de Configuration. Il s'agit des mêmes notions que IC et IT mais restreinte au mouvement de changement de Configuration.

ICP / ITP : Image Cible de Position et Image de Transition de Position. Il s'agit des mêmes notions que IC et IT mais restreinte au mouvement de changement de Position.

ICX / ITX : Image Cible de X et Image de Transition de X. Il s'agit des mêmes notions que IC et IT mais restreinte au mouvement de changement de X (soit Position, soit Configuration).

ITXM : Durant le flux des ITX qui composent une transition, il s'agit de l'image pour laquelle le mouvement est le plus important.

INPG : Institut National Polytechnique de Grenoble.

IPL : Inner Plexiform Layer (ou couche Plexiforme interne). Il s'agit de la couche de neurones de la rétine où l'on trouve le phénomène de persistance rétinienne.

LIG : Laboratoire d'Informatique de Grenoble.

LPC : Langue française Parlée Complétée, appelée aussi parfois code LPC.

LSF : Langue des Signes Française.

Magicien D'Oz : Protocole expérimental consistant à simuler l'existence d'un outil automatisé grâce à un magicien qui réalise lui-même les tâches de l'outil, afin de pouvoir mener des études d'usage, d'ergonomie et d'interaction avant même le complet développement d'un prototype.

MAGOZ : Ensemble de données issu de la campagne du Magicien d'Oz, à partir duquel les corpus MAGOZ R, J et B sont produits. Chacun représente un gant et un codeur différents.

Pari de Dupe : scénario de jeu dans lequel une série de paris (avec les mises et les gains correspondants) garantit un gain ou une perte minimum quelque soit le résultat faisant l'objet du pari.

PMS : Processus de Markov à Saut.

Powerset : Ensemble des parties d'un ensemble.

Presbyacousie : hypoacousie causée par l'âge.

SVM : Support Vector Machine (Séparateur à Vastes Marges). Algorithme de classification binaire basé sur la recherche d'un hyperplan séparateur optimal par rapport à la distance extra-classe.

TELMA : téléphonie à l'usage des malentendants.

Titre : Reconnaissance automatique des gestes de la Langue Française Parlée Complétée.

Résumé : Le LPC est un complément à la lecture labiale qui facilite la communication des malentendants. Sur le principe, il s'agit d'effectuer des gestes avec une main placée à côté du visage pour désambigüiser le mouvement des lèvres, qui pris isolément est insuffisant à la compréhension parfaite du message. Le projet RNTS TELMA a pour objectif de mettre en place un terminal téléphonique permettant la communication des malentendants en s'appuyant sur le LPC. Parmi les nombreuses fonctionnalités que cela implique, il est nécessaire de pouvoir reconnaître le geste manuel du LPC et de lui associer un sens. L'objet de ce travail est la segmentation vidéo, l'analyse et la reconnaissance des gestes de codeur LPC en situation de communication. Cela fait appel à des techniques de segmentation d'images, de classification, d'interprétation de geste, et de fusion de données. Afin de résoudre ce problème de reconnaissance de gestes, nous avons proposé plusieurs algorithmes originaux, parmi lesquels (1) un algorithme basé sur la persistance rétinienne permettant la catégorisation des images de geste cible et des images de geste de transition, (2) une amélioration des méthodes de multi-classification par SVM ou par classifieurs unaires via la théorie de l'évidence, assortie d'une méthode de conversion des probabilités subjectives en fonction de croyance, et (3) une méthode de décision partielle basée sur la généralisation de la Transformée Pignistique, afin d'autoriser les incertitudes dans l'interprétation de gestes ambigus.

Mots clés : Langue française Parlée Complétée, code LPC, reconnaissance de gestes, vision par ordinateur, segmentation d'images, évaluation du mouvement, rétine, traitement vidéo, classification, SVM, HMM, fonctions de croyance, Transformée Pignistique Partielle, Langue des Signes Américaine, fusion de modalités.

*****

Title: Automatic recognition of French Cued Speech gestures.

Abstract: Cued Speech facilitates hearing-impaired people communication by completing lip-reading. Basically, its purpose is to add manual gestures nearby the face in order to disambiguate the lip motion which is not self-sufficient for a complete understanding of the message. The goal of **T**elephony for **H**earing **IM**paired **P**roject is to elaborate a terminal which allows communication based on French Cued Speech. Amongst the manifoldness of functionalities it requires, it is mandatory to automatically recognize French Cued Speech manual gestures. The subject of this work is the segmentation, the analysis and the recognition of Cued Speech gestures. It requires image and video processing techniques as well as data fusion, classification and gesture recognition techniques. In order to achieve this goal, we have developed several original algorithms, such as (1) a bio-inspired filter which quantifies the amount of motion in a video by integrating retinal processing, (2) a new combination technique for multi-classification via SVMs or unary classifiers based on belief theories, from which a transform from belief function to probability is derived, (3) a partial decision method based on the generalisation of the Pignistic Transform, in order to authorize some uncertainty when processing ambiguous gestures.

Key words: French Cued Speech, gesture recognition, machine perception, pattern analysis, motion evaluation, retina, video processing, classification, SVM, HMM, belief function, Partial Pignistic Transform, American Sign Language, modality fusion.