Reconnaissance de gestes en vision par ordinateur

Jérôme Martin

N° attribué par la bibliothèque

THÈSE

pour obtenir le grade de

DOCTEUR DE L´INPG

Spécialité : «Imagerie Vision Robotique»

préparée au laboratoire GRAVIR – IMAG

dans le cadre de l´école doctorale «mathématiques et informatique»

présentée et soutenue publiquement

par

Jérôme MARTIN

le 13 juillet 2000

Titre :

RECONNAISSANCE DE GESTES EN VISION PAR ORDINATEUR

Directeur de thèse :
James L. CROWLEY

JURY

Présidente	Mme Marie–Paule CANI, INP Grenoble
Rapporteurs	M Joseph MARINI, LIMSI–CNRS Paris Orsay (non membre du jury)
	M Christian ROUX, ENST Bretagne
Examinateurs	Mme Sylvie GIBET, LIMSI–CNRS Paris Orsay
	M Olivier BERNIER, France Telecom R&D
	M James L. CROWLEY, INP Grenoble

Préface

L´implémentation des techniques présentées dans cette thèse et leurs expérimentations sont basées sur l´utilisation de l´environnement multi–languages Ravi, développé au sein du projet Prima, par Augustin Lux, Bruno Zoppis, Claude Poizat et Christophe Le Gal [LZ97, Zop97, LZPL].

Les illustrations présentent dans cette thèse, à l´exception de celle précisant le contraire, sont l´œuvre de Marie–Claude Tourtet.

Enfin, pendant mes années de thèse, on m´a souvent posé la question : « tu fais quoi dans la vie? ». Je répondais tout naturellement que je préparais une thèse en informatique et obtenais en réponse : « c´est quoi une thèse? » ou bien « mais tu fais quoi en fait? ». Pour répondre à cette difficile question, également posée par certains de mes relecteurs, j´ai décidé d´illustré chaque fin de chapitre par les planches de la bande dessinée de « Le petit Nicolas en thèse ». Les dessins sont de J. J. Sempé, les textes de G. Taviot [Pet]. J´espère que les auteurs ne m´en voudront pas.

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Introduction

1 Contexte : Vers une interaction Homme–Machine gestuelle par la vision artificielle de l´utilisateur

Le domaine de la vision par ordinateur a longtemps consisté à interpréter les objets dans une image d´une scène. Il s´agissait d´un processus de traitement de l´information dont l´entrée est constituée d´une ou plusieurs images. Le système apporte un certain nombre de connaissances, des connaissances physiques tels que la gravitation impliquant qu´un objet doit être posé sur un support horizontal, des connaissances géométriques définissant l´objet en terme de lignes ou de surfaces et des connaissances de haut niveau décrivant la fonction de l´objet dans la scène ou bien son existence. La présence d´une chaise dans une scène d´intérieur est plus probable que celle d´un arbre. Une approche classique de la reconnaissance se limitait à la reconnaissance d´objets polyhédriques souvent représentés par un modèle géométrique de type CAO.

Poussée par les progrès scientifiques et technologiques, la recherche en vision par ordinateur s´est orientée vers la compréhension de scènes comportant tout type d´objets et, en particulier vers l´analyse de scènes comportant des humains. La disposition de caméra et de cartes d´acquisition vidéo dans le commerce a poussé son introduction dans le domaine de l´interaction homme–machine. L´interaction homme–machine imagine de nouveaux moyens de communication avec un système informatique où l´utilisateur n´est plus réduit au traditionnel couple clavier/souris. Les interfaces deviennent perceptuelles [1], elles voient l´utilisateur dans son environnement. De son côté, la vision par ordinateur découvre un nouveau terrain d´expérimentation dans laquelle la difficulté est la reconnaissance et la compréhension d´un objet très complexe : l´utilisateur.

Plutôt que de nous intéresser à la reconnaissance du mouvement et des gestes du corps

1. On parle alors d´*interfaces utilisateur perceptuelles* ou *perceptual user interface* (PUI) par opposition aux traditionnelles *interface utilisateur graphique* GUI

humain ou même sur le visage, nous nous sommes concentrés sur la main. Elle est effet très riche, d´un point de vue du nombre de degrés de liberté[2] puisqu´elle en compte 28 : 22 pour les doigts et 6 pour le mouvement de la main complète. La figure 1 présente quelques mouvements permis par la main. Mais également du point de vue «organe de communication». Nous verrons dans le premier chapitre de cette thèse que le canal de communication gestuel est l´un des plus riche. Il présente aussi un grand intérêt du point de vue des interactions homme–machine puisqu´il permet la transformation des interfaces graphiques classiques en interface digitale [Bér00]. Dans les interfaces graphiques classiques, la main agit sur un dispositif physique, la souris par exemple, permettant à son tour d´agir sur le dispositif logique. Dans une interaction digitale, la main agit directement sur le dispositif logique. De plus, la dextérité de la main permet la définition de signes pour le contrôle d´interface. Ces signes utilisent à la fois la configuration de la main et son mouvement.

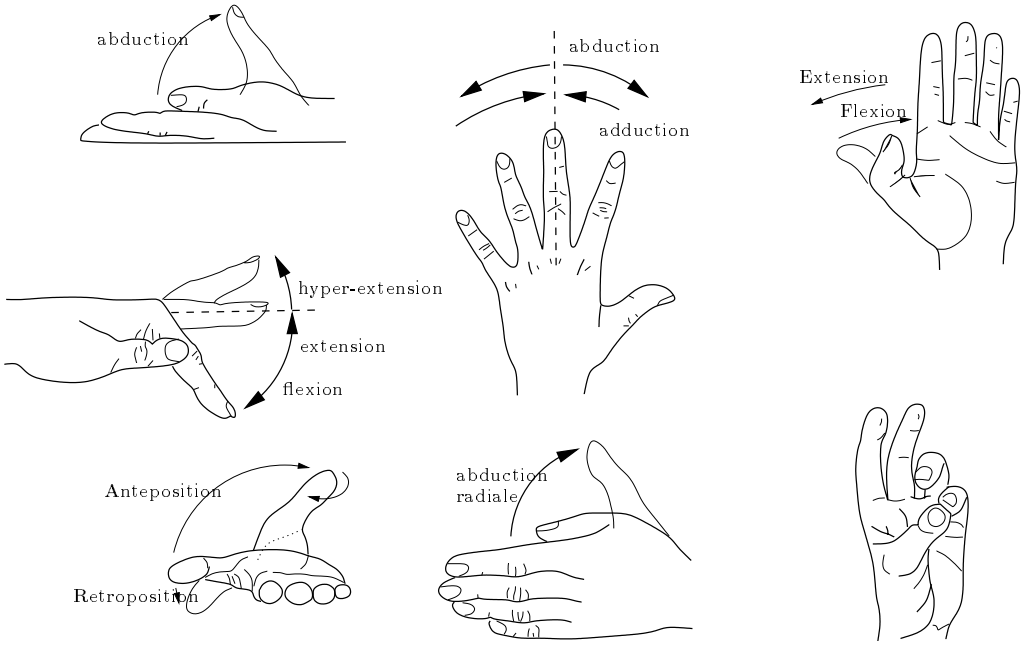


FIG. 1 – ***Mouvement de la main.*** *(extrait de [Stu92])*

2. *degrees of freedom* ou DOF

2 Sujet de recherche et approche

Le sujet de nos travaux de recherche est la conception et le développement de techniques liées à la vision par ordinateur pour la reconnaissance de gestes. Contrairement à une approche «top–down» [Bér00] qui identifierait les besoins dans le domaine de l´interaction homme–machine par une étude de l´utilisateur et de son activité. Nous avons choisi une approche «bottom–up». Nous cherchons à concevoir des techniques de vision par ordinateur et de statistiques pour la reconnaissance de geste puis leur application à l´interaction homme–machine. Nous pensons en effet que la conception de tel système doit être réalisée dans les deux sens. Dans un premier temps, une démonstration technique de la faisabilité de la reconnaissance indépendemment de l´utilisation dans un système d´interaction homme–machine. Puis, l´étude de l´utilisation des besoins applicatifs pour l´adaptation des techniques.

Dans le cadre de cette thèse, nous nous intéressons à l´étude de techniques pour le développement d´un système de reconnaissance de gestes. Nous proposons les trois étapes permettant à un système de comprendre les gestes, plus particulièrement ceux de la main, qu´un utilisateur effectue face à une caméra. La première étape est l´analyse de l´image. Nous présentons des techniques basées sur la vision par apparence pour extraire les caractéristiques spatiales et de configuration de la main. La vision par apparence, contrairement à l´approche de modélisation tridimensionnelle de la main ainsi que de sa cinématique, permet la modélisation d´une configuration de main à partir de ce qui est directement observable dans l´image telle que la couleur, le contour ou un ensemble de pixels. La seconde étape permet l´interprétation de gestes dynamiques. Une séquence d´images d´une main effectuant un geste permet la définition d´une trajectoire en considérant un ensemble de paramètres dans un espace donné. Nous cherchons à évaluer alors des techniques statistiques permettant la classification d´une trajectoire parmi un ensemble de modèles connus et liés à l´application visée. La troisième étape d´interprétation définit deux applications de reconnaissance de gestes et d´activités humaines dans un environnement intelligent. Cette étape permet la définition d´un ensemble de gestes utilisés dans l´ensemble du manuscrit comme base d´expérimentation.

3 Organisation du manuscrit

Cette section présente le manuscrit chapitre par chapitre en développant les aspects proposés par chacun d´eux.

Le chapitre 1 présente l´**interaction homme–machine gestuelle**. Dans un premier temps, le canal gestuel est étudié d´un point de vue psychologique et anthropologique. Trois fonctionnalités distinctes mais complémentaires définissent les gestes. La **fonction ergotique** est une fonction d´action matérielle, de modification et de trans-

formation de l´environnement. La **fonction épistémique** est la fonction tactilo–proprio–kinesthésique de perception. La **fonction sémiotique** est une fonction d´émission d´information à destination de l´environnement. Celle–ci semble être la plus appropriée pour l´interaction homme–machine, elle est étudiée en profondeur. Plusieurs types de gestes sémiotiques sont présentés et illustrés par des exemples. Le lien entre la fonction sémiotique et ergotique est présentée, elle correspond, du point de vue système informatique, à une différence entre faire et faire–faire.

Après l´étude du geste, son utilisation est étudiée pour l´interaction. Des études sont proposées montrant l´utilisabilité du geste comme moyen de communication avec un système informatique multimodal. Des exemples de systèmes informatiques utilisant les gestes sont proposés. Ces exemples sont divisés en trois catégories. La **reconnaissance de la langue des signes** est une catégorie d´application mais aussi un domaine important pour l´utilisation des gestes. Elle permet d´étudier un ensemble de gestes riches et précis. Sa reconnaissance est un apport important pour la communauté des sourds–muets et des linguistes qui l´étudient. Les systèmes de **réalité virtuelle** plongent l´utilisateur dans un environnement d´images synthèse. Dans cet environnement, l´utilisateur manipule par des gestes physiques des objets virtuels. La **réalité augmentée** laisse l´utilisateur dans le monde physique dans lequel l´utilisateur ou son environnement sont augmentés de fonctionnalités informatiques. L´utilisateur effectue alors des gestes physiques sur des objets physiques ou virtuels.

Deux catégories de gestes sont présentés [HP95]. Un **geste statique** correspond à la configuration de la main à un instant donné tandis que le **geste dynamique** correspondand à un changement de cette configuration dans le temps. Une classification plus précise [Edw97] distingue la configuration et la position. Un geste peut être défini par une trajectoire dans un espace de caractéristiques. Ces caractéristiques représentent la configuration et la position de la main. Nous proposons de décomposer le problème de reconnaissance et d´interprétation de gestes en trois parties. L´**étape d´analyse** calcule les caractéristiques de la main. L´**étape de reconnaissance** est une analyse spatio–temporelle de la trajectoire permettant sa classification parmi un ensemble de trajectoires connues. L´**étape d´interprétation** effectue la correspondance entre le geste et l´action à réaliser par le système.

La reconnaissance est intimement liée à la nature de l´application, trois classes de techniques sont présentées et illustrées d´exemples. L´approche de reconnaissance de gestes de dessins s´appuie sur des périphériques, tels qu´une **tablette graphique ou une souris**, fournissant des coordonnées 2D. Le **gant numérique** permet d´obtenir des mesures précises sur la position des doigts. Cette technique souffre du lien entre l´utilisateur et l´ordinateur. La **vision par ordinateur** permet de libérer l´utilisateur de ce lien. Une, ou plusieurs, caméras observent l´utilisateur ou une partie de celui–ci. Deux approches sont alors envisageables. Le première consiste à reconstruire un modèle 3D de la main à partir des images. La seconde consiste à utiliser directement les images ou des éléments extraits de l´image.

Dans le chapitre 2, nous considérons l´**extraction de caractéristique**. Cette extraction correspond à l´étape d´analyse. Deux types de caractéristiques sont étudiées. Les **caractéristiques spatiales** correspondent à la position de la main dans l´image. Les caractéristiques auxquelles nous nous intéressons sont la position de la main dans le plan, son orientation et sa taille. Nous considérons la localisation par segmentation permettant de différencier les pixels de la main des autres pixels. La localisation par la couleur s´appuie sur la chrominance particulière de la main pour la distinguer des autres objets. Un modèle de la chrominance par histogramme et par gaussienne sont discutés. La localisation par différence d´images consiste à effectuer une différence pixel à pixel entre deux images. Cette localisation permet de déterminer les objets en déplacement lorsque deux images successives sont considérées. La différence avec une image de fond permet la détection des objets apparus depuis l´acquisition de l´image de fond. La localisation par apparence consiste à comparer la manifestation visuelle de la main avec un ensemble de manifestations possibles. La corrélation permet de mesurer la similitude entre deux images. Cette localisation s´appuie donc sur une manifestation exacte de l´objet à localiser et ne prend pas en compte les changements d´orientation ou d´échelle. Une extension est proposée pour considérer des motifs de référence à plusieurs orientations et plusieurs échelles. Les techniques proposées présentent chacunes des avantages et des inconvénients. Nous proposons un système de suivi basé sur une architecture dans laquelle un superviseur active et coordonne des modules visuels.

L´**extraction de caractéristiques de la configuration** est un vecteur de mesures permettant de caractériser la configuration d´une main. La solution la plus directe est de prendre directement l´image de la main. Cependant, cette solution n´est pas réalisable en pratique. L´analyse en composantes principales ou transformation de Karhunen–Loeve, permet d´extraire un sous–espace optimal d´une distribution d´images. Dans ce sous–espace, une image est représentée par un vecteur de petite dimension. Ce vecteur donne une description concise de la configuration de la main et permet une généralisation en supprimant les informations inutiles. L´analyse en composantes principales permet de réduire la dimensionalité de l´espace. Cependant, cette réduction optimise la reconstruction et non la discrimination. Le discriminant de Fisher permettant cette réduction pour la discrimination est présenté. Les invariants de Hu sont une seconde méthode pour extraire les caractéristiques de l´image de la main. Ces invariants sont calculés à partir des moments d´ordre supérieur. Les invariants sont calculées pour être indépendants en similitude et rotation. Nous montrons qu´ils permettent de classifier des configurations de mains différentes. Cette classification est étudiée au chapitre 3.

Au chapitre 3, nous présentons une **classification automatique de configuration de mains**. Elle s´appuie sur les caractéristiques extraites au chapitre précédent. Les classifications euclidienne et bayésienne sont proposées et expérimentées. Une autre, fondée sur la distance à l´espace propre, est également présentée. Cette classification permet

la reconnaissance des gestes statiques. Elle peut également être utilisée dans le cas d´une reconnaissance des gestes dynamiques fondée sur une approche symbolique des gestes.

Le chapitre 4 correspond à l´étape de **reconnaissance des gestes dynamiques**. Il s´agit ici d´un problème difficile de classification de séquences temporelles de caractéristiques. Deux difficultés apparaissent : la segmentation temporelle du geste et la reconnaissance de la dynamicité. Dans ce chapitre, trois méthodes sont proposées. La première est une classification par **automates d´états finis**. Nous considérons ici un geste représenté par une séquence de symboles. Chaque symbole représente une configuration de main donnée. Les états de ces automates sont associés à des symboles et les transitions représentent un changement de configurations. La reconnaissance d´un ensemble de gestes est obtenue par un système utilisant un ensemble d´automates où chaque automate correspond à un geste particulier du vocabulaire gestuel de l´application.

Les modèles de Markov cachés sont utilisés pour la reconnaissance de séquences temporelles d´observations. Nous étudions dans ce chapitre leur utilisation pour la reconnaissance de gestes. Nous nous intéressons aux trois problèmes spécifiques. Le premier concerne l´architecture du modèle : un modèle complet est–il plus adapté qu´un modèle gauche–droite? La détermination du nombre optimal d´états est également un problème critique lors de la construction de modèles de Markov cachés. Nous proposons une méthode automatique pour les déterminer. Enfin, nous nous intéressons à la nature des séquences d´observations. Nous opposons en particulier les observations discrètes aux observations vectorielles et continues. L´ensemble de ces problèmes est expérimentalement étudié pour la reconnaissance de gestes de dessins, basés sur le langage *Unistroke*.

Puis, une méthode originale basée sur un algorithme de **reconnaissance statistique de trajectoires** est proposée. La trajectoire d´un geste est observée à travers une fenêtre temporelle. Dans cette fenêtre, les caractéristiques sont représentées par une signature dans un espace de caractéristiques créé à partir d´une analyse en composantes principales. La reconnaissance locale de ces signatures utilise un histogramme multidimensionnel. Enfin, nous proposons un algorithme de reconnaissance statistique de signatures de gestes.

Le **chapitre 5** propose une application pour la reconnaissance d´activités humaines. Il présente la combinaison d´un capteur d´éléments d´activités et la reconnaissance par modèles de Markov cachés. Le capteur utilise une description spatio–temporelle du mouvement et fournit une carte de probabilité pour chaque classe d´activités considérée. Une règle de décision permet la transformation de cette carte en un symbole discret utilisé en entrée du système de modèles de Markov cachés. Dans ce système, chaque activité est représentée par un modèle, la reconnaissance de l´activité est effectuée en considérant le modèle ayant obtenu la plus forte probabilité. Des premiers résultats encourageants sont proposés.

L´environnement MONICA **est présenté au chapitre 6**. Il s´agit d´un environnement intelligent et interactif dans lequel les ordinateurs participent aux activités de l´utilisateur. L´interaction dans de tels environnements se fait suivant les modes humains : la voix, le geste et le mouvement. La description matérielle et logicielle de l´environnement est présentée. Un ensemble d´applications liées à cet environnement est proposé, parmi lesquels le *Tableau Magique*, un tableau blanc augmenté. Cet environnement est un formidable banc d´expérimentation pour la reconnaissance des activités et des gestes de l´utilisateur.

Le **dernier chapitre** présente les **conclusions et perspectives** relatives au travail effectué dans le cadre de cette thèse.

Un **ensemble d´annexes** propose un complément théorique au calcul de l´analyse en composantes principales ainsi que des résultats complémentaires.

Le petit Nicolas en thèse [Pet]
Le directeur de thèse

«Pour commencer une thèse, il faut avoir un patron. Un patron, c´est un monsieur très, très fort qui me pose un problème et qui va m´aider à le résoudre.»

Chapitre

Une Interaction Homme–Machine gestuelle ?

« Qu´est–ce qu´un geste ? » « Un système informatique peut–il utiliser les gestes comme périphériques d´interaction ? » « Quelles applications peuvent être alors conçues ? » « Comment reconnaître un geste ? » Ce sont les questions auxquelles nous allons tenter de répondre dans ce chapitre. Dans un premier temps, nous définissons le geste et comparons les différentes fonctionnalités associées avec ce canal de communication à l´aide d´exemples. Puis, nous nous intéressons à son utilisation pour l´interaction homme–machine. Nous présentons des études qui ont mis en évidence l´utilisabilité d´une telle interaction. Ces études comparent, dans des applications multimodales, l´utilisation des gestes par rapport à d´autres moyens de communication telle que la parole. Trois catégories d´applications profitant de l´interaction gestuelle sont présentées et illustrées de prototypes existants. Une définition informatique et mathématique des gestes est alors proposée dans laquelle un geste est représenté par une trajectoire dans un espace de caractéristiques. La reconnaissance de ces trajectoires s´effectue en trois étapes : analyse, reconnaissance et interprétation. Un état de l´art des techniques de reconnaissance de gestes est présenté. Il s´articule autour de trois classes principales : les techniques utilisant les gestes de dessins, celles basées sur des gants numériques et les techniques visuelles. Nous motivons enfin l´utilisation de ces dernières.

1 Communication gestuelle et Interaction Homme–Machine

Les recherches effectuées dans le domaine de l´Interaction Homme–Machine visent à améliorer les performances des utilisateurs d´un système informatique, non pas en s´intéressant au système lui–même, mais plutôt à son utilisation. Les gestes, mais aussi la parole, apparaissent comme des moyens spontanés pour une personne de communiquer avec son environnement. Ils sont faciles à utiliser, rapides et correspondent à une réalité humaine. Il semble donc légitime de faire évoluer les systèmes informatiques pour prendre en compte ces moyens de communication plus naturels que l´utilisation des périphériques classiques : clavier et souris.

Cette section présente le lien entre la communication gestuelle et l´interaction homme–machine. Dans un premier temps, nous étudions la communication gestuelle. Les trois fonctionnalités du canal gestuel, proposées par Cadoz, sont présentées. La fonction sémiotique, qui est la fonction la plus importante, est développée par la définition d´une taxonomie et illustrée d´exemples. Puis, nous nous intéressons aux interactions homme–machine pouvant utiliser la communication gestuelle. Après une étude de l´utilisabilité de celle–ci, nous présentons trois domaines dans lesquels la communication par geste a un rôle important. Nous donnons enfin une définition des gestes, non plus d´un point de vue psychologue, mais du point de vue d´un concepteur d´un système informatique utilisant le geste comme périphérique d´entrée.

1.1 Communication Gestuelle

Le geste est souvent assimilé «à un mouvement d´une partie du corps (en particulier des mains, des bras ou de la tête) que l´on fait avec ou sans intention de signifier quelque chose» [Hac98]. Pour De Marconnay [dM91], «le geste englobe tous les mouvements des mains permettant de communiquer des informations significatives et pertinentes». Parmi les cinq modes de communication (l´ouïe, la vue, la parole, le toucher et le geste) illustrés à la figure 1.1, le geste semble être l´un des plus riches. Pour Cadoz [Cad94], il est «le plus singulier et le plus riche des canaux de communication». Mais plus qu´un moyen de communication comme l´est la parole, le canal gestuel est aussi un moyen d´action et de perception du monde physique. Le canal gestuel est alors associé à trois fonctionnalités.

1.1.1 Trois fonctionnalités du canal gestuel

Cadoz [Cad94] s´intéresse au canal gestuel associé à la main et, plus particulièrement à la communication «instrumentale»[1]. Il considère trois fonctions distinctes mais

1. il s´agit de «la relation communicationnelle établie entre l´instrumentiste musicien et son instrument».

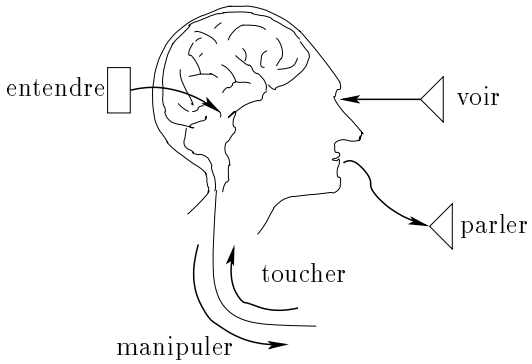


FIG. 1.1 – ***Les cinq modes de communication de l´être humain.*** *(d´après [Mig95])*

complémentaires intervenant à des degrés différents dans chacune des deux autres : une fonction d´action matérielle, de modification et de transformation de l´environnement, nommée fonction *ergotique*, une fonction *épistémique* de perception de l´environnement et une fonction d´émission d´information à destination de l´environnement, la fonction *sémiotique*.

CADOZ précise ces fonctions du geste de la manière suivante :

> **le geste ergotique** agit sur le monde. La main est en prise directe avec la matière, elle peut la modeler, la transformer, la briser, ... La poterie en est un exemple.
>
> **le geste épistémique** , par le sens *tactilo–proprio–kinesthésique*, donne des informations relatives à la température, la pression, l´état de surface d´un objet, sa dureté, sa mollesse, sa forme, son orientation, son poids. Il s´agit du sens du *toucher* ; la main est alors un organe de perception.
>
> **le geste sémiotique** produit un message informationnel à destination de l´environnement. La main devient un organe d´expression. Cette fonction regroupe les gestes qui accompagnent la parole, la langue des signes, les gestes qui incluent une symbolique avec des règles, tels que ceux d´un chef d´orchestre.

La fonction sémiotique du geste est celle qui est la plus riche et la plus complexe, affinons–la au travers d´une taxonomie.

1.1.2 Taxonomie du geste sémiotique

La fonction sémiotique permet au geste d´être un moyen d´expression. Il peut être utilisé en complément de la parole, nous parlons alors de *gestes co–verbaux* et la communication se place dans le domaine de la multi–modalité. Il peut, au contraire, être le seul moyen d´expression, comme dans la langue des signes ou lorsque l´environnement est trop bruyant pour que deux personnes puissent s´entendre. McNeil [McN85] précise que l´action de parler est souvent accompagnée dans notre culture par des mouvements des bras et des mains. Ils anticipent la parole de quelques milli–secondes. Geste et parole partagent donc des étapes de traitement et font partie de la même structure communicative ou expressive [McN85, Fey87, McN87]

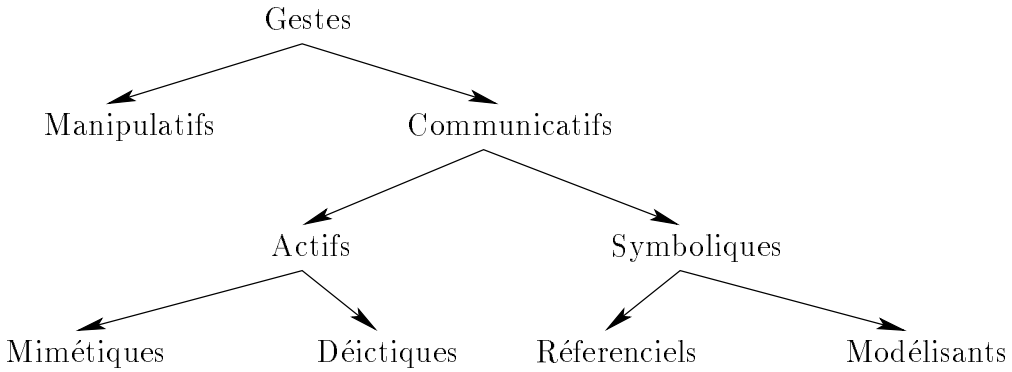


Fig. 1.2 – ***Taxonomie des gestes de*** Quek *[Que94]. Quek décompose les gestes en deux classes : les gestes manipulatifs, correspondant à la fois aux fonctions ergotique et épistémique de* Cadoz *[Cad94], et les gestes communicatifs à la fonction sémiotique.*

Les gestes sémiotiques peuvent être classifiés selon leurs apports informatifs. Suivant les auteurs, cette classification diverge. Nous avons choisi de présenter ici la taxonomie proposée par Quek. Quek étant un chercheur dans le domaine de l´interaction homme–machine, sa taxonomie est directement liée aux applications qui nous concernent, contrairement aux linguistes, psychologues et anthropologues. Cependant, nous faisons, quand cela est possible, référence à la classification de Ekman et Freisen[EF73]. La taxonomie de Quek [Que94] est présentée par la figure 1.2. Dans cette taxonomie, les *gestes manipulatifs* sont précisés, ils correspondent à la fois aux fonctions *ergotique* et *épistémique* de Cadoz [Cad94]. Les gestes sémiotiques ou communicatifs sont décomposés en classes : les

gestes symboliques[2] et les *gestes actifs*[3].

a) Gestes symboliques

Pour QUEK, les *gestes symboliques* sont «un type de mouvement de sténographie ayant un rôle linguistique»[4] [Que94]. Ils disposent d´une *dichotomie de transparence/opacité*[5] les rendant incompréhensifs pour les non–initiés. Ainsi, les langages des signes sont symboliques puisqu´il faut être initié pour les comprendre, au même titre que les langues parlées. CUXAC [Cux99] remarque cependant que beaucoup de gestes de la langue des signes ont une iconicité[6] de telle sorte qu´ils peuvent être interprétés par des signeurs[7] d´origines différentes ou, dans certains cas, par des non–signeurs sans informations complémentaires. Cette catégorie de gestes peut à son tour être divisée en gestes *référentiels* et gestes *de modélisation*.

(*a*) (*b*)

FIG. 1.3 – ***Exemples de deux gestes symboliques***. *(*a*) le geste de continuation : mouvements circulaires répétés de la main pointant sur le côté. Ce geste a un sens différent suivant le contexte pouvant indiquer «continue de me donner des exemples», «continue de faire défiler la page», «continue d´avancer», «fais tourner», , ... (*b*) le geste demandant du feu : mouvement de pliage / dépliage du pouce imitant l´utilisation d´un briquet.*

2. QUEK les nomment *symbol gestures*.
3. *act gestures* pour QUEK.
4. «*Symbol* gestures are a kind of motion shorthand, serving in a linguistic role».
5. *transparency—opacity dichotomy*.
6. c´est–à–dire ayant une origine pictorale.
7. personne «parlant» une langue des signes.

Gestes référentiels Ces gestes font directement référence à un objet ou un concept. QUEK donne l´exemple du geste de frottement circulaire du pouce et de l´index pour référencer l´argent. Il est possible de faire référence à un objet en montrant sa forme ou le volume qu´il occupe dans l´espace, dans ce dernier cas, un mouvement est réalisé. Les gestes représentant une figure ou la forme du référant sont nommés *pictographes* [EF73]. Lorsqu´ils permettent d´esquisser une pensée, les gestes sont des *idéographes* [EF73]. Le geste, consistant à avoir le pouce levé tandis que les autres doigts sont fermés, est en France au moins, un idéographe ; il a le sens de bien.

Gestes Modélisants Les gestes de modélisation sont souvent conjoints à d´autres moyens de communication, la parole par exemple ; ils modélisent un état et/ou l´opinion de la personne l´exprimant. Ainsi, dans une soirée, une personne peut dire : «As–tu vu son mari ?» en tenant ses mains à une certaine distance de son corps pour indiquer qu´il est gros.

Pour CUXAC [Cux99], l´expression faciale permet de donner des indications complémentaires sur l´opinion du communiquant. Une personne indiquant un certain espace entre son pouce et son index modélise une taille cependant son expression faciale peut exprimer son opinion : «cette taille», «gros comme ça» ou bien «petit comme ça». Bien qu´il se place dans le contexte de la langue des signes, ceci reste vrai pour les gestes, l´expression faciale est alors souvent remplacée par la parole ou encore l´intonation de la voix. Un autre exemple de geste de modélisation est celui de continuation (mouvements circulaires répétés de la main pointant sur le côté ou le haut — illustré par la figure 1.3*a*) ayant un sens différent suivant le contexte pouvant indiquer par exemple «continue de me donner des exemples», «continue de faire défiler la page» ou bien «continue d´avancer». Dans cet exemple, l´expression du visage et les mouvements de tête permettent de préciser la vitesse du déroulement.

Ambiguïté culturelle Il faut noter que le sens d´un geste peut avoir un sens différent selon la culture et l´origine du communiquant. Ainsi, le geste représenté par la figure 1.4 dans lequel la configuration des doigts représente un O, a un sens différent selon que nous sommes nord–américain, français ou anglais. Pour un américain, ce signe a le sens de «bien», «ok» ou «parfait». En France, il correspond au chiffre 0 ou à l´absence du référant. Pour un anglais, ce signe est une insulte vulgaire. Attention donc avec qui vous communiquez par gestes ! Le livre de AXTELL [Axt91] donne une liste de gestes ayant une signification différente selon le pays.

b) Gestes actifs

Un geste *actif* est un mouvement réalisé en lien direct avec son interprétation. Il est réalisé en support de la parole. Deux classes divisent les gestes actifs : les gestes *mimétiques* et *déictiques*.

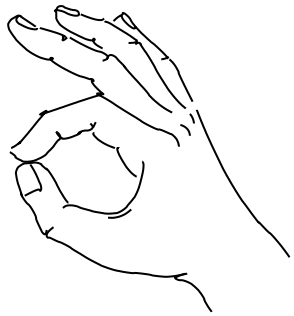


FIG. 1.4 – ***Exemple de geste ayant un sens différent selon les cultures.*** *La configuration des doigts de cet exemple, représentant un O, a un sens différent selon que nous sommes nord–américain, français ou anglais. Pour un américain, ce signe a le sens de «bien», «ok» ou «parfait». En France, il correspond au chiffre* 0 *ou à l´absence du référant, ceci correspond à une quantité nulle. Pour un anglais, ce signe est une insulte vulgaire. En plongée sous–marine, il s´agit du code international pour «ok», même pour les anglais !*

Gestes mimétiques Ils sont souvent exprimés par le mime de l´utilisation référent, sa position spatiale ou temporelle par rapport à d´autres. Ils sont caractérisés par leur iconicité. Une personne, avec une cigarette dans la bouche, faisant un geste de plier/déplier le pouce indique qu´elle veut du feu en imitant l´utilisation d´un briquet (cf. figure 1.3*b*). Il s´agit alors d´un geste *kinétographe* [EF73]. Il permet également de représenter le mouvement du référent à l´aide d´un déplacement des mains. Braffort [Bra96] donne l´exemple suivant : «Après le huitième verre, il titubait `[geste]` !». Le geste permet de montrer le mouvement qu´il effectuait, la vitesse du geste ajoute le rythme du titubage.

Il est également possible de mimer le rythme d´un évènement avec un geste *rythmique* [EF73]. Un exemple est le geste de la tête accompagnant la phrase «...et il marchait, marchait, ...». Le geste de la tête, ainsi que la voix, indique le rythme de la marche.

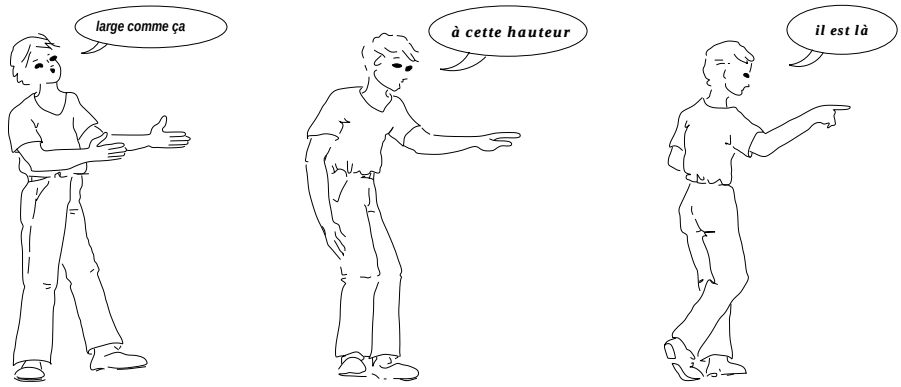


Fig. 1.5 – ***Exemples de gestes mimétiques.***

Enfin, il est possible de mimer la relation spatiale par un geste *spatial* [EF73]. Montrer un écart entre les deux mains permet de définir la relation spatiale entre deux objets représentés chacun par une main. Les mains peuvent représenter des objets différents, comme par exemple dans le geste accompagnant la phrase «La voiture est passée à ça `[geste]` de moi», ou bien deux bouts d´un même objet : «large comme ça `[geste]`]». Un des deux référents peut être omis, il est dans ce cas soit implicite ou soit correspondre au locuteur. Dans la phrase «à cette hauteur `[geste]`», le référent est implicitement le sol. Le geste du locuteur accompagnant la phrase «il est là `[geste]`» permet de spécifier la position relative de l´objet par rapport au locuteur. Ce geste peut, par exemple, signifier à gauche. Ce geste est en fait à mi–chemin entre un geste mimétique consistant à mimer la relation spatiale du locuteur à l´objet et un geste déictique de désignation d´une position.

Identiquement au geste spatial, il est possible de définir un geste *temporel*, définissant la relation temporelle entre deux objets. Cette relation est souvent exprimée par rapport au locuteur. Il représente le présent, le passé se trouvant derrière lui et le futur devant. Encore plus que le geste spatial, cette catégorie de gestes peut être classée dans les gestes de désignation.

Gestes déictiques Les gestes déictiques ou de pointage sont classifiés en trois groupes selon le contexte.

Les gestes déictiques *spécifiques* sont effectués lorsque le sujet fait allusion à un objet ou à un lieu en particulier. Dans la phrase «il [désignation] m´a dit ...», le geste de désignation d´une personne est un spécifique puisque le locuteur fait directement référence à la personne pointée. Les gestes déictiques *génériques* permettent l´identification d´une classe d´objets en indiquant un de ses représentants. Montrer un objet en demandant s´il en reste permet juste de définir la classe d´objets recherchée. Il est évident qu´il ne s´agit pas de l´objet montré lui–même. Avec un geste déictique *métonymique*, l´auteur du geste référence une entité liée à l´objet qu´il pointe. En pointant une photographie de gratte–ciel, il peut faire référence à la ville de New York. Pointer un objet permet aussi d´indiquer sa fonction, comme par exemple désigner sa tête pour indiquer la réflexion.

Ekman et Freisen [EF73]	Quek [Que94]
bâtons	(sans correspondance)
idéographes	symboliques modélisants
déictiques	actifs déictiques
rythmiques	actifs mimétiques
kinétographes	actifs mimétiques
spatiaux	actifs mimétiques
pictographes	symboliques référentiels

Tab. 1.1 – ***Comparaison des taxonomies de*** Ekman ***et*** Freisen ***[EF73] et de*** Quek ***[Que94].***

c) Gestes de battements

Une classe de geste non représentée dans la classification de Quek est celle de *battements* ou *bâtons* [EF73]. Ceux–ci marquent, accentuent et donnent de l´importance à un mot ou une phrase du discours. Ils sont souvent redondants avec l´expression de la voix.

WEXELBLAT [Wex97] ajoute la notion de *gestes publics* et *gestes privés*. L´ensemble des gestes que nous avons présentés dans les sections précédentes est de nature publique : ils apportent un complément d´information au discours, lorsqu´ils ne le remplacent pas. Les gestes de battements sont d´avantage d´ordre privé. Ils permettent au locuteur de s´exprimer sans apporter de sens au discours. Ils peuvent d´ailleurs facilement être supprimés.

1.1.3 Sémiotique ou ergotique ?

Dans le contexte d´interaction homme–machine, le geste sémiotique semble le plus approprié. Il s´agit, dans ce contexte, de la production d´un message informationnel à un système informatique. Cependant, comme le souligne MIGNOT [Mig95], la distinction entre la fonction ergotique et la fonction sémiotique est parfois difficile. Dans une application de «*manipulation directe*»[8], comme celles présentées par PAVLOVIĆ et al. [PSH96, SHP+96] ou MIGNOT [Mig95, DMV93], le déplacement d´un objet est–il un geste ergotique ou sémiotique? Selon que nous nous plaçons du point de vue de l´utilisateur ou du concepteur, cette opinion est différente. Pour l´utilisateur, le geste lui permet de manipuler ou déplacer son objet (virtuel), il s´agit donc d´un geste ergotique. Du point de vue du concepteur du système, le geste permet de fournir au système des informations (des coordonnées par exemple) permettant à ce dernier de déplacer ou modifier l´objet ; il s´agit donc d´un geste sémiotique. MIGNOT préfère une distinction entre *communiquer* et *manipuler*. Elle est à rapprocher de la distinction proposée par POUTEAU *et al* [PRP94], entre une logique de *faire*, dans laquelle «les objets réagissent aux opérations de manipulation», et une logique de *faire–faire*, où «l´opérateur indique à l´exécutant informatique les opérations à effectuer».

1.2 Nouvelles interactions homme–machine gestuelles

L´objectif de recherche dans le domaine des interactions homme–machine est de développer des modèles, des concepts, des outils et des méthodes pour réaliser des systèmes qui répondent aux besoins et aux aptitudes des utilisateurs. Une des aptitudes des utilisateurs est l´emploi des gestes selon les modes précisés à la section précédente. Nous présentons ici l´utilisation et la création d´applications gestuelles.

Dans un premier temps, une étude de l´utilisabilité du geste est proposée, elle s´appuie principalement sur les interacions multimodales et des expérimentations de type *Magicien d´Oz*. Un ensemble d´exemples de systèmes utilisant les gestes comme interaction est proposé dans trois catégories : reconnaissance de la langue des signes, gestes dans le domaine de la réalité virtuelle et dans celui de la réalité augmentée.

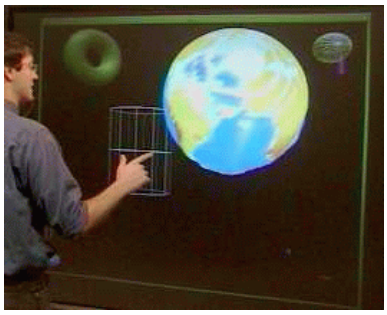
8. Dans un système de *manipulation directe*, l´utilisateur contrôle directement les objets, ceux–ci étant souvent des métaphores d´objets physiques tels le bureau et les fichiers. Ce type de manipulation a été créé par le Xerox Alto et popularisé par les interfaces du Apple Macintosh et de Microsoft Windows.

1.2.1 Utilisabilité des gestes dans les interactions homme–machine

Avant de considérer les techniques permettant la reconnaissance de gestes, il convient, dans un premier temps, d´étudier l´utilisabilité des gestes dans des systèmes informatiques. Cette étude permet de vérifier si, lorsqu´une telle interaction est disponible dans un système, elle est réellement employée par les utilisateurs, et si elle apporte une amélioration par rapport aux interactions classiques. L´absence d´amélioration et d´emploi entraînerait alors l´abandon du domaine.

Les études, que nous citons ici, s´appuient sur des interactions multimodales et sur des expérimentations de type *Magicien d´Oz*. Définissons dans un premier temps ces deux concepts.

«Put that ...»
«Prends ça ...»



«there»
«mets le ici»

FIG. 1.6 – ***Interactions multimodales dans le système « Put that there » de*** BOLT ***[Bol80]***. *(extrait de [LZG98], d´après [Bol80])*

Interactions multimodales : Le concept d´interaction multimodale réfère aux «canaux sensoriels de l´être humain» [Cae91]. Pour DAUCHY *et al* [DMV93], l´idée est de proposer à l´utilisateur un large éventail de moyens de communication selon ses besoins. Un système informatique multimodal est capable de communiquer avec ses utilisateurs en respectant les modalités de la communication humaine. Cette communication s´effectue par la vision (voir l´utilisateur et lui afficher des informations), par la parole (entendue et produite) et par le geste (mouvement, désignation, écriture et dessin). Ces systèmes doivent être équipés du matériel et des logiciels adaptés à l´acquisition, à la restitution et à la compréhension des énoncés multimodaux. Ainsi, une station à caractère multimodal

doit être pourvue de microphones pour la reconnaissance de la parole, de caméras et de gants numériques pour l´acquisition des mouvements et des gestes, de palettes graphiques et de scanner pour la compréhension de l´écriture, de haut–parleurs et de synthétiseurs de vocaux pour la production de sons, de musiques et de message vocaux, de moniteurs, projecteurs et lunettes spéciales pour la visualisation de graphiques, images naturelles ou de synthèse. Un bon exemple de système multimodal est le système «Put that there»[9] développé par BOLT [Bol80]. Il a été le premier à proposer un système dans lequel l´utilisateur pouvait utiliser conjointement la parole et les gestes de désignation pour commander des évènements de manipulation d´objets graphiques sur un écran géant.

FIG. 1.7 – ***Environnement physique de la technique du magicien d´Oz****. Par le biais de microphones et de caméras, le comportement d´un utilisateur (*a*) face au système est étudié. Les fonctions manquantes du système sont simulées par un compère (*b*). (d´après [Hau89])*

Magicien d´Oz : La technique du *Magicien d´Oz*[10] consiste à étudier, par le biais de microphones et de caméras, le comportement d´un utilisateur face au système. Les fonctions manquantes du système sont simulées par un compère. La figure 1.7 présente la disposition physique du Magicien d´Oz. L´enregistrement vidéo et sonore de l´utilisateur

9. «Mets ça ici» ou «Prends ça ... mets–le ici»
10. Oz Wizard

ainsi que les évènements informatiques permettent ensuite aux concepteurs du système, aux psychologues et aux ergonomes d´étudier les problèmes auxquels l´utilisateur a été confronté. Dans les expérimentations suivantes, elles permettront d´étudier l´utilisation des gestes et de répondre aux questions :

- le geste est–il utilisé?
- dans quelle situation l´est–il?
- quel type de geste est utilisé?
- quel lien existe–t–il entre le geste et la parole?
- ...

Expérimentations : HAUPTMANN *et al* [HMS88, Hau89] ont évalué l´efficacité des gestes pour manipuler des objets graphiques. La technique du Magicien d´Oz a été utilisée pour observer les utilisateurs devant manipuler un cube pour le faire correspondre à un modèle en utilisant la voix et/ou le geste. Cette étude apporte des résultats intéressants sur la coopération des modalités mais surtout sur l´utilisation des mains comme moyen de manipulation d´objets informatiques. Ainsi, nous apprenons que les utilisateurs réalisent des gestes avec deux ou trois doigts. Les gestes se font dans les trois dimensions. Une autre étude, réalisée par DAUCHY [DMV93] montre que la réalisation de tâches complexes, comme par exemple une rotation ou la combinaison rotation/déplacement, ont tendance à être multimodales. La parole est utilisée pour nommer l´action et les gestes pour spécifier les paramètres. Dans les autres cas, la parole et le geste sont souvent redondants et seul l´un des deux modes est suffisant. Le geste est souvent porteur de la commande tandis que la parole n´est que commentaire. Il apparaît que les utilisateurs préfèrent des gestes simples et rapides plutôt que des commandes multimodales complexes pour les tâches simples ! Il est en effet souvent plus facile d´exécuter une manipulation que d´énoncer les opérations oralement.

1.2.2 Utilisation de gestes

Trois principales catégories d´applications profitent des débuts de la reconnaissance de gestes. La reconnaissance des langues des signes est un domaine important de la reconnaissance des gestes, comme en témoigne le grand nombre d´articles qui y sont consacrés lors du colloque sur le gestes : «Gesture Workshop» de mars 1999 à Gif–sur–Yvette [BGG+99]. Deux autres courants de l´interaction homme–machine utilisent des langues de commandes gestuelles : la réalité virtuelle et la réalité augmentée.

Dans cette section, nous présentons quelques systèmes de reconnaissance de gestes de la langue des signes. Les périphériques utilisés par ces systèmes sont aussi bien les gants numériques que la vision par ordinateur. Cependant, comme nous le verrons, les systèmes reconnaissant le plus grand nombre de signes de la langue des signes sont ceux utilisant les gants numériques. BRAFFORT [Bra96] aréférencé 32 systèmes de reconnaissance de la langue des signes, parmi lesquels 90% utilisent des gants numériques.

Nous nous intéressons ensuite à la réalité virtuelle, et plus particulière à l´utilisation des gestes pour la réalité virtuelle. Les systèmes de réalité virtuelle plongent les utilisateurs dans un environnement 3D virtuel. Celui–ci peut être éducatif, pour modéliser des phénomènes physiques (la description de molécules chimiques) ou reproduire des lieux ayant existés (tel que le palais de Ramses II). Nécessitant une interaction 3D, ces systèmes utilisent principalement les gants numériques. Les interfaces «sans fil» sont peu répandus dans ce domaine puisque bien souvent l´utilisateur porte, en plus du gant numérique, un casque de vision 3D.

La réalité augmentée est la dernière catégorie présentée ici. Elle vise, selon MACKAY à «fusionner le monde physique réel et le monde électronique et informatisé»[11][Mac98]. Selon les applications et la volonté de construire une interface non–intrusive, ces systèmes utilisent aussi bien les gants numériques que la vision par ordinateur.

a) Reconnaissance de langues des signes

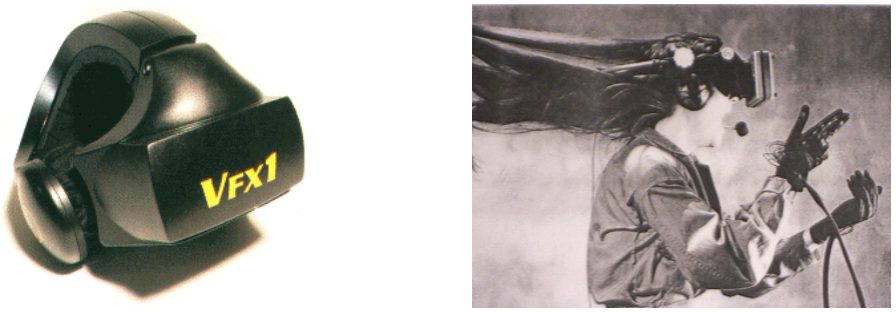
L´importance de la langue des signes, par rapport aux autres types d´interactions, est sa richesse et la précision de son sens. Selon EDWARDS [Edw97], deux motivations principales existent pour l´étude des langues des signes. La première est une considération pratique : dans de nombreux cas, la reconnaissance automatique de langues des signes est un apport important pour la communauté des sourds–muets. Un système de conversion signes–paroles permettrait à des signeurs de discuter avec des non signeurs ou bien de proposer des «téléphones de la langue des signes» tel celui étudié par les laboratoires d´HITACHI [Ohk95]. L´étude de la langue des signes est effectuée aussi bien dans le domaine de la reconnaissance que de la génération [BGG+99]. Une seconde application est la création de documents en langue des signes. Les documents composés de signes sont, pour EDWARDS, davantage lisibles pour les signeurs puisqu´ils sont écrits dans la première langue alors que les textes sont plus naturels pour les langues parlées. La seconde raison est la proposition d´une structure d´entrée gestuelle. Une troisième considération, non mentionnée par EDWARDS, est l´étude linguistique de la langue [Cux99].

b) Gestes en réalité virtuelle

La réalité virtuelle plonge l´utilisateur dans un nouveau monde où les objets sont électroniques mais les actions réelles. La figure 1.8 montre un exemple de réalité virtuelle. L´utilisateur effectue une action physique (pédaler) et réelle sur des objets virtuels. Le décor projeté dans le casque de visualisation est entièrement généré par des images de synthèse. Les périphériques classiques de la réalité virtuelle sont le gant numérique et des lunettes de projection. Ceux–ci sont présentés par les figures 1.9 et 1.16[12].

11. La définition de MACKAY est «linking real and virtual worls»
12. la figure se trouve à la page 34

FIG. 1.8 – ***Exemple de réalité augmentée.*** *L´utilisateur effectue une action physique réelle cependant le décor, projeté dans le casque, est généré par des images de synthèse. (extrait de [Kru91])*



(*a*) (*b*)

FIG. 1.9 – ***Périphériques pour la réalité augmentée.*** *(*a*) photographie d´un casque de projection. (*b*) Dispositifs complets sur un utilisateur : gants et lunettes (extrait de [Kru91]). Le schéma et une photographie d´un gant numérique sont proposés par la figure 1.16 (page 34)*

Les applications principales de la réalité virtuelle sont les applications de simulation. Le système CAVE[13] est certainement l´environnement de réalité virtuelle le plus connu [CNSD93]. RHEINGOLD [Rhe91] définit ces environnements dans lesquels une personne est «entourée par une représentation tri–dimensionnelle, générée par ordinateur, et est capable de se déplacer dans un monde virtuel et le découvrir sous différents angles, l´atteindre de l´intérieur, le saisir et le redimensionner.»[14]

Un autre exemple est `MDScope`, une application proposée par SHARMA *et al* [SHP+96, PSH96]. Elle fournit un environnement pour la simulation et la visualisation de composés biomoléculaires en biologie structurelle. Ce système utilise la double–modalité, geste et parole, pour le contrôle des molécules.

c) Gestes en réalité augmentée

La *réalité augmentée* ou *réalité informatisée*, pour sa part, laisse l´utilisateur dans un monde physique dans lequel les actions peuvent être virtuelles [WMG93]. Ce concept est issu de la constatation que les ordinateurs fournissent des fonctionnalités très intéressantes telles que la capacité de reproduire, de correction automatique ou de traduction automatique. Mais la manipulation de ces objets électroniques est limitée par les périphériques d´entrée. La manipulation des objets physiques profite, pour sa part, de notre habileté et de notre habitude [Bér94]. Nos mains nous permettent des manipulations difficilement reproductibles avec un système informatique. Cette double constatation a introduit le concept de réalité augmentée qui «mélange le système électronique dans le monde physique plutôt que de tenter de le remplacer»[15] [Mac98]. Pour MACKAY [Mac96], un système de réalité augmentée permet le «mélange des deux mondes». La réalité augmentée est l´enrichissement par une réalité virtuelle du monde réel [FMS93].

Il existe trois méthodes non exclusives, pour réaliser une application de réalité augmentée [Mac98, Mac96] :

augmenter l´utilisateur : celui–ci porte des appareils spéciaux tel qu´un casque de réalité augmentée ou un gant numérique.

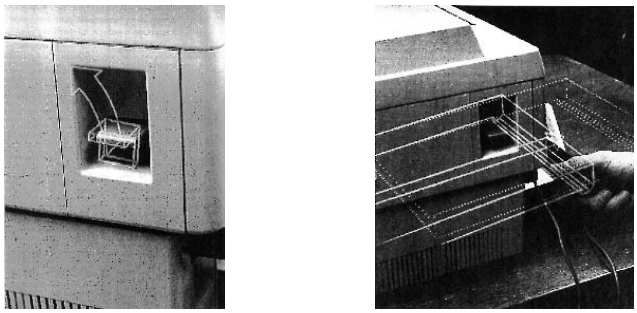
augmenter les objets : dans cette méthode, des dispositifs, tels que des capteurs, sont incorporés dans l´objet.

augmenter l´environnement : il dispose de capteurs indépendants permettant la projection et la capture des objets et des utilisateurs.

13. CAVE est l´acronyme récursif de «CAVE Automatic Virtual Environment»

14. «surronded by a three–dimensional computer–generated representation, and is able to move around in the virtual world, to reach into it, grab it, and reshape it.»

15. «merge electronic systems into the physical world instead of attempting to replace them»



(*a*) (*b*)

FIG. 1.10 – ***Maintenance d´une imprimante avec le système de réalité augmentée*** KARMA. *L´utilisateur voit les manipulations à effectuer (*a*) et les pièces internes de l´imprimante (*b*). (extrait de [FMS93])*

Augmenter l´utilisateur : le système KARMA[16] [FMS93] propose d´augmenter l´utilisateur pour la maintenance d´imprimantes. L´utilisateur est équipé de lunettes spéciales dans lesquelles il voit le monde réel sur lequel sont superposés des objets virtuels. La figure 1.10 présente la vision que l´utilisateur a de la scène. Les flèches superposées à l´image permettent à l´utilisateur de connaître les manipulations à effectuer (cf. figure 1.10*a*) et affiche les pièces internes de l´imprimante (cf. figure 1.10*b*).

L´utilisateur de `Charade` [BBL93] est augmenté d´un gant numérique relié à une station de travail pilotant un logiciel de présentation assistée par ordinateur. Ce gant autorise deux types de manipulations : la première permet de positionner un pointeur sur l´écran correspondant à la projection de la main ; la seconde donne des directives au système telles que projeter la diapositive suivante ou lancer la vidéo. Le système a été créé de telle sorte que les gestes privés, c´est–à–dire ceux effectués par le locuteur pendant sa présentation, ne trompent pas le système. En fait, les gestes de commande sont très contraints et ne peuvent être réalisés accidentellement. Ils correspondent à une configuration particulière de la main, un mouvement au cours duquel la configuration est changée en une configuration finale. La figure 1.11(*b*) illustre l´exemple du geste «chapitre suivant». Le geste de positionnement du pointeur est lui dirigé vers l´écran.

Augmenter les objets consiste à ajouter des composants électroniques et des logiciels spécialisés dans les objets. WEISER parle d´*informatique partout* ou d´*informatique*

16. **K**nowledge–based **A**ugmented **R**eality for **M**aintenance **A**ssistance

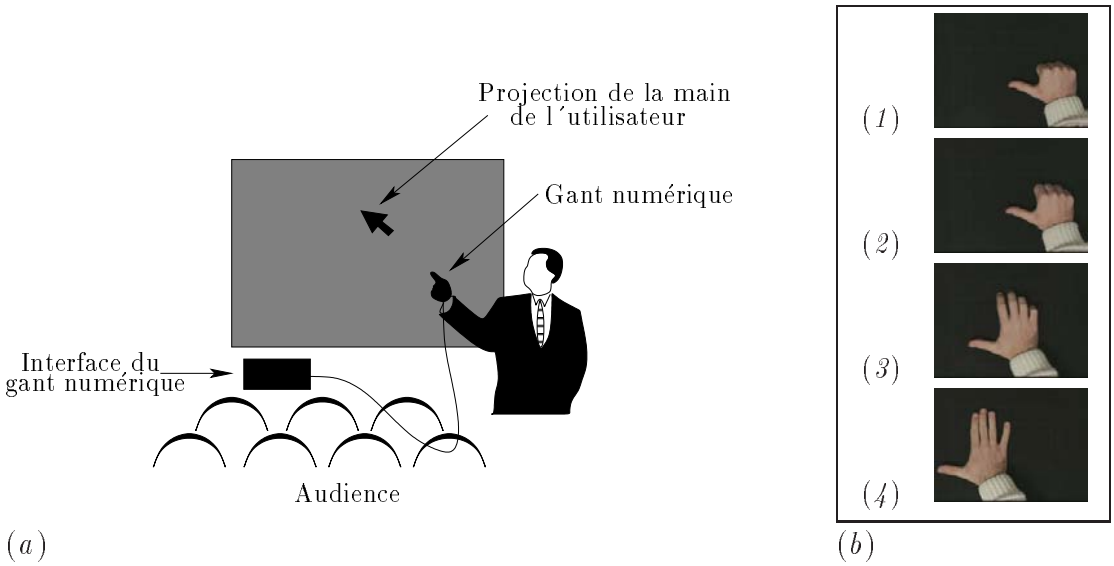


FIG. 1.11 – ***Le Système*** `Charade`. *(a) L´utilisateur du système est augmenté d´un gant numérique relié à une station de travail pilotant un logiciel de présentation assistée par ordinateur. (b) Exemple du geste «chapitre suivant». (d´après [BBL93])*

intégrée [17]. Dans cette nouvelle forme d´interaction, l´informatique est incorporée dans tous les objets de notre vie ordinaire : tasse, chaise, *etc.* Des centaines d´ordinateurs seront alors présents dans une pièce et communiqueront par réseaux locaux ou infra–rouge. Le projet «Adaptative House» [Moz98, Moz] vise à créer une maison dans laquelle tous les composants, interrupteurs, lampes, chauffage, sont connectés entre eux et permettent une prise en charge complète de la maison par l´informatique. Ce domaine de recherche n´ayant plus de lien avec la reconnaissance de gestes, nous n´allons pas plus loin dans son étude.

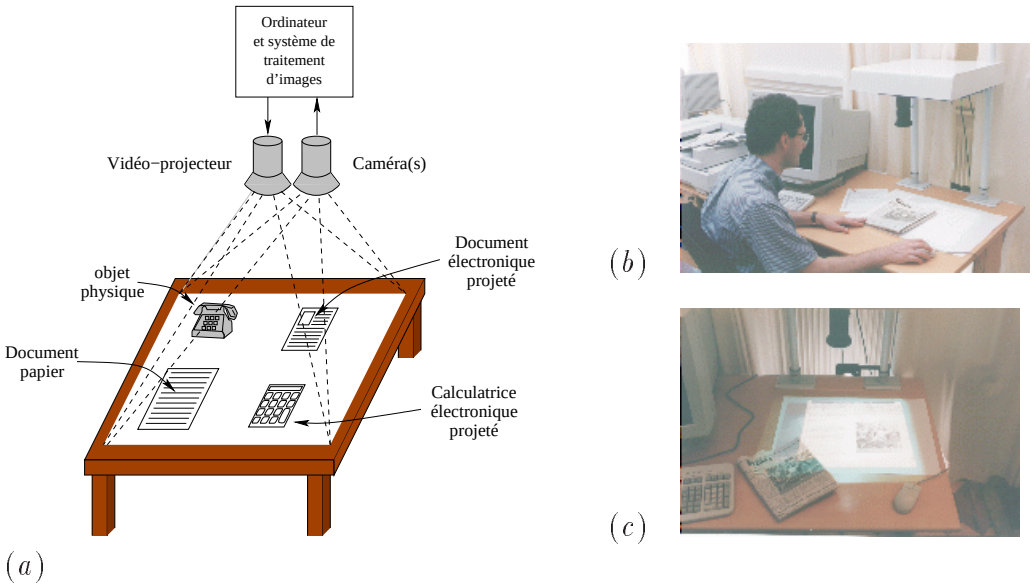
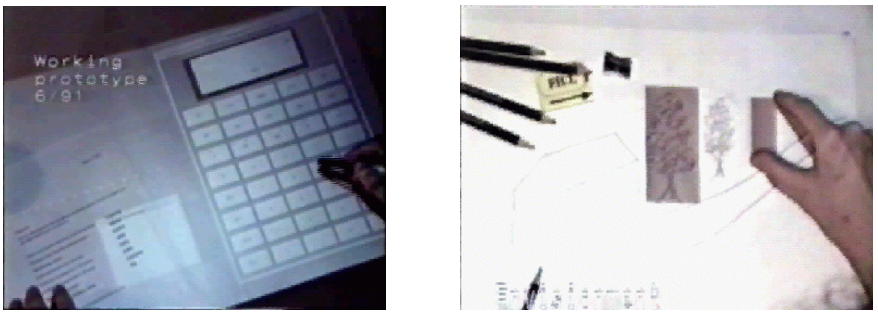


FIG. 1.12 – ***Configuration du bureau digital.*** *Le bureau physique est enrichi d´objets et de fonctions électroniques par l´intermédiaire d´une caméra et d´un vidéo–projecteur. Ce dernier permet la projection d´objets informatiques sur le bureau tandis que la caméra est utilisée pour l´acquisition des objets physiques et des gestes réalisés par l´utilisateur. (*a*) Schéma explicatif (d´après [NW92]). (*b*) et (*c*) vues différentes (extraites de [XTS98])*

17. *ubiquitous computing*

Augmenter l´environnement consiste à ajouter à l´environnement des capteurs et des effecteurs permettant, par exemple, de localiser un utilisateur, sans l´avoir augmenté d´un badge actif, et de lui projeter des informations. Un exemple d´environnement développé par WELLNER [Wel91a, Wel91b, NW92, Wel93b] est le «`Bureau Numérique`» ou «`DigitalDesk`». Le bureau physique est enrichi d´objets et de fonctions électroniques par l´intermédiaire d´une caméra et d´un vidéo–projecteur. Le vidéo–projecteur permet la projection d´objets informatiques sur le bureau tandis que la caméra est utilisée pour l´acquisition des objets physiques et des gestes réalisés par l´utilisateur. La figure 1.12 montre un schéma du système ainsi que deux vues de celui–ci.



(*a*) (*b*)

FIG. 1.13 – ***Deux utilisations du*** *«`Bureau Numérique`» (*a*) Addition de chiffres par «glisser–déposer» sur une calculatrice virtuelle. (*b*) «Copier–coller» d´un arbre dessiné au crayon (extrait de [Wel91b])*

Par l´intermédiaire d´un «glisser–déposer»[18], l´utilisateur peut additionner des chiffres inscrits sur une feuille de papier avec une calculatrice informatique projetée sur le bureau physique (cf. figure 1.13*a*). Il peut également effectuer un dessin sur une feuille de papier à l´aide d´un crayon physique puis effectuer un «copier–coller» du dessin physique sur la feuille (cf. figure 1.13*b*) pour finalement effacer les deux dessins, le dessin réel et le dessin virtuel, avec sa gomme physique. D´autres bureaux numériques ont été développés pour des applications particulières [Mac98, Mac96] telle que l´édition de StoryBoard [MP94], l´aide à la gestion de contrôle aérien [MFFM98]. Dans le chapitre 6, nous verrons le tableau magique dans lequel nous avons augmenté un tableau blanc mural.

18. il s´agit d´une opération consistant, dans les interfaces à manipulation directe, à sélectionner un objet avec la souris, déplacer le pointeur en maintenant le bouton de la souris et à relâcher le bouton à l´endroit où on veut le déposer.

Dans le «`Bureau Numérique`», seul le bureau est augmenté. Si toute la pièce est augmentée, nous parlons alors d´*environnement interactif*. La présentation des environnements ainsi que le prototype que nous développons sont proposés au chapitre 6.

1.2.3 Définition de gestes

Après avoir défini, dans la section 1.1, le geste d´un point de vue socio–psychologique et avoir considéré son utilisation dans des catégories d´interaction homme–machine, intéressons–nous aux gestes d´un point de vue pratique. Considérons le développeur d´un système utilisant le geste comme moyen de communication et définissons la notion de geste.

a) Classes d´utilisation des gestes

Les premières définitions des gestes peuvent être celles de EDWARDS [Edw97]. Il définit trois types de gestes différents :

les gestes naturels pour les personnes. Ils sont généralement employés lors d´interactions entre elles. Ils présentent un grand intérêt pour la communication médiatisée[19].

les gestes synthétiques utilisés et spécialement définis dans des applications de communication homme–machine. Ils ont parfois été choisis pour leur facilité de reconnaissance au détriment du confort d´utilisation.

les gestes pour l´interaction en environnement virtuel correspondent à des gestes de manipulations d´objets dans une scène virtuelle. Ils sont naturels dans le sens où ils sont réalisés par les personnes dans la vie quotidienne ; cependant, l´absence d´objets physiques les classe dans une catégorie particulière.

Nous pouvons également ajouter une définition des gestes pour l´interaction en réalité augmentée. Ces gestes sont une fusion des gestes synthétiques et des gestes pour l´interaction en environnement virtuel :

les gestes pour la réalité augmentée . Ils correspondent à des gestes de manipulations d´objets dans une scène réelle et augmentée. Ils sont à la fois naturels puisqu´ils sont réalisés par les personnes dans la vie quotidienne mais ils peuvent s´effectuer sur des objets physiques ou des objets virtuels. De plus, ils peuvent être également complètement synthétiques, comme par exemple, dessiner la lettre `C` dans le «`Bureau Numérique`» pour obtenir la calculatrice virtuelle.

19. Il s´agit d´une communication entre personnes mais où l´informatique prend une importance comme par exemple un suivi automatique de l´intervenant lors d´une vidéo–conférence.

Cette taxonomie est intéressante car elle permet de définir un ordre de complexité dans la considération des gestes. Les gestes naturels sont généralement ignorés. Ils présentent un intérêt pour les personnes en communication mais n´en présentent aucun pour le système. Dans des applications médiatisées où le système réagit à certains gestes des utilisateurs, telle que celle présentée par GONG *et al.* [Gon97, HB98, HB99], l´application est alors une application de réalité augmentée. La caméra est augmentée de fonctions permettant son déplacement au travers de commandes gestuelles.

La reconnaissance des gestes synthétiques est très simple, les gestes sont choisis pour être facilement discriminés. Ils sont généralement directement liés avec le périphérique et l´algorithme employé (cf. section 2).

Les gestes pour l´interaction en environnement virtuel sont difficiles à reconnaître car les classes ne sont pas réellement définies. Nous nous plaçons dans un contexte dans lequel l´utilisateur est libre des gestes qu´il réalise. Les gestes pour la réalité augmentée présentent la même difficulté, mais une difficulté supplémentaire est la distinction entre les gestes à destination du système et les gestes naturels. Il convient, souvent, de connaître si l´objet manipulé est réel ou virtuel. S´il est réel, il faut alors le reconnaître pour vérifier quelles interactions informatiques existent

b) Classes de dynamicité

Une seconde classification des gestes concerne leur dynamicité. HUANG et al [HP95] distinguent deux catégories de gestes : les gestes statiques et les gestes dynamiques. HARLING et EDWARDS [HE96, Edw97] proposent une classification plus précise des gestes en quatre catégories :

- Configuration statique, position statique (SPSL)[19]
- Configuration dynamique, position statique (DPSL)[19]
- Configuration statique, position dynamique (SPDL)[19]
- Configuration dynamique, position dynamique (DPDL)[19]

Ils considèrent les gestes de la langue des signes mais cette classification reste vraie pour les gestes liés aux interactions homme–machine. Cependant, la distinction entre configuration et position n´est pas nécessaire. Il est possible de définir un état de la main, à un instant t_i, par le couple «position–configuration». Ce couple est en réalité un vecteur v_{t_i} à n dimensions. Ces dimensions représentent des mesures de la position ou de la configuration. Ainsi, un geste peut être défini par l´ensemble des vecteurs v_t avec $i \in \{1 \dots T\}$. Cet ensemble de vecteurs est une trajectoire, paramétrée par le temps et définie dans l´intervalle $I = \{1, \dots T\}$, dans l´espace des mesures de v.

La figure 1.14 présente un exemple de deux gestes dans un espace de mesures limité à trois dimensions. Ces dimensions peuvent, par exemple, être la position de la main dans l´espace et les deux trajectoires correspondre aux gestes «bonjour» et «au revoir».

19. Respectivement «*static posture, static location*» ; «*dynamic posture, static location*» ; «*static posture, dynamic location*», «*dynamic posture, dynamic location*»

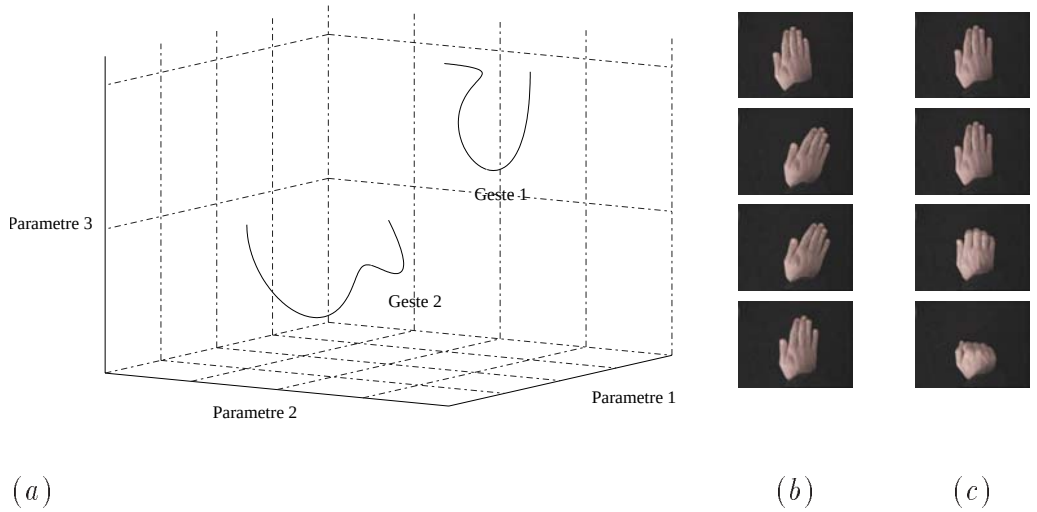


FIG. 1.14 – ***Exemple de trajectoire de deux gestes dans un espace de mesures à trois dimensions***. (a) *trajectoire des deux gestes représentés par les images* (b) *et* (c). *(d´après [Mar95b] et [PSH97])*

c) Étapes de la reconnaissance de gestes

La reconnaissance de gestes est découpée en trois étapes [Mar95b] :

1. **étape d´analyse** dans laquelle sont calculés les paramètres de la main, c´est–à–dire la position et la configuration. Cette étape permet la création de la trajectoire du geste dans l´espace des mesures.

2. **étape de reconnaissance** ; l´analyse spatio–temporelle de la trajectoire permet la classification de la trajectoire parmi l´ensemble des trajectoires connues du système. Un symbole est généré pour représenter le geste

3. **étape d´interprétation** ; il s´agit du dernier niveau. Elle fait partie de l´application utilisant l´interpréteur de geste. Le symbole est utilisé pour effectuer les actions correspondant au geste.

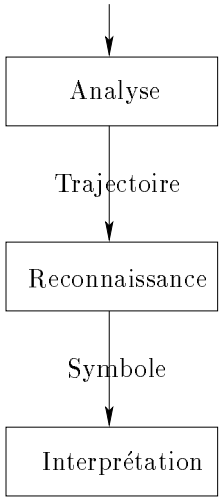


FIG. 1.15 – ***Étapes de la reconnaissance de gestes.*** *L´étape d´analyse calcule les paramètres de la main créant une trajectoire. L´analyse spatio–temporelle de la trajectoire lors de l´étape de reconnaissance permet la classification de la trajectoire. Cette classe est définie par un symbole. L´étape d´interprétation effectue la correspondance entre le geste et l´action à réaliser par le système.*

2 Comment reconnaître les gestes ?

Cette section présente un état de l´art des techniques d´analyse de gestes. HUANG et PAVLOVIĆ [HP95, PSH97] déterminent quatre classes principales de techniques d´analyse de gestes :

- les techniques utilisant des gestes de dessins ;
- les techniques basées sur des gants ;
- les techniques visuelles ;
- les autres techniques.

Dans cette étude, nous nous intéressons plus particulièrement aux techniques visuelles mais présentons brièvement les techniques utilisant des tablettes graphiques et des gants numériques. Les «autres techniques», étant marginales et basées souvent sur des capteurs particuliers, ne sont pas répertoriées ici.

2.1 Analyse de gestes de dessins

L´approche de reconnaissance de gestes de dessins s´appuie souvent sur l´utilisation d´une tablette graphique. Elle mesure la position du stylo et retourne ses coordonnées à l´ordinateur. Certaines tablettes graphiques évoluées retournent également une mesure de la pression exercée sur le stylo. Dans de nombreux systèmes de reconnaissance de gestes de dessins, le couple tablette–stylo est remplacé par la souris dans ce cas, les boutons de la souris permettent d´augmenter les gestes. Les écrans tactiles sont également des périphériques permettant la reconnaissance de gestes de dessins.

Les gestes réalisés avec ces périphériques sont des marques définissant des commandes dans différentes applications. Ils ont pour objectifs de remplacer les traditionnels menus, en particulier dans des systèmes où la place est limitée pour de telles interfaces comme par exemple sur les *assistants personnels*, PalmPilot© ou Newton© [20]. Ils peuvent également aboutir à la reconnaissance de textes ou de chiffres [LX96].

GRANDMA [21] est une application développée par RUBINE [Rub91, Rub92]. Il utilise des gestes effectués sur un écran tactile pour dessiner, copier, déplacer ou effacer des formes géométriques. Une seconde application de RUBINE, nommée GSCORE permet l´édition de partitions musicales.

RUBINE [Rub91] précise que l´avantage d´un système basé sur les gestes de dessins donne la possibilité de spécifier en même temps l´objet, la commande et des paramètres additionnels. L´objet est sélectionné par le début du geste, la forme du geste détermine la commande, la taille et l´orientation déterminent des paramètres de la commande. Cependant, ces gestes restent limités à des mouvements sur un plan.

20. Il s´agit de l´assistant développé par Apple Computer Inc

21. GRANDMA est l´acronyme de «*Gesture Recognizers Automated in a Novel Direct Manipulation Architecture*», c´est–à–dire «Système de reconnaissance automatique de geste dans une nouvelle architecture à manipulation directe»

2.2 Utilisation de gants numériques

Avant de proposer des applications utilisant un gant numérique, présentons brièvement son fonctionnement. Des fibres optiques placées sur chaque doigt permettent de mesurer les angles de deux flexions des doigts. L´intensité du signal lumineux envoyé dans la fibre et son intensité au bout du doigt permettent de déterminer ces angles. De plus, un capteur de position et d´orientation de la main est ajouté. Il est composé de deux modules. Le premier, l´émetteur, est positionné dans un endroit fixe et émet trois champs électromagnétiques orientés. Le second, le récepteur, est fixé sur le gant. La réception de ces champs magnétiques produit trois courants induits dont les valeurs sont proportionnelles à la position et l´orientation du récepteur par rapport à l´émetteur. L´un des gants le plus utilisé est le *DataGlove*© [22]. Le mot dataglove est souvent utilisé pour désigner un gant numérique. Ce gant est illustré par la figure 1.16.

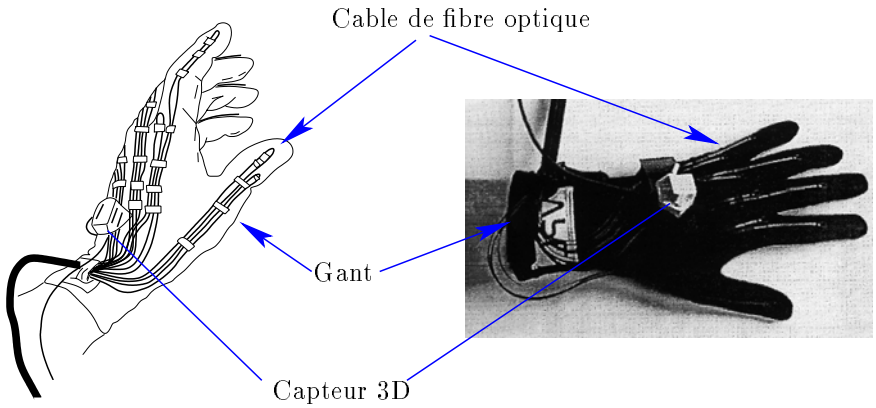


FIG. 1.16 – ***Gant numérique* DataGlove *de la société* VPL** *(extrait de [Bra96] et [MHO91])*

Depuis longtemps, HUANG et PAVLOVIĆ [HP95] l´estiment à la fin des années 1970, le gant numérique permet la reconnaissance de gestes pour différentes applications. Une des applications privilégiées est la reconnaissance des signes. BRAFFORT [Bra96] a identifié 31 systèmes différents utilisant un gant numérique. Comme nous l´avons vu précédemment à la section 1.2.2, les gestes de la langue des signes ont une richesse et une imprécision. Un système de reconnaissance de ces gestes des signes doit également nécessiter une richesse et une précision, de quelques degrés, que peut lui apporter un gant numérique. *Glove–Talk*

22. de la société VPL

est le premier système de reconnaissance de signes. FELS et HINTON [FH93] y utilisent un ensemble de cinq réseaux de neurones pour traduire un vocabulaire de 203 mots. La traduction est alors énoncée par un synthétiseur vocal.

Le système `Charade` [BBL93], proposé par BAUDEL et BEAUDOUIN-LAFON, utilise des gestes effectués avec un gant numérique pour contrôler une présentation assistée par ordinateur. Le système fonctionne en temps réel et est capable de reconnaître 16 commandes. Toutes les commandes sont constituées de trois phases : une configuration de la main au début du geste, la direction du mouvement et la configuration de la main à la fin du geste. Cette application est également un exemple réussi de réalité augmentée. Elle est décrite plus précisément à la section 1.2.

Le système, développé par MORITA, HSAHIMOTO et OHTERU [MHO91], permet la reconnaissance des gestes effectués par un chef d´orchestre. Un gant numérique est utilisé pour la main gauche et la baguette de direction est équipée d´une lumière infra–rouge. Une caméra, équipée d´un filtre infra–rouge et positionnée face au chef d´orchestre, détecte le mouvement de la baguette. La fusion de la reconnaissance du geste de la main gauche et du mouvement de la baguette pilote un synthétiseur MIDI simulant ainsi l´interaction du chef d´orchestre avec son orchestre.

TUNG et KAK [TK95] proposent un système de programmation automatique de robots par observation d´un opérateur. Ce dernier effectue une opération d´assemblage, le système apprend le geste de l´utilisateur et le reproduit.

ONISHI *et al* [OTK93] utilisent seulement un ensemble de capteurs Promothéus© attachés aux extrémités des doigts et à la paume. Un réseau de neurones récurrent permet la reconnaissance de gestes pour la manipulation d´objets graphiques. Les gestes considérés sont des gestes pour la rotation, la translation, l´agrandissement et le rétrécissement pour lesquels seule la position spatiale de la main est nécessaire.

2.3 Reconnaissance visuelle de gestes

L´analyse de gestes en vision par ordinateur est la solution technique la plus naturelle. Elle libère l´utilisateur de l´emploi de périphériques, tels le gant numérique ou le stylo, et s´appuie sur la technique par laquelle nous, humains, percevons les gestes. Cette solution est la plus difficile à mettre en oeuvre compte–tenu des limitations actuelles en vision par ordinateur. Cependant, des solutions sont apportées en admettant des restrictions telle que l´utilisation de marqueurs (passifs ou actifs), des fonds uniformes ou en définissant un vocabulaire réduit de gestes.

Ces solutions sont souvent basées sur le modèle «pipeline» classique en vision par ordinateur [KS98] :

1. acquisition
2. segmentation
3. extraction de caractéristiques
4. classification

Lors de la première étape d´*acquisition*, les images prises par la caméra sont digitalisées et préparées pour les traitements ultérieurs. Cette préparation peut également inclure un changement de codage du format des images, comme le propose LYONS et PELLETIER [LP99]. La seconde étape sépare la région de l´image contenant la main du fond. La troisième étape extrait des caractéristiques de la région segmentée permettant la différenciation des gestes considérés. Enfin, la classification est effectuée. PAVLOVIĆ *et al* [PSH97] représentent un système d´interprétation de gestes par le schéma de la figure 1.17. La grammaire permet de refléter la syntaxe des commandes gestuelles mais également le lien avec d´autres types de modalités telle que la parole.

KOHLER [KS98] a identifié 40 systèmes de reconnaissance visuelle de gestes, divisées en trois catégories :

1. des marqueurs ou des gants marqués ;
2. des modèles 3D ;
3. l´apparence de l´image.

2.3.1 Approches basées sur des marqueurs ou des gants marqués

Cette approche permet de simplifier le problème de détection de la configuration de la main. Des marqueurs sont positionnés sur les extrémités des doigts ou bien des gants de couleurs sont portés par l´utilisateur. Ces marqueurs sont facilement détectables dans les images vidéos par des algorithmes classiques en vision par ordinateur.

DAVIS et SHAH [DS93, DS94] utilisent des marques blanches collées sur les doigts. Les gestes sont exécutés face à la caméra. Dans chaque image, les marques sont extraites en utilisant une segmentation. Le seuil de la segmentation est défini automatiquement en utilisant un histogramme. Le centre des marques segmentées est calculé. L´ensemble de ces centres permet de définir les vecteurs de direction et la magnitude. Un geste est alors modélisé par le n–uplet : $< d_i, m_i >_{i=1,...5}$ où d_i est la direction et m_i la magnitude du mouvement du i^{e} doigt. Ce vecteur est transformé en un code de mouvement de 5 bits déterminant le mouvement de chacun des doigts. Le i^{e} bit est mis à 1 si le i^{e} doigt à bougé, c´est–à–dire si la magnitude du mouvement est supérieur à une valeur prédéfinie. La reconnaissance d´un geste est effectuée par la recherche du code de mouvement dans une table.

YACHIDA et IWAI [YI96, IWYY96] proposent un système de reconnaissance de langue des signes en temps réel. Les utilisateurs prennent un gant coloré composé de 12 parties : 2 par doigts, la paume et le poignet, la figure 1.18*a* montre ce gant. Chaque partie porte une couleur différente, permettant de facilement les extraire et les différencier. À partir de ces zones de couleurs, quatre caractéristiques sont calculées :

- l´aire des différentes régions ;
- les vecteurs entre la zone du poignet et les zones des doigts ;
- les vecteurs entre la partie haute et la partie basse d´un doigt ;
- les vecteurs entre les extrémités des doigts.

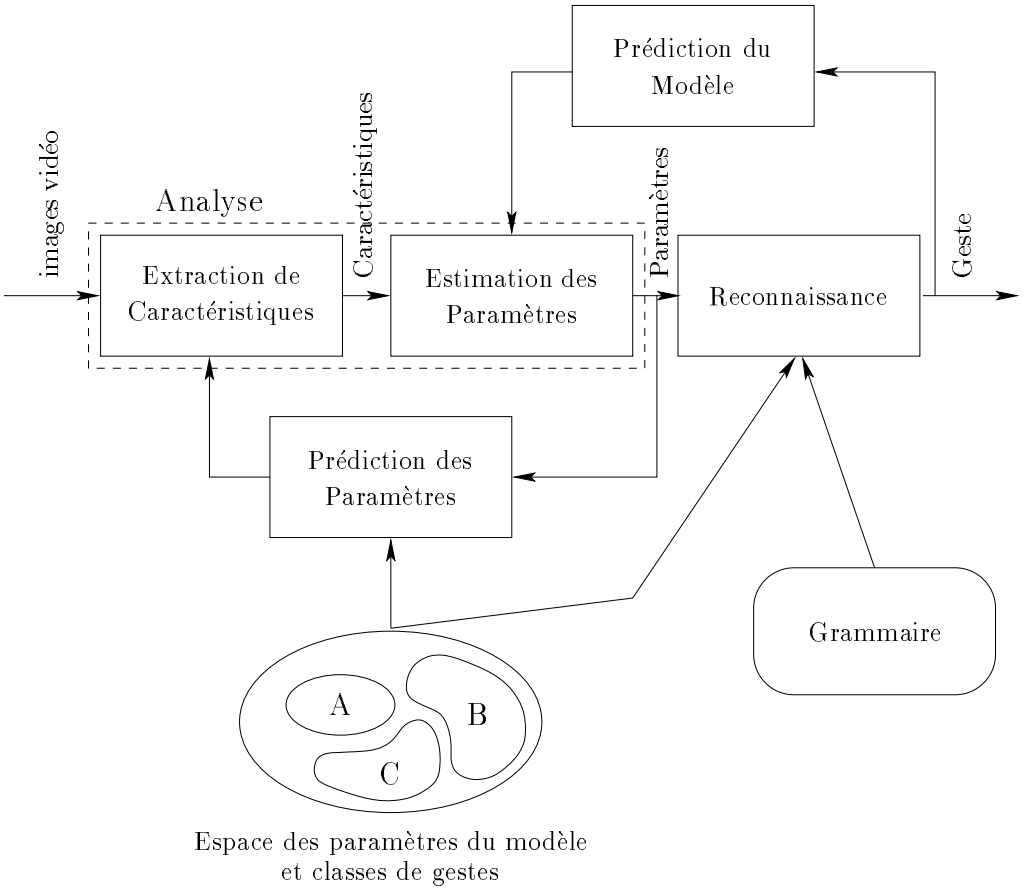
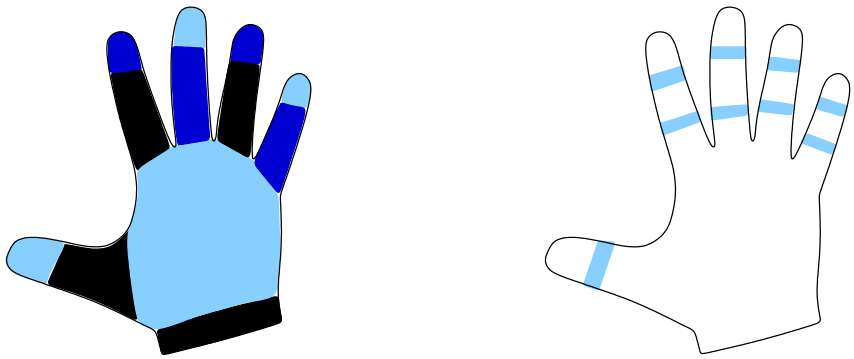


FIG. 1.17 – ***Système d´interprétation de gestes en vision par ordinateur***. *Les images des gestes sont acquises par une ou plusieurs caméras. Celles–ci sont traitées par l´étape d´analyse dans laquelle les paramètres du modèle sont estimés. A partir des paramètres estimés et de connaissances de haut–niveau (telles celles données par une grammaire), les gestes observés sont inférés dans l´étape de reconnaissance. (d´après [PSH97])*



(*a*) (*b*)

FIG. 1.18 – ***Exemple de gants colorés.*** (a) *Le gant coloré est composé de* 12 *parties, chaque partie à une couleur différente (d´après [YI96])* (b) *seules les jointures de doigts sont colorées avec une couleur différente par doigt (d´après [Hol97])*

La reconnaissance est effectuée avec un arbre de décision. L´arbre est calculé automatiquement à l´aide d´un algorithme ID3.

HOLDEN [Hol97] utilise également un gant de couleur pour la reconnaissance de la langue des signes australienne. Dans ce système, seules les jointures de doigts sont colorées différemment pour chaque doigt, la figure 1.18*b* présente ce gant. Un algorithme incrémental permet de reconstruire le modèle 3D de la main à partir de la position de la jointure. Un système expert permet la classification des gestes.

2.3.2 Approches fondées sur un modèle 3D de la main

La seconde approche utilisée pour la reconnaissance est la construction d´un modèle 3D de la main. La reconstruction 3D des scènes est une technique classique en vision par ordinateur [Fau93]. La difficulté est de faire correspondre le modèle avec le contenu de l´image (ou des images). Deux options sont possibles :

- estimation du modèle, puis mise en correspondance du modèle sur l´image ;
- extraction d´éléments caractéristiques (segments de droite, points) puis estimation du modèle.

Lorsque les paramètres correspondant aux angles des doigts et l´orientation de la main sont estimés, la classification est effectuée.

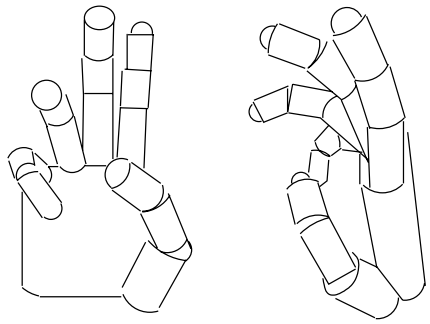


FIG. 1.19 – ***Configuration d´une main sous deux points de vue***. *Les doigts sont modélisés par des cylindres et leur extrémité par des sphères. (d´après [RK93])*

Un système complet de reconnaissance nommé `DigitEyes` a été construit par REGH et KANADE [RK93]. Il utilise un modèle cinématique de la main à base de cylindres. La main est modélisée par 16 formes : 3 pour chacun des cinq doigts, ces formes correspondent aux phalanges, et une forme pour la paume. La figure 1.19 montre un tel modèle pour une main sous deux points de vue différents. La cinématique de la main est la suivante :

- les quatre doigts ont chacun 4 degrés de liberté[23], ils sont considérés se déplacer dans un plan avec un degré supplémentaire pour l´abduction ;
- le pouce, par sa grande dextérité, est modélisé par 5 degrés ;
- 6 paramètres représentent la paume de la main.
- le point d´ancrage[24] de chaque doigt dans le plan de la paume. Ce point est considéré rigide.

L´état de la main est alors défini par un vecteur à 27 coordonnées. L´estimation de l´état est calculée par corrections incrémentales entre chaque image. Un cycle de correction est défini par :

1. estimation du vecteur d´état pour l´image suivante ;
2. acquisition de l´image et extraction de lignes correspondant aux doigts ;
3. mesure du vecteur d´erreur entre le modèle et les lignes de l´image
4. mise à jour du modèle par minimisation du vecteur d´erreur. L´algorithme de GAUSS–NEWTON est utilisé pour résoudre la minimisation non–linéaire des moindres carrés.

23. *degrees of freedom* ou DOF
24. anchor point

Nirei *et al* [NSMO94] présentent un système dans lequel la main est modélisée par 21 segments et 20 jointures. Le modèle 3D est mis en correspondance avec l´image de la main en minimisant l´erreur du flot optique et en maximisant le recouvrement entre l´image et la projection du modèle dans l´image. Cette minimisation et cette maximisation sont résolues par un algorithme génétique.

Le système Grefit[25], développé par Nölker et Ritter [NR97, NR99], détecte les extrémités des doigts dans des images de niveaux de gris. La détection est effectuée à l´aide d´une séquence de trois réseaux de neurones de type llm[26]. Les images sont, dans un premier temps, codée en utilisant des filtres de Gabor dans trois orientations et cinq positions. Le résultat est un vecteur à 35 dimensions. Les réseaux llm permettent de réaliser une correspondance entre le vecteur d´entrée à 35 dimensions représentant l´image et le vecteur de sortie à deux dimensions représentant la position du doigt. Les trois réseaux sont hiérarchiques, le premier calcule la position d´un doigt sur une image de taille 80×80. Une sous–image de taille 40×40 est extraite, elle est centré sur la position du doigt. Le troisième réseau calcule la position sur une image de taille 24×24. Un nouvel ensemble de cinq réseaux de neurones, un par doigt, est entraîné pour la transformation des positions des extrémités des doigts en configuration de la main en trois dimensions. Le modèle est construit en tenant compte des dimensions et des mouvements possibles d´une main humaine. La cinématique est inversée au moyen de réseaux de neurones. Les erreurs moyennes entre la position réelle des doigts et la position calculée ou reconstruite sont inférieures à $1\,cm$ pour les doigts et $\frac{1}{2}\,cm$ pour le pouce pour une main de $19\,cm$.

2.3.3 Approches fondées sur l´apparence visuelle de la main

Une approche émergente de la vision par ordinateur est fondée sur l´apparence des objets. Dans ce contexte, il n´y a pas de modèle 3D construit comme au paragraphe précédent mais un modèle constitué des apparences possibles de l´objet sous différents points de vue et différentes conditions. Ces modèles sont généralement constitués des images elles–mêmes ou bien de paramètres extraits des images. Ils peuvent être des contours, des moments d´images. Nous verrons dans le chapitre 2 un état de l´art des approches fondées sur l´apparence visuelle de la main.

2.3.4 Conclusion

Trois approches visuelles pour la reconnaissance existent. L´approche fondée sur des gants colorés ou des marqueurs pose le problème de contraindre l´utilisateur à porter des dispositifs spéciaux. Dans de nombreuses applications, cette contrainte n´est pas envisageable. Celles–ci se basent sur la notion «Viens comme tu es»[27] de Krueger [Kru91]

25. **G**esture **RE**cognition based on **FI**nger **T**ips
26. Local Linear Mapping network
27. «Come As You Are»

dans lesquels l´utilisateur entre dans l´environnement et en sort sans avoir à se vêtir de marques ou de gants.

L´approche basée sur un modèle 3D nécessite la construction de la main à partir des images. Cette technique présente deux inconvénients majeurs. Il est dans un premier temps nécessaire d´inverser la cinématique de la main pour recalculer le modèle 3D à partir de sa projection dans le plan image. Ce calcul, malgré les améliorations apportées pour le réduire [NR99], est souvent long. Le système `DigitEyes` fonctionne à 10Hz, c´est–à–dire un temps de réponse de 100ms, insuffisant pour une application avec un utilisateur. Pour qu´un système soit réellement utilisable, il doit avoir un temps de réponse inférieur à 50ms [Bér00]. Bérard parle d´«*interaction fortement couplée*». Le second problème est l´auto–occlusion de la main, en particulier des doigts, pouvant entraîner la construction d´un modèle erroné et rendant la reconnaissance difficile ou impossible. Cette occlusion peut être supprimée par l´utilisation de plusieurs caméras impliquant alors la mise en correspondance des images résultantes.

L´approche fondée sur l´apparence semble être la technique la plus adéquate. Elle est fondée sur un apprentissage par l´exemple d´images de la main dans les configurations utiles pour l´application. De ces images est extrait un ensemble de caractéristiques permettant la représentation de cette configuration. Contrairement à l´approche basée sur une modélisation 3D, il n´est pas nécessaire de faire un apprentissage sur toutes les apparences mais uniquement sur celles intervenant dans l´application. Ces techniques sont extrêmement rapides car elles ne nécessitent souvent qu´un seul parcours de l´image. De plus, une seule caméra est nécessaire alors que l´approche basée sur un modèle 3D impose souvent l´utilisation d´au moins une caméra supplémentaire permettant de lever les ambiguïtés.

3 Synthèse du chapitre

Dans ce chapitre, nous nous sommes intéressés à la conception de système d´interaction homme–machine gestuelle. Nous avons défini la communication gestuelle d´un point de vue psychologique et anthropologique. Parmi les trois fonctionnalités du canal gestuel proposées par Cadoz, nous avons approfondi la définition du geste sémiotique qui semble être la fonction la plus intéressante pour un système informatique. Nous avons présenté et illustré par des exemples chacune des fonctions de la taxonomie du geste sémiotique. Les fonctions sémiotique et ergotique sont difficilement distingables. Elles dépendent du point de vue de l´utilisateur ou du concepteur et peuvent être rapprochées d´une distinction entre «faire» et «faire–faire». Du point de vue d´un concepteur d´interaction gestuelle, il s´agit d´un geste sémiotique : l´utilisateur demande au système de réaliser une tâche. Cette étude des gestes nous permet de nous interroger sur leur application pour une interaction homme–machine.

L´objectif de la recherche dans le domaine des interactions homme–machine est le

développement de modèles, de concepts, d´outils et de méthodes pour la réalisation de systèmes répondant aux besoins et aux aptitudes des utilisateurs. C´est la raison pour laquelle, les chercheurs font appel aux ergonomes et aux psychologues. L´aptitude des utilisateurs dans l´emploi de gestes permet la définition des modèles et concepts d´interaction gestuelle. Les études proposées dans ce chapitre ont montré que l´utilisation des gestes, en particulier dans des applications multimodales, présente un grand intérêt pour la facilité d´interaction. Les utilisateurs ont tendance à effectuer des gestes pour les opérations de manipulation plutôt que d´énoncer les opérations oralement ou par les interfaces classiques «fenêtres, icônes, menus, pointeur»[28]. Les trois principales catégories d´applications profitant des débuts de la reconnaissance de gestes sont proposées.

Un domaine important est la reconnaissance de la langue des signes. Elle présente une triple motivation. D´un point de vue pratique, elle apporte une aide pour la communauté des sourds–muets en permettant la création de systèmes de conversion automatique de la langue. Dans un deuxième temps, elle permet l´étude de la langue des signes au sens linguistique. La troisième est que d´un point de vue scientifique, la langue des signes est riche et précise, elle est composée d´un vocabulaire et d´une grammaire [Cux99]. De plus, elle est universelle à un pays, elle permet donc une comparaison plus facile des techniques de reconnaissance sur un vocabulaire commun. Seules les modes d´acquisition peuvent alors changer.

En interaction homme–machine, deux courants utilisent une interaction gestuelle : la réalité virtuelle et la réalité augmentée. La réalité virtuelle plonge l´utilisateur dans un monde virtuel dans lequel il effectue des gestes réels sur des objets virtuels. La réalité augmentée laisse l´utilisateur dans le monde physique. Dans cette réalité, l´utilisateur peut être augmenté par un gant numérique, comme dans l´application `Charade` [BBL93], pour interagir avec le système informatique par des gestes synthétiques. L´environnement peut également être augmenté et les objets physiques supportent alors des fonctions informatiques ou bien des objets informatiques sont fusionnés à l´environnement, comme c´est le cas dans le `Bureau Numérique` de Wellner [Wel91a, Wel91b, NW92, Wel93b].

Un geste, du point de vue du concepteur, peut être vu selon différentes catégories. Il est possible, dans un premier temps, de distinguer les gestes statiques et les gestes dynamiques. Les gestes statiques ne nécessitent que l´étude de la configuration de la main à un instant donné tandis que les gestes dynamiques impliquent l´étude sur un temps borné. Il est également possible de distinguer le mouvement de la main du mouvement des doigt et faire une opposition entre configuration et position. Nous préférons définir un geste par le changement de configurations statiques dans le temps. Ce geste statique est alors représenté à la fois par la configuration et la position. Mathématiquement, nous définissons un geste comme une courbe paramétrée par le temps dans un espace représentant l´ensemble des gestes statiques possibles. Un geste statique est alors un vecteur de mesures de paramètres représentant la position et la configuration de la main. Nous

28. «*Windows, Icons, Menus, Pointing device*» ou WIMP

proposons de décomposer la reconnaissance et l´interprétation des gestes en trois parties : analyse, reconnaissance et interprétation. L´étape d´analyse calcule les paramètres de la main créant une trajectoire. L´analyse spatio–temporelle de la trajectoire lors de l´étape de reconnaissance permet la classification de la trajectoire. Cette classe est définie par un symbole. L´étape d´interprétation effectue la correspondance entre le geste et l´action à réaliser par le système.

Après l´étude du geste et sa définition d´un point de vue concepteur de systèmes à communication gestuelle, il convient de choisir une méthode de reconnaissance des gestes. La reconnaissance est intimement liée à la nature de ce système, nous avons donc présenté trois classes de techniques que nous avons illustrées d´exemples. L´approche de reconnaissance de gestes de dessins s´appuie sur l´utilisation d´un périphérique fournissant des coordonnées en deux dimensions. Ce périphérique est souvent une tablette graphique, cependant des écrans tactiles ou, plus simplement, la souris peuvent être utilisés. La technique basée sur des gants numériques permet d´obtenir des mesures précises sur la position des doigts. Cette technique souffre cependant du lien de l´utilisateur à la machine par un câble. Elle est avantageuse lorsque l´utilisateur n´est pas censé se déplacer, comme par exemple dans les systèmes de réalité virtuelle où, bien souvent, c´est le décor qui se déplace. Dans de nombreuses applications, cette contrainte n´est pas possible. La vision par ordinateur permet de libérer l´utilisateur de ce lien. Les systèmes utilisant ces caméras permettent soit de reconstruire le modèle 3D de l´objet observé [Roh94] soit d´effectuer des correspondances entre images ou paramètres d´images. Cette dernière technique est appelée «vision par apparence». Elle nous semble plus judicieuse car, contrairement à l´approche basée sur des modèles 3D, il n´est pas nécessaire de modéliser la main, les algorithmes sont plus rapides car aucune inversion de la cinématique n´est nécessaire.

Dans la suite de ce manuscrit, nous étudions les trois étapes de la reconnaissance et de l´interprétation des gestes. Dans un premier temps, au chapitre 2, nous proposons l´analyse, s´appuyant sur la vision par apparence. Nous présentons des techniques permettant l´extraction de la position et de la configuration. Avant de voir la seconde étape de reconnaissance, nous proposons, au chapitre 3, une étape de classification des configurations de mains. Celle–ci permet la reconnaissance des gestes statiques. L´étape de reconnaissance de gestes dynamiques est étudiée au chapitre 4 à travers trois techniques : automates d´états finis, modèles de Markov cachés et une technique originale de reconnaissance statistique. Enfin, l´étape d´interprétation est illustrée, dans les chapitres 5 et 6, par une application de reconnaissance d´activités et de gestes dans un environnement interactif.

Le petit Nicolas en thèse [Pet]
La Recherche

«Au début, c´est tellement compliqué, on n´y comprend rien.»

THEORÈME 1. - *On a l'inégalité*

$$\sum_{\sigma\in G}\frac{|K_\sigma|(|K_\sigma|-1)}{k^2}\log\left(\frac{|K_\sigma|-1}{k\,e\,\lambda_\sigma}\right)+\frac{K-1}{k^2}\sum_{\sigma\in G}\sum_{i\in K_\sigma}\|\log|\sigma\,a_i\|\|$$
$$\leq\left(1-\frac{1}{K}\right)\frac{2D}{K}\sum_{i=1}^{K}h(a_i)+\frac{D}{K}\left(1+\frac{|G'|}{2D}+\log\frac{K}{2}\right)$$

«On peut passer des heures et des heures à chercher sans rien trouver. Dans ces moments–là, mon papa et ma maman sont drôlement inquiets et quand ma maman demande si c´était une bonne idée de faire faire une thèse au petit (c´est moi), mon papa ouvre la bouche sans parler, il agite les bras, et il s´en va lire le journal dans le salon.»

Chapitre 

Étape d´analyse : Extractions de Caractéristiques

Ce chapitre présente l´extraction de caractéristiques de la main. Nous appelons caractéristiques *un ensemble de mesures permettant de déterminer la configuration de la main et sa position dans l´espace. Nous proposons des techniques permettant l´extraction de ces deux types de caractéristiques. Dans le cas de l´extraction de caractéristiques spatiales, nous nous intéressons à des algorithmes de localisation par segmentation et par apparence. Nous proposons également une fusion de ces algorithmes permettant une localisation plus stable.*

Deux méthodes sont proposées pour l´extraction de caractéristiques de configuration. La première consiste à effectuer une analyse en composantes principales des images de mains. Elle permet de définir un sous–espace propre dans lequel une image de main est représentée par un vecteur. Nous proposons également les discriminants de Fisher *comme une alternative à l´analyse en composantes principales. Les invariants de* Hu *permettent également la représentation des images de mains par un vecteur de mesures. Les mesures sont ici les valeurs des sept premiers invariants. Ils présentent l´avantage d´être indépendants de la position, orientation et taille de la main dans l´image.*

1 Introduction

L´extraction de caractéristiques d´une image est une des préoccupations de la recherche en vision par ordinateur. Dans le concept de MARR [Mar82], la première étape est la production d´une description d´une ou plusieurs images en termes d´attributs bi–dimensionnels à partir d´un processus d´extraction de caractéristiques. Ce niveau de représentation est appelé *première ébauche*[1]. Cette première ébauche permet la création d´une ébauche 2,5D au cours de laquelle des propriétés tri–dimensionnelles sont calculées à partir des attributs de la première ébauche. L´ébauche 2,5D permet alors la description 3D de la scène. Pour HORAUD et MONGA [HM83], la première ébauche est la *segmentation des images*, il s´agit de la base à tout système de vision. De nombreuses recherches lui sont consacrées, de la détection de contours à l´extraction de régions. Une approche plus récente de la vision par ordinateur ne cherche plus à reconstruire la scène tri–dimensionnelle pour la reconnaître, il s´agit de la vision par apparence. Dans cette approche, les objets sont représentés par un ensemble de caractéristiques permettant la mise en correspondance avec des objets connus. Parmi ces caractéristiques, nous pouvons citer celles proposées dans notre laboratoire:

- points d´intérêts [Sch96];
- statistiques de couleurs [SW95, Col96];
- champs réceptifs [Sch97, Col99];
- détecteurs appris [Gua98].

Dans le contexte de la reconnaissance de gestes et d´activités, l´approche d´extraction de caractéristiques reste valide. Cependant deux courants s´opposent. Le premier considère les gestes ou activités dynamiques, l´extraction des caractéristiques est spatio–temporelle. Elle se fait sur une séquence complète d´images ou quelques images successives. La seconde approche ne considère que les caractéristiques dans une image. L´aspect dynamique du geste est alors uniquement considéré pendant l´étape de reconnaissance

1.1 Caractéristiques spatio–temporelles

BOBICK et DAVIS [BD96, Bob96, DB97] proposent une représentation spatio–temporelle d´activités humaines. Ils définissent les «*images d´énergie de mouvement*»[2] et les «*images de l´historique du mouvement*»[3]. Les premières, binaires, représentent la position du mouvement dans une séquence d´images. Dans les secondes, les pixels sont des valeurs correspondant à l´âge du mouvement. Ils sont calculés par un simple remplacement et un opérateur décalage. Si $D(x, y, t)$ est un booléen indiquant si l´intensité lumineuse du point (x, y) à l´instant t a changé depuis l´instant $t-1$ alors, les pixels de l´image H de l´historique du mouvement à l´instant t sont définis par:

1. primal sketch
2. «*Motion Energy Motion*» ou MEI
3. «*Motion History Image*» ou MHI

$$H_\tau(x,y,t) = \begin{cases} \tau & \text{si } D(x,y,t) = 1 \\ \max\left(0, H_\tau(x,y,t-1) - 1\right) & \text{sinon} \end{cases} \tag{2.1}$$

Dans cette équation, τ est l´âge considéré dans l´historique. L´historique au point (x,y) est affecté à la valeur τ si un mouvement est détecté en (x,y), c´est–à–dire si $D(x,y,t) = 1$. Sinon, l´âge du mouvement augmente et la valeur de $H(x,y,t)$ est décrémentée. L´âge du mouvement en (x,y) est en fait déterminé par :

$$|H_\tau(x,y,t) - \tau| \tag{2.2}$$

L´image E d´énergie de mouvement est alors définie par le seuillage de l´image H à zéro, c´est–à–dire :

$$E_\tau(x,y,t) = \bigcup_{i=0}^{\tau} D(x,y,t-i) = \begin{cases} 1 & \text{si } H_\tau(x,y,t) > 0 \\ 0 & \text{sinon} \end{cases} \tag{2.3}$$

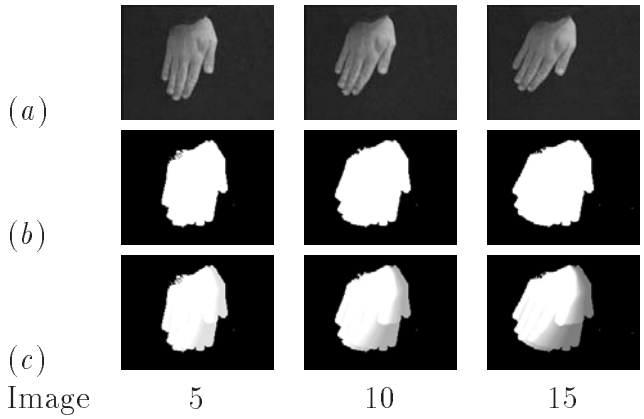


FIG. 2.1 – ***Exemple d´image d´énergie et image de l´historique du mouvement*** *(a) Images clés du geste. (b) images d´énergie. (c) image de l´historique du mouvement. (d´après [Bob96])*

La figure 2.1 montre les images d´énergie et les images de l´historique du mouvement d´un geste de la main. Une description statistique des images est effectuée avec les sept moments de HU [Hu62]. Une action est classifiée en choisissant le modèle d´action qui minimise la distance de MAHALANOBIS.

Cutler et Turk [CT98] proposent l´utilisation du flot optique pour la reconnaissance de sept gestes de bras. Ils considèrent le fond statique et les changements de luminosité faibles. Le flot optique calculé entre deux images successives est segmenté en zone par un algorithme «*K–mean*». Un ensemble de règles permet la reconnaissance des gestes. Les règles considèrent le nombre de zones, la direction du mouvement, la relation entre les deux zones et la taille du mouvement.

Chomat [CC99, Cho00, CMC00] propose également une approche basée sur une analyse spatio–temporelle d´une séquence d´images. Chaque activité est caractérisée par un histogramme multidimensionnel des projections de voisinages locaux sur une base de champs réceptifs. Ces histogrammes donnent une estimation de la densité de probabilité nécessaire à un processus de reconnaissance basé sur une règle de Bayes. Le résultat de la technique est une carte de probabilité pour chaque élément d´activité à reconnaître. Nous reprenons cette technique dans le chapitre 5 et l´étendons à la reconnaissance de l´activité complète.

1.2 Caractéristiques dans une seule image

Freeman et Roth [FR95, FTOK96, FAB+98] utilisent des histogrammes orientés pour caractériser une configuration de main. Cette description est indépendante de la luminosité et de la translation. Elle s´appuie sur l´orientation des valeurs des pixels obtenue par le calcul de la direction du gradient d´une image. L´orientation locale est fonction de la position (x, y) et de l´intensité lumineuse, $I(x, y)$ en ce point :

$$\theta(x,y) = \tan^{-1}\left(\frac{I(x,y) - I(x-1,y)}{I(x,y) - I(x,y-1)}\right) \tag{2.4}$$

L´histogramme orienté est un vecteur à N coordonnées dans lequel la i^{e} coordonnée donne le nombre d´orientations $\theta(x, y)$, où $\theta(x, y)$ compris entre :

$$\frac{360^\circ}{N}\left(i - \frac{1}{2}\right) \quad \text{et} \quad \frac{360^\circ}{N}\left(i + \frac{1}{2}\right)$$

N est alors le nombre d´orientations possible pour les directions du gradient. Ainsi, l´histogramme $\Phi(i)$ est calculé par :

$$\Phi(i) = \sum_{x,y} \begin{cases} 1 & \text{si } \left|\theta(x,y) - \frac{360^\circ}{N} i\right| < \frac{360^\circ}{N} \\ 0 & \text{sinon} \end{cases} \tag{2.5}$$

La figure 2.2 présente un ensemble de configurations de main, les images d´orientation et les histogrammes orientés. La reconnaissance est alors effectuée par une simple distance euclidienne entre la configuration candidate et les configurations cibles. Cette technique a été utilisée dans plusieurs jeux parmi lesquels le très classique «*Caillou, papier, ciseaux*» ou `RobotHand`, un jeu ressemblant à un jeu de tétris. Dans ces jeux, le nombre de

FIG. 2.2 – ***Exemple d´histogrammes orientés.*** *(a) Images d´origine. (b) Image d´orientation. (c) Histogramme orientés en coordonnées polaires. (extrait de [FAB+98])*

configurations est réduit et les configurations sont très différentes les unes des autres (cf. figure 2.2*a*). FREEMAN *et al* [FAB+98] montrent que des gestes différents peuvent avoir des histogrammes orientés proches. De plus, la main doit être dominante dans l´image pour que sa configuration intervienne majoritairement dans l´histogramme orienté.

Une autre alternative est la modélisation de l´apparence de la main. COOTES et TAYLOR [CT92, CTCG92] modélisent des objets déformables par leur contour. Ce contour est représenté par un ensemble de points répartis uniformément autour de l´objet. Le «*modèle de distribution de points*»[4] est créé à partir d´un ensemble de contours du même objet. Dans un premier temps, un alignement aux moindres carrés des exemples est réalisé puis une analyse en composantes principales de points est effectuée. Le résultat est un contour moyen et un ensemble de vecteurs représentant les principaux modes de variations autour de la moyenne. Un nouveau contour est alors approximé par :

$$c = \bar{c} + Pb \tag{2.6}$$

où c est un contour représenté par la concaténation des coordonnées de ses points, $\bar{c}$ est le contour moyen, P est la matrice contenant les vecteurs des modes de variation et b est un vecteur de paramètres du contour. Le vecteur $\bar{c}$ et la matrice P étant connus, un contour c peut être approximé par le vecteur b, calculé par :

4. «*Point Distribution Model*» ou PDM

$$b = P^T(c - \bar{c}) \tag{2.7}$$

Les exemples de la base d´apprentissage peuvent être utilisés pour déterminer la distribution des vecteurs b pour chaque classe de configuration, i, en terme de sa moyenne $\bar{b_i}$ et de sa matrice de covariance K_i. La classification est alors obtenue en cherchant la classe minimisant la distance de MAHALANOBIS [Mar95a, ATLC95] :

$$\hat{i} = \arg_i \max (b - \bar{b_i})^T K_i^{-1} (b - \bar{b_i}) \tag{2.8}$$

HEAP et HOGG [HH96a, HH96b] ont étendu cette approche à la modélisation tri–dimensionnelle de la main à partir d´images à résonance magnétique (IRM). Cependant, le modèle construit n´est pas utilisé pour la reconnaissance des configurations mais seulement pour le suivi.

Dans la suite de ce chapitre, nous considérons les caractéristiques dans une seule image. Elles sont découpées en deux types : les caractéristiques spatiales définissant la position, l´orientation et la taille de la main dans l´image puis les caractéristiques représentant sa configuration.

2 Extraction de Caractéristiques spatiales : localisation de la main

La localisation de la main constitue la première étape pour extraire ses caractéristiques. Nous étudions, dans cette section, deux classes d´algorithmes de localisation : par segmentation et par apparence. À partir de celles–ci, et pour chacune des techniques, nous estimons les caractéristiques. Enfin, nous proposons l´utilisation d´un *système multi–modules adaptatif* [CM97, MDC98] permettant la coopération des différentes techniques proposées pour obtenir une localisation plus fiable et plus robuste.

2.1 Localisation par segmentation

La segmentation permet de différencier les pixels appartenant à la main des autres. Nous proposons deux méthodes. La première utilise le mouvement pour localiser la main. La seconde s´appuie sur la couleur particulière de la peau.

2.1.1 Localisation par différence d´images

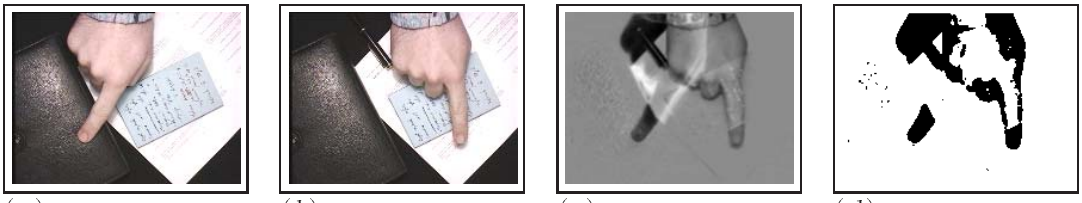
La technique de différence d´images est bien connue en vision par ordinateur, elle consiste à effectuer une différence pixel à pixel entre deux images. La différence du pixel de coordonnées (x, y) est :

$$D(x,y) = \Big|I(x,y) - I'(x,y)\Big|, \quad \forall x,y \tag{2.9}$$

Dans cette équation, I et I' sont deux images, elles peuvent être successives dans une séquence ou bien I' peut être une image de référence. Nous voyons ces deux possibilités dans la suite.

a) Différence d´images successives

Si nous considérons la différence entre deux images successives dans une séquence, le résultat est une image dont les pixels différents de zéro correspondent aux objets ayant bougé. La figure 2.3 montre la différence entre deux images successives : l´image 2.3a à l´instant t et 2.3b l´image à l´instant $t + \Delta t$. L´image 2.3c est l´image de différence, le fond gris correspond à une valeur de pixel 0, un niveau plus foncé correspond à une valeur négative et un niveau plus clair à une valeur positive. L´image 2.3d est l´image de valeur absolue de l´image de différence, le niveau zéro est blanc et une valeur non nulle est grise/noire.

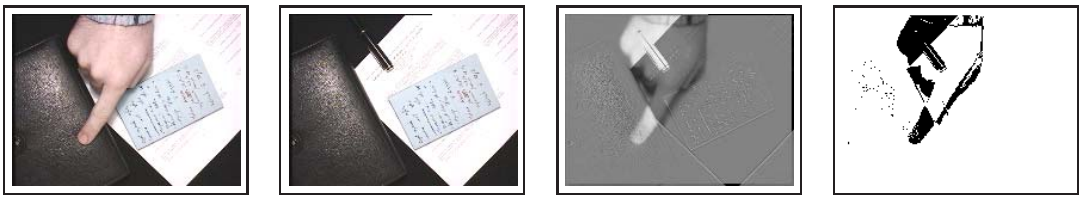


(a) (b) (c) (d)

FIG. 2.3 – ***Différence entre deux images successives.*** *Différence entre* (a)*, l´image à l´instant t, et* (b)*, l´image à l´instant* $t+\Delta t$*.* (c) *est l´image de différence, le niveau de gris correspond à la valeur* 0*. Un niveau plus foncé correspond à une valeur négative et un niveau plus clair à une valeur positive.* (d) *est l´image seuillée de l´image de différence, le niveau zéro est blanc et une valeur non nulle est noire.*

b) Différence avec une image de fond

Il est également possible de faire la différence avec une image contenant le fond, c´est–à–dire prise au moment de l´initialisation ou quand aucun objet ne se trouve dans le champ de la caméra. La différence permet alors la détection des objets apparus, et en particulier la main.



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

FIG. 2.4 – ***Différence d´image avec une image de fond.*** *Différence entre* (a)*, l´image à l´instant t, et* (b)*, l´image de fond prise à un moment où aucune main ne se trouvait dans le champ de la caméra.* (c) *est l´image de différence et* (d) *est l´image de valeur absolue de l´image de différence.*

c) Discussion

La différence d´image est une technique simple permettant de faire rapidement une estimation de la position d´un objet en mouvement. Cette estimation permet de réduire la zone de recherche d´un autre algorithme. Cependant, cette technique impose des contraintes sur l´environnement :

1. la caméra doit être fixe sous peine de détecter l´image entière comme objet en mouvement. Cette contrainte ne présente pas de difficulté dans des applications fixant une région particulière de la scène telle que le «`Bureau Numérique`» de WELLNER [Wel91a, Wel91b, NW92] où seul le bureau doit être observé.
2. les sources lumineuses doivent être constantes et, également, fixes. Un changement de luminosité, même local, entraîne la détection de la zone de changement comme étant en mouvement. Un changement de luminosité peut être provoqué par l´allumage d´une lampe, le passage d´un individu créant une ombre ou le passage d´un nuage. Le problème est particulièrement crucial pour la différence avec une image de fond, puisque l´image a été prise dans des conditions particulières. Pour éliminer ce problème, une mise à jour de l´image de fond est nécessaire. La différence entre deux images successives étant moins sensible au changement de luminosité, il peut être fait l´hypothèse de conservation de l´intensité lumineuse [Cho00].

Elle impose également des limitations :

1. En utilisant la technique de différence d´images successives et lorsque le mouvement est faible, la zone détectée est petite et ne contient qu´une

partie de l´objet que nous cherchions.
2. Si deux objets sont en mouvement, un calcul supplémentaire est nécessaire pour différencier les deux objets. Il est simple lorsque les objets sont éloignés, un calcul de zones connexes est suffisant. Lorsque les objets sont superposés, il faut tenir compte d´autres critères tel que leur texture ou leur couleur.

2.1.2 Localisation par couleur

La localisation par couleur est une approche classique dans le domaine de localisation de visages ou de mains [Bic95, Ess96, Mas98, Bér00]. Elle s´appuie sur la segmentation de couleur et suppose que celle de l´objet à segmenter soit discriminante, c´est–à–dire que l´entourage ne présente pas la même couleur. L´objet est localisé en cherchant la couleur des pixels la plus proche de celle *a priori* de l´objet. Nous cherchons alors les pixels p dont la probabilité de la couleur $c(p)$ est maximale, sachant que celle de l´objet est modélisée par le modèle $\mathcal{M}_{\text{coul}}$. Cette probabilité conditionnelle est notée :

$$P(c(p) \mid \mathcal{M}_{coul}) \tag{2.10}$$

Classiquement, une couleur est modélisée, pour chaque pixel de l´image, par un triplet de valeurs sur chacun des canaux de couleur rouge, vert et bleu. Ce triplet permet de représenter toutes les luminosités d´une même couleur. Par exemple, les deux triplets $< 255\ 0\ 0 >$[5] et $< 100\ 0\ 0 >$[5] représentent la couleur rouge avec une luminosité différente. Afin de s´affranchir de ce problème, Schiele et Waibel [SW95] proposent de normaliser les valeurs du triplet de couleur par la luminosité, nous parlons alors de chrominance :

$$p_r = \frac{p_R}{p_L} \qquad p_v = \frac{p_V}{p_L} \qquad p_b = \frac{p_B}{p_L} \tag{2.11}$$

Dans cette équation, les valeurs p_R, p_V et p_B sont respectivement les composantes rouge, verte et bleue du pixel p. Les valeurs p_r, p_v et p_b sont les composantes normalisées par la luminance p_L du pixel p définie par :

$$p_L = p_R + p_V + p_B \tag{2.12}$$

Il est à noter que les trois composantes normalisées p_r, p_v et p_b sont linéairement dépendantes :

$$p_r + p_v + p_b = \frac{p_R + p_V + p_B}{p_L} = 1 \tag{2.13}$$

Ainsi, deux composantes sont suffisantes pour représenter la chrominance d´un pixel. Dans la suite, la chrominance du pixel p est représentée par le couplet de deux des composantes chrominatiques normalisées :

5. les valeurs des pixels sont généralement comprises entre 0 et 255, 255 étant la valeur de saturation.

$$c(p) =< p_r, p_v > \tag{2.14}$$

a) Modélisation de la chrominance

Le modèle $\mathcal{M}_{\text{coul}}$ de la chrominance de l´objet, utilisé dans l´équation 2.10, doit être défini. Celui–ci peut être calculé à partir d´un échantillon de pixels E d´apprentissage. La chrominance de l´objet à localiser peut être représentée par deux modèles : un histogramme ou un modèle gaussien.

Histogramme de chrominance [SB91] Swain et Ballard [SB91] montrent que l´histogramme de chrominance est un modèle fiable pour la reconnaissance d´entités colorées. Ils expérimentent différents type d´histogrammes dont ceux créés à partir des composantes rouge et verte normalisée. Ces composantes normalisées correspondent à la constance de couleur la plus simple. Une cellule de coordonnées (r, v) de l´histogramme à deux dimensions h_E donne le nombre de pixels de l´échantillon E ayant une chrominance de composantes rouge r et vert v. Cet histogramme permet de définir la probabilité de l´équation 2.10 par

$$p(c(p)|\mathcal{M}_{\text{coul}}) = \frac{1}{n_E} h(c(p)) = \frac{1}{n_E} h(p_r, p_v) \tag{2.15}$$

où n_E est le nombre total de pixels de l´ensemble E. L´algorithme pour calculer l´histogramme h_E de l´échantillon de pixels E est donné par l´algorithme 2.1.

Algorithme 2.1 Algorithme de calcul de l´histogramme de chrominance à partir de l´échantillon E

1. Initialisation
   pour tout (r, v), cellule de l´histogramme **faire**
   $h(r, v) = 0$
   fin pour

2. Calcul de l´histogramme à partir de l´échantillon E
   pour tout pixel p de l´échantillon E **faire**
   $h(p_r, p_v) = h(p_r, p_v) + 1$
   fin pour

Modèle gaussien Le modèle de la chrominance $\mathcal{M}_{\text{coul}}$ est représenté par une fonction de probabilité gaussienne f définie par :

$$f(\vec{x}) = \frac{1}{\sqrt{2\pi|\Lambda|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})\Lambda^{-1}(\vec{x}-\vec{\mu})^T} \tag{2.16}$$

Le vecteur $\vec{x}$ est la variable aléatoire à deux dimensions représentant le couple de chrominance ; la moyenne $\vec{\mu}$ et la matrice de covariance Λ sont les paramètres du modèle gaussien. Ils correspondent aux moments centraux d´ordre 0 et 1 de la distribution des chrominances dans l´ensemble E. Ces moments sont calculés selon les équations suivantes :

$$\vec{\mu} = \begin{bmatrix} \mu_r \\ \mu_v \end{bmatrix} \qquad \Lambda = \begin{bmatrix} \sigma_r^2 & \sigma_{rv} \\ \sigma_{rv} & \sigma_v^2 \end{bmatrix} \tag{2.17}$$

avec

$$\begin{aligned} \mu_a &= \tfrac{1}{N} \textstyle\sum_{p\in E} p_a \\ \sigma_{ab} &= \tfrac{1}{N} \textstyle\sum_{p\in E} \Big((p_a - \mu_a)(p_b - \mu_b)\Big) \\ &= \tfrac{1}{N} (\textstyle\sum_{p\in E} p_a p_b - \mu_a \mu_b) \end{aligned} \tag{2.18}$$

Dans ces équations a et b prennent les valeurs r et v.

b) Discussion

Le premier avantage de cette technique par rapport à celle de la différence d´images est de limiter les contraintes sur l´environnement :

1. la caméra peut être en mouvement ;
2. le changement de la luminosité est atténué par l´utilisation de la chrominance plutôt que de la couleur.

Cependant, cette technique nécessite la construction d´un modèle à partir d´un exemple. De plus, ce modèle est différent selon les types d´éclairage. Bérard [Bér00, CB97] propose une initialisation automatique. La détection du clignement des paupières fournit une estimation de la position des yeux. Cette dernière sert de référence à l´extraction d´un motif de couleur situé entre les deux yeux.

Störring *et al* [MS99] montrent le changement de chrominance en fonction de celui de la lumière. Il est donc possible, selon ce principe, de mettre à jour automatiquement un modèle de couleur en fonction de celle d´un objet fixe dans la scène. Dans le cadre de l´environnement intelligent Monica, il est possible de créer le modèle de couleur de l´utilisateur lorsque celui–ci franchit la porte et en se basant sur la couleur des montants de porte [Le]. Elle permet également l´ajout de plusieurs teintes différentes représentant par exemple des couleurs de visages pour des personnes différentes ou bien sous différents éclairages. Il est également possible d´ajouter une teinte en négatif, permettant de rejeter des objets tel le fond.

Le choix entre l´une des deux méthodes est moins important, les modèles étant équivalents : il est possible de transformer une gaussienne en histogramme et vice–versa. Dans le cas d´un histogramme composé de plusieurs pics, l´équivalent est alors une mélange de gaussiennes[6]. L´avantage de l´histogramme lors de la détection est la recherche dans un tableau contrairement à la gaussienne impliquant un calcul complexe.

2.2 Localisation par apparence

La localisation par apparence, et plus généralement la vision par apparence [Col99, Cho00, Sch97], se fonde sur la manifestation visuelle d´un objet captée par une caméra. L´*espace d´apparence* d´un objet est l´ensemble des manifestations de cet objet vu sous tous les éclairages et tous les points de vue possibles. Cette condition ne pouvant être appliquée, nous utilisons en pratique les manifestations pouvant effectivement être observées. Dans cette section, la localisation par apparence consiste à comparer la manifestation dans l´image avec une manifestation (ou un ensemble de manifestations) particulière ou avec un modèle de toutes les manifestations possibles.

2.2.1 Corrélation

La corrélation est une opération de traitement du signal souvent utilisée pour comparer deux signaux ou pour calculer leur déphasage. Appliquée à la vision par ordinateur, la corrélation mesure la similitude entre deux images de même taille. Dans un premier temps, nous présentons le principe de localisation par corrélation. Nous présentons ensuite l´opérateur de corrélation et proposons deux méthodes de calcul. Nous discutons enfin de problèmes liés à l´utilisation et proposons des solutions.

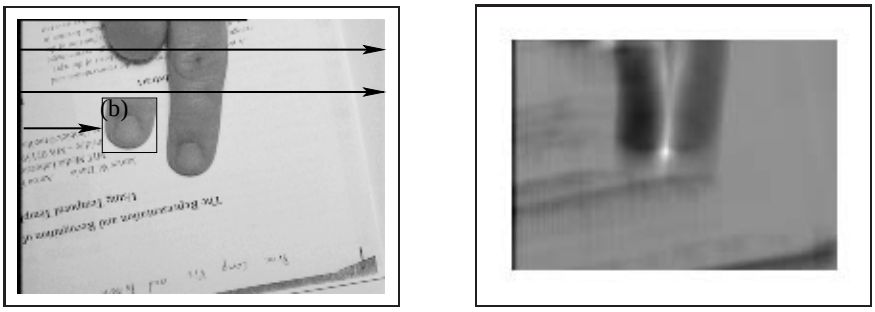
a) Principe

La corrélation est une mesure de similitude entre deux images de même taille. Cette mesure permet la recherche d´un motif de référence dans une image [Mar94, MC95, Bér94]. La localisation du motif dans une image s´effectue par le parcours de toutes les sous–images ayant la même taille que le motif. Pour chaque sous–image, la mesure de corrélation est calculée. Le motif de référence est l´emplacement où la mesure a été maximale. Nous parlons alors de *pic de corrélation*. La figure 2.5 illustre ce principe.

b) Mesures de corrélation

La mesure de similitude la plus simple à calculer est la distance euclidienne entre le motif de référence noté M, de taille $m \times n$, et la partie de l´image I centrée à la position (i, j) et également de taille $m \times n$. Cette distance euclidienne est la *somme des différences*

6. mixture of gaussians



(a) (c)

FIG. 2.5 – ***Localisation par mesure de corrélation***. *La mesure de corrélation entre les sous–parties de l´image (*a*) et le motif de référence (*b*) permet le calcul d´une carte de corrélation (*c*). Dans cette carte, plus les pixels sont clairs et plus la similitude est grande. La recherche du* pic de corrélation, *c´est–à–dire la position où la mesure de corrélation est maximale donne la position du motif de référence.*

des carrés et notée SSD[7]. La similitude est parfaite lorsque la valeur de la mesure SSD est nulle. La définition de cette mesure est :

$$SSD(i,j) = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} \Big(I(i+u, j+v) - M(u,v) \Big)^2 \tag{2.19}$$

En pratique, la mesure SSD n´est pas optimale [Mar94]. Elle est en effet très sensible aux changements de luminosité. Lorsque la lumière globale change dans la scène, le niveau de tous les pixels est également modifié. Pour résoudre ce problème, il faut normaliser la mesure de corrélation par l´énergie du motif et de l´image. Cette énergie prend en compte la luminosité générale de l´image. La formule de la corrélation normalisée est notée NCC[8] :

$$NCC(i,j) = \frac{\sum_{u=0}^{m-1} \sum_{v=0}^{n-1} I(i+u, j+v) M(u,v)}{\sum_{u=0}^{m-1} \sum_{v=0}^{n-1} I^2(i+u, j+v) \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} M^2(u,v)} \tag{2.20}$$

La valeur de corrélation est comprise entre 0 et 1. Elle est égale à 1 lorsque le motif et l´image sont identiques à un coefficient de luminosité près.

7. SSD est l´abréviation de *Sum of Squared Difference*
8. NCC est l´abréviation de *Normalized Cross-Correlation.*

c) Discussion

Nous discutons ici des problèmes liés à l´utilisation de la corrélation: le choix de la méthode de corrélation, l´optimisation du calcul de localisation par réduction de la zone de recherche et les problèmes dus aux changements d´orientation, d´échelle et de déformation.

Choix de la méthode de corrélation Nous avons montré [Mar94, MC95, AG92] que la corrélation SSD est plus stable en présence de bruits. Toutefois, BÉRARD [Bér00] note que l´utilisation de la corrélation dans des applications où la luminosité n´est pas contrôlée fait préférer la corrélation NCC. Son prototype, la `fenêtre perceptuelle`, montre de très bon résultat. Dans ce prototype, les mouvements du visage sont capturés de manière non intrusive et permettent le contrôle d´une interface graphique. Le système utilise la corrélation pour suivre le mouvement du visage en prenant une cible telle que le sourcil.

Réduction de la zone de recherche Le calcul de la localisation peut être réduit en considérant une *zone de recherche*[9]. Cette notion de *zone de recherche* s´appuie sur le suivi de l´objet et non sur la localisation dans une image isolée.

Le suivi a pour objectif de déterminer la position d´un objet dans chacune des images de la séquence vidéo. Celui–ci peut être effectué par la localisation dans chacune des images indépendamment les unes des autres ; on peut utiliser aussi la connaissance de la position dans les images précédentes. Ces positions peuvent être utilisées de deux manières : en calculant le déplacement maximal qu´effectue l´objet entre deux images successives ou bien en émettant une prédiction sur la prochaine position par rapport aux positions précédentes.

Le déplacement maximal entre deux images est défini en fonction de la taille du motif m[10] et la taille de la zone de recherche t[10] [Bér00] :

$$d_{max}(t,m) = \frac{t-m}{2} \tag{2.21}$$

La fréquence de fonctionnement du suivi est inversement proportionnelle au nombre de mesures de corrélation à effectuer :

$$F(t,m) = \frac{k}{(t-m+1)^2)}$$

La vitesse de déplacement maximale V_{max} est

9. *region of interest (ROI).*

10. Nous considérons un motif et une zone de recherche carrés afin de simplifier les calculs, cependant ceux–ci peuvent être étendus au cas général.

$$V_{max}(t,m) = d_{max}(t,m)F(t,m) = k\frac{t-m}{2(t-m+1)^2}$$

Le choix de la taille de la zone de recherche est donc un compromis entre la vitesse maximale autorisée par le système de suivi et la vitesse maximale possible de l´objet en mouvement. Si nous considérons le geste de sélection d´une cible, la vitesse de la main en mouvement peut être déterminée par la loi de FITTS [Bér94, Fit53]. À partir des positions précédentes, une prédiction sur la suivante peut être calculée en utilisant un filtre de KALMAN [Kal60, BL93, WB97] ou par l´algorithme de `Condensation` [BI98b].

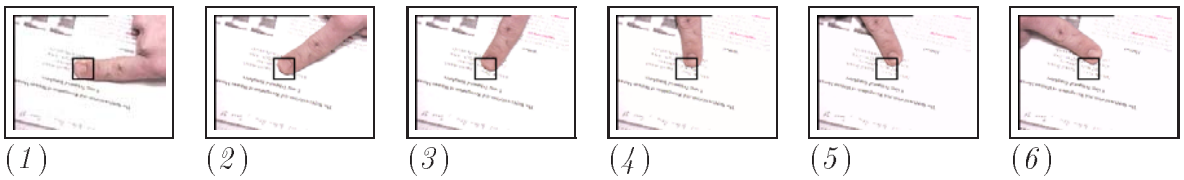


FIG. 2.6 – ***Perte de l´objet suivi après la mise à jour du motif de corrélation.*** *La mise à jour du motif de corrélation pose le problème de perte de l´objet suivi. Lors de la rotation du doigt, le motif de référence est de plus en plus approximatif, nous arrivons alors au cas où le motif de référence ne correspond plus à l´objet à suivre. (d´après [Bér94])*

Problème du changement d´orientation et d´échelle En théorie, les objets que nous cherchons à localiser doivent être dans les mêmes conditions d´orientation et d´échelle. Cependant, lorsque la modification est faible, la mesure de similarité reste valable. Pour de plus grandes rotations, DARRELL et PENTLAND [DP92], ainsi que BÉRARD [Bér94], proposent la mise à jour du motif. Lorsque le résultat de corrélation est inférieur à un seuil prédéterminé, un nouveau motif est extrait de l´image à l´emplacement de la dernière détection. Cette mise à jour du motif pose le problème de perte de l´objet à suivre. La figure 2.6 illustre cette perte sur l´exemple de suivi de l´extrémité du doigt en rotation.

Il est possible d´utiliser un ensemble de motifs à différentes orientations ou échelles [Dev98, CB96]. Dans le cas du changement d´orientation, nous considérons des motifs de référence pour l´objet dans différentes orientations. La figure 2.7 illustre un exemple de neuf motifs pouvant être utilisés pour la recherche d´un doigt. Les motifs ont été pris tous les 20° entre $-80°$ et $80°$.

FIG. 2.7 – ***Exemple de neuf motifs de référence de l´extrémité d´un doigt dans différentes orientations***. *Ces motifs peuvent être utilisés pour la localisation de l´extrémité de doigt. Les motifs ont été pris tous les* 20° *entre* −80° *et* +80°.

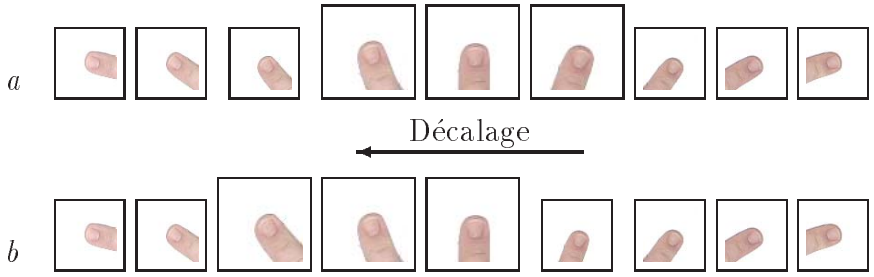


FIG. 2.8 – ***Glissement de la série motifs de référence vers la gauche***. *La corrélation effectuée sur les trois motifs les plus gros de* (a) *a donné un résultat supérieur pour le motif de gauche. Pour la prochaine image, les motifs de référence seront les trois motifs décalés vers la gauche montrés en* (b).

Lors du suivi, la corrélation est calculée pour trois motifs d´orientation croissante. Tant que le motif central obtient le meilleur résultat de corrélation, la série des trois motifs de référence est conservée. Si le premier motif de référence (respectivement le dernier) obtient le meilleur résultat de corrélation, une nouvelle série de trois motifs est sélectionnée en glissant vers la gauche (respectivement la droite) dans la liste des motifs. La figure 2.8 illustre le cas où la corrélation est effectuée sur les trois motifs les plus gros de 2.8*(a)*. Celle–ci a donné un résultat supérieur pour le motif de gauche. Pour la prochaine image, les motifs de référence seront les trois motifs décalés vers la gauche, montrés en 2.8*(b)*.

Cette technique permet de résoudre le problème de changement du motif lorsqu´une rotation est repérée. Afin de prendre également en compte le changement d´échelle, il serait nécessaire d´ajouter des motifs à différentes échelles. Nous obtiendrions alors une matrice de motif dans laquelle nous nous déplacerions. Cette matrice implique un travail de recherche important, il faudrait alors effectuer une corrélation avec 9 motifs : 3 pour les orientations multiplié par 3 pour les échelles. Ce lourd calcul prendrait en compte plusieurs orientations et plusieurs échelles mais ne tiendrait pas compte des déformations possibles de l´objet à suivre. De plus, le décalage n´est vraiment utile que si le mouvement effectif est proche du mouvement entre les deux motifs de référence.

Problème de déformation Nous nous intéressons au suivi de la main, déformable par définition. La corrélation, ainsi présentée à la section 2.2.1, ne permet d´en effectuer une bonne localisation que si elle reste dans une configuration stable, celle correspondant au motif de référence. Black et Jepson [BJ96a, BJ96b] proposent l´utilisation de vecteurs propres en combinaison avec une transformation affine de l´image. L´analyse en composantes principales d´images de mains, présentée à la section 3.1, permet de définir une transformation $\mathcal{T}$ de l´image ι en un vecteur ϕ de dimension inférieure au nombre de pixels de l´image ι (2.25) :

$$\phi = \mathcal{T}(\iota)$$

La fonction de reconstruction de l´image $(T)^{-1}$ est définie par:

$$\tilde{\iota} = \mathcal{T}^{-1}(\phi)$$

L´erreur de reconstruction entre ι et $\tilde{\iota}$ est définie par :

$$\begin{aligned} \varepsilon &= \|\tilde{\iota} - \iota\| \\ &= \sum_j (\tilde{\iota} - \iota)^2 \\ &= \sum_j (\mathcal{T}^{-1}(\phi) - \iota)^2 \end{aligned}$$

Cette fonction d´erreur permet de vérifier que l´image ι «ressemble» aux images ayant servi à définir la fonction $\mathcal{T}$.

Black et Jepson [BJ96a] ajoutent à cette transformation, une transformation affine $\mathcal{A}$ représentant le mouvement de l´objet à suivre. Ainsi, l´objectif est de déterminer les paramètres de la transformation afin de minimiser l´erreur :

$$\varepsilon(\mathcal{A}) = \sum_j \left(\mathcal{T}^{-1}(\phi) - \mathcal{A}(\iota)\right)^2$$

La transformation $\mathcal{A}$ est la composition d´une rotation d´angle θ, d´une translation dx et dy et d´un changement d´échelle de valeur ds :

$$\begin{aligned} \mathcal{A} &= R(\theta)T(dx,dy)S(ds) \\ &= \begin{pmatrix} \cos(\theta) & \sin(\theta) & dx \\ -\sin(\theta) & \cos(\theta) & dy \\ 0 & 0 & ds \end{pmatrix} \end{aligned}$$

2.3 Estimation des caractéristiques spatiales

Nous nous plaçons dans une configuration de reconnaissance dans un plan. Ainsi, les caractéristiques spatiales d´une main sont ses positions, sa taille et son orientation. Nous voyons, dans cette section, comment obtenir ces caractéristiques à partir des localisations de la section précédente.

2.3.1 Segmentation

L´utilisation de la segmentation permet facilement de déterminer l´ensemble de ces paramètres. Après l´étape de segmentation, par chrominance ou par différence, nous disposons d´une image I de niveaux de gris. Cette image contient, dans le cas de la segmentation par chrominance, la probabilité du pixel d´avoir une chrominance de peau. Pour la différence, les pixels sont nuls s´ils correspondent au fond et une valeur non nulle dans le cas contraire. Il est possible de seuiller[11] ces images. L´image créée est alors une carte indiquant la position des pixels de la main.

Les moments centraux de cette carte permet de calculer le centre de la main, sa taille et son orientation. Soit $I(i,j)$ l´image à deux dimensions ; la moyenne et la matrice de covariance sont définies[12] par :

$$\vec{\mu} = \begin{bmatrix} \mu_i \\ \mu_j \end{bmatrix} \qquad \Lambda = \begin{bmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{bmatrix} \tag{2.22}$$

11. Seuiller une image correspond à en créer une nouvelle dans laquelle les pixels sont nuls s´ils sont inférieurs au seuil et valent 1 dans les autres cas.

12. Nous renvoyons le lecteur à l´équation (2.17) pour une définition de μ_i et σ_{ij}

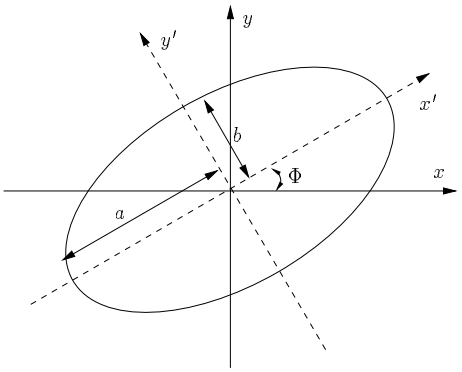


FIG. 2.9 – ***Paramètres de l´ellipse***. *Les dimensions a et b sont respectivement l´axe semi–majeur et semi–mineur de l´ellipse ; l´angle* Φ *donne son orientation.*

La position de la main est directement $\vec{\mu}$. L´orientation et la taille peuvent être calculées à partir de la matrice de covariance Λ en calculant l´ellipse issue de cette covariance :

Paramètres de l´ellipse

$$\begin{aligned} a &= \sqrt{\frac{\sigma_i^2 + \sigma_j^2 + \sqrt{(\sigma_i^2 - \sigma_j^2)^2 + 4\sigma_{ij}^2}}{\frac{N}{2}}} \\ b &= \sqrt{\frac{\sigma_i^2 + \sigma_j^2 - \sqrt{(\sigma_i^2 - \sigma_j^2)^2 + 4\sigma_{ij}^2}}{\frac{N}{2}}} \\ \Phi &= \tfrac{1}{2}\tan^{-1}\left(\frac{2\sigma_{ij}}{\sigma_i^2 - \sigma_j^2}\right) \end{aligned} \tag{2.23}$$

Dans ces formules, N est le nombre de pixels, c´est–à–dire le moment d´ordre 0. L´ambiguïté sur l´angle Φ peut être levée en considérant qu´il s´agit toujours de l´angle entre l´axe x et l´axe semi–majeur, par définition, nous avons donc toujours : $a \geq b$. De plus, la valeur de la fonction arc tangente est prise telle que :

$$-\frac{\pi}{2} \leq \tan^{-1}(x) \leq \frac{\pi}{2}$$

2.3.2 Corrélation

L´utilisation directe de la corrélation ne permet de calculer que la position du motif de référence. Si nous considérons l´utilisation de plusieurs motifs de référence, nous pouvons calculer une orientation et une échelle approximative en considérant la position du motif dans la matrice. Ong *et al* [OMG98] utilisent cette technique pour déterminer la direction de la tête de l´utilisateur. Ils utilisent 133 images permettant de couvrir une zone comprise entre -90° et $+90^\circ$ horizontalement et -30° et $+30^\circ$ verticalement. Une image est prise tous les 10°.

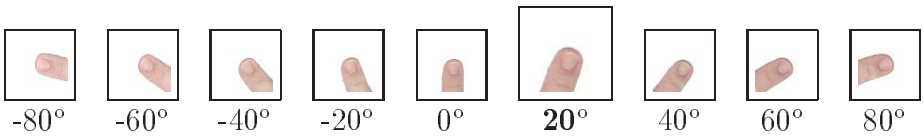


Fig. 2.10 – ***Calcul de l´orientation à partir de la corrélation***. *Le plus grand motif est celui ayant donné la valeur de corrélation la plus grande. L´orientation est alors approximée à 20°. La même technique peut être utilisée pour approximer la taille.*

2.4 Système multi–modules adaptatif [CM97, MDC98]

Les techniques proposées dans cette section présentent chacune des avantages et des inconvénients. Il convient donc de faire coopérer ces techniques afin d´obtenir une localisation et un suivi plus fiables et plus robustes. L´idée est d´obtenir le meilleur de chacune des techniques.

2.4.1 Architecture SERVP [CB94]

Le système de suivi proposé est basé sur une architecture dans laquelle un superviseur active et coordonne des modules visuels. L´architecture Servp[13] [CB94] a été développée comme une approche synchrone à l´intégration de modules pour la vision active[14]. Elle a été utilisée dans la construction de plusieurs systèmes incluant un système de navigation

13. Servp est l´acronyme de *Synchronous Ensemble of Reactive Visual Processes*, *Ensemble synchrone de modules visuels réactifs*

14. Pour une définition de la vision active, nous renvoyons le lecteur au livre de Blake et Yuille [BY92]

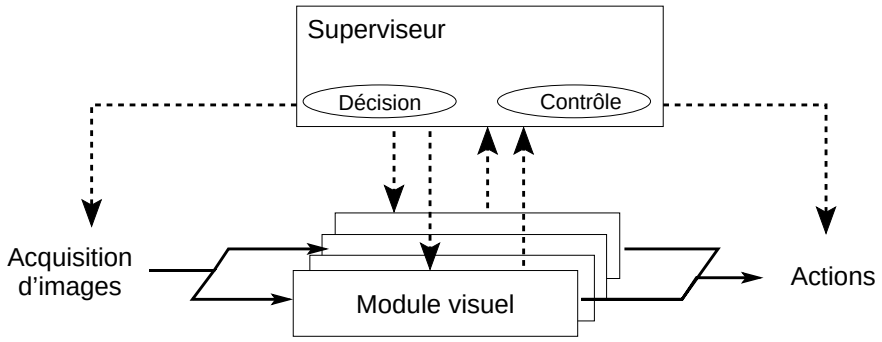


FIG. 2.11 – ***Architecture*** SERVP. *Elle a été développée comme une approche synchrone à l´intégration de modules pour la vision active. Un superviseur active et coordonne des modules visuels. Les traits pleins correspondent à la circulation de données tandis que les traits pointillés sont des commandes ou des évènements de sortie.*

d´un robot mobile [AJC97], le suivi de visage pour la communication vidéo [CB97] ou encore la détection et le suivi d´individu [CC95]. La figure 2.11 illustre ce superviseur.

Le système développé a utilisé le squelette logiciel *Chord* [Jon97, AJC97]. Le système *Chord* inclut des facilités pour la création de systèmes répartis ainsi que des opérateurs pour combiner et gérer les modules visuels.

2.4.2 Modules visuels

Un module visuel est graphiquement représenté par une boîte avec des ports d´entrée et de sortie. Le module lit des données (images et paramètres) en entrée, effectue des calculs et écrit les résultats sur les ports de sortie. Le module réagit à des commandes telles que l´initialisation, le démarrage et l´arrêt de l´exécution, et la terminaison. Il peut également générer des messages d´évènements, basés sur les résultats des calculs. Ceux–ci comportent des conditions d´exception qui déclenchent la modification du système auprès du superviseur. À la fin de son exécution, le module communique un message de succès ou d´échec au superviseur. La figure 2.12 donne la représentation graphique d´un module visuel.

Afin de contrôler l´exécution et la relation entre les modules visuels, *Chord* définit un ensemble de «combinateurs» binaires et de «modificateurs» [Jon97]. Les combinateurs permettent de définir l´ordre d´exécution des deux modules visuels. Ces combinateurs sont le combinateur «séquentiel» (*seq*), le combinateur «séquentiel conditionnel» (*if*), le combina-

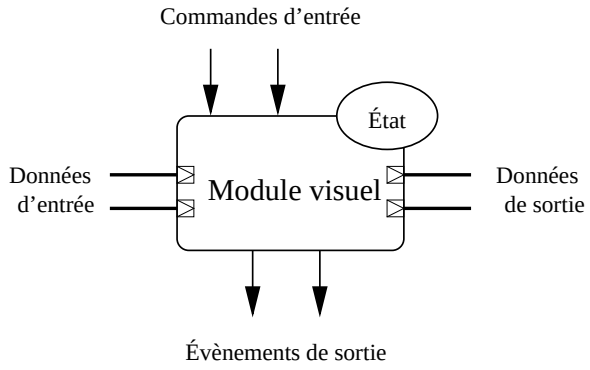


FIG. 2.12 – ***Modèle générique d´un module visuel dans*** **Chord**. *Un module visuel est graphiquement représenté par une boîte avec des ports d´entrée et de sortie. Le module lit des données (images et paramètres) en entrée, effectue des calculs et écrit les résultats sur les ports de sortie. Le module réagit à des commandes telles que l´initialisation, le démarrage et l´arrêt de l´exécution, et la terminaison. Il peut également générer des messages d´évènements, basés sur les résultats des calculs.*

teur «et logique» (*and*), le combinateur «ou logique» (*or*) et le combinateur «surveillant» (*watch*). Les modificateurs interviennent sur l´exécution ou le résultat des modules : négation, boucle, fonctionnement asynchrone ou synchrone. La composition de ces différents opérateurs permet de réaliser des graphes de contrôle, ou automates d´états finis.

La partie décisionnelle de l´architecture peut être exprimée par un ensemble de règles. Celles–ci sont exécutées par le chaînage avant et réagissent aux commandes et messages des modules visuels. Le superviseur reçoit des messages depuis les modules visuels concernant l´état du module, ainsi que les évènements générés.

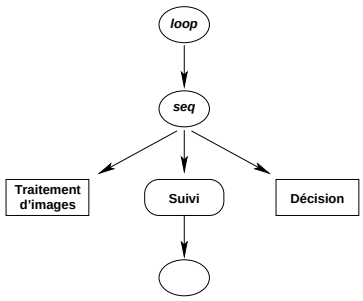
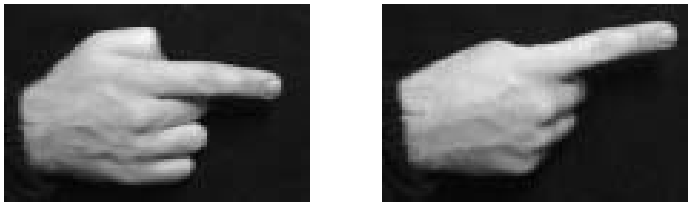


FIG. 2.13 – ***Graphe de contrôle du système de suivi dans le formalisme*** **Chord**. *Dans ce graphe, les cercles représentent les opérateurs de* Chord. *Le système exécute une boucle sur la séquence de trois modules visuels :* traitement d´image, suivi *et* décision. *Le module de* suivi, *représenté par une boîte arrondie, est également décrit par un graphe* Chord *et composé des traitements de vision bas–niveau*

La figure 2.13 présente le graphe de contrôle du système de suivi de main. Dans ce graphe, les cercles représentent les opérateurs de *Chord*. Le système exécute une boucle sur la séquence de trois modules visuels : *traitement d´image*, *suivi* et *décision*. Le module de *suivi*, représenté par une boîte arrondie, est également décrit par un graphe *Chord* et est composé des traitements de vision bas–niveau, présentés dans les sections précédentes : localisation par différence d´images, par segmentation de chrominance et par corrélation. Les traitements de vision sont suivis par un module d´estimation récursif basé sur un filtre de KALMAN [Kal60, WB97, BL93].

3 Extractions de caractéristiques de configurations

Dans cette section, nous nous intéressons à la seconde extraction de caractéristiques. Nous cherchons à déterminer un vecteur de mesures permettant de caractériser la configuration d´une main. La solution la plus évidente est de prendre directement l´image de la main. Cependant, la taille de ce vecteur (82944 valeurs si nous considérons les images utilisées dans cette thèse, c´est–à–dire 192×144 sur les trois canaux rouge, vert et bleu) ainsi que le nombre de valeurs sans intérêt, représentant le fond par exemple (dans les images de la figure 2.14, environ 60% des pixels sont des pixels du fond) ne permettent pas l´utilisation directe de l´image comme caractéristiques de la configuration. De plus, une même configuration peut avoir plusieurs aspects différents. Dans la figure 2.14, les deux images présentent la même configuration «*pointer*», mais leur apparence est assez différente.



(*a*) (*b*)

FIG. 2.14 – ***Deux images de main de la même configuration contenant une part importante de fond.*** *Les deux images de main présentant la même configuration* «pointer». *Dans ces images environ* 40% *des pixels correspondent à des pixels de la main, les* 60% *restants sont utilisés pour le fond. Les deux images présentent la même configuration, cependant l´apparence de ces mains est assez différente.*

Dans la suite de cette section, nous étudions deux techniques permettant de réduire la taille du vecteur de caractéristiques et de calculer des vecteurs proches pour des configurations identiques.

3.1 Analyse en composantes principales

L´analyse de matrices en composantes principales, également connue sous le nom de transformation de KARHUNEN–LOEVE, permet d´extraire un sous–espace optimal d´une distribution de points. Cette base orthonormée permet d´obtenir une décomposition li-

néaire de vecteurs. SIROVICH et KIRBY [SK87] ont montré que l´analyse en composantes principales d´un ensemble d´images de visages permettait d´obtenir un nouvel espace de projection dans lequel une image de 5000 pixels pouvait être réduite en un vecteur d´une cinquantaine de dimensions. TURK et PENTLAND [TP90, TP91] ont popularisé cette technique pour la reconnaissance de visages. Ils ont montré que la représentation des visages dans l´espace des composantes principales, appelée également «*facespace*», permettait de créer des classes distinguant les visages. Un nouveau visage peut être facilement classifié en projetant son image dans l´espace de composantes principales et en déterminant la classe de visages minimisant la distance de la projection avec la classe. Cette technique est à l´origine de nombreux systèmes de reconnaissance de visages [Ess96, Mas98, VIS, Kru]. Elle a été également utilisée pour la reconnaissance et la classification d´objets rigides [Col99], pour l´estimation de position d´un robot mobile par appariement d´images [Pou98, PC98] ou par l´appariement de données d´un capteur télémétrique laser [Wal97] ou encore pour la compression d´images vidéo [VSC99, Sch00]. Selon le même modèle que TURK et PENTLAND, nous pouvons reconnaître et classifier des images de mains [Mar95b, BM96, EW97].

Nous nous proposons de décrire l´utilisation de l´analyse en composantes principales pour la reconnaissance d´images de mains. Nous montrons que le vecteur de projection d´une image dans un espace de composantes principales donne une bonne description de la configuration de la main. Puis, nous abordons les problèmes de normalisation des images de mains en taille, orientation et position.

3.1.1 Analyse en Composantes Principales d´images de mains

Un vecteur e et un scalaire λ sont respectivement le vecteur et la valeur propre d´une matrice carrée $C = [c_{ij}]$ s´ils vérifient l´équation :

$$C.e = \lambda.e \tag{2.24}$$

Dans le contexte de l´analyse en composantes principales d´images, la matrice C est la matrice de covariance de m vecteurs ι_i. Les vecteurs ι_i sont les vecteurs à $N \times M$ dimensions des images d´intensité lumineuse I_i représentées sous forme de colonnes. Chaque vecteur ι_i est normalisé par la soustraction de l´image moyenne $\bar{\iota}$. Soit donné l´ensemble des m images ι_i, $i = 1 \ldots m$ constituant la distribution de la base d´apprentissage du modèle, la moyenne est calculée par :

$$\bar{\iota} = \frac{1}{m} \sum_{i=1}^{m} \iota_i$$

L´image normalisée de ι_i est notée $\hat{\iota}_i$:

$$\hat{\iota}_i = \iota_i - \bar{\iota}$$

Ces images normalisées forment la matrice A, concaténation des images :

$$A = [\hat{\iota}_1 \hat{\iota}_2 \dots \hat{\iota}_m]$$

La matrice de covariance C de dimensions $m \times m$ est alors définie par :

$$C = \frac{1}{m} A.A^T$$

En utilisant l´équation (2.24), nous obtenons le système linéaire :

$$C.E = E.\Lambda$$

La matrice E est constituée de n vecteurs propres e_i à n dimensions :

$$E = [e_1 e_2 \dots e_n]$$

et Λ est la matrice diagonale contenant les valeurs propres :

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}$$

Les vecteurs propres et les valeurs propres sont calculés par les transformations de HOUSEHOLDER ou de JACOBI [Kre93, PTVF92]. Ces deux méthodes cherchent à diagonaliser la matrice

$$E^{-1}.C.E = \Lambda$$

La matrice E permet de définir une transformation de l´espace image vers l´espace propre, la projection ϕ de l´image ι est donnée par :

$$\phi = \mathcal{T}(\iota) = E^T.(\iota - \bar{\iota}) \tag{2.25}$$

Le vecteur ϕ est un vecteur à n dimensions représentant le même contenu que l´image initiale ι. De même, nous pouvons définir la transformation–image permettant le passage de l´espace propre vers l´espace image. L´image reconstruite $\tilde{\iota}$ à partir du vecteur ϕ est défini par la transformation inverse partielle $\mathcal{T}^*$:

$$\tilde{\iota} = \mathcal{T}^*(\phi) = E.\phi + \bar{\iota} \tag{2.26}$$

Pour des images n´appartenant pas à la base d´apprentissage ou lorsque le nombre de dimensions de l´espace propre est réduit, la transformation correspond à une perte d´information, ainsi

$$\tilde{\iota} \neq \iota$$

La distance euclidienne ε entre l´image initiale et l´image reconstruite par la transformation $\mathcal{T}^{-1}$ est

$$\varepsilon = \|\tilde{\iota} - \iota\| \quad (2.27)$$
$$= \sum_j (\tilde{\iota}_j - \iota_j)^2 \quad (2.28)$$

Il s´agit de l´erreur résiduelle de reconstruction [MP95a, MP95b]. Cette erreur est également référencée comme la «*distance à l´espace propre*»[15] et peut être schématisée par la figure 2.15.

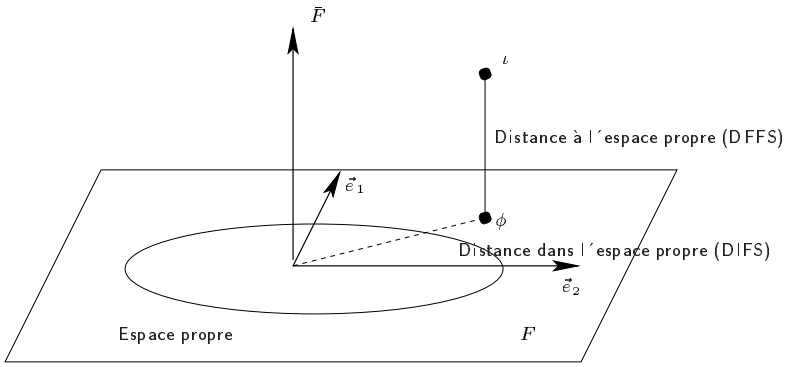


FIG. 2.15 – ***Distance à l´espace propre et distance dans l´espace propre.*** *L´espace propre est l´espace à deux dimensions nommé F. L´image ι se projette sur le point ϕ dans le plan F. La* «distance à l´espace propre»[15] *est représentée par le trait plein. La* «distance dans l´espace propre»[16] *est la distance euclidienne entre le point projeté et la moyenne de toutes les projections, elle est représentée en pointillés. L´ellipse représente la covariance de la distribution des points projetés dans l´espace propre. La dimension $\bar{F}$ est le complément orthogonal au sous–espace propre F. (d´après [MP95a])*

La figure 2.15 représente la «*distance à l´espace propre*». L´espace propre est l´espace à deux dimensions nommé F. L´image ι se projette sur le point ϕ dans le plan F. La

15. en anglais : «*distance–from–feature–space*»

«*distance à l´espace propre*» est représentée par le trait plein. La «*distance dans l´espace propre*»[16] est la distance euclidienne entre le point projeté et la moyenne de toutes les projections :

$$\bar{\phi} = \frac{1}{m} \sum_{i=1}^{m} \phi_i$$

L´ellipse représente la covariance de la distribution des points projetés dans l´espace propre. La quantité ε permet de mesurer la fiabilité de la projection [Wal97], permettant ainsi de déterminer le nombre de vecteurs propres suffisants pour une bonne représentation de l´image. Elle permet également de déterminer si une image est bien représentée par l´espace propre [TP91].

3.1.2 Vecteur de projection d´une image de main, un descripteur de la configuration

Cette section propose de montrer quelques résultats expérimentaux sur l´utilisation de l´analyse en composantes principales pour l´extraction de caractéristiques d´une image de main.

Considérons les 8 configurations de mains données par la figure 2.16. Elles constituent les commandes gestuelles dans des applications telles que celles présentées par Hauptmann *et al* [HMS88, Hau89] ou dans le démonstrateur Charade [BBL93].

Les 40 images de chaque configuration ont été prises dans les mêmes conditions d´éclairage et de paramètres de caméra. L´orientation de la main dans chacune des images a été gardée relativement constante, seules de petites déformations de la main, tel que le déplacement des doigts, sont prises en compte. Afin de considérer les tailles différentes de la main résultant directement de la configuration, les images sont normalisées en taille et ramenées à une taille $x \times x$. Dans les expériences suivantes, nous considérons x égal à 16, 32 et 64.

L´espace propre, issu des 40 images normalisées, est défini par la moyenne et les vecteurs propres représentés par la figure 2.17. La première image est la moyenne, les suivantes sont les 9 premiers vecteurs et valeurs propres associées.

3.1.3 Choix du nombre de vecteurs propres

L´analyse en composantes principales a permis la définition d´un changement de base pour l´espace de caractéristiques. Cependant, la dimension de cette base est identique à celle d´origine. Le problème est à présent de réduire celle–ci autant que possible mais sans perdre trop d´information. La première réduction est la suppression de tous les vecteurs dont la valeur propre associée est nulle. Ceci permet une réduction sans perte

16. en anglais : «*distance–in–feature–space*»

«*pointer*» «*stop*» «*gauche*» «*droite*»

«*poing*» «*main*» «*V*» «*L*»

FIG. 2.16 – ***Exemple de huit configurations de main.*** *Ces configurations permettent la commande gestuelle dans des applications telles celles présentées par* HAUPTMANN et al *[HMS88, Hau89] ou dans le démonstrateur* CHARADE *[BBL93].*

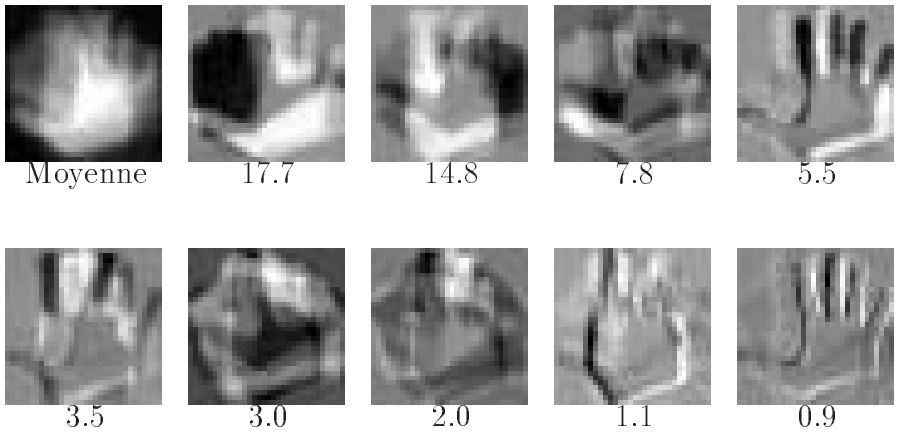


FIG. 2.17 – ***Moyenne, vecteurs et valeurs propres d´un ensemble d´images de configuration de mains***. *L´espace propre est créé à partir des* 40 *séquences pour les huit configurations. Les images de mains sont normalisées à une taille* 32×32.

d´information. La solution généralement adoptée [KS90, BM96, Wal97, Pou98] est de sélectionner le nombre de vecteurs telle que la fraction de la variance totale représente un pourcentage donné d´information. Cette fraction est donnée par :

$$q_K = \frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{2.29}$$

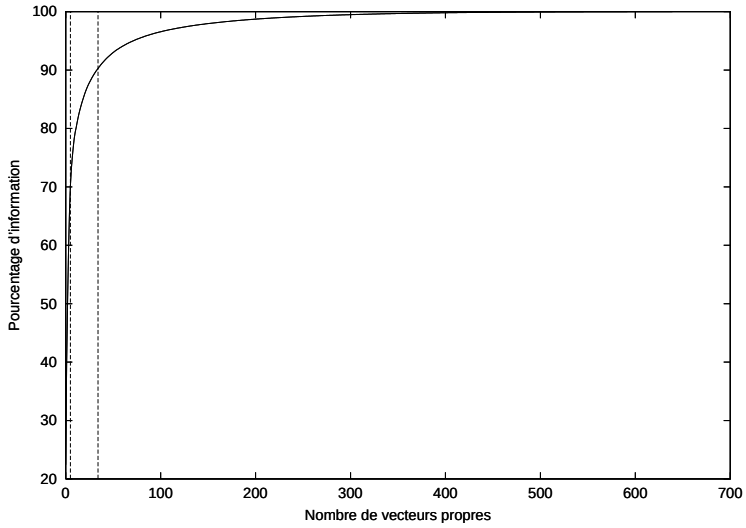


FIG. 2.18 – ***Pourcentage d´information en fonction du nombre de vecteurs propres.*** *L´espace propre a été créé à partir de* 40 *séquences pour* 8 *configurations. Les images sont normalisées à une taille* 32×32. *Les* 5 *premiers vecteurs représentent* 70% *de l´information et les* 34 *premiers plus de* 90%.

Le terme q_K donne le pourcentage d´information contenu dans les K premiers vecteurs, N est le nombre total de vecteurs et λ_i la i^{e} valeur propre. La figure 2.18 donne la courbe du pourcentage d´information en fonction du nombre de vecteurs propres conservés. Dans cette figure, les 5 premiers vecteurs représentent 70% de l´information et les 34 premiers plus de 90%.

3.1.4 Discriminant linéaire de FISHER

L´analyse en composantes principales permet de réduire la dimensionnalité de l´espace. Cependant, cette réduction de dimension optimise la reconstruction [Sch00] et non

pas la discrimination. Le discriminant linéaire de FISHER permet de réduire la dimensionnalité de l´espace en optimisant le facteur de discriminalité entre classes. Ainsi, si nous cherchons à séparer deux classes dans espace $\mathcal{E}$ à d dimensions, il est possible de réduire cet espace à une dimension en projetant les points sur une ligne [DH73]. Nous cherchons ici à déterminer l´orientation de la ligne maximisant la séparation des classes. Étant donné un point x de l´espace à d dimension, la projection de ce point sur la ligne est définie par la combinaison linéaire :

$$y = w^T x \tag{2.30}$$

Dans cette équation y est un scalaire et w les coordonnées de la droite de projection dans l´espace $\mathcal{E}$. Nous cherchons donc à déterminer les coordonnées, principalement la direction, de w.

Pour calculer cette droite, nous considérons une mesure de séparation entre les points projetés. Cette mesure est la différence entre les points moyens. Si nous notons μ_i la moyenne des points de classe i constituée de n_i exemples, elle est définie par :

$$\mu_i = \frac{1}{n_i} \sum_{x \in \mathcal{C}} x$$

Le point moyen projeté est donné par :

$$\begin{aligned} \tilde{\mu}_i &= \frac{1}{n_i} \sum_{y \in P(\mathcal{C})} y \\ &= \frac{1}{n_i} \sum_{x \in \mathcal{C}} w^T x = w^T \mu_i \end{aligned}$$

Le discriminant linéaire de FISHER est la fonction linéaire $w^T x$ pour laquelle le critère $J(w)$ est maximal :

$$J(w) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{\sigma}_1^2 + \tilde{\sigma}_2^2} \tag{2.31}$$

Ce critère mesure la distance entre les moyennes des points projetés des deux classes divisée par la somme des variances des deux classes :

$$\tilde{\sigma}_i^2 = \sum_{y \in P(\mathcal{C})} (y - \tilde{\mu}_i)^2$$

Pour expliciter w dans le critère discriminant $J(w)$, les *matrices de dispersion*[17] S_i sont définies par :

17. *scatter matrices*

$$S_i = \sum_{x \in \mathcal{C}} (x - \mu_i)(x - \mu_i)^T \tag{2.32}$$

Nous définissons alors :

$$\begin{aligned} \tilde{s}_i^2 &= \sum_{x \in \mathcal{C}} (w^T x - w^T \mu_i)^2 \\ &= \sum_{x \in \mathcal{C}} w^T (x - \mu_i)(x - \mu_i)^2 w \\ &= w^T S_i w \end{aligned}$$

La matrice S_w est la *matrice de dispersion intra–classe* [18], elle est définie par :

$$S_w = S_1 + S_2 \tag{2.33}$$

Ainsi

$$\tilde{\sigma}_1^2 + \tilde{\sigma}_2^2 = w^T (S_1 + S_2) w = w^T S_w w$$

De même, nous pouvons définir

$$\begin{aligned} (\tilde{\mu}_1 - \tilde{\mu}_2)^2 &= (w^t \mu_1 - w^t \mu_2)^2 && (2.34) \\ &= w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w && (2.35) \\ &= w^T S_B w && (2.36) \end{aligned}$$

où S_B est la *matrice de dispersion inter–classe* [19] définie par :

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2) \tag{2.37}$$

Ainsi, la fonction de critère J peut être redéfinie par

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{2.38}$$

Cette équation est connue en physique par le *quotient de* RAYLEIGH. Le vecteur w maximisant J doit alors satisfaire la condition :

$$S_B w = \lambda S_W w \tag{2.39}$$

soit

18. *within–class scatter matrix*
19. *between–class scatter matrix*

$$S_W^{-1} S_B w = \lambda w \tag{2.40}$$

Ceci définit w comme vecteur propre de $S_W^{-1} S_B$, une solution simple est de définir w par :

$$w = S_W^{-1}(\mu_1 - \mu_2) \tag{2.41}$$

puisque, seule la direction de w est importante et que S_B est dans la direction de $\mu_1 - \mu_2$. L´équation (2.41) définit le *discriminant linéaire de* FISHER.

Dans le cas de c classes, la généralisation du discriminant linéaire de FISHER implique $c-1$ discriminants. La projection de l´espace $\mathcal{E}$ à d dimensions vers l´espace à $c-1$ dimensions est définie par :

$$y = W^T x \tag{2.42}$$

où W est une matrice $d \times (c-1)$.

Les matrices de dispersion sont alors :

$$S_W = \sum_{i=1}^{c} S_i \tag{2.43}$$

$$S_B = \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^T \tag{2.44}$$

où μ est le *vecteur moyen total*[20] :

$$\mu = \frac{1}{\sum_{i=1}^{c} n_i} \sum_{i=1}^{c} n_i \mu_i \tag{2.45}$$

La fonction critère à maximiser est alors :

$$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|} \tag{2.46}$$

comme précédemment, ceci revient à satisfaire la condition :

$$S_B w_i = \lambda_i S_W w_i$$

où w_i est la i^{e} colonne de la matrice W, correspondant à la projection sur la dimension i.

La figure 2.19 présente, pour deux classes différentes gaussiennes, les dimensions principales sélectionnées par le calcul de l´analyse en composantes principales et du discriminant

20. *total mean vector*

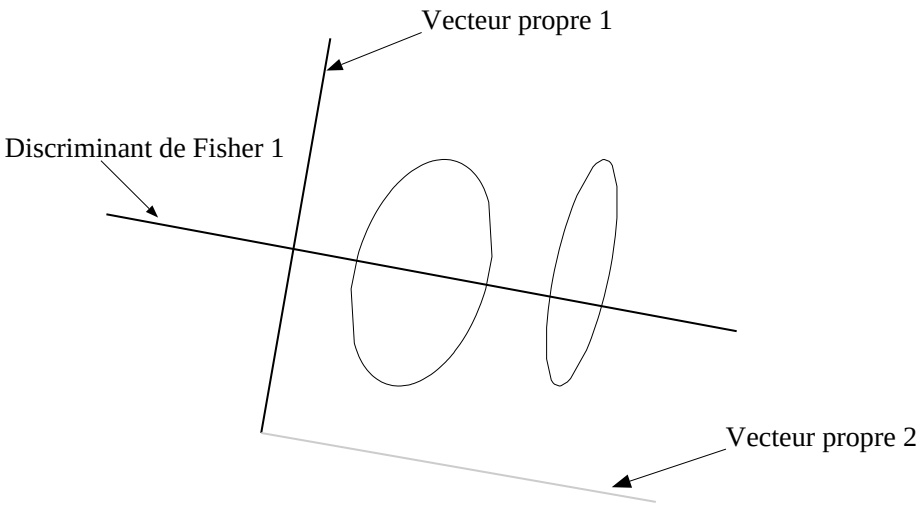


FIG. 2.19 – ***Dimensions principales de l´analyse en composantes principales et du discriminant linéaire de*** FISHER. *Les dimensions sont calculées pour deux classes gaussiennes différentes. L´axe principal des vecteurs propres, celui associé à la plus grande valeur propre, est noté en trait plein. Le vecteur propre principal ne permet pas de discriminer les deux classes contrairement au vecteur du discriminant de* FISHER. *(d´après [WC96])*

linéaire de FISHER. L´axe principal des vecteurs propres, celui associé à la plus grande valeur propre, est noté en trait plein. Cette figure montre que le vecteur propre principal ne permet pas de discriminer les deux classes contrairement au vecteur du discriminant de FISHER. BELHUMEUR *et al* [BHK96, BHK97] ont comparé, dans le cadre de la reconnaissance de visages, l´analyse en composantes principales, le discriminant de FISHER et la corrélation. Ils montrent que, sous des changements de luminosité, d´expressions faciales et le port de lunettes, l´analyse en composantes principales obtenait les résultats les plus mauvais avec un taux d´erreur compris entre 25% et 45%. Avec un taux d´erreur inférieur à 10%, la technique utilisant le discriminant de FISHER se trouvait en tête. WENG et CUI [CW96, WC96] préfèrent également l´utilisation du discriminant de FISHER, nommé «*caractéristique la plus discriminante*»[21] par opposition à l´analyse en composantes principales ou «*caractéristiques les plus expressives*»[22].

3.2 Invariants de Moments de HU

Les moments sont depuis longtemps utilisés dans toutes les sciences pour calculer la position du centre d´une distribution mais aussi sa variance. En vision par ordinateur, les moments permettent de calculer la position et l´orientation d´objets telle que la main, comme nous l´avons vu dans la section 2.3.

Une première utilisation de moments pour la reconnaissance de motifs géométriques a été proposée par HU [Hu62]. HU se proposait de reconnaître des caractères alphabétiques indépendamment de leur position, taille et orientation. Le calcul des moments permet à DUDANI *et al* [DBM77] de classifier automatiquement des vues d´avions. DAVIS et BOBICK [Dav96, DB97] utilisent les moments de HU pour la reconnaissance d´activités humaines. Ils construisent une image historique du mouvement. Dans cette image, appelée «*Motion History Image*» ou MHI, les pixels sont des valeurs scalaires correspondant à l´âge du mouvement. Elles capturent une apparence du mouvement. Les descriptions statistiques des images par des moments de HU sont utilisées pour comparer des activités candidates avec des activités cibles en utilisant une distance de MAHALANOBIS. Cette technique a été utilisée avec succès pour la reconnaissance des activités utilisée dans l´application `KidsRoom` [BDI97, BID+97, BID+].

Dans la suite de cette section, nous présentons la théorie des moments. Nous montrons comment obtenir des invariants en similitude et rotation. Enfin, nous présentons graphiquement notre utilisation des moments de HU pour la classification de configurations de mains.

21. «*Most Discriminant Feature*» ou MDF
22. ou «*Most Expressive Features*» ou MEF

3.2.1 Théorie des moments

Soit $\rho(x,y)$ la distribution plane de l´intensité d´une image, les moments d´ordre $p+q$ sont définis par :

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q \rho(x,y)dxdy$$

Il est à noter [Hu62] que les moments de tous ordres existent et que la séquence des moments m_{pq} est uniquement déterminée par la distribution $\rho(x,y)$ et inversement, $\rho(x,y)$ est uniquement déterminé par m_{pq}. TEAGUE [Tea80] précise qu´en utilisant un nombre suffisant de moments, il est possible en théorie de reconstruire l´image[23].

Le moment central μ_{pq} est défini par

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q \rho(x,y)d(x-\bar{x})d(y-\bar{y})$$

Dans cette formule, $\bar{x}$ et $\bar{y}$ correspondent au centre de la distribution :

$$\bar{x} = \frac{m_{10}}{m_{00}} \qquad \bar{y} = \frac{m_{01}}{m_{00}}$$

Les moments centraux sont invariants aux translations.

a) Fonction génératrice du moment

Considérons à présent la *fonction génératrice du moment*[24] $G(u,v)$ définie par [Kre93] :

$$G(u,v) = E(e^{ux+vy}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{ux+vy}\rho(x,y)dxdy$$

Cette fonction est telle que :

$$\frac{dG}{d^p x d^q y}(0,0) = E(X^p Y^q) = m_{pq}$$

En développant l´exponentielle en une série, nous obtenons

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_{p=0}^{\infty}\frac{1}{p!}(ux+vy)^p\rho(x,y)dxdy$$

Soit, en inter–changeant l´intégration et la sommation et en considérant le binôme de NEWTON défini par :

23. «by using a sufficiently large number of image moments we in principle recapture all the image information»

24. en anglais : *moment generating fonction*

$$\begin{aligned}(ux+yv)^p &= \sum_{r=0}^{p} C_p^r (ux)^p (vy)^{p-r} \\ &= (x^p, x^{p-1}y, x^{p-2}y^2, \ldots, y^p)(u,v)^p\end{aligned}$$

nous obtenons :

$$\begin{aligned}G(u,v) &= \sum_{p=0}^{\infty} \frac{1}{p!} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (ux+vy)^p \rho(x,y)dxdy \\ &= \sum_{p=0}^{\infty} \frac{1}{p!} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x^p, x^{p-1}y, x^{p-2}y^2, \ldots, y^p)(u,v)^p \rho(x,y)dxdy \\ &= \sum_{p=0}^{\infty} \frac{1}{p!} (m_{p0}, m_{p-1,1}, \ldots, m_{0p})(u,v)^p\end{aligned}$$

Introduisons également la transformation T :

$$T = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

telle que

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

c´est–à–dire

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha & \gamma \\ \beta & \delta \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix} \tag{2.47}$$

En développant, nous obtenons :

$$\begin{aligned}ux+vy &= (\alpha u' + \gamma v')x + (\beta u' + \delta v')y \\ &= (\alpha x + \beta y)u' + (\gamma x + \delta y)v'\end{aligned} \tag{2.48}$$

Ceci permet de définir la relation d´invariant :

$$ux + vy = u'x' + v'y' \tag{2.49}$$

Nous pouvons alors définir le moment transformé :

$$m'_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x'^p y'^q \rho'(x',y')dx'dy' \tag{2.50}$$

ainsi que la fonction de génération transformée :

$$G'(u',v') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_{p=0}^{\infty}\frac{1}{p!}(u'x'+v'y')^p\rho'(x',y')dx'dy'\frac{1}{|J|} \tag{2.51}$$

où $|J|$ est la valeur absolue du Jacobien de la transformation T.

L´incorporation du moment transformé de l´équation (2.50) dans l´équation (2.51) donne :

$$G'(u',v') = \sum_{p=0}^{\infty}\frac{1}{p!}(m'_{p0}, m'_{p-1,1}, \ldots, m'_{0p})(u',v')^p$$

b) Invariants

Le polynôme de coefficient $(x^p, x^{p-1}y, x^{p-2}y^2, \ldots, y^p)$ est un *invariant algébrique* de poids w si

$$(x'^p, x'^{p-1}y', x'^{p-2}y'^2, \ldots, y'^p)(u',v')^p = \Delta^w(x^p, x^{p-1}y, x^{p-2}y^2, \ldots, y^p)(u,v)^p$$

où Δ est le déterminant de la transformation (2.47) :

$$\Delta = \begin{vmatrix} \alpha & \gamma \\ \beta & \delta \end{vmatrix} = \alpha\delta - \beta\gamma$$

Lorsque w est nul, l´invariant est dit absolu.

Dans l´espace des moments, l´invariant devient :

$$(m'_{p0}, m'_{p-1,1}, \ldots, m'_{0p})(u',v')^p = |J|\Delta^w(m_{p0}, m_{p-1,1}, \ldots, m_{0p})(u,v)^p \tag{2.52}$$

c) Invariant en similitude

Considérons le cas de la transformation de similitude définie par la matrice de transformation T :

$$T = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \qquad \Rightarrow \qquad \Delta = \begin{vmatrix} \alpha & 0 \\ 0 & \alpha \end{vmatrix} = \alpha^2$$

Nous avons donc

$$x'^p y'^q = \alpha^{p+q} x^p y^q$$

L´invariant des moments découle de l´équation (2.52) :

$$\mu'_{pq} = \alpha^{p+q+2}\mu_{pq} \tag{2.53}$$

A l´ordre 0, nous avons

$$\mu' = \mu'_{00} = \alpha^2 \mu_{00} = \alpha^2 \mu \tag{2.54}$$

En combinant, les équations (2.53) et (2.54), nous obtenons l´invariant absolu:

$$\frac{\mu_{pq}}{\mu^{(\frac{p+q}{2})} + 1} \qquad \text{pour } p+q \geq 2$$

d) Invariant en rotation

Considérons la transformation de rotation d´angle θ définie par la matrice de transformation T:

$$T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Nous avons alors

$$J = \begin{vmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{vmatrix} = 1$$

Les invariants de moments sont alors identiques aux invariants algébriques:

$$\begin{aligned} (x'^p, x'^{p-1}y', x'^{p-2}y'^2, \ldots, y'^p)(u', v')^p & \\ = \; & (m'_{p0}, m'_{p-1,1}, \ldots, m'_{0p})(u', v')^p \\ = \; & (m_{p0}, m_{p-1,1}, \ldots, m_{0p})(u, v)^p \\ = \; & (x^p, x^{p-1}y, x^{p-2}y^2, \ldots, y^p)(u, v)^p \end{aligned} \tag{2.55}$$

Définissons le changement de variable:

$$\begin{bmatrix} U \\ V \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, \qquad \begin{bmatrix} U' \\ V' \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix}$$

Cette substitution donne une nouvelle relation de rotation:

$$U' = Ue^{-i\theta}, \qquad V' = Ve^{i\theta}$$

Ceci défini la relation d´invariance:

$$(I'_{p0}, I'_{p-1,1}, \ldots, I'_{0p})(U', V')^p = (I_{p0}, I_{p-1,1}, \ldots, I_{0p})(U, V)$$

soit

$$(I'_{p0}, I'_{p-1,1}, \ldots, I'_{0p})(Ue^{-i\theta}, Ve^{i\theta})^p = (I_{p0}, I_{p-1,1}, \ldots, I_{0p})(U, V)$$

Ainsi, puisque les coefficients des deux polynômes doivent être les mêmes, nous obtenons:

$$I'_{p0} = e^{ip\theta} I_{p0} I'_{p-1,1} = e^{i(p-2)\theta} I_{p-1,1} \dots I'_{1,p-1} = e^{-i(p-2)\theta} I_{1,p-1} I'_{0p} = e^{-ip\theta} I_{0p}$$

L´équation (2.55) et la substitution de variable permettent d´écrire :

$$(I_{p0}, \dots I_{0p})(U,V)^p = (x^p, \dots, y^p)(u,v)^p = (m_{p0}, \dots, m_{0p})(u,v)^p$$

et

$$(I'_{p0}, \dots I'_{0p})(Ue^{-i\theta}, Ve^{i\theta})^p = (x'^p, \dots, y'^p)(u,v)^p = (m'_{p0}, \dots, m'_{0p})(u,v)^p$$

Ces égalités permettent de définir $I_{p-r,r}$ et $I'_{p-r,r}$ en fonction de $\mu_{p-r,r}$ et $\mu'_{p-r,r}$

En considérant les moments du second et troisième ordre, l´ensemble des invariants suivant est défini :

Invariants du second ordre

$$\begin{aligned} S_1 &= \mu_{20} + \mu_{02} \\ S_2 &= (\mu_{20} - \mu_{02})^2 - 4\mu_{11}^2 \end{aligned} \tag{2.56}$$

Invariants du troisième ordre

$$\begin{aligned} S_3 &= (\mu_{30} - 3\mu_{12})^2 + (\mu_{03} - 3\mu_{21})^2 \\ S_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{03} + \mu_{21})^2 \\ S_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})\Big[(\mu_{30} + \mu_{12})^2 - 3(\mu_{03} + \mu_{21})^2\Big] \\ &\quad (3\mu_{21} - \mu_{03})(\mu_{03} + \mu_{21})\Big[3(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2\Big] \\ S_6 &= (\mu_{20} - \mu_{02})\Big[(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2\Big] + \\ &\quad 4mu_{11}(\mu_{30} + \mu_{12})(\mu_{03} + \mu_{21}) \end{aligned} \tag{2.57}$$

3.2.2 Résultats expérimentaux

Dans ces expérimentations, nous considérons les configurations de main vues en section 3.1.2 et la figure 2.16. Les invariants S_1 et S_2 ont été calculés pour chacune des 320 images de configuration. La figure 2.20(a) présente les valeurs de ces invariants. Les 8 configurations considérées sont fléchées sur la figure.

Cette figure montre que les valeurs des invariants pour une configuration donnée restent dans un sous–espace proche. Par contre, les configurations «*pointer*» et «*droite*», et «*L*» et «*gauche*» ne peuvent être facilement dissociées. L´utilisation d´un troisième invariant permet de lever une ambiguïté. La figure 2.20(b) montre les valeurs du troisième invariant en fonction du second pour l´ensemble des images considérées. Cette figure montre que l´ajout du troisième invariant ne permet pas de distinguer les configurations «*L*» et «*gauche*». Il faudra ajouter un quatrième invariant pour distinguer «*pointer*» et «*droite*». Dans le chapitre 3, nous étudierons la classification de ces 8 configurations.

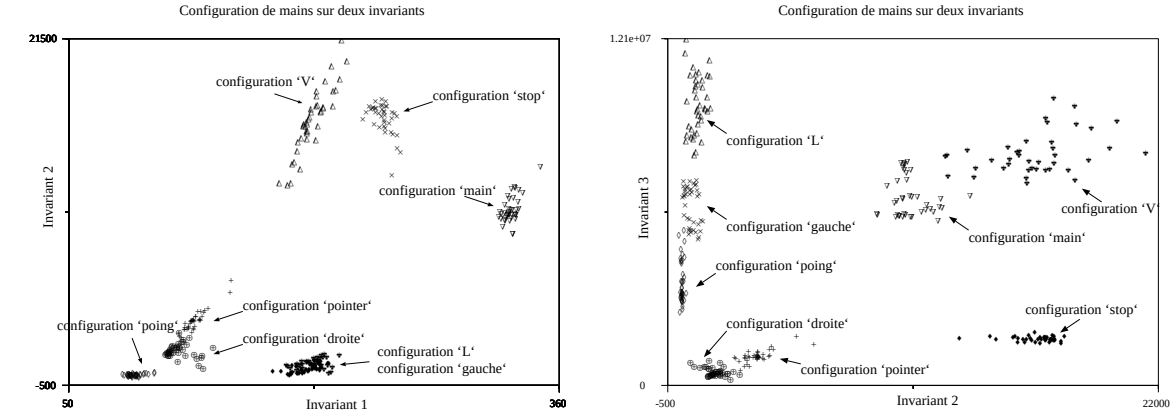


FIG. 2.20 – ***Valeurs des invariants pour les 8 configurations de mains considérées***. *Les 8 configurations considérées sont fléchées sur la figure, elles contiennent chacunes* 40 *exemples.* (a) *valeurs des deux premiers invariants.* (b) *valeurs des deux seconds invariants.*

4 Résumé du chapitre

Dans ce chapitre, nous avons présenté des méthodes d´extraction d´une image les caractéristiques de la main. Celles–ci sont à la fois les paramètres spatiaux de la main, parmi lesquels sa position, son orientation et sa taille, ainsi que les paramètres déterminant sa configuration.

Nous avons appuyé l´extraction des caractéristiques spatiales sur la localisation de la main. Nous présentons deux algorithmes de segmentation. La segmentation par chrominance s´appuie sur la teinte particulière de la chrominance de peau. Le déplacement de la main est utilisé par la seconde technique de segmentation. La localisation par apparence s´appuie sur la manifestation visuelle de l´objet à localiser. Cette opération s´effectue à l´aide d´une mesure de similarité entre l´image et un ensemble de manifestations de l´objet sous différentes conditions d´éclairage et de points de vue. Les méthodes de localisation présentent toutes des faiblesses et des points forts. La combinaison de celles–ci dans un système multi–modules est présentée. Ce système repose sur l´architecture SERVP dans laquelle un superviseur active et coordonne l´ensemble des méthodes de localisation. La décision d´activation s´effectue par la description d´un ensemble de règles exécutées par un algorithme de chaînage avant.

Nous avons ensuite proposé deux approches pour la détermination d´un vecteur de mesure, permettant de caractériser la configuration d´une main. L´analyse en composantes principales d´images de mains permet de définir un sous–espace de l´espace des pixels. Dans cet espace, une image de main est réduite à un vecteur de petite taille codant la configuration de la main. Cette analyse est étendue à l´utilisation du discriminant linéaire de FISHER. Celui–ci détermine un sous–espace optimisant le facteur de discriminalité, contrairement à l´analyse en composantes principales qui optimise le facteur de reconstruction. Les invariants de HU sont une seconde méthode pour extraire des caractéristiques de la configuration de mains. Ils sont calculés à partir des moments de la distribution.

Dans les chapitres suivants, nous nous appuyons sur ces caractéristiques pour reconnaître les gestes statiques et dynamiques. Dans un premier temps, le chapitre 3 présente l´extraction des caractéristiques de la main, utilisée pour la classification automatique de configurations. Puis, au chapitre 4, nous présentons la reconnaissance de gestes dynamiques sur des gestes d´écriture de caractères stylisés et sur des changements de configurations.

Le petit Nicolas en thèse [Pet]

La découverte (1/4)

«Des fois, c´est super, parce que je découvre des trucs que mon patron m´avait demandés. Évidemment, ça peut arriver à n´importe quelle heure, et mes parents ne sont pas toujours ravis.»

Chapitre 

Reconnaissance de configurations

Dans ce chapitre, la classification automatique de configuration de mains est présentée. Elle s´appuie sur les caractéristiques extraites au chapitre précédent. Les classifications euclidienne et bayésienne sont proposées et expérimentées. Une autre, basée sur la distance à l´espace propre, est également présentée.

1 Introduction

Les sections 3.1 et 3.2 du chapitre 2 ont montré deux solutions permettant d´extraire d´une image de main un vecteur de caractéristiques représentant de manière unique une configuration de main donnée, et en autorisent une classification automatique. Cette classification consiste à maximiser ou minimiser une fonction discriminante. La figure 3.1 représente le schéma bloc d´un système de classification par maximisation d´une fonction discriminante. Les fonctions $g_i(\vec{X})$ sont les fonctions discriminantes pour les classes C_i. La décision est, dans cet exemple, effectuée en choisissant la classe dont la fonction discriminante est maximale :

$$I = \arg \max_{1 \leq i \leq n} g_i$$

Dans la suite de cette section, deux fonctions discriminantes sont présentées, la fonction euclidienne et la fonction bayésienne. Nous notons

- $\mathcal{E}_i = \left\{ \vec{O}_i^j \right\}$, l´ensemble des N vecteurs caractéristiques définissant la classe C_i ;

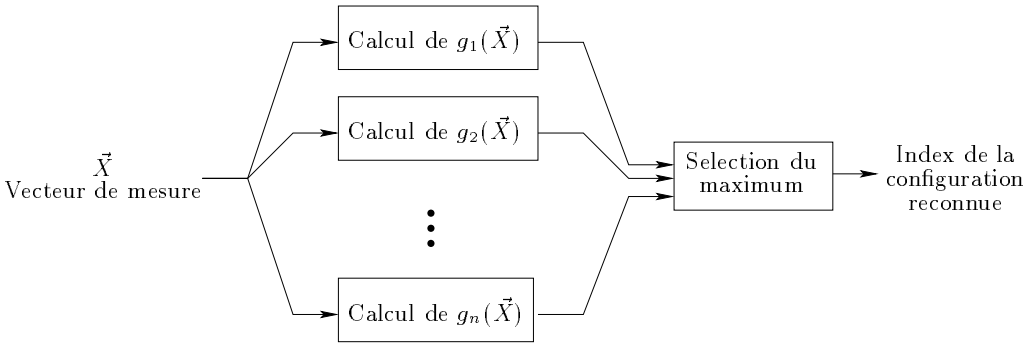


FIG. 3.1 – ***Schéma d´un système de classification.*** *La classification s´effectue sur un vecteur de mesure $\vec{X}$ parmi n classes. Les fonctions $g_i(\vec{X})$ sont les fonctions discriminantes pour les classes C_i. La décision est, dans cet exemple, effectuée en choisissant la classe dont la fonction discriminante est maximale.*

- $\vec{O}_i^j$, le j$^{\text{e}}$ exemple de la classe C_i ;
- $\vec{\mu}_i$, la moyenne des N vecteurs caractéristiques $\vec{O}_i^j$ de la classe C_i :

$$\vec{\mu}_i = \frac{1}{N} \sum_{j=1}^{N} \vec{O}_i^j$$

- Λ_i, la matrice de covariances :

$$\Lambda_i = \frac{1}{N-1} \sum_{j=1}^{N} (\vec{O}_i^j - \vec{\mu}_i)(\vec{O}_i^j - \vec{\mu}_i)^T$$

- $\vec{X}$, le vecteur de caractéristiques inconnu à classifier ;
- $\vec{\tilde{X}}_i$, le vecteur centré défini par :

$$\vec{\tilde{X}} = \vec{X} - \vec{\mu}_i$$

- d, la taille des vecteurs caractéristiques $\vec{O}_i^j$, $\vec{\mu}_i$, $\vec{X}$ et $\vec{\tilde{X}}_i$.

2 Classification euclidienne

La fonction de classification euclidienne est définie par

$$g_i(\vec{X}) = \sqrt{\vec{X}_i^T \vec{X}_i}$$

Elle permet de choisir la classe la plus proche du vecteur $\vec{X}$ dans l´espace de caractéristiques. La surface de décision entre deux classes est un hyper–plan situé à équi–distance des deux classes. La surface de décision entre deux classes C_1 et C_2 est l´ensemble des vecteurs $\vec{X}$ vérifiant l´équation :

$$g_1(\vec{X}) = g_2(\vec{X})$$

La figure 3.2 présente graphiquement les surfaces de décision entre trois classes dans un espace à deux dimensions.

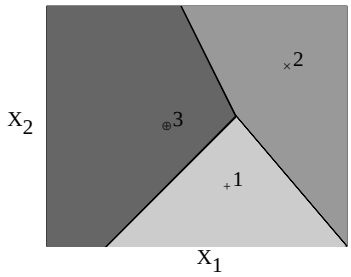


FIG. 3.2 – ***Surface de décisions euclidienne entre trois classes.***

3 Classification bayésienne

La classification bayésienne s´appuie sur la règle de BAYES qui lie la probabilité *a posteriori* à la probabilité *a priori* et la fonction de densité des classes conditionnelles :

$$P(C_i|\vec{X}) = \frac{P(\vec{X}|C_i)P(C_i)}{P(\vec{X})} \tag{3.1}$$

Dans cette formule, on note

- $P(C_i|\vec{X})$ est la probabilité *a posteriori* de la classe C_i étant donné le vecteur caractéristique $\vec{X}$;
- $P(\vec{X}|C_i)$ est la probabilité conditionnelle du vecteur caractéristique $\vec{X}$ sachant que la classe est C_i ;
- $P(C_i)$ est la probabilité *a priori* de la classe C_i ;
- $P(\vec{X})$ la probabilité du vecteur caractéristique :

$$P(\vec{X}) = \sum_{k=1}^{N} P(\vec{X}|C_k)P(C_k)$$

La surface de décision pour un vecteur caractéristique $\vec{X}$ entre deux classes C_1 et C_2 est définie par l´égalité

$$P(C_1|\vec{X}) = P(C_2|\vec{X})$$

Ainsi, la décision est la classe C_1 lorsque

$$P(C_1|\vec{X}) > P(C_2|\vec{X})$$

dans le cas contraire C_2 est choisi.
Nous pouvons donc définir la fonction discriminante par

$$g_i(\vec{X}) = P(C_i|\vec{X}) \tag{3.2}$$

en appliquant la règle de Bayes (3.1), la fonction g_i est définie par

$$g_i(\vec{X}) = \frac{P(\vec{X}|C_i)P(C_i)}{P(\vec{X})} \tag{3.3}$$

La valeur $P(\vec{X})$ étant constante quelque soit la classe C_i, elle peut être éliminée dans l´équation (3.3). De plus, le calcul du logarithme de g_i permet de séparer les deux probabilités :

$$\log\left(g_i(\vec{X})\right) = \log\left(P(\vec{X}|C_i)\right) + \log\left(P(C_i)\right)$$

Si les données sont supposées suivre une distribution normale, la probabilité du vecteur de caractéristiques $\vec{X}$, sachant la classe C_i, est définie par :

$$P(\vec{X}|C_i) = \frac{1}{\sqrt[d]{2\pi}\sqrt{|\Lambda_i|}} \exp\left[-\frac{1}{2}(\vec{X} - \vec{\mu}_i)^T \Lambda_i^{-1}(\vec{X} - \vec{\mu}_i)\right] \tag{3.4}$$

La fonction discriminante est alors

$$g_i(\vec{X}) = -\frac{1}{2}(\vec{X} - \vec{\mu}_i)^T \Lambda_i^{-1}(\vec{X} - \vec{\mu}_i) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|\Lambda_i|) + \log(P(C_i))$$

Dans cette fonction, le terme $\frac{d}{2}\log(2\pi)$, ainsi que le facteur multiplicatif $-\frac{1}{2}$ peuvent être éliminés car ils sont constants. Si, de plus, nous considérons l´ensemble des classes équiprobables, le terme $\log(P(C_i))$ peut également être supprimé. Nous obtenons alors la fonction discriminante bayésienne :

$$g_i(\vec{X}) = (\vec{X} - \vec{\mu}_i)^T \Lambda_i^{-1}(\vec{X} - \vec{\mu}_i) + \log(|\Lambda_i|) \tag{3.5}$$

Cette fonction correspond à la distance de MAHALANOBIS, $d_M(\vec{X})$, corrigée d´un terme fonction de la taille de la classe :

$$g_i(\vec{X}) = d_M(\vec{X}) + \log(|\Lambda_i|)$$

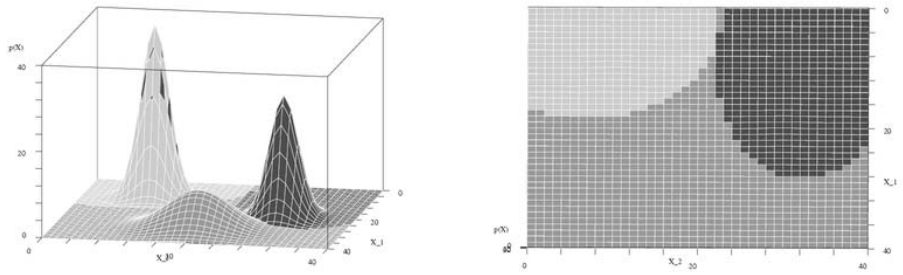


FIG. 3.3 – ***Surfaces de décision entre trois classes de loi normale***. *La figure de gauche présente les trois classes et celle de droite une vue du dessus. Ce graphique permet de visualiser les courbes d´équiprobabilité.*

La figure 3.4 montre la surface de décision bayésienne pour trois classes C_1, C_2 et C_3 définies par leur centre respectif μ_1, μ_2 et μ_3 ainsi que par les covariances représentées dans cette figure par les ellipses. Celles–ci représentent les surfaces d´équidistance $g_i(\vec{X}) = cte$.

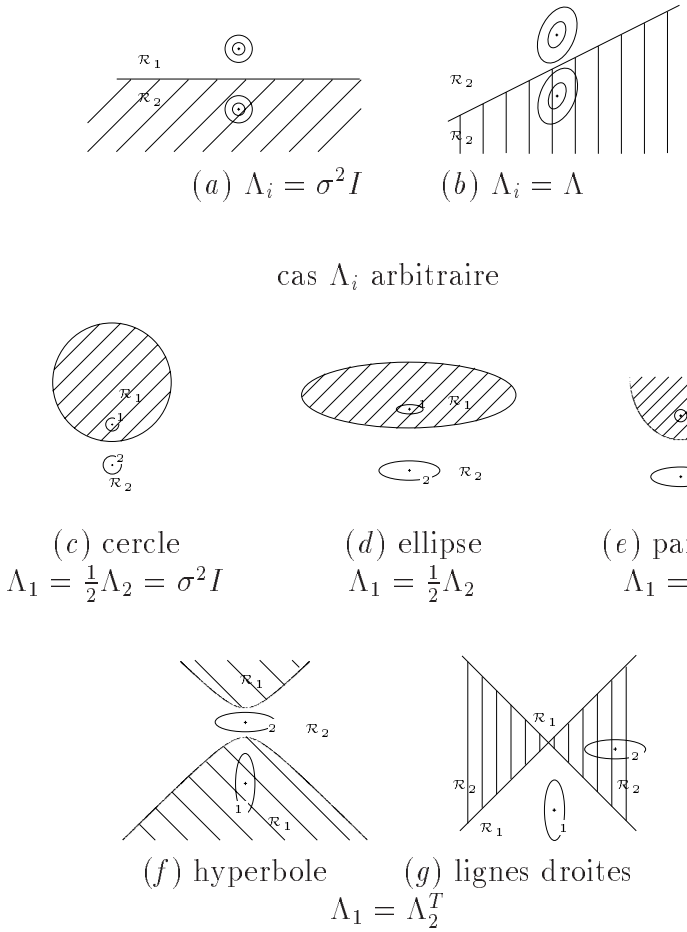


FIG. 3.4 – ***Surfaces de décision entre deux classes définies par une loi normale***. *Les ellipses représentent pour chaque classe les courbes d´équidistance au sens de la distance de* MAHALANOBIS. *(inspirée de [DH73])*

4 Expérimentations de classification

4.1 Classification de gestes représentés par les moments de Hu

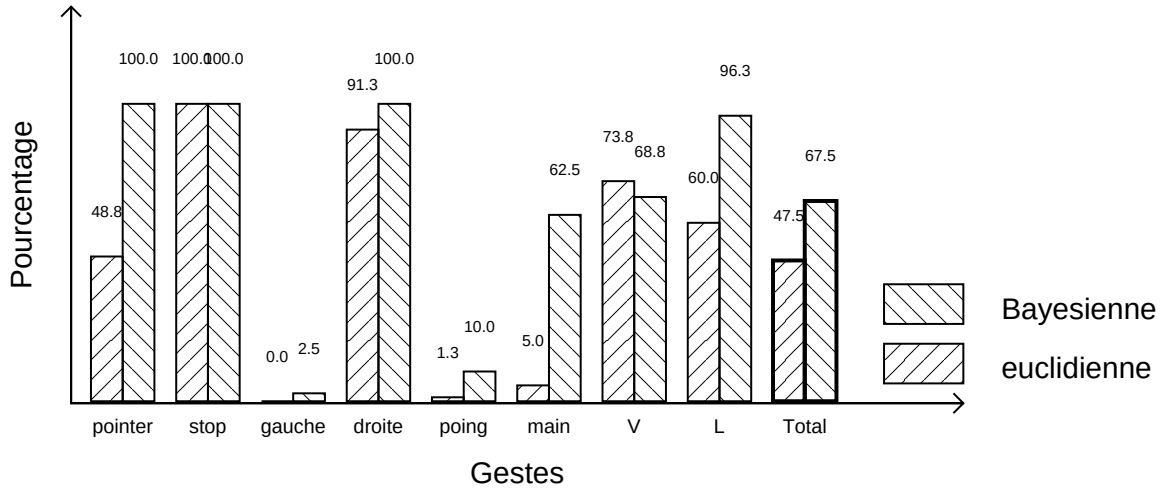


FIG. 3.5 – ***Classification de gestes par moments de*** Hu ***et par distance euclidienne ou bayésienne***

Dans cette section, nous classifions les gestes à partir des sept moments de Hu. La classification est effectuée en utilisant la distance euclidienne et bayésienne. Les résultats de ces classifications sont proposés dans la figure 3.5. Cette figure montre une reconnaissance supérieure en utilisant la classification de bayésienne. L´ensemble des gestes est mieux reconnu. Cependant de nombreux gestes ne sont pas reconnus, ils sont souvent confondus avec des gestes proches. Le geste «*gauche*» est ainsi reconnu comme étant de la classe «*droite*». Les moments de Hu étant invariants en similitude, les deux types de gestes donnent les mêmes moments. L´ajout d´une mesure permet de reconnaître cette symétrie.

4.2 Classification de main par distance à l´espace propre

Nous avons vu dans la section 3.1.1 du chapitre 2 (équation 2.27) que la «*distance à l´espace propre*» permet de mesurer la fiabilité de la projection. Nous pouvons utiliser cette quantité afin de mesurer si une image de configuration de main est bien représentée par l´espace propre, c´est–à–dire si elle est «similaire» aux images ayant permis la construction de l´espace. Ainsi, si un espace propre est créé pour chacune des configurations, il est possible de déterminer l´espace propre pour lequel une nouvelle image est la mieux représentée.

Soit $\mathcal{E}_i$ l´ensemble des configurations considérées. Les transformations $\mathcal{T}_i$ et $\mathcal{T}_i^*$ dénotent la projection et la reconstruction partielle d´une image. Ces transformations sont associées à l´ensemble $\mathcal{E}_i$. La quantité ϵ_i mesure la distance à l´espace propre dans $\mathcal{E}_i$ pour une image ϕ :

$$\epsilon_i = \|\mathcal{T}^*\Big(\mathcal{T}(\phi)\Big) - \phi\| \tag{3.6}$$

La classification consiste à chercher l´indice $\hat{i}$ minimisant l´erreur de reconstruction pour toutes les classes C_i :

$$\hat{i} = \arg_i \min \epsilon_i \tag{3.7}$$

4.2.1 Expérimentations sur la taille de la normalisation

Considérons les 8 classes de gestes décris dans la section 3.13.1.2 et la figure 2.16. Pour chacune de ces configurations, un espace propre est créé. Ces espaces sont notés $\mathcal{E}_{\text{pointer}}$, $\mathcal{E}_{\text{stop}}$, $\mathcal{E}_{\text{gauche}}$, $\mathcal{E}_{\text{droite}}$, $\mathcal{E}_{\text{poing}}$, $\mathcal{E}_{\text{main}}$, $\mathcal{E}_{\text{V}}$ et $\mathcal{E}_{\text{L}}$.

La figure 3.6 donne le taux de classification en fonction de la taille des images lors de la classification des configurations de mains d´une seconde séquence contenant 40 images de chacune des 8 configurations. L´expérimentation a été effectuée avec une taille d´image carrée de 8, 16, 32 et 64 pixels. Cette figure montre que la taille 8×8 est insuffisante pour une bonne classification. Nous réduisons le nombre de pixels de telle sorte que les doigts ou les écarts entre ceux–ci disparaissent. Le nombre de confusions entre la configuration «*V*» et «*pointer*» illustre parfaitement ce problème : les doigts sont fusionnés lors de la normalisation. Le nombre de confusions pour une taille d´images de 64×64 est plus petit : seules deux confusions existent.

5 Résumé du chapitre

Nous nous somme intéressés dans ce chapitre à la reconnaissance de configurations de main. Elle s´appuie sur les vecteurs de cacactéristiques mesurés dans le chapitre 2, au cours de l´étape d´analyse. Nous avons proposé trois méthodes de classification. Les deux

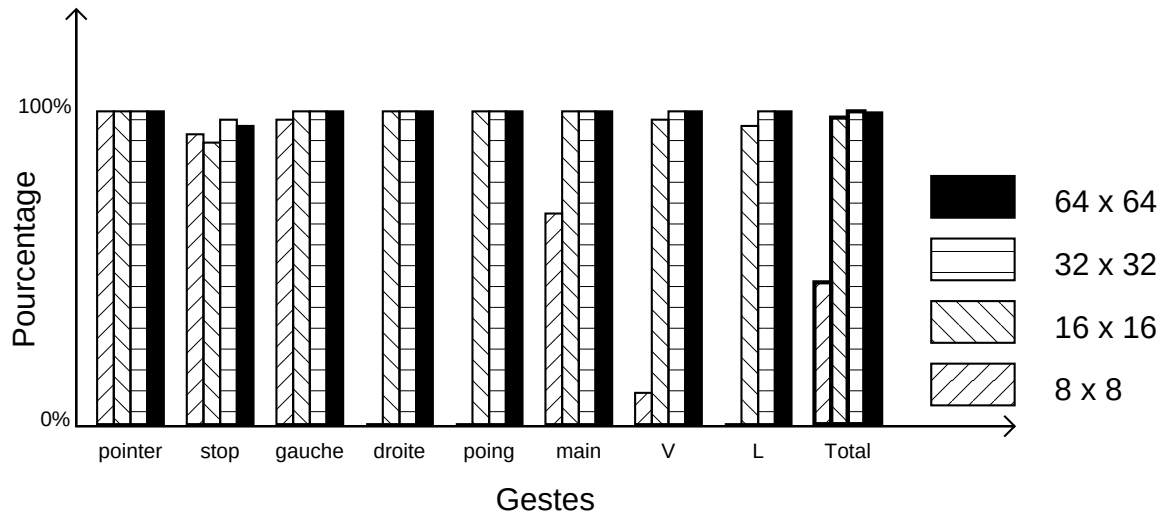


FIG. 3.6 – ***Évolution du pourcentage de classification par distance à l´espace propre par rapport à la taille des images.***

premières s´appuient sur la mesure d´un vecteur de caractéristiques à une classe représentée par son centre et, éventuellement, sa covariance. La distance euclidienne mesure la distance entre ce vecteur et la moyenne, tandis que, la distance bayésienne est pondérée par la covariance. L´expérimentation réalisée sur les vecteurs de caractéristiques calculés à partir des moments de HU montre que la classification bayésienne donnent de meilleurs résultats. Ils sont cependant faibles avec seulement 67.5% de reconnaissance. Il reste en effet de nombreuses confusions, les configurations «`gauche`» et «`poing`» ne sont pas reconnues. La raison principale de ces confusions est l´invariance des moments à la symétrie. De plus, la différence entre les configurations «`poing`» et «`pointer`» est faible et peu prise en compte par les moments. Nous pouvons remarquer que, en l´absence de ces deux configurations, la reconnaissance atteint 88% avec la classification bayessienne.

La troisième méthode de classification est basée sur l´analyse en composantes principales des images de mains. Il s´agit ici de mesurer l´erreur de reconstruction d´une image. Cette erreur est une mesure de similarité entre une image et les images ayant servi à construire le sous–espace propre. Nous avons ici expérimenté cette méthode sur des images de tailles différentes. Il apparaît que la reconnaissance est très bonne dès que la taille des images est supérieure à 16×16 pixels.

Dans le chapitre précédent, nous avons extrait d´une image de main un vecteur de caractéristiques permetttant de représenter une configuration de main. Ce chapitre a présenté des méthodes de classification de ces vecteurs. Ainsi, nous sommes capables de représenter une configuration de main par le vecteur de mesure ou bien par un symbole tel que «`droite`» ou «`gauche`». Cette classification est très utile lorsque les gestes sont dans les catégories «configuration statique», c´est–à–dire SPSL ou SPDL, proposées par HARLING et EDWARDS[1]. Lorsque la configuration est dynamique, convient–il mieux de faire une classification à chaque instant puis faire une reconnaissance dynamique sur ces classes ou bien effectuer directement la reconnaissance dynamique à partir des vecteurs de mesures ? La classification, comme tout processus diminuant le nombre de dimensions d´un espace, entraîne une perte d´information. La décision pouvant être erronée, celle–ci introduit alors un bruit supplémentaire dans le système de reconnaissance dynamique. Ainsi, il semble plus avantageux d´utiliser tout le vecteur de caractéristiques. Dans le chapitre suivant, la reconnaissance des gestes dynamiques est effectuée. La première méthode utilise la classe de geste, tandis que les deux suivantes effectuent la reconnaissance directement sur les vecteurs de caractéristiques.

1. Cette classification des gestes est proposée au chapitre 1, section 1.2.3, page 29

Le petit Nicolas en thèse [Pet]

La découverte (2/4)

«Ils se demandent si je deviens pas complètement fou, mais ma maman sait que mon papa n´aime pas qu´elle lui dise.»

Chapitre 

Étape de Reconnaissance : Classification des Gestes Dynamiques

Dans notre approche, un geste est représenté par une succession d´images dans lesquelles sont extraites les caractéristiques. Ce chapitre propose plusieurs techniques permettant de reconnaître l´aspect dynamique d´un geste issu de séquences de caractéristiques. Nous abordons trois approches : l´utilisation d´automates d´états finis pour reconnaître une suite de symboles, son extension aux modèles de Markov cachés puis une reconnaîssance probabiliste des gestes basée sur la description des séquences de caractéristiques par des trajectoires dans l´espace propre de caractéristiques locales. Dans un premier temps, une section introductive présente un état de l´art.

1 Introduction

La reconnaissance de gestes dynamiques est un problème difficile. Plusieurs difficultés apparaissent, parmi lesquelles la segmentation temporelle du geste et la reconnaissance de la dynamicité. La première difficulté pose les questions : «quand commence le geste?» et «quand finit-il?». Cette difficulté est largement augmentée par la coarticulation des gestes. Le commencement d´un geste peut, en effet, coïncider avec la terminaison du précédent. La reconnaissance de la dynamicité pose également le problème de reconnaître un geste

lorsque celui–ci a une durée variable. Enfin, comment mettre en correspondance deux gestes exécutés à deux vitesses différentes ?

Dans ce chapitre, nous nous concentrons sur la seconde difficulté : reconnaître des gestes de durée variable. Nous plaçant dans le domaine de l´interaction gestuelle pour les nouvelles interfaces homme–machine, nous faisons l´hypothèse que les gestes sont naturellement segmentés par un temps de latence ou de pause entre eux.

Aux chapitres 1 et 2, nous avons proposé des techniques permettant l´extraction de caractéristiques de la main dans une image. Ces caractéristiques sont de deux types : les caractéristiques spatiales correspondant à la position, orientation et taille de la main dans l´image ainsi que des caractéristiques de configurations reflétant la position relative des doigts. Selon la nature de l´application visée, le vecteur caractéristique associé à une image est composé d´un sous–ensemble de ces deux types de caractéristiques. L´extension de cette définition de l´image à la séquence d´images permet de représenter un geste par une trajectoire dans l´espace des caractéristiques choisi. Dans la suite de ce chapitre, nous noterons la trajectoire dans l´espace des caractéristiques $\mathcal{S}$ du geste g par :

$$\mathcal{T}_{\mathcal{S}}(g) = \{\vec{m}_{t_1}, \ldots, \vec{m}_{t_{T_g}}\} \tag{4.1}$$

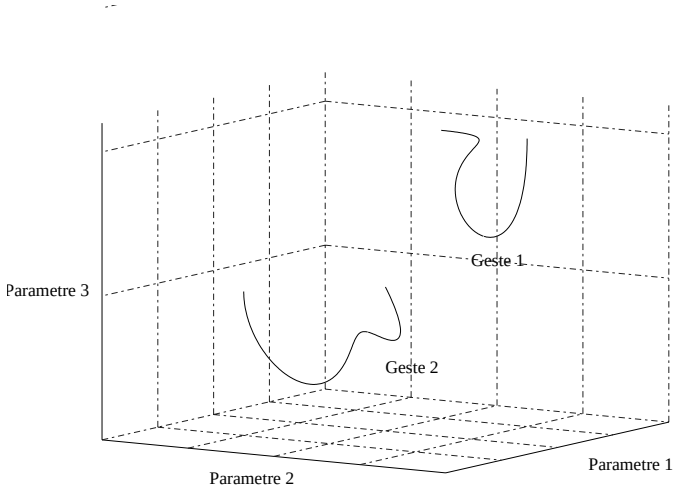


FIG. 4.1 – ***Exemples de trajectoires de deux gestes dans un espace de caractéristiques à trois dimensions.*** *(d´après [PSH97] et [Mar95b])*

Dans cette équation, $\vec{m}_{t_i}$ est le vecteur de caractéristiques à l´instant t_i dans l´espace $\mathcal{S}$, T est la durée du geste (en nombre d´images). La figure 4.1 illustre un exemple de deux trajectoires de deux gestes dans un espace de caractéristiques à trois dimensions.

Pour PAVLOVIĆ *et al* [PSH97], dans le contexte de l´interprétation visuelle de gestes, un geste est défini de la façon suivante :

> «Un geste de la main est processus stochastique dans l´espace des paramètres du modèle de geste $\mathcal{M}_\mathcal{T}$ sur un intervalle de temps défini I»[1]

La reconnaissance de gestes dynamiques est une étape de décision consistant à associer à une séquence de classe inconnue la classe la plus probable ou représentant mieux cette trajectoire. Plusieurs méthodes permettent cette décision. Nous en présentons trois existantes dans cette introduction avant de passer à celle que nous avons proposée.

1.1 Comparaison ou programmation dynamique

La *comparaison dynamique*[2] permet d´effectuer une mise en correpondance de deux trajectoires en considérant un décalage possible entre les points à mettre en correspondance. Le principe de cette technique est d´étirer ou de réduire le signal dans le temps. Cet alignement temporel entre les deux trajectoires est non linéaire, autorisant des «étirements» ou des «réductions» locales de trajectoires.

La figure 4.2 illustre le principe de mise en correspondance de deux séquences unidimensionnelles X et Y. Celles–ci sont définies par :

$$\begin{aligned} X &= x_1, x_2, \ldots, x_{T_X} \\ Y &= y_1, y_2, \ldots, y_{T_Y} \end{aligned} \tag{4.2}$$

où T_X et T_Y sont la taille des séquences. La correspondance des points de X et de Y, notée $c(k) = (i(k), j(k))$, permet de définir une *fonction de déformation temporelle*[3] F :

$$F = c(1), c(2), \ldots, c(k), \ldots, c(K) \tag{4.3}$$

Cette fonction, également appelée *chemin*, commence avec la mise en correspondance des points initiaux : $c(1) = (1, 1)$ et se termine avec les points finaux $c(K) = (T_X, T_Y)$. Entre les deux, le déplacement d´une correspondance $c(k-1)$ à une correspondance $c(k)$ se fait selon trois possibilités :

1. un déplacement vertical permet de mettre en correspondance deux points successifs de Y avec un point de X (cf. exemple $c(3)$ sur la figure 4.2), la trajectoire Y est ici étirée ;

1. «A hand gesture is a stochastic process in the gesture model parameter space $\mathcal{M}_\mathcal{T}$ over a suitable defined time interval I.»
2. *Dynamic Time Warping* ou DTW
3. time warping function

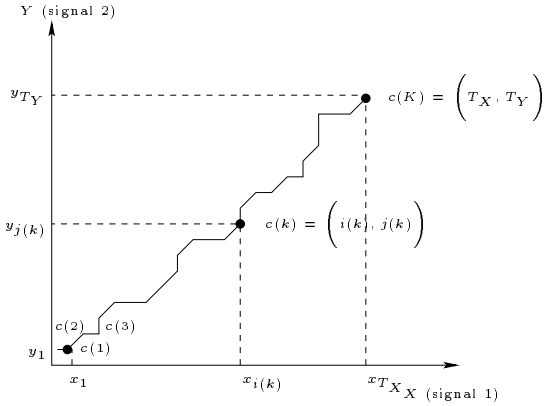


FIG. 4.2 – ***Principe de mise en correspondance de deux séquences par comparaison dynamique.*** *(d´après [HAJ90])*

2. un déplacement horizontal met en correspondance un point de Y avec deux points successifs de X (cf. exemple $c(2)$ sur la figure 4.2), la trajectoire X est étirée ;

3. un déplacement oblique signifie la mise en correspondance d´un point de Y avec un point de X (cf. exemple $c(1)$ sur la figure 4.2). Il n´y a pas ici d´étirement de trajectoire.

Il convient donc à présent de déterminer la séquence optimale de correspondance entre les deux trajectoires. La fonction de coût entre les deux signaux pour un chemin F correspond à la somme des distances euclidiennes $d\Big(c(k)\Big)$ entre chaque paire de correspondants :

$$D(X,Y,F) = \sum_{k=1}^{K} d\Big(c(k)\Big) \tag{4.4}$$

Il faut donc trouver la fonction F qui minimise cette distance :

$$F^* = \arg\min_F D(X,Y,F) \tag{4.5}$$

Une méthode simple consiste à calculer exhaustivement de tous les chemins possibles, puis de prendre le minimum. La comparaison dynamique décompose le problème de décision à K dimensions en K problèmes de décisions monodimensionnelles. Considérons $G(c(k))$, la distance minimale sur le chemin de $c(1)$ à $c(k)$. Cette distance peut être calculée récursivement par :

$$G\Big(c(k)\Big) = \min_{c(k-1)} G\Big(c(k-1)\Big) + d\Big(c(k)\Big) \tag{4.6}$$

Puisque le déplacement de $c(k-1)$ vers $c(k)$ n´est possible que dans trois directions, cette distance devient :

$$G\Big(c(k)\Big) = G(i,j) = \min \left\{ \begin{array}{l} G(i-1,j) \\ G(i-1,j-1) \\ G(i,j-1) \end{array} \right\} + d(i,j) \tag{4.7}$$

Dans cette solution, le calcul de la distance n´est plus effectué sur tous les chemins possibles mais uniquement en chaque point du chemin optimal.

Darrell et Pentland [DP92, DP93] ont utilisé cette technique pour la reconnaissance de deux gestes de la main. Un geste est représenté par la séquence des valeurs obtenues en calculant le score de corrélation d´une image avec un ensemble d´images modèle. La figure 4.3 présente les résultats de corrélation pour les images du geste «bonjour» (cf. figure 4.3*b*) avec les 4 images de références (cf. figure 4.3*b*). Ces images de référence ont été déterminées automatiquement en considérant les images telles que la corrélation n´est jamais inférieure à un seuil prédéfini. La concaténation des résultats de corrélation de la figure 4.3*a* permet de définir la signature d´un geste tel que présenté par la figure 4.4. Ils utilisent l´algorithme de comparaison dynamique pour mettre en correspondance des signatures de gestes. L´une des deux signatures est un modèle du geste, la seconde est le geste à reconnaître.

Nagaya, Seki et Oka [NSO96] utilisent l´algorithme de *programmation dynamique continue*[4] pour la mise en correspondance du geste à reconnaître et l´ensemble des modèles. Un geste est représenté par une trajectoire définie par un ensemble de points. La distance euclidienne entre le point de la trajectoire candidate et le point du modèle le plus proche est calculée. Ces distances sont sommées pour tous les points de tous les modèles par l´algorithme de programmation dynamique. La trajectoire choisie est celle qui minimise la distance totale.

1.2 Algorithme de `Condensation`

L´algorithme de `condensation` permet la propagation d´une densité probabiliste conditionnelle au cours du temps[5]. Cette méthode s´appuie sur un échantillonnage aléatoire des trajectoires pour permettre leur classification dans un cadre probabiliste. Cet algorithme stochastique a été introduit dans la communauté de la vision par ordinateur par Isard et Blake [IB96, BI98b, BI98a] pour le suivi d´individu.

4. Continuous Dynamic Programming
5. `Condensation` est l´acronyme de ***Conditionnal Density Propagation***

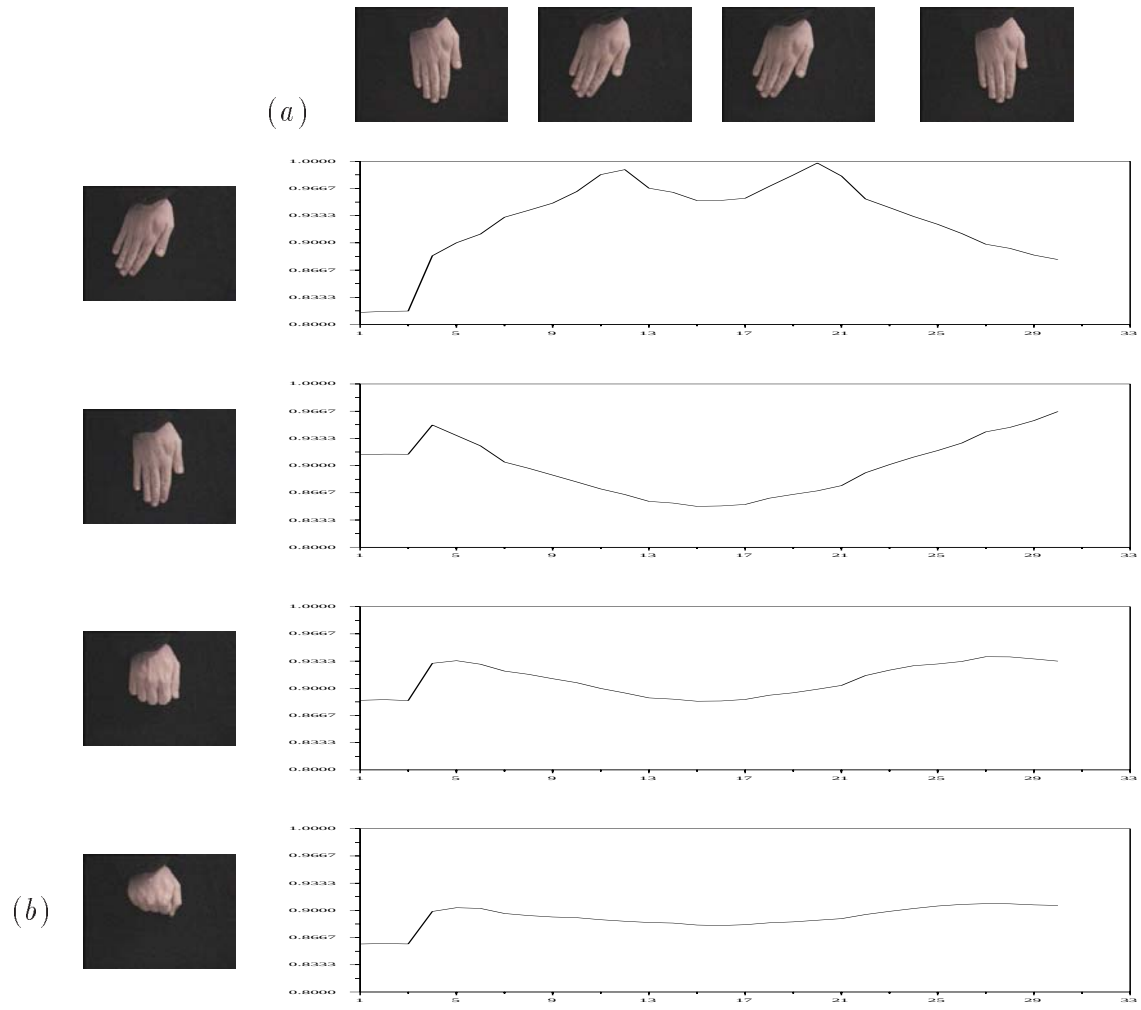


FIG. 4.3 – ***Résultats de corrélation pour les images d´un geste (a) avec un ensemble d´image de références (b).*** *(d´après [DP92])*

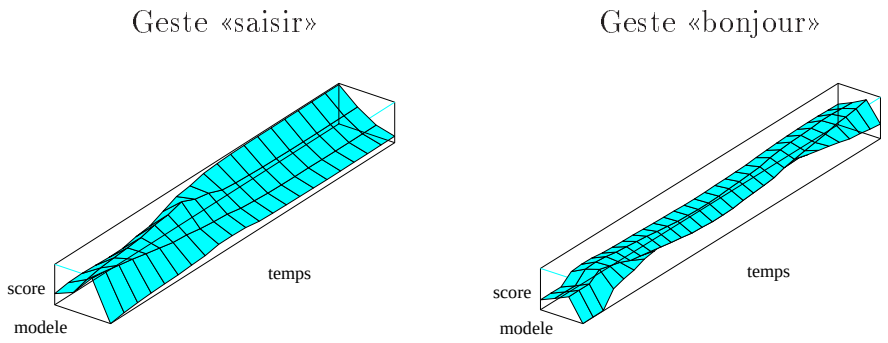


FIG. 4.4 – ***Signature des gestes «saisir» et «bonjour»***. *Ces signatures sont obtenues par la concaténation des résultats de corrélation telles celles de la figure 4.3. (d´après [DP92])*

L´état du modèle à l´instant t est noté x_t et son histoire est $\mathbf{X}_t = \left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, y_t\right)$ où $y_t \in \{1, \ldots, N\}$ est une étiquette définissant le modèle courant. Les caractéristiques dans l´image sont notées z_t et son histoire $\mathbf{Z}_t = \left(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\right)$. L´algorithme de `Condensation` permet de calculer la probabilité, $\mathbf{X}_t$, de l´histoire des états à l´instant t à partir de la connaissance de cette probabilité à l´instant $t-1$:

$$p\left(\mathbf{X}_t \mid \mathbf{X}_{t-1}\right) = p\left(\mathbf{x}_t \mid y_t, \mathbf{X}_{t-1}\right) P\left(y_t \mid \mathbf{X}_{t-1}\right) \tag{4.8}$$

où

$$\begin{aligned} P\left(y_t \mid \mathbf{X}_{t-1}\right) &= P\left(y_t = j \mid \mathbf{x}_{t-1}, y_{t-1} = i\right), \qquad \text{et} \\ p\left(\mathbf{x}_t \mid y_t, \mathbf{X}_{t-1}\right) &= p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}, y_{t-1} = i, y_t = j\right) \end{aligned} \tag{4.9}$$

L´algorithme de `Condensation` est donné par l´alogorithme 4.1 et représenté par la figure 4.5.

BLACK et JEPSON [BJ98] utilisent cette méthode pour reconnaître des marques dessinées à la main sur un tableau blanc[6]. Ils définissent un état par les paramètres : $s_t = (\mu, \phi, \alpha, \rho)$ où :

μ est un entier désignant le modèle correspondant ;

6. cf. le *Tableau magique* au chapitre 6

Algorithme 4.1 Algorithme de `Condensation`. La figure 4.5 présente cet algorithme sous forme graphique (d´après [BI98b])

Construction de l´état n selon la méthode :

1. **Sélection** d´un état $s_{t-1}^{(j)}$ à l´instant $t-1$ en utilisant la loi de probabilité de l´état notée π_{t-1}^{j}
2. **Prédiction** du nouvel état $\mathbf{s}_t^{(n)} = (x_t^{(n)}, y_t^{(n)})$ à l´instant t à partir de la probabilité $p(\mathbf{X}_t \mid \mathbf{X}_{t-1} = s_{t-1}^{(j)})$:
   (a) Prédiction de l´étiquette $y_t^{(n)}$ à partir des probabilités de transition d´états
   $$P(y_{t-1}^{(j)} = j \mid \mathbf{X}_{t-1} = s_{t-1}^{(j)})$$
   (b) Prédiction des paramètres $x_t^{(n)}$ de l´état à l´instant t
   $$p(x_t^{(n)} \mid \mathbf{X}_{t-1} = s_{t-1}^{(j)}, y_{t-1}^{(j)} = j)$$
3. **Mise à jour** de la vraissemblance de l´observation $\mathbf{Z}_t$ conditionnellement à l´état courant :
   $$\pi_t^{(n)} = p(\mathbf{Z}_t \mid \mathbf{X}_t = s_t^{(n)})$$

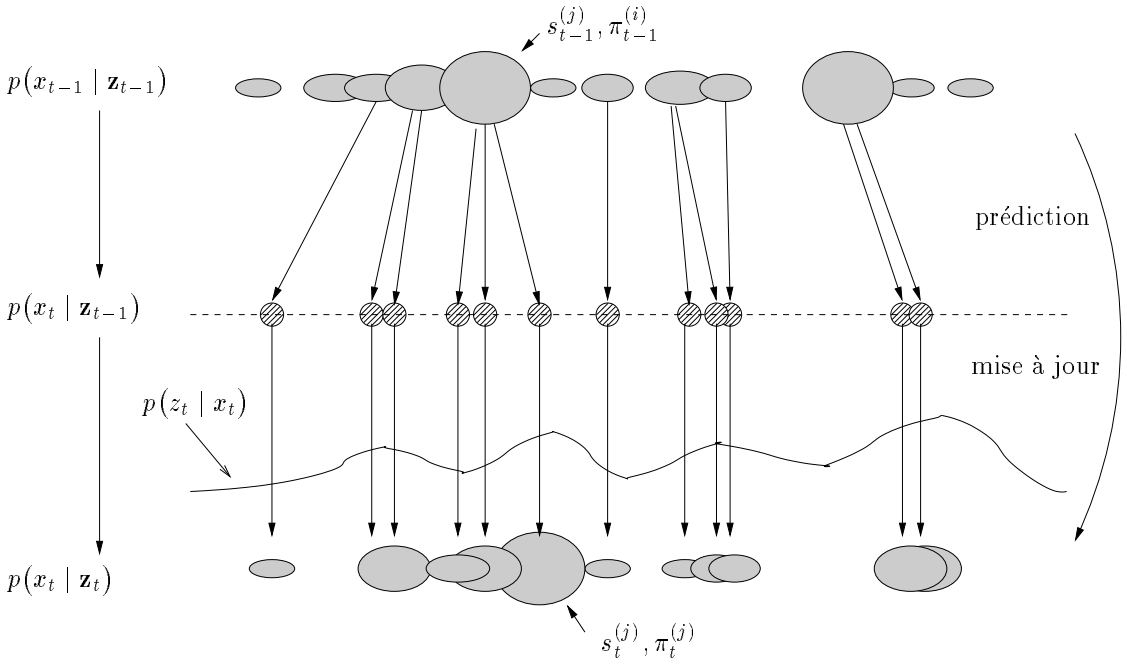


FIG. 4.5 – ***Une étape dans l´algorithme*** `Condensation`. *(d´après [BI98b])*

ϕ est la position dans le modèle permettant l´alignement du modèle avec les données ;
α est le paramètre d´amplitude verticale du modèle ;
ρ est le paramètre d´amplitude horizontale du modèle, c´est–à–dire temporelle.

1.3 Réseaux de neuronnes

Les réseaux de neurones, également appelés réseaux connexionnistes, sont fréquemment utilisés pour les problèmes de classification. Ils sont constitués de cellules appelés *neurones*, regroupés en couches. Chaque cellule d´une couche est liée à toutes les cellules de la couche précédente.

1.3.1 Perceptron Multicouches

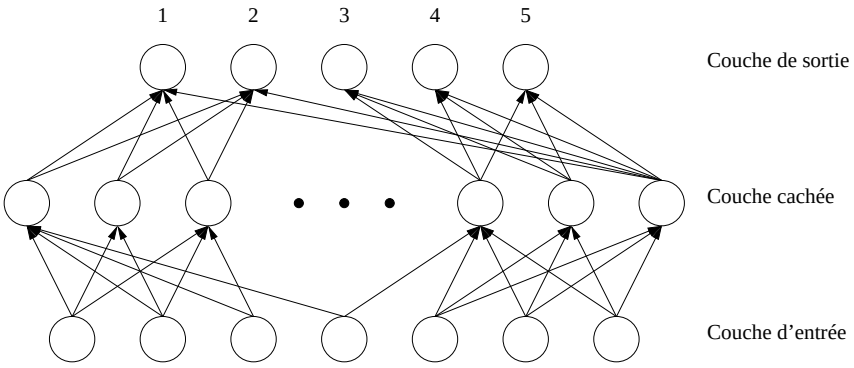


FIG. 4.6 – ***Exemple de Perceptron Multicouches (PMC) composé d´une seule couche cachée.***

Les perceptrons multicouches sont les réseaux de neurones les plus simples, ils sont entièrement déterminés par les matrices représentant les valeurs des liens entre les neurones. Après normalisation, les données sont fournies à la première couche appelée couche d´entrée. La dernière couche, la couche de sortie, fournit les résultats. Le réseau de neurones est entraîné en utilisant un *algorithme de propagation arrière*[7] permettant une

7. back propagation algorithm

minimisation de l´erreur quadratique entre la sortie attendue et la sortie obtenue. La couche de sortie répondra 1 pour le i^{e} neurone et 0 pour les autres neurones lorsque la i^{e} classe est déterminée comme correspondant à la donnée d´entrée. La figure 4.6 présente un exemple de Perceptron Multicouches (PMC) avec une seule couche cachée.

La classification de données temporelles ne peut directement être effectuée avec un perceptron multicouches classique. Il est possible de normaliser les observations sur l´axe temporel ou d´utiliser des variantes des perceptrons multicouches tels que les réseaux récurents proposés dans les sections suivantes.

1.3.2 Réseaux récurrents

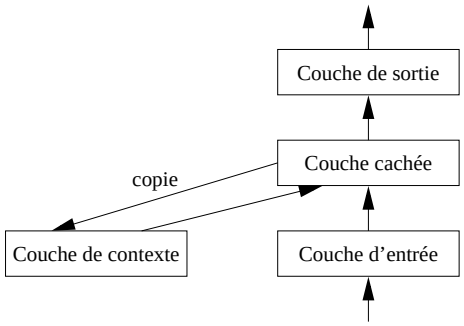


FIG. 4.7 – ***Architecture des réseaux de neurones récurrents*** *(d´après [MT91])*

Les réseaux récurrents prennent en compte l´historique des données en entrée. Les valeurs de neurones de la couche cachée sont copiées dans la couche de contexte à l´instant t. A l´instant $t+1$, ces valeurs sont réintroduites dans la couche cachée. L´apprentissage de tels réseaux s´effectue avec un algorithme de propagation arrière.

Ce type de réseau a été utilisé par MURAKAMI et TAGUCHI [MT91] pour la reconnaissance de dix signes de la langue des signes japonaise. Les données sont issues d´un gant numérique. Un réseau de neurones est utilisé pour reconnaître la configuration de la main initiale. Puis, un réseau récurrent reconnait les signes dynamiques. Dans cette étude, le taux de reconnaissance est de 98% pour le signeur ayant entraîné le réseau et 77% pour d´autres signeurs.

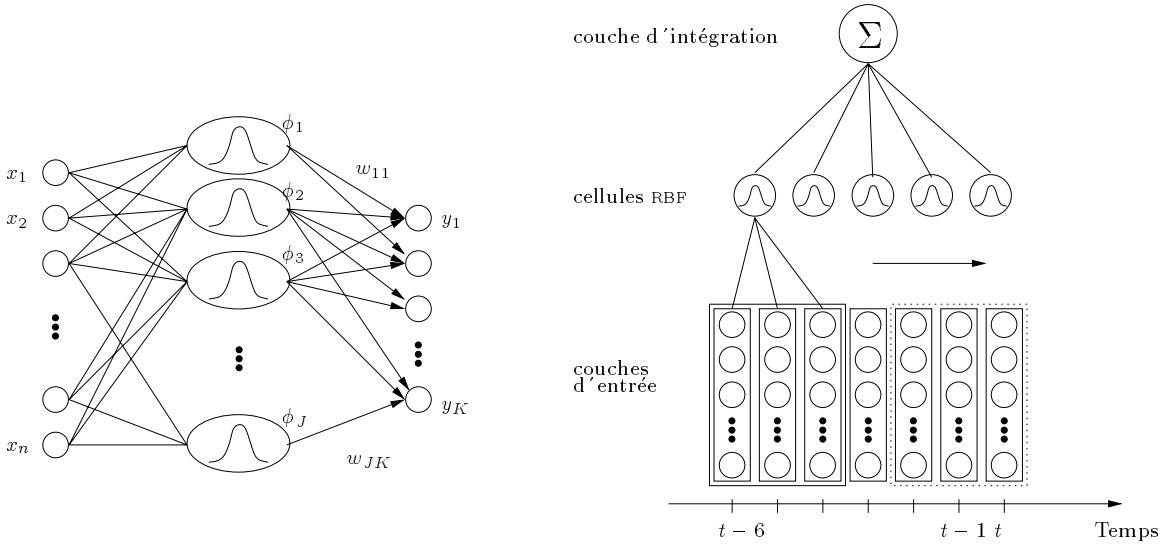


FIG. 4.8 – ***Architecture des réseaux*** RBF ***et*** TDRBF. *(a) Réseau* RBF *avec un vecteur d´entrée de taille n, J cellules cachées et un vecteur de sortie de taille K (d´après [Tar98]). (b) Structure d´un réseau* TDRBF *pour une seule classe avec une fenêtre temporelle de taille 3 et une intégration de 5 fenêtres. (d´après [HB98, HB99])*

1.3.3 Réseaux RBF

Les réseaux de type RBF [8] sont des réseaux à deux couches [Tar98]. Un exemple de l´architecture d´un réseau RBF est donné par la figure 4.8. Les cellules de la couche de sortie forment une combinaison linéaire des fonctions de bases calculées par les cellules de la couche cachée. L´activation de ces cellules est déterminée par la distance euclidienne entre le vecteur d´entrée $\vec{x}$ et un ensemble de prototypes. La correspondance non–linéaire de la première couche est construite par un ensemble de *fonctions de bases* [9] dont les centres correspondent aux vecteurs des prototypes dans l´espace d´entrée. Ces fonctions sont généralement choisies comme des fonctions gaussiennes. L´activiation de la j^{e} cellule cachée est donnée par la fonction :

$$\phi_j(\vec{x}) = \exp\left(-\frac{\|\vec{x} - \vec{\mu}_j\|^2}{2\sigma_j^2}\right) \tag{4.10}$$

Dans cette formule, $\vec{x}$ est le vecteur d´entrée, $\vec{\mu}_j$ est le vecteur du j^{e} prototype et σ_j est la taille de la gaussienne pour ce prototype. La réponse de la seconde couche est donnée par la combinaison linéaire pondérée de l´activiation des cellules de la couche cachée :

$$y_k = \sum_{j=1}^{J} w_{jk}\phi_j + w_{0k} \tag{4.11}$$

Howell et Buxton [HB98, HB99] proposent l´utilisation d´une variante des réseaux RBF prennant en compte la dynamicité temporelle des gestes. Un mécanisme de délai temporel est ajouté afin de manipuler le contexte temporel. Les réseaux TDRBF [10] combinent les données d´une fenêtre temporelle et d´un seul vecteur de données pour un réseau RBF. Une procédure d´apprentissage permet l´ajout et l´ajustement de cellules RBF. Les expérimentations réalisés par Howell et Buxton concernent la reconnaissance de gestes dans le contexte d´*interaction visuelle médiatisée*[11]. Les gestes réalisés face à une caméra permettent de la contrôler dans des applications de conférence vidéo. Quatre gestes sont considérés : deux gestes déictiques consistant à pointer à gauche ou à droite et deux gestes sémiotiques de mouvement rapide de la main[12] au dessus ou au dessous de la tête. Ils obtiennent, dans un premier temps, entre 69% et 75% pour un apprentissage effectué par une seule personne et une reconnaissance avec deux personnes différentes. Le résultat est amélioré à 100% lorsque les gestes ayant un facteur de confiance faible sont éliminés.

8. Radial Basis Function
9. basis functions
10. *Time–Delay Radial Basis Functions network*
11. *Visually Mediated Interaction* ou VMI
12. Howell et Buxton nomment ce geste *geste de vague* (*waving gesture*»)

1.3.4 Conclusion sur les réseaux de neurones

L´utilisation des réseaux de neurones pose deux problèmes fondamentaux liés à l´entraînement. Le premier est le temps de calcul nécessaire pour l´entraînement. Il convient de faire converger un modèle pouvant contenir plusieurs centaines de neurones. La convergence du réseau de MURAKAMI et TAGUCHI [MT91] nécessite plusieurs jours de calcul. Le second problème est lié à l´aspect «boîte noire» du réseau. On construit un réseau en choisissant le nombre de couches et de neurones par couche ; le réseau est entrainé à partir d´un ensemble d´exemples. À la fin de l´entrainement, soit le réseau est capable de reconnaître soit il n´a pas convergé.

2 Automate d´états finis [MC97]

En utilisant la classification des configurations présentée au chapitre 3, il est possible de définir un geste par une séquence de configurations. Ce geste peut être reconnu par un automate d´états finis construit selon le modèle suivant :

- les états correspondent à une configuration particulière de la main ; par exemple celles présentées à la figure 2.16, page 73 ;
- des états intermédiaires correspondant aux configurations intermédiaires, dus, en particulier, aux co–articulations, sont créés. Ces états permettent également de gérer les erreurs de reconnaissance de configuration ;
- les transitions entre états sont obtenues lorsqu´une nouvelle configuration est reconnue. Une transition vers un état intermédiaire intervient lorsque celle–ci n´est par reconnue.

Pour tenir compte de la variation temporelle de l´exécution des gestes des transitions réflexives sont permises dans tous les états. La figure 4.9 présente un automate d´états finis représentant un geste composé de deux configurations. Le double cercle représente l´état initial et le triangle l´état final. L´état C_1 correspond à la première configuration, l´état C_2 à la seconde configuration et l´état T est un état intermédiaire. Les étiquettes des arcs de transition ont la signification suivante :

C_i : la configuration i est reconnue ;
$\neg C_i$: la configuration i n´est pas reconnue ; ou bien une configuration, qui n´est pas la configuration i, est reconnue ;
$X \wedge Y$: combinaison des deux conditions X et Y.

Un système de reconnaissance de gestes utilise un ensemble d´automates d´états finis. Chaque automate correspond à un geste particulier. Lors de la reconnaissance d´un geste inconnu, le système fournit à chaque automate les configurations du geste à reconnaître. Dès qu´un automate atteint un état final, le geste correspondant est alors reconnu.

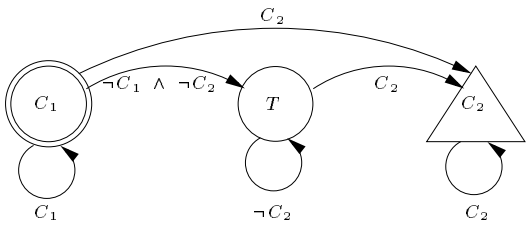


FIG. 4.9 – ***Automate d´états finis représentant un geste composé de deux configurations.***

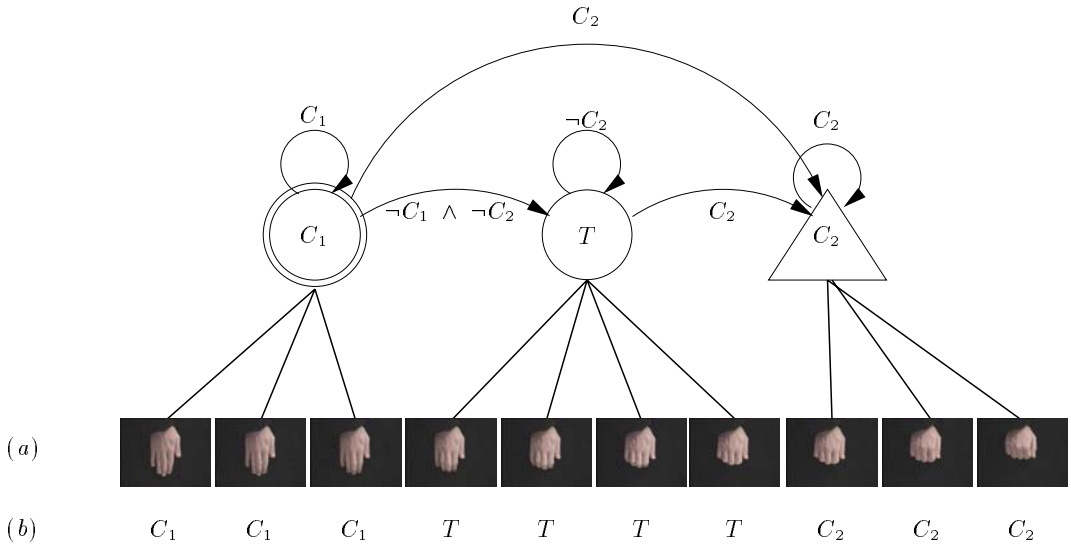


FIG. 4.10 – ***Exemple de reconnaissance d´un geste avec un automate d´états finis.*** *Le geste «saisir» est représenté par les images (*a*), les étiquettes (*b*) sont les états associés aux images.*

La nécessité d´utiliser un automate d´états finis par geste à reconnaître implique que la compléxité de calcul du processus de reconnaissance augmente linéairement avec le nombre de gestes. Cependant, ce système peut facilement être parallélisable. Une difficulté majeure de cette approche est la nécessité de construire manuellement tous les automates d´états finis représentant les gestes à reconnaître. Ceci peut être relativement complexe dans le cas où un grand nombre de gestes est souhaité.

L´avantage de cette approche est de s´appuyer sur une technique disposant d´une théorie. Un certain nombre d´opérateurs, permettant par exemple la composition d´automates, existe.

Cette technique peut être améliorée de deux manières. La première est de considérer les transitions entre états pondérées par une probabilité. La seconde amélioration est de considérer une méthode d´apprentissage permettant de déterminer automatiquement les transitions, ainsi que leur pondération. Ceci revient en fait à ne plus considérer des automates à états finis mais des chaînes de Markov. Nous avons proposé une introduction sur les chaînes de Markov dans [Mar00]. Dans la section suivante, nous considérons l´utilisation de modèles de Markov cachés.

3 Modèles de Markov Cachés

Les modèles de Markov cachés sont depuis longtemps adoptés par la communauté de reconnaissance de parole [RJ86, HAJ90]. Leur succès dans ce domaine et la structure linguistique proche [Cux99] ont permis le développement des modèles de Markov cachés dans le domaine de la reconnaissance et l´interprétation de la langue des signes [Sta95, YX94, Bra96]. Principalement utilisés avec des gants numériques [Bra96, HHH97], cette technique commence à être appliquée dans des systèmes de reconnaissance visuelle des gestes.

Un premier essai de système de reconnaissance de gestes, à partir de modèles de Markov cachés, aété réalisé par YAMATO *et al* [YOI92]. Ils se proposaient de reconnaître six gestes issus du tennis réalisés par trois sujets différents, en utilisant un modèle discret.

STARNER [Sta95, SP95] utilise des modèles de Markov pour reconnaître 40 mots de la langue des signes américaine. Dans ce système, les mains sont suivies en temps–réel par une caméra couleur. Le processus de suivi extrait la position des mains, leur forme et leur trajectoire. STARNER obtient un taux de reconnaissance de 99% lorsque l´utilisateur est équipé de gant de couleur, facilitant ainsi le processus de suivi et d´extraction basé sur la couleur. Le taux de reconnaissance est de 92% lorsque le processus s´appuie sur la couleur de la peau.

MORIMOTO *et al* [MYD96] explorent l´utilisation de modèles de Markov cachés pour la reconnaissance de quatre mouvements de la tête : «*oui*», «*non*», «*peut–être*» et «*bonjour*». Les gestes sont définis par :

oui cycle de mouvements de tanguage[12] ;

non cycle de mouvements de déviation[12];
peut–être cycle de mouvements de roulis[12] ;
bonjour mouvement unique de tanguage.

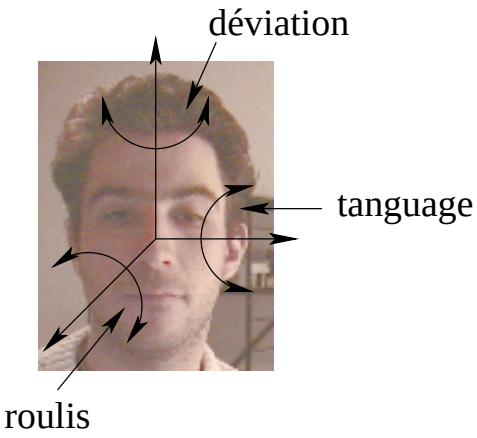


FIG. 4.11 – ***Définition des rotations d´axes déterminant les gestes de tête.*** *(d´après [MYD96])*

La figure 4.11 définit les rotations d´axes permettant de définir les quatre gestes.

Plus récemment, PODDAR *et al* [PSOS98] proposent l´utilisation des modèles de Markov cachés pour reconnaître les gestes d´un présentateur de la météo. Trois gestes sont considérés : «*pointer*», «*entourer*» et «*désigner une zone*». Les deux premiers gestes sont réalisés en pointant un doigt alors que la main est ouverte lors du geste de désignation. Le système proposé fonctionne sur des images télévisées et un taux de reconnaissance de 80%, en considérant uniquement les informations visuelles, est obtenu. Le taux augmente à 92% lorsque le système est couplé à un système de reconnaissance de parole. Dans ce cas, le modèle de Markov permet également la fusion des données multi–modales.

NAG *et al* [NWF86] proposent l´utilisation de modèles de Markov cachés pour la reconnaissance de chiffres écrits sur une tablette graphique. Ils utilisent des modèles discrets de type «gauche–droite». Les séquences d´observations sont les angles d´inclinaison des lettres, quantifiées en 128 étiquettes. Les résultats obtenus dans ce système sont compris entre 90.5% et 98.5% selon le nombre d´états dans les modèles.

Dans la suite de cette section, après une courte[13] introduction, nous abordons deux

12. respectivement *pitch*, *yaw* et *curl*
13. Une introduction plus complète est proposée dans [Mar00]

problèmes spécifiques aux modèles. Nous considérons le problème de détermination du type de modèle : discret ou continu. Le second est celui de choisir le nombre optimal d´états du modèle. Plusieurs méthodes de sélection sont présentées et nous proposons une méthode de sélection automatique. Ces deux problèmes sont validés sur des expérimentations de reconnaissance de gestes d´écriture d´un alphabet schématique.

3.1 Définition

Les modèles de Markov cachés sont des automates finis stochastiques. La suite des états constitue une chaîne de Markov non observable directement, elle est dite cachée. L´état du système à l´instant t détermine la loi de X_t. Un modèle de Markov caché, composé de N états, est représenté par l´ensemble des paramètres suivants :

- une matrice de transition entre états notée :
$$A = (a_{ij})_{i,j\in\{1\ldots N\}} ;$$
- pour chaque état $i \in \{1 \ldots N\}$, une loi de probabilité b_i à observations X discrètes ou continues. Le vecteur des lois de probabilité est noté :
$$B = \Big(b_1(X), \ldots, b_N(X)\Big) ;$$
- la loi de probabilité de l´état initial
$$\Pi = \Big(\pi_1, \ldots, \pi_N\Big).$$

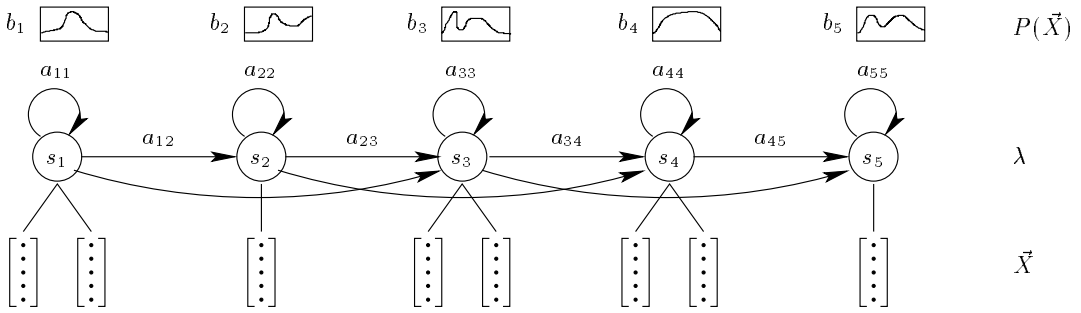


FIG. 4.12 – ***Représentation graphique d´un modèle de Markov caché à*** 5 ***états***

Le modèle de Markov correspondant est noté :

$$\lambda = (A, B, \pi) \tag{4.12}$$

3.1.1 Apprentissage

La phase d´apprentissage d´un modèle de Markov caché consiste à modifier les paramètres pour que le modèle ait une grande probabilité de générer les données d´apprentissage $\mathcal{A}$. Nous cherchons donc à maximiser la vraisemblance :

$$P = \prod_{k=1}^{k=K} P(O^k|\lambda) \tag{4.13}$$

où K est le nombre d´exemples des données d´apprentissage et

$$\mathcal{A} = (O^1, \ldots, O^K)$$

Les formules de Baum–Welch [HAJ90, You93] sont utilisées pour réestimer les paramètres a_{ij}, b_i et π_i, à partir des exemples. L´algorithme de Baum–Welch est un algorithme de type *restauration–maximisation*[14] [CC92]. Pendant l´étape de *prévision*, les données cachées sont substituées par leur espérance conditionnelle sachant l´observation O^k et le modèle λ. La seconde étape *maximise* la vraisemblance complète.

3.1.2 Reconnaissance : classification d´une séquence d´observations

La méthode classiquement utilisée pour classifier une observation dans plusieurs classes $\mathcal{C}_{l_{1\leq l\leq V}}$ est de représenter chaque classe par un unique modèle de Markov caché λ_l. Ce modèle est entraîné, selon la méthode présentée à la section précédente, à partir d´un ensemble d´exemples du geste considéré $\mathcal{A}_l = (A_l, B_l, \Pi_l)$.

La classification d´une séquence d´observations inconnue O consiste à déterminer la classe $\mathcal{C}_{l^*}$ parmi les N classes $(\lambda_1, \ldots, \lambda_N)$ tel que :

$$l^* = \arg \max_{1\leq l\leq N} P(\mathcal{C}_l \mid O)$$

où, plus exacte, si nous considérons que λ_l est le modèle de Markov caché représentant la classe $\mathcal{C}_l$

$$l^* = \arg \max_{1\leq l\leq N} P(\lambda_l \mid O) \tag{4.14}$$

En pratique, seule la probabilité $P(O \mid \mathcal{C}_l)$ est calculable à partir des algorithmes *avant–arrière* (cf. [Mar00]). On rappelle que ces algorithmes permettent, pour chaque modèle de Markov caché λ_l, de calculer la probabilité $P(O \mid \lambda_l)$.

La règle de Bayes permet de calculer la probabilité d´obtenir la classe $\mathcal{C}_l$, sachant que l´observation est O :

14. *expectation–maximization*

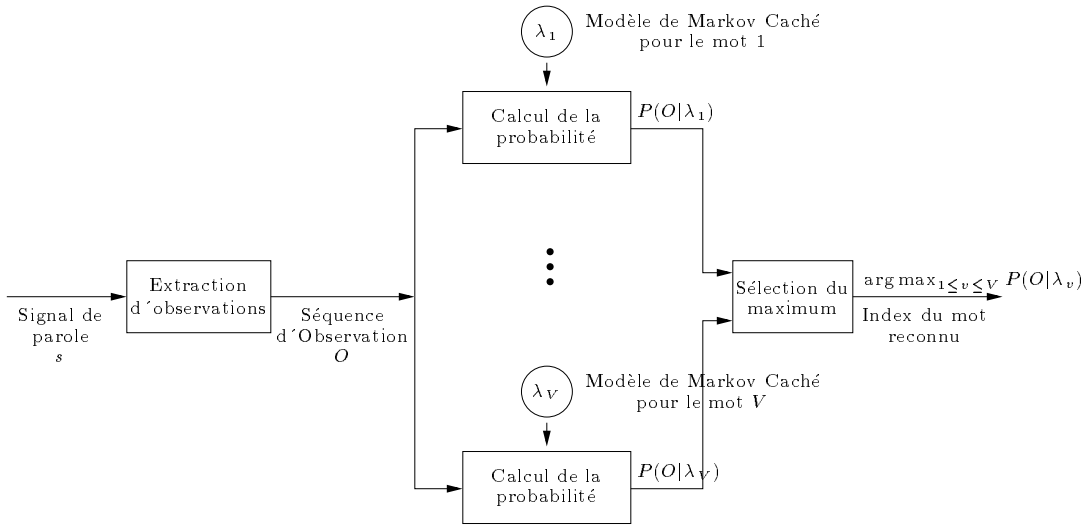


FIG. 4.13 – ***Processus de reconnaissance d´un mot isolé, basé sur un modèle de Markov caché.*** *Ce processus utilise un modèle de Markov caché par mot du vocabulaire (composé de v mots). Le signal de parole d´un mot m est traduit en une séquence d´observations O en utilisant une table de correspondance. La probabilité de la séquence est calculée avec chacun des modèles. Le mot reconnu est celui dont le modèle a donné la probabilité la plus grande.*

$$P(\lambda_l \mid O) = \frac{P(O \mid \lambda_l)}{P(O)} P(\lambda_l) \tag{4.15}$$

Sans connaissance *a priori* sur la distribution des classes et des séquences d´observations. Nous pouvons considérer toutes les classes équi–probables. Ceci implique :

$$\exists k \in \mathbf{R}^+ \qquad P(\lambda_l \mid O) = k.P(O \mid \lambda_l) \tag{4.16}$$

Ainsi, la classe affectée à O est définie par :

$$l^* = \arg \max_{1 \leq l \leq N} P(O \mid \mathcal{C}_l)$$

Dans le contexte de la reconnaissance de gestes, nous cherchons à estimer la classe $\mathcal{C}(g)$ du geste maximisant la probabilité des classes, sachant la trajectoire de celui–ci dans l´espace de caractéristiques $\mathcal{S}$:

$$\mathcal{C}(g) = \arg \max_{1 \leq l \leq N} P(\mathcal{C}_l \mid \mathcal{T}_{\mathcal{S}}(g)) \tag{4.17}$$

$$= \arg \max_{1 \leq l \leq N} P(\mathcal{T}_{\mathcal{S}}(g) \mid \mathcal{C}_l) \tag{4.18}$$

où

- $\mathcal{C}_l$ est représenté par le modèle de Markov caché λ_l ;
- $\mathcal{T}_{\mathcal{S}}(g)$ constitue la séquence d´observation à reconnaître.

3.2 Problèmes spécifiques aux modèles de Markov cachés

La diversité des modèles de Markov cachés oblige à effectuer de nombreux choix de modèles. Parmi les choix possibles, nous considérons dans cette section deux architectures de modèles : complet ou «gauche–droite». Il convient aussi de choisir entre modèles continus et modèles discrets, c´est–à–dire la nature des observations. La détermination du nombre optimal d´états est également un problème essentiel lors de la définition d´un modèle de Markov cachés. Dans la suite, nous verrons une méthode automatique pour choisir ce nombre.

3.2.1 Architecture du modèle

Selon la nature de la matrice de transition, il est possible de définir plusieurs architectures de modèles de Markov cachés. Nous considérons deux types d´architectures. La plus générale est l´architecture «*fortement connectée*» ou complète. Chaque état est atteignable depuis tous les autres états en un nombre fini de transitions. Le modèle «*gauche–droite*» est plus adapté lorsque les séquences d´observations ont des propriétés qui changent successivement au cours du temps. Nous proposons des exemples de ces deux architectures dans [Mar00].

3.2.2 Détermination du type de modèle

Deux types principaux de modèles de Markov cachés existent. Un modèle de Markov caché est dit *discret* ou *symbolique* si les observations sont à valeur dans un ensemble fini $\mathcal{V} = (v_1, \ldots, v_N)$. Cet ensemble fini est alors nommé «*vocabulaire*» et les observations sont considérées comme des «*mots*». Dans ce cas, la probabilité d´une observation, étant donné l´état, est donnée par la matrice $B = b_j(k)_{1 \leq j \leq N}$. La probabilité $b_j(k)$ correspond à la probabilité d´observer le symbole v_k, le k^{e} mot du vocabulaire $\mathcal{V}$ dans l´état q_j à l´instant t:

$$b_j(k) = P(O_t = v_k \mid s_t = q_j) \tag{4.19}$$

Si les observations ne peuvent être réduites à un vocabulaire, le modèle est continu et les observations sont à valeurs dans l´ensemble réel. La distribution de probabilité continue $b_j(x)$ d´observations d´un symbole est utilisée à la place de la probabilité $b_j(k)$. La probabilité $b_j(x)dx$ est la probabilité que l´observation soit comprise entre x et $x + dx$. La densité de probabilité peut être généralisée au cas vectoriel et est alors définie par $b_j(\vec{X})$. Celle–ci peut prendre plusieurs formes [Sta95], parmi lesquelles la densité de probabilité gaussienne:

$$b_j(\vec{X}) = \frac{1}{\sqrt[n]{2\pi}\sqrt{|\Lambda_j|}} \exp^{\frac{1}{2}\left(\vec{X}-\vec{\mu}_j\right)^T \Lambda_j^{-1}\left(\vec{X}-\vec{\mu}_j\right)} \tag{4.20}$$

Dans cette équation $\vec{\mu}_j$ et Λ_j sont respectivement la moyenne et la matrice de covariance associées à l´état q_j. Dans la suite, les densités de probabilité d´observations seront considérées gaussiennes.

Le choix des modèles discrets s´impose si les observations sont définies dans un vocabulaire. Dans le cas contraire, il est toujours possible d´opérer une quantification vectorielle ou scalaire des données en utilisant un *dictionnaire* [15]. Cette méthode a été retenue dans un grand nombre de systèmes de reconnaissance de la parole ou de gestes [MYD96, YX94, NWF86]. Nag *et al* [NWF86] justifient l´utilisation des modèles discrets par la facilité d´utilisation. Il n´est en effet pas nécessaire de choisir une loi de probabilité en faisant l´hypothèse sur la distribution des données.

L´avantage des modèles continus est le nombre réduit de paramètres justifié par l´utilisation de densités paramétriques pour les lois conditionnelles. Une conséquence directe de cette réduction de paramètres est le nombre moins important de séquences d´apprentissage nécessaires pour entraîner le modèle de Markov caché [Dur99, MD00]. Un second avantage est l´absence de discrétisation ou de quantification des données. Comme toute opération de discrétisation, celle–ci entraîne une perte d´information et une instabilité aux frontières de zone.

15. *codebook*

3.2.3 Détermination du nombre optimal d´états

Dans beaucoup d´approches ou de systèmes utilisant les modèles de Markov cachés, la détermination du nombre d´états est traitée par des méthodes exhaustives ou heuristiques.

La méthode exhaustive ou *a priori* consiste à entraîner et évaluer les modèles de Markov cachés pour tous les nombres d´états inférieurs à un nombre prédéfini d´états noté $N_{\max}$. Le nombre d´états finalement choisi est celui qui maximise le critère d´optimalité. Cette méthode est extrèmement coûteuse puisque toutes les $N_{\max}^N$ possibilités doivent être couvertes.

La méthode heuristique ou *a posteriori* consiste à déterminer si le problème a une structure intrinsèque où peuvent intervenir les états cachés. Étant cachés, rien ne garantit que les états sélectionnés correspondent réellement aux états calculés. Dans l´algorithme d´entraînement de Baum–Welch, nous n´avons pas la possibilité d´intervenir sur le choix des états en fonction des observations. Cependant, ce choix peut se révéler judicieux dans des cas simples mais complètement inadéquat dans des cas plus complexes. Dans la section 3.3, nous verrons des exemples de choix de méthode heuristique.

Nous avons également proposé [Dur99, MD00] une méthode automatique de sélection du nombre d´états. Elle a été introduite par Biernacki *et al* [BCG98]. Étant donné un ensemble d´entraînement $\mathcal{A}$, la méthode cherche à estimer le couple :

$$(\hat{N}, \hat{\lambda}) = \arg\max N, \lambda f(\mathcal{A} \mid \lambda, N) \tag{4.21}$$

dans cette équation, nous avons :

$$f(\mathcal{A} \mid \lambda, N) = \int_{\Phi_{\lambda,N}} f(\mathcal{A} \mid \lambda, N, \phi)\rho(\phi \mid \lambda, N)d\phi \tag{4.22}$$

avec

- $f(\mathcal{A} \mid \lambda, N, \phi) = \prod_{O \in \mathcal{A}} f(O \mid \lambda, N, \phi)$;
- $\Phi_{\lambda,N}$ est l´espace des paramètres du modèle de Markov λ à N états cachés ;
- $\rho(\phi \mid \lambda, N)$ est la densité de probabilité a priori du paramètre ϕ.

L´équation (4.22) étant inconnue, nous pouvons l´approximer ou approximer son logarithme à l´aide d´un critère. Un exemple de tel critère est le *Critère d´Information Bayesien* [16]. Le critère BIC est défini par :

$$\mathrm{BIC}(\lambda, N) = \log f(\mathcal{A} \mid \lambda, N, \hat{\phi}) - \frac{\nu_{\lambda,N}}{2} \log\left(\mathrm{card}(\mathcal{A})\right) \tag{4.23}$$

Dans l´équation (4.23), $\nu_{\lambda,N}$ est le nombre de paramètres indépendants du modèle λ à N états et $\hat{\phi}$ est l´estimateur de maximum de vraisemblance de ϕ :

16. *Bayesian Infromation Criterion* ou BIC.

$$\hat{\phi} = \arg\max_{\phi} f(\mathcal{A} \mid \lambda, N, \phi) \tag{4.24}$$

L´estimateur $\hat{\phi}$ peut être tiré des formules de BAUM–WELCH[17]. Le critère BIC peut être vu comme la différence entre un terme mesurant l´adéquation entre les données au modèle et d´un terme pénalisant un grand nombre de paramètres indépendants. Dans le cas, d´un modèle de Markov discret, la densité de probabilité $f(\mathcal{A} \mid \lambda, N)$ peut être remplacée par la probabilité $P(\mathcal{A} \mid \lambda, N)$.

3.3 Expérimentations sur *Unistroke*

Dans cette section, nous expérimentons les résultats de la section précédente sur la reconnaissance du geste d´écriture d´un alphabet stylisé. Les trois problèmes considérés précédemment sont expérimentés.

3.3.1 Unistroke

Les expérimentations réalisées dans cette section portent sur la reconnaissance du *geste d´écriture* de lettres. Les lettres sont extraites de l´alphabet d´*Unistroke*, c´est–à–dire de l´agenda électronique PALMPILOT commercialisé par 3COM. Ces lettres ressemblent au dessin des lettres majuscules classiques. Elles ont été simplifiées et stylisées pour permettre une écriture avec un seul trait continu et faciliter leur reconnaissance. Six lettres de cet alphabet ont été considérées : `A`, `E`, `H`, `L`, `O` et `Q`. Leur choix a été motivé par les deux critères suivants :

- choix de similarité entre lettres. Les lettres `H` et `L` présentent la même ambiguïté au début du dessin. Les lettres `O` et `Q` ne diffèrent que de la petite barre supplémentaire du `Q` (cf. figure 4.14).
- choix de lettres complexes. Le dessin des lettres `E`, `O` et `Q` ne se résument pas à de simples segments de droites comme pour les lettres `A` et `L`

Les données de l´expérimentation sont un ensemble de 50 séquences vidéo pour chacune des lettres. La figure 4.15 présente 5 images clés pour des séquences au cours desquelles les lettres `A` et `O` sont dessinées.

Chacune des lettres a été réalisée par la même personne et effectuée, comme pour le PALMPILOT, dans un cadre. Les lettres ont une taille qui varie peu. Elles sont donc normalisées en amplitude. Les dessins ont également été réalisés à vitesse constante. Cette restriction nous permet de faire l´hypothèse que les observations sont effectuées à des intervalles temporels réguliers [Dur99]. Un modèle de Markov est un modèle à temps discret où les observations sont effectuées à chaque top d´horloge. Ne pas respecter cette contrainte revient à supprimer ou ajouter des observations à des instants aléatoires, ce qui

17. Les formules de réestimation de BAUM–WELCH sont données dans [Mar00, RJ86]

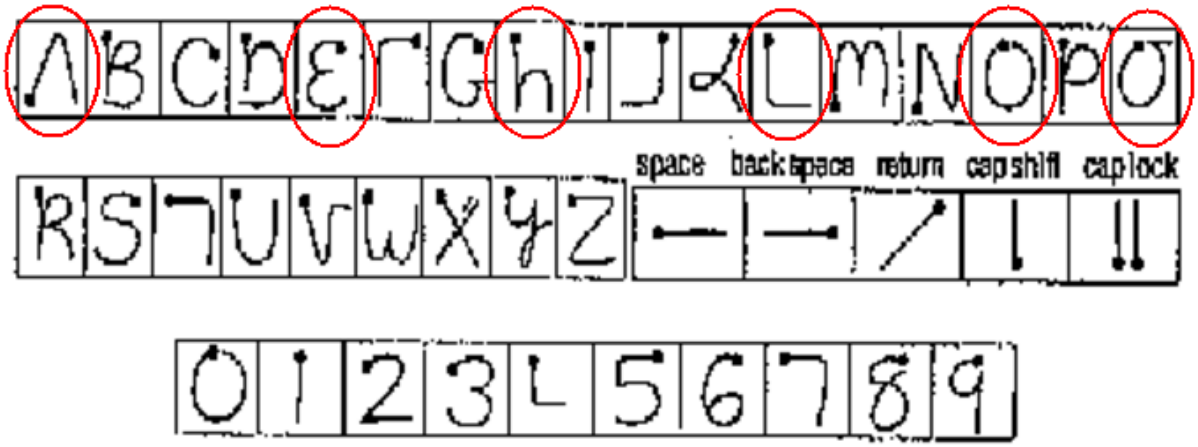


FIG. 4.14 – ***Alphabet Unistroke de l´agenda électronique*** PALMPILOT ***commercialisé par 3***COM. *Les lettres encerclées sont celles considérées dans nos expérimentations. (Source : [3Co])*

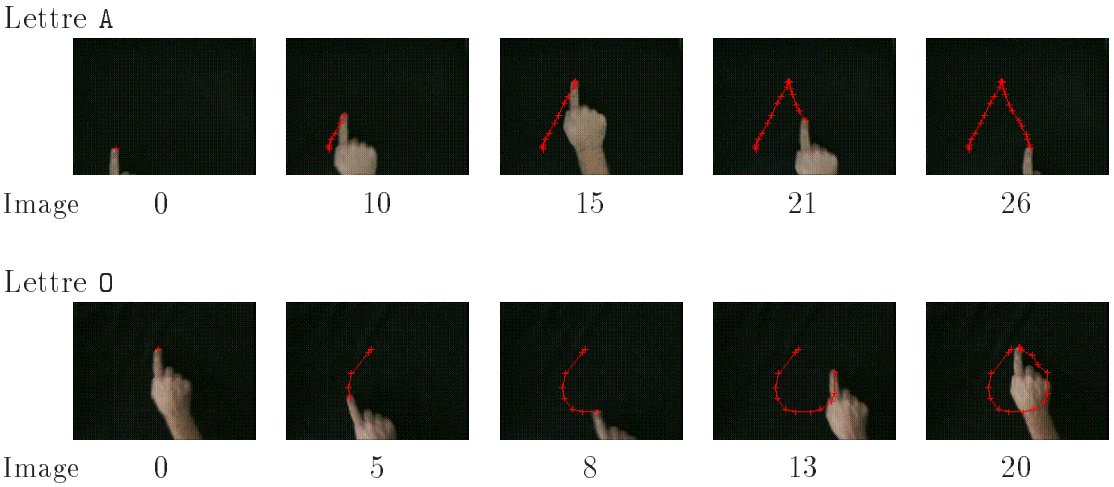


FIG. 4.15 – ***Images de séquences au cours desquelles les lettres A et O sont dessinées.***

invalide l´hypothèse markovienne. Dans les cas où les observations ne sont pas effectuées à intervalles réguliers, il est possible d´interpoler les données.

3.3.2 Extraction des paramètres

À partir des images de la séquence vidéo, la position du doigt est extraite en utilisant le système présenté à la section 2.4. Ainsi, à partir d´une séquence composée de T images, nous obtenons une séquence d´observation $O = (o_1, \ldots, o_T)$ dans laquelle l´observation o_t à l´instant discret t (t correspond à l´indice de l´image dans la séquence vidéo) :

$$\forall t, 1 \leq t \leq T \qquad o_t = \begin{pmatrix} x_t \\ y_t \end{pmatrix} \tag{4.25}$$

Dans cette équation, x_t et y_t sont les coordonnées, dans l´image, de l´extrémité du doigt. De ces coordonnées, plusieurs vecteurs caractéristiques peuvent être extraits. CAMPBELL *et al* [CBA+96] proposent une étude et une comparaison sur des caractéristiques pour la reconnaissance de 18 gestes de *T´ai Chi* en 3 dimensions. L´étude s´intéresse à huit vecteurs différents :

1. positions des mains dans l´image (x, y, z) ;

2. vitesses en coordonnées cartésiennes (dx, dy, dz) ;
3. positions en coordonnées polaires (r, θ, z) ;
4. vitesses en coordonnées polaires avec vitesse angulaire $(dr, d\theta, dz)$;
5. vitesses en coordonnées polaires avec vitesse tangente $(dr, rd\theta, dz)$;
6. vitesse instantannée et courbure locale $(ds, \log(\rho), dz)$ et $(ds, \log(\rho ds), dz)$;
7. vitesses en coordonnées cartésiennes et polaires $(dr, d\theta, dx, dy, dz)$.

Chacun de ces huit vecteurs présente des avantages et des inconvénients comme les présente la table 4.1.

Notes sur le tableau 4.1 (page 128)

1. les coordonnées polaires sont centrées sur le visage ; ainsi, les caractéristiques mélangent la position des mains et du visage ;
2. *DTW* ou *Dynamic Time Warping* est la propriété des modèles de Markov cachés permettant de compenser au sein d´un état de faibles variations temporelles sur les observations. La considération de dérivées de mesures ne permet plus l´utilisation de cette propriété.

L´expérimentation effectuée par Campbell *et al* considère 108 séquences de gestes parmi lesquelles 18 correspondent à une translation et 18 une translation et une rotation. À chaque geste est associé un modèle de Markov caché, comme nous l´avons présenté à la section 3.1.2. Les modèles ont une topologie *gauche–droite* et sont constitués de 5 états. Cependant de par l´architecture des modèles, certains états ont été supprimés lors de l´apprentissage. Les résultats obtenus sont très bons, entre 74% et 98%, alors qu´aucune translation ou rotation n´est effectuée. La valeur de 74% est obtenue avec les caractéristiques de vitesse instantannée et de courbure locale. Lorsque la translation et la rotation sont considérées, l´utilisation des coordonnées cartésiennes ou polaires se montrent, comme nous pouvions nous y attendre, complètement inadéquates. Le meilleur résultat est obtenu pour les caractéristiques $(dr, d\theta, dz)$.

Pour la reconnaissance de gestes de dessin d´*Unistroke*, nous considérons donc les vitesses en coordonnées polaires. De plus, désirant être également invariants dans la vitesse d´exécution du geste, nous considérons uniquement la caractéristiques $d\theta$.

3.3.3 Expérimentations sur les modèles discrets

Pour considérer des modèles de Markov discrets, il convient de définir une transformation permettant de passer des valeurs réelles à un vocabulaire. Nous avons décidé de segmenter le cercle trigonométrique en 8 secteurs égaux. Pour que les frontières entre les

	Vecteurs	Avantages	Inconvénients
1	(x, y, z)	contient le plus de «*contexte*», permet l´utilisation de DTW^2 des modèles de Markov pour compenser les variations de vitesse	sensible aux translations et rotations
2	(dx, dy, dz)	invariant en translation	sensible aux rotations ne permet l´utilisation de DTW^2
3	(r, θ, z)	invariant en translation permet l´utilisation de DTW^2	inhomogène[1]
4	$(dr, d\theta, dz)$	invariant en translation et rotation	inhomogène[1] ne permet pas l´utilisation de DTW^2
5	$(dr, rd\theta, dz)$	invariant en translation et rotation	inhomogène[1] ne permet pas l´utilisation de DTW^2
6	$(ds, \log(\rho), dz)$ $(ds, \log(\rho ds), dz)$	invariant en translation et rotation homogène[1]	contient le moins de contexte ne permet pas l´utilisation de DTW^2
7	$(dr, d\theta, dx, dy, dz)$	idem que 2 et 4 Redondance des caractéristiques	idem que 2 et 4

TAB. 4.1 – ***Comparaison de différents vecteurs caractéristiques pour la reconnaissance de*** 18 ***gestes de T´ai Chi***. *Les notes sont données à la page 127 (d´après [CBA+96])*

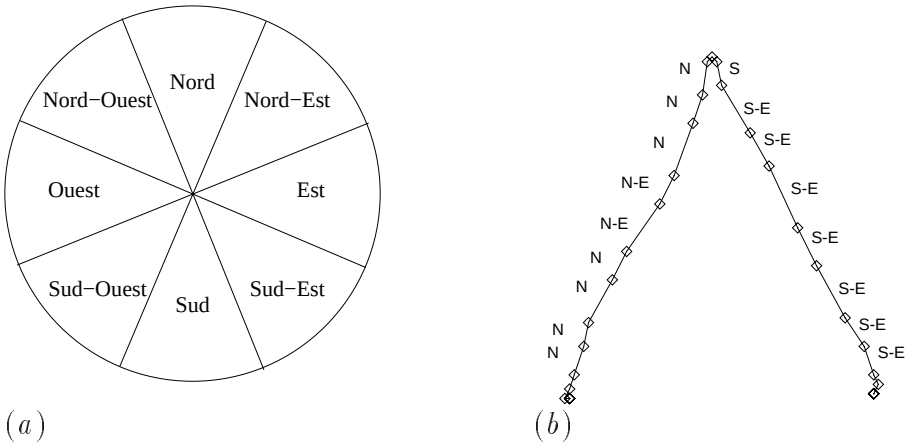


FIG. 4.16 – ***Discrétisation du cercle trigonométrique en*** 8 ***secteurs égaux et exemple de séquence de lettre.*** *(*a*) Les secteurs permettent de transformer les vecteurs d´observations continues en observations symboliques. (*b*) Exemple du dessin de la lettre* `A` *et les symboles associés à chaque segment.*

secteurs ne correspondent pas à des orientations de lettres sélectionnées, nous avons effectué une rotation de $\frac{\pi}{8}$. La figure 4.16 présente les secteurs sur le cercle trigonométrique ainsi que l´exemple de la lettre A.

a) Expérimentations sur l´architecture du modèle

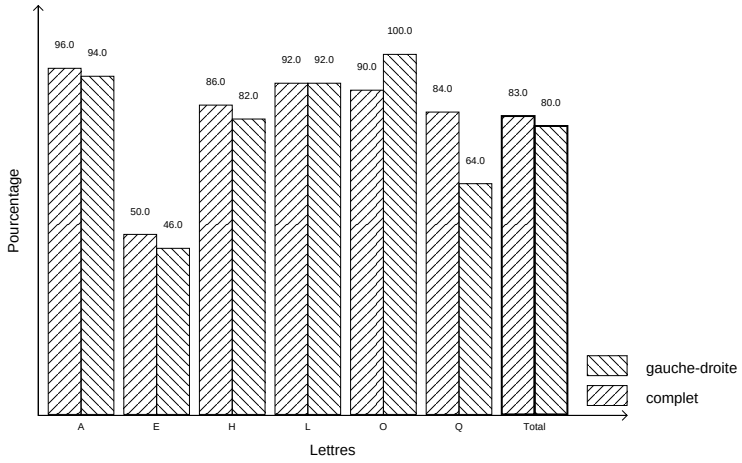


FIG. 4.17 – ***Résultats de reconnaissance selon l´architecture des modèles de Markov cachés discrets.***

Nous avons considéré dans ces expérimentations chacune des deux architectures «gauche–droite» et complète. Elles sont composées de 2 états pour les lettres A et L, 4 pour E, H et O, et 5 pour Q. Il s´agit d´un choix selon une heuristique, présentée à la section 3.2.3. Les modèles ont été entraînés à partir d´une base de 25 séquences d´observations par lettre. Une nouvelle série de 25 séquences est utilisée pour la reconnaissance. Puis, nous avons inversé les deux bases pour avoir une nouvelle expérimentation indépendante de la base d´apprentissage. La figure 4.17 donne les résultats de cette expérimentation. Il apparaît sur ce graphique que l´architecture complète obtient, globalement, de meilleurs résultats. La lettre E n´est pas reconnue dans les deux cas. Cependant, elle est parfaitement reconnue dans la seconde expérimentation. La lettre Q, avec l´architecture «gauche–droite», est peu reconnue, les erreurs correspondent souvent à une confusion avec la lettre O. Cette confusion montre la limitation de ce type d´architecture. Dans la suite, nous étudierons donc les modèles de type complet.

b) Étude du nombre d´états

L´objectif de cette étude est de comparer les méthodes permettant de déterminer le nombre d´états maximisant le taux de reconnaissance.

Méthode directe Le nombre d´états, dans cette méthode, est fixé. Nous avons considéré ici que 5 états semblaient être un bon compromis entre un nombre trop petit d´états impliquant une généralisation trop grande du modèle de Markov et un grand nombre correspondant au contraire à un modèle trop spécialisé.

Méthode exhaustive Cette méthode consiste à essayer toutes les combinaisons d´états. Le nombre d´états essayé est compris entre 1 et 10. Le résulat donne 2 états pour les lettres `A` et `L`, 4 pour les lettres `O` et `Q` et 5 pour les lettres `H` et `E`.

Méthode heuristique Nous faisons ici l´hypothèse que les états sont associés à des caractéristiques particulières du dessin de la lettre. Ceci ne permet pas de trouver le nombre d´états assurant la meilleure reconnaissance. Dans le cas d´*Unistroke*, nous choisissons de faire correspondre un segment de droite ou un arc de cercle par état. Nous prenons donc 2 états pour les lettres `A` et `L`, 4 pour les lettres `E`, `H` et `O` et 5 la lettre `Q`.

Méthode	A	E	H	L	O	Q
Directe	5	5	5	5	5	5
Exhaustive	2	5	5	2	4	4
Heuristique	2	4	4	2	4	5
Automatique	2	2	2	2	2	3

TAB. 4.2 – ***Résumé du nombre d´états sélectionnés pour chaque lettre selon les différentes méthodes pour une architecture complète***

Méthode automatique Le *critère d´information bayésien* BIC est utilisé pour sélectionner automatiquement le nombre d´états. Le calcul du critère BIC est effectué pour chaque lettre et le nombre d´états maximisant le critère BIC est sélectionné. La figure 4.18 montre, pour chacune des lettres, la valeur du critère BIC en fonction du nombre d´états. Nous choisissons donc 2 états pour les lettres `A`, `L`, `E` et `O` et 3 états pour les lettres `H` et `Q`. Cependant, le choix pour les lettres `H` et `Q` peut être 2 compte–tenu de la proximité des résultats du critère BIC pour 2 et 3 états. La table 4.2 rappelle le nombre d´états

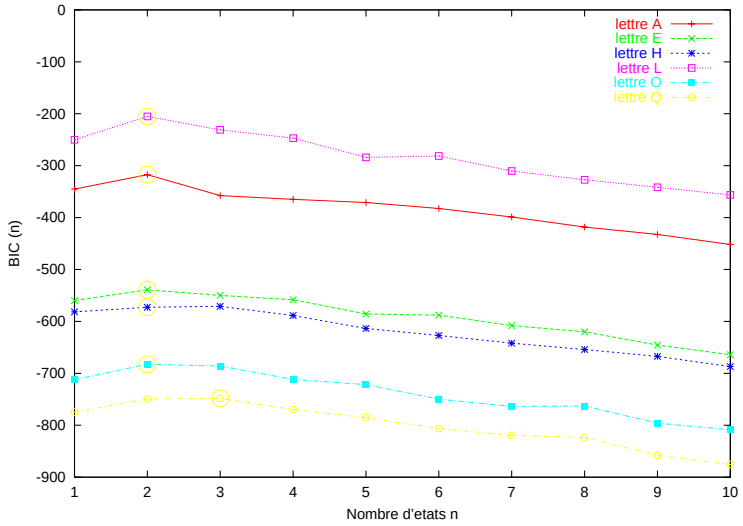


FIG. 4.18 – ***Critère d´information bayésien pour chaque lettre considérée.*** *Le maximum du critère* BIC *pour chaque lettre est cerclé.*

sélectionnés pour chacune des méthode. La coïncidence du nombre d´états pour les deux lettres «simples» A et L est intéressante à relever. Elle suggère que dans certains cas simples comme ceux–ci, on puisse trouver intuitivement le nombre d´états.

Méthode	A	E	H	L	O	Q	globale
Directe	96%	50%	88%	92%	90%	94%	85%
Exhaustive	92%	100%	100%	80%	100%	92%	94%
Heuristique	96%	50%	86%	92%	90%	84%	83%
Automatique	96%	100%	100%	92%	92%	94%	96%

TAB. 4.3 – ***Résultats de reconnaissance selon les méthodes de sélection du nombre d´états pour les modèles discrets et une architecture complète.***

La table 4.3 donne, pour chaque méthode et pour chaque lettre, le nombre de confusions. La méthode exhaustive donne les meilleurs résultats. Cependant cette méthode est lourde en temps de calcul. La méthode automatique donne un bon compromis entre le coût de calcul pour déterminer le nombre d´états et le résultat de la reconnaissance. Un autre avantage est de pouvoir sélectionner le modèle ayant le moins de paramètre.

c) Étude de la robustesse de la reconnaissance

Lors du choix des paramètres à la section 3.3.2, nous avons choisi une représentation permettant d´éliminer les variations dues aux changements d´amplitudes. Dans cette section, nous vérifions cette hypothèse en testant les modèles de Markov cachés avec des séquences de tests dans lesquelles l´amplitude et la vitesse des tracés sont exagérement[18] modifiées. Nous avons entraîné les modèles avec les données de la section précédente. Les ensembles de séquences de test ont été réalisés par une seconde personne. Puis, dans un second ensemble, nous augmentons et diminuons l´amplitude du tracé de 25% et faisons varier la vitesse de 25%. Les taux de reconnaissance varient selon la lettre, mais nous pouvons énoncer quelques résultats globaux :

- changer la personne qui effectue le tracé n´altère pas le taux de reconnaissance. Nous observons même une améliortion de 3% (de 94% à 97%) du résultat, ce que nous attribuons bien entendu au hasard ;

18. Nous considérons ici des modifications qui ne sont pas seulement dues à la répétition d´un geste avec l´incertidute impliquée.

- changer l´amplitude du tracé diminue légèrement le taux de reconnaissance qui passe à 90%. Cette diminution est attribuée à la très mauvaise reconnaissance de la lettre `E` ;
- changer la vitesse du tracé diminue plus sensiblement le taux de reconnaissance qui passe à 78%. La raison de cette diminution s´explique par la transgression de l´hypothèse markovienne effectuée dans cette section. En effet, changer la vitesse du tracé revient à supposer que la vitesse d´échantillonage des trajectoires n´est pas constante.

Nous avons constaté que certaines lettres sont plus affectées que d´autres pas ces changements. Il semble que les lettres les moins sensibles soient les lettres composées de segments de droite tels que le `A` et `L`. Au contraire, la lettre `E`, composée de plusieurs arcs de cercle, est moins reconnue dans les conditions de variation. Ce fait s´explique par la méthode de représentation choisie, amplifiant les incertudes sur les quartiers d´angles lorsque la résolution change.

d) Interprétation des états cachés dans le cas d´une méthode de sélection heuristique

Dans la section précédente, nous avons proposé une méthode heuristique pour associer un état à un dessin de la lettre. Cependant, cette méthode n´est justifiée que si nous sommes capables *a posteriori* de vérifier que cette association est valide [Dur99]. Cette vérification est possible en utilisant l´algorithme de VITERBI[19]. Celui–ci permet de trouver la séquence d´états la plus probable associée à une séquence d´observations. Dans le cas des lettres `A` et `L`, le modèle possède deux états cachés. Les matrices de probabilité de transitions $A_{\mathtt{A}}$, de l´état initial $\Pi_{\mathtt{A}}$ et des observations $B_{\mathtt{A}}$ pour la lettre `A` sont les suivantes :

$$A_{\mathtt{A}} = \begin{bmatrix} 0.72 & 0.28 \\ 0.28 & 0.72 \end{bmatrix}$$

$$\Pi_{\mathtt{A}} = \begin{bmatrix} 0.96 \\ 0.04 \end{bmatrix}$$

$$B_{\mathtt{A}} = \begin{array}{c} \begin{array}{cccccccc} SO & S & SE & E & NE & N & NO & O \end{array} \\ \begin{bmatrix} 0.00 & 0.12 & 0.07 & 0.00 & 0.12 & 0.46 & 0.00 & 0.21 \\ 0.00 & 0.02 & 0.75 & 0.04 & 0.05 & 0.08 & 0.00 & 0.05 \end{bmatrix} \end{array}$$

Le modèle défini par ces matrices génère, avec une forte probabilité, des séquences d´états formées de l´état 1 puis l´état 2. De plus, La matrice des probabilités d´une

19. cf. une description de l´algorithme dans [Mar00]

observation sachant l´état nous montre que l´état 1 est associée au symbole `N` et l´état 2 au symbole `SE`. L´étude d´un exemple de lettre particulier permet de vérifier ceci. La séquence des symboles et des états sont données respectivement par $\mathcal{S}_s$ et $\mathcal{S}_e$:

$\mathcal{S}_s =$	O	N	N	N	NE	NE	N	O	S	S	SE	SE	SE	SE	SE	SE	SE	SE	S	N	O
$\mathcal{S}_e =$	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1

Pour cet exemple, le passage de l´état 1 à l´état 2 correspond effectivement au changement de direction du tracé. Le passage de l´état 2 à l´état 1, à la fin de la séquence, correspond à la fin du tracé de la lettre au cours duquel le doigt se déplace légèrement dans des directions quelconques.

Il est également possible de déterminer la correspondance des états pour la lettre `L`. Les séquences ci–dessous représentent les matrices de probabilité de transitions $A_{\mathtt{L}}$, de l´état initial $\Pi_{\mathtt{L}}$ et des observations $B_{\mathtt{L}}$:

$$A_{\mathtt{L}} = \begin{bmatrix} 0.80 & 0.20 \\ 0.23 & 0.77 \end{bmatrix}$$

$$\Pi_{\mathtt{L}} = \begin{bmatrix} 0.92 \\ 0.07 \end{bmatrix}$$

$$B_{\mathtt{L}} = \begin{array}{c} \begin{array}{cccccccc} SO & S & SE & E & NE & N & NO & O \end{array} \\ \begin{bmatrix} 0.00 & 0.73 & 0.00 & 0.04 & 0.00 & 0.00 & 0.08 & 0.13 \\ 0.00 & 0.10 & 0.02 & 0.80 & 0.00 & 0.00 & 0.01 & 0.06 \end{bmatrix} \end{array}$$

Le modèle de la lettre `L` génère également des séquences d´états formées de l´état 1 puis l´état 2. Les symboles associés aux états sont `S` pour l´état 1 et `E` pour l´état 2. L´étude d´un exemple confirme cette constatation :

$\mathcal{S}_s =$	NO	S	S	S	S	S	S	S	S	S	S	S	S	SO	O	E	E	E	E	E	E	O
$\mathcal{S}_e =$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2

Dans cet exemple, le passage de l´état 1 à l´état 2 correspond au passage au dessin de la seconde barre de la lettre. L´étude des autres lettres est plus difficile, sans doute parce que le nombre d´étapes intervenant dans le tracé des lettres est moins évident. Nous pouvons cependant noter :

- pour la lettre `H`, l´état 1 est associé à la barre verticale et l´état 2 au pont de la lettre ;
- pour la lettre `O`, l´état 1 correspond à la partie supérieure de la lettre tandis que l´état 2 à la partie basse ;

– pour la lettre Q, les changements d´états correspondent plus ou moins au changement de courbure de la lettre. Cependant, contrairement à notre hypothèse, la barre finale n´est pas représentée par un état particulier.

3.3.4 Expérimentations sur les modèles continus

Dans ces expérimentations, la caractéristique $d\theta$ est directement utilisée. De plus, ne disposant pas d´information *a priori* sur la loi de probabilité des observations, notre choix se porte sur les lois gaussiennes. Ces lois permettent de modéliser de nombreux problèmes. L´expérimentation réalisée a consisté, comme précédemment, à utiliser 25 séquences pour l´apprentissage et 25 pour tester. Nous obtenons des résultats très mauvais, le taux de reconnaissance global est inférieur à 20%. Ceci s´explique de deux manières. La première est que la caractéristique $d\theta$ n´est pas une fonction continue. Le second point est le nombre de données disponibles insuffisant. Il ne permet pas une convergence correcte des modèles. Un modèle continu composé de 5 états nécessite l´estimation de 10 paramètres (pour chaque état, nous avons la moyenne et la variance à estimer). Nous convenons que pour l´estimation d´un paramètre, nous avons besoin de 15 à 20 exemples. Il faut donc entre 150 et 200 séquences d´apprentissage pour estimer correctement le modèle de Markov caché.

Les positions (x, y) du doigt étant naturellement normalisées, il est possible de les utiliser directement comme entrée des modèles de Markov cachés [Dur99].Les données d´apprentissage sont les mêmes que celles utilisées avec les modèles discrets. La section **??** a montré que le critère BIC permet une bonne estimation du nombre d´états. Nous définissons donc les modèles à 7 états pour les lettres A, L, O et Q, 8 pour la lettre H et 10 pour la lettre E. Le nombre d´état nécessaire est donc plus élevé que pour les modèles discrets. Le choix des paramètres explique cette augmentation, ils sont moins adaptés que les angles pour représenter différents tracés des lettres. Les taux de reconnaissance obtenus sont de 100% pour les lettres du premier ensemble de test. Ils sont de 0% pour l´ensemble de test dans lequel de fortes variations d´amplitude ou de vitesse du tracé sont présentes. La vitesse du tracé influence la reconnaissance de la même manière que dans le cas discret. Elle met en défaut l´hypothèse markovienne. Les variations en position et en amplitude des tracés entraînent un écart par rapport à la moyenne de la loi gaussienne.

3.4 Conclusion

Les résultats présentés dans cette section mettent en évidence l´importance du choix des paramètres des modèles de Markov cachés. Nous avons étudié trois types de paramètres. Nous avons, dans un premier temps, considéré deux architectures de modèle : les modèles complets et les modèles «gauche–droite». L´architecture d´un modèle de Markov caché est déterminée par la matrice de transitions d´états. Dans le cas du modèle «gauche–droite», cette matrice est triangulaire supérieure et n´autorise que des change-

ments croissants d´états. Toutes les transitions sont autorisées pour le modèle complet. Nous avons expérimenté la reconnaissance des lettres d´*Unistroke* avec les même séquences de tracés. Cette expérience montre la supériorité du modèle complet.

Le second point étudié concerne la détermination du nombre d´états cachés de manière à maximiser le taux de reconnaissance global. L´utilisation du critère BIC donne une bonne estimation du nombre d´états nécessaire. Il s´agit d´une méthode rapide donnant de bons résultats. La méthode heuristique, consistant à choisir les états en fonction du problème et des séquences d´observations, fonctionne correctement dans les cas simples. Nous avons montré expérimentalement que cette méthode peut être employée pour les lettres composées de segments de droite, le choix des états a été confirmé par l´utilisation de l´algorithme VITERBI. Cependant, cette méthode ne fonctionne pas pour les cas plus complexes. La méthode énumérative garanti de trouver la meilleure solution mais est extrêmement lourde à mettre en place.

Enfin, nous nous sommes intéressés à la nature des observations. Deux types sont à considérer : les modèles discrets et les modèles continus. Le choix de paramètres discrets engendre une perte d´information pouvant nuire à la reconnaissance. L´utilisation des modèles continus est donc plus adéquate. Le choix d´un paramètre continu étant invariable avec les changements de translations et de rotations est difficile à déterminer. Dans un premier temps, nous avons choisi d´utiliser la vitesse en coordonnées polaires $d\theta$. Cette mesure s´est avérée complètement inadaptée. Nous pensons que la nature de la fonction non continu est une source de confusion lors de la classification. De plus, le nombre de séquences d´apprentissage semble insuffisant. Nous avons montré, dans une seconde expérience, la supériorité des modèles continus. Dans cette expérience, dans laquelle les observations sont directement les positions (x, y) du doigt, les taux de reconnaissance atteignent 100%. Cependant, étant donné la nature des observations, l´ajout de variations d´amplitude des gestes rend cette reconnaissance impossible.

Dans la section suivante, nous présentons une seconde approche pour la reconnaissance des trajectoires ou séquences d´observations. Dans cette approche, les observations sont continues et nous nous appuyons sur une technique statistique de reconnaissance par histogrammes multidimmensionnels.

4 Reconnaissance statistique de trajectoires

4.1 Introduction

4.1.1 Représentation de densité de probabilité par histogrammes multidimensionnels

SCHIELE [Sch97, SC98] proposait et démontrait, dans son étude doctorale, une nouvelle technique pour la reconnaissance d´objets à partir de l´union de statistiques de vecteurs de caractéristiques locales. Ces vecteurs étaient obtenus par la projection de

l´image sur un ensemble d´opérateurs travaillant sur un voisinage d´images tels que les dérivées gaussiennes ou les filtres de Gabor. Dans son approche, il représentait l´union des statistiques par des histogrammes multi–dimensionnels. Ses travaux ont montré que les histogrammes multi–dimensionnels proposaient un moyen sûr et précis pour la reconnaissance d´un grand nombre d´objets à partir d´images prises sous différents aspects. Il obtenait des résultats compris entre 90% et 100% de reconnaissance sur la base de Columbia [NNM96] composé de 72 objets sous 100 vues différentes.

Considérant un ensemble de mesures M, le changement d´apparence d´un objet o est modélisé par la densité de probabilité sur les mesures M :

$$p(M \mid o_n, C) \tag{4.26}$$

où o_n est l´objet, C décrit les changements de l´apparence des objets. Ceux–ci incluent les rotations et translations de l´image et de l´objet, les changements de scène (occultations partielles et changement de fond) et les conditions d´enregistrement (modifications d´éclairage, flou, bruits de signal, erreurs de discrétisation). L´ensemble de mesures M est l´ensemble de caractéristiques locales m_k défini par :

$$M = \bigcup_k m_k \tag{4.27}$$

Pour représenter la densité de probabilité d´un objet, SCHIELE propose l´utilisation d´histogrammes multi–dimensionnels. Leur avantage est de bien représenter l´ensemble d´apprentissage. Cette propriété est intéressante puisqu´elle permet de conserver toute l´information et, en particulier l´information de discriminalité. La réduction du nombre de cellules de l´histogramme ou l´augmentation de la taille de l´ensemble d´apprentissage permet la généralisation de la reconnaissance à des objets non présents dans l´ensemble d´apprentissage. SCHIELE [Sch97, page 52] souligne que cette généralisation est possible lorsque le nombre d´exemples est du même ordre que le nombre de cellules de l´histogramme.

Ainsi, la fonction de densité probabiliste d´un objet sous un ensemble de mesure M est représentée par plusieurs histogrammes multi–dimensionnels. L´histogramme pour des conditions particulières C_p est donné par

$$H(M \mid o_n, C_p) \tag{4.28}$$

Dans le contexte de la reconnaissance d´objets, il s´agit de mesurer la probabilité $p(o_n \mid m_k)$ d´un objet o_n à partir d´un vecteur de mesures locales m_k. La probabilité $p(o_n \mid m_k)$ peut être calculée à partir de la règle de BAYES :

$$p(o_n \mid m_k) = \frac{p(m_k \mid o_n)p(o_n)}{p(m_k)} = \frac{p(m_k \mid o_n)p(o_n)}{\sum_i p(m_k \mid o_i)p(o_i)} \tag{4.29}$$

avec
- $p(o_n)$, la probabilité *a priori* de l´objet o_n ;
- $p(m_k)$, la probabilité *a priori* du vecteur de mesures locales m_k ;
- $p(m_k \mid o_n)$, la densité probabiliste de l´objet o estimée à partir des histogrammes multidimensionnels.

Dans la plupart des cas, un seul vecteur de mesures n´est pas suffisant pour la reconnaissance d´objets. En utilisant K vecteurs indépendants, c´est–à–dire $M = \bigcup_{k=1}^{K} m_k$, la probabilité de l´objet est calculée par :

$$p(o_n \mid M) = p(o_n \mid \bigwedge_k m_k) \tag{4.30}$$

$$= \frac{p(\bigwedge_k m_k \mid o_n)p(o_n)}{\sum_i p(\bigwedge_k m_k \mid o_i)p(o_i)} \tag{4.31}$$

$$= \frac{\prod_k p(m_k \mid o_n)p(o_n)}{\sum_i \prod_k p(m_k \mid o_i)p(o_i)} \tag{4.32}$$

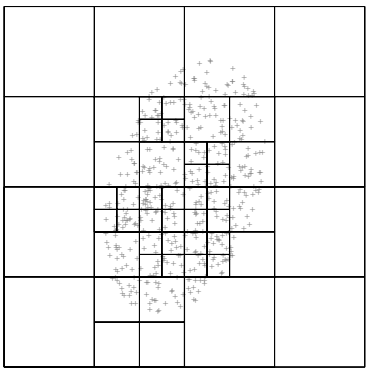


FIG. 4.19 – ***Représentation d´un histogramme bi–dimenseionnel sous forme de quad–tree***. *Histogramme simulé à partir du tirage aléatoire de* 500 *points selon une probabilité gaussienne. Une zone est redécoupée si elle contient plus de* 50 *points.*

CHOMAT [CC99, Cho00] étend cette approche à la reconnaissance d´activités. Les

mesures locales sont calculées sur des séquences d´images. Il s´agit de réponses de la projection de voisinages spatio–temporelles sur des bases de champs réceptifs. Chaque activité est caractérisée par un histogramme à 12 dimensions. La nature des mesures locales est telle que les histogrammes multidimensionnels présentent de nombreuses zones creuses, c´est–à–dire des zones dans lesquelles peu de données sont disponibles. Pour minimiser le coût mémoire des histogrammes, CHOMAT propose une représentation par «*quad–tree*». La finesse du découpage est relative à la densité des données dans les zones. La figure 4.19 présente un découpage en quad–tree de données générées par une loi gaussienne. Le technique du quad–tree est de découper une zone en quatre lorsque celle–ci contient un nombre de données supérieur à un seuil prédéfini. Il s´agit d´un algorithme récursif de découpage. L´exemple proposé par la figure 4.19 découpe une zone lorsque celle–ci contient plus de 50 points.

4.1.2 Algorithme probabiliste de reconnaissance

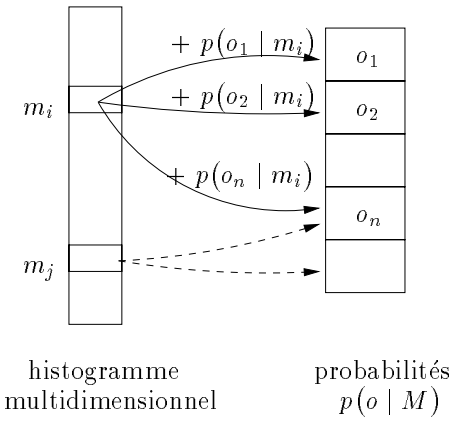


FIG. 4.20 – ***Structure d´un algorithme probabiliste de reconnaissance.*** *La mesure locale m_i est utilisée pour le calcul de la probabilité $p(o_n \mid m_i)$ pour tous les objets o_n. (d´après [Sch97])*

SCHIELE propose un algorithme probabiliste de reconnaissance. Cet algorithme utilise une mesure locale m_i pour le calcul de la probabilité $p(o_n \mid m_i)$ pour tous les objets o_n. L´algorithme répète le calcul d´indices m_k jusqu´à l´obtention par un des objets d´une probabilité suffisante. La figure 4.20 présente graphiquement cet algorithme.

Cette section étend l´approche statistique de SCHIELE pour la reconnaissance statistique de trajectoires de gestes. Les mesures locales correspondent à des descripteurs locaux sur une fenêtre ou voisinage temporel. La fonction de densité de probabilité de ces descripteurs est représentée par un histogramme multidimensionnel. L´approche s´articule autour de deux étapes : une étape d´apprentissage et une étape de reconnaissance.

4.2 Signature d´une fenêtre temporelle

Considérant la trajectoire d´un geste définie par l´équation (4.1) :

$$\mathcal{T}_{\mathcal{S}}(g) = \{\vec{m}_{t_1}, \ldots, \vec{m}_{t_{T_g}}\}$$

la fenêtre temporelle de taille s définie à l´instant t_i est le vecteur de caractéristiques $\vec{m}_{t_i}$ et son historique récent. Cette fenêtre temporelle est notée :

$$\vec{w}_{t_i} = (\vec{m}_{t_{i-s+1}}, \ldots, \vec{m}_{t_i}) \tag{4.33}$$

Dans cette équation $\vec{m}_{t_x}$ est le vecteur de mesure à l´instant t_x et s la taille de la fenêtre temporelle sélectionnée. La valeur de s est un paramètre important de notre approche et doit être «petite» afin de considérer la séquence comme étant locale. Elle définit la période de temps pendant laquelle un vecteur de paramètres est considéré suffisamment récent pour contribuer à la séquence. Dans la suite, nous utiliserons des valeurs comprises entre 5 et 20.

La collection de toutes les fenêtres temporelles de toutes les trajectoires d´une base d´apprentissage permet le calcul d´un sous–espace propre en utilisant une analyse en composantes principales. Cet espace, noté $\mathcal{M}$, est réduit aux m vecteurs propres les plus dominants.

La projection d´une fenêtre temporelle w_t sur l´espace $\mathcal{M}$ est la *signature de la fenêtre*. Elle est notée γ_t. La projection des fenêtres temporelles de la trajectoire $\mathcal{T}_{\mathcal{S}}$ définit une nouvelle trajectoire : la *signature du geste*. Elle est notée

$$\mathcal{T}_{\mathcal{M}} = (\gamma_{t_s}, \ldots \gamma_{t_{T_g}}) \tag{4.34}$$

4.3 Densité de probabilité des signatures de fenêtre temporelle

La signature de la fenêtre temporelle à l´instant t est définie par γ_t, la fonction de densité de probabilité du geste g_t^n, à l´instant t et considérant cette signature, est :

$$p(g_t^n \mid \gamma_t, t) \tag{4.35}$$

La règle de BAYES, nous permet d´obtenir :

$$p(g_t^n \mid \gamma_t, t) = \frac{p(\gamma_t \mid g_t^n, t)p(g_t^n \mid t)}{p(\gamma_t, t)} = \frac{p(\gamma_t \mid g_t^n, t)p(g_t^n \mid t)}{\sum_i p(\gamma_t \mid g_t^i, t)p(g_t^i \mid t)} \tag{4.36}$$

avec

- $p(g_t^i \mid t)$ est la probabilité *a priori* du geste g_i à l´instant t ;
- $p(\gamma_t \mid g_t^i, t))$ est la probabilité *a priori* de la signature γ_t à l´instant t, sachant que le geste est g_i ;
- $p(\gamma_t, t)$ est la probabilité de la signature à l´instant t.

Si, nous considérons, pour simplifier, les signatures et les gestes indépendants du temps, nous obtenons :

$$\begin{aligned} p(g_t^i \mid t) &= p(g^i) \\ p(\gamma_t \mid g^i, t) &= p(\gamma_t \mid g^i) \end{aligned} \tag{4.37}$$

Ainsi, l´équation (4.36) s´écrit :

$$p(g_t^n \mid \gamma_t, t) = \frac{p(\gamma_t \mid g^n)p(g)}{\sum_i p(\gamma_t \mid g^i)p(g^i)} \tag{4.38}$$

Sans connaissance *a priori* de la probabilité des gestes, il est classique de considérer chaque classe équiprobable :

$$p(g^n) = \frac{1}{N} \tag{4.39}$$

Il est également possible de considérer que les probabilités sont proportionnelles au nombre d´échantillons de l´ensemble d´apprentissage :

$$p(g^n) = \frac{N_n}{\sum_{i=1}^{N} N_i} \tag{4.40}$$

où N_i dénombre le nombre de signatures pour la classe du geste g^i.

Une analyse plus fine du problème permet l´estimation de la probabilité. Dans le cadre de la reconnaissance du dessin des lettres d´*Unistroke* et la considération de la langue française, la probabilité de la lettre `A` est plus grande que celle de la lettre `Q`. Par exemple, dans le premier chapitre du «*Petit Prince*» (cf. annexe C), le texte est composé de $5,9\%$ de lettres `A` et $0,6\%$ de `Q`.

4.4 Représentation de la fonction de densité probabiliste par histogrammes multidimensionnels

La fonction de densité de probabilité $p(\gamma_t \mid g^n)$ peut être représentée par un histogramme multidimensionnel. Un histogramme est construit en divisant l´espace $\mathcal{M}$ contenant les signatures de tous les gestes en cellules équidistantes. Le nombre d´axes

de l´espace $\mathcal{M}$ détermine le nombre de dimensions de l´histogramme. La résolution de l´histogramme, c´est–à–dire le nombre de cellules par dimension, peut être réduite afin de généraliser la reconnaissance, au prix cependant d´une perte de discriminalité. Ainsi, elle est un paramètre important de notre approche.

L´utilisation de trajectoires discrètes avec une période d´échantillonage donnée implique que les points entre ces échantillons ne sont pas capturés par l´histogramme. Plus le nombre de cellules est important et plus le nombre de données est nécessaire pour remplir l´histogramme. Ceci est particulièrement crucial avec la règle de BAYES puisqu´une cellule vide donne une probabilité nulle. Afin de réduire le nombre de trajectoires nécessaire au remplissage de l´histogramme, celui–ci est lissé par un filtre gaussien.

La probabilité de la signature γ_t, sachant le geste g^n est définie par :

$$p(\gamma_t \mid g^n) = \frac{1}{N^n}\, h^n(\gamma_t) \tag{4.41}$$

où N^n est le nombre d´exemples du geste g^n et h^n est l´histogramme associé au geste. L´incorporation des équations (4.41) et (4.40) dans la règle de BAYES (4.38) donne la définition de la probabilité du geste g^n, sachant la signature γ_t, en fonction des histogrammes :

$$p(g^n \mid \gamma_t) = \frac{h^n(\gamma_t)}{\sum_i h^i(\gamma_t)} \tag{4.42}$$

4.5 Expérimentations

Nous considérons ici deux expérimentations. La première consiste en la reconnaissance d´*Unistroke*. En second, la technique est appliquée à la reconnaissance d´expressions du visage.

4.5.1 Unistroke

Les expérimentations suivantes concernent la reconnaissance d´*Unistroke*. Les données utilisées sont présentées dans la section 3.3.1. Nous considérons dans cette expérimentation, les trois paramètres relatifs à notre approche :

- la taille s de la fenêtre temporelle ;
- le nombre d de dimensions de l´histogramme, c´est–à–dire le nombre de dimensions conservées de l´espace propre $\mathcal{M}$;
- le nombre c de cellules par dimension dans l´histogramme.

La table 4.4 donne les résultats de reconnaissance pour différentes valeurs pour les trois paramètres de notre approche. Comme nous pouvions nous y attendre, la comparaison des résultats montrent que le taux de reconnaissance augmente avec ces paramètres. La taille de l´histogramme apporte peu de changement dans le taux de reconnaissance. Ainsi, il est préférable de choisir le nombre de dimensions égal à 10, permettant de réduire l´espace de

s	
5	77.2%
10	81.5%
15	82.9%

(*a*)

d	
2	74.8%
3	80.7%
4	84.6%

(*b*)

c	
10	79.5%
15	80.5%
20	80.0%

(*c*)

TAB. 4.4 – ***Résultats de reconnaissance sur Unistroke pour différentes valeurs de paramètres.*** *(*a*) Taux de reconnaissance selon la valeur de la taille de la fenêtre. (*b*) Taux de reconnaissance selon le nombre de dimensions de l´histogramme. (*c*) Taux de reconnaissance selon le nombre de cellules par dimension de l´histogramme.*

mémoire occupé. Une meilleure reconnaissance est obtenue avec une fenêtre temporelle de taille 15. Cependant, cette taille empêche la reconnaissance de séquences d´une longueur inférieure à 15. De plus, l´augmentation de cette taille limite la localité de la fenêtre temporelle.

La figure 4.21 présente les résultats pour chaque geste en utilisant les paramètres ayant, précédemment, donné les meilleurs résultats. La lettre `O` obtient le meilleur résultat, les courbes spécifiques de la lettre ne se retrouvent pas ailleurs. Seule la lettre `H` dispose de courbes équivalentes, elle représente 100% des erreurs de classification de la lettre `O`. La lettre `H` est la lettre la moins reconnue, elle présente en effet de grandes similitudes avec la lettre `L` et la lettre `O`. La similitude entre la lettre `L` et la lettre `H` est le début des deux lettres, commençant par une barre verticale. Cette similude est à l´origine de 63% des erreurs. La similitude entre la lettre `H` et la lettre `O` est la courbure, 22% des erreurs correspondent à cette confusion.

4.5.2 Expérimentation sur la reconnaissance d´expression de visages

Une seconde expérience a été réalisée en collaboration avec HALL [Hal98, MHC98, MHC99]. Cinq expressions de visages sont considérées : «*colère*», «*interrogation*», «*dégoût*», «*joie*» et «*surprise*». Les expressions sont données sous la forme de séquences d´images de passage d´un état neutre vers l´expression correspondante. Elles sont représentées dans la figure 4.22.

Après normalisation des images de visages en taille, position et orientation, un sous–espace propre de visages est construit à partir d´une analyse en composantes principales. Cette opération est décrite à la section 3.1. L´espace construit est utilisé comme espace de caractéristiques et un visage exprimant une émotion y est représenté par une trajectoire.

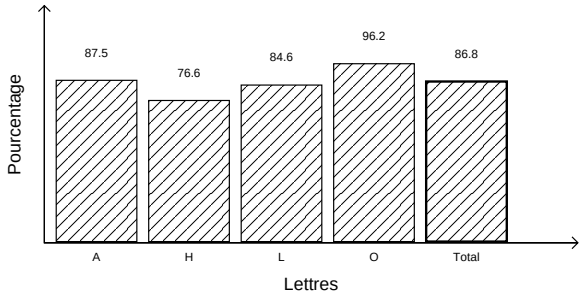


FIG. 4.21 – ***Reconnaissance par lettre selon les lettres.*** *Utilisation des paramètres optimaux : la taille de la fenêtre temporelle est* 15*, le nombre de dimensions de l´histogramme est* 4 *et le nombre de cellules par dimension de l´histogramme est* 20*.*

(*a*) (*b*) (*c*) (*d*) (*e*)

FIG. 4.22 – ***Images des expressions de visages considérées.*** *Cinq expressions sont considérées :* (a) *«colère»,* (b) *«interrogation»,* (c) *«dégoût»,* (d) *«joie»* et (e) *«surprise»*

Les vecteurs propres associés aux plus grandes valeurs propres correspondent aux changements les plus importants. Les petits mouvements tel que lever les sourcils sont associés à des vecteurs propres correspondant à des valeurs plus petites. Ainsi, parmi l´ensemble des vecteurs propres définissant l´espace, seuls ceux permettant la distinction des visages sont conservés. Dans cette expérience, nous avons conservé quatre vecteurs : les 2^{e}, 7^{e}, 8^{e} et 11^{e}. Nous avons créé et rempli un histogramme à quatre dimensions et contenant 16 ou 24 cellules par dimension. L´histogramme est alors lissé par un filtre gaussien de taille 3.

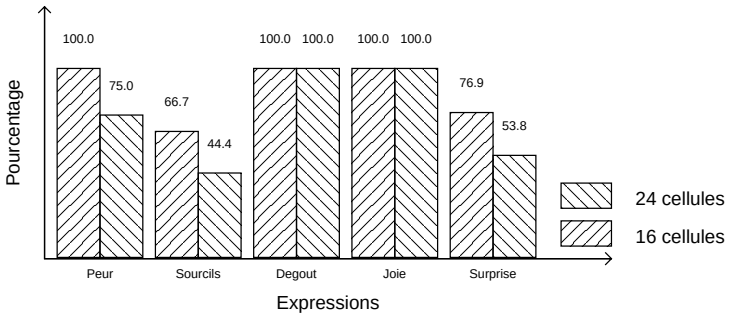


FIG. 4.23 – ***Résultats de reconnaissance des expressions du visage***. *Deux expérimentations sont effectuées : le nombre de cellules par dimension de l´histogramme est* 16 *ou* 24.

La figure 4.23 montre les résultats de reconnaissance avec les deux nombres de cellules. Le premier bilan de cette expérience est la reconnaissance plus importante en utilisant l´histogramme composé de 16 cellules par dimension. Ceci permet de réduire le coût mémoire. Le nombre d´images de la base d´apprentissage est trop petit pour remplir suffisamment bien l´histogramme comportant 24 cellules par dimension. Ainsi, de nombreuses probabilités sont nulles même après le lissage de l´histogramme. La reconnaissance des expressions «*colère*», «*dégoût*» et «*joie*» est totale, toutes les expressions ont été correctement reconnues. Les résultats sont moins bons pour les expressions «*interrogation*» et «*surprise*» car elles sont plus difficilement distinguables l´une de l´autre. Pour les autres expressions, tout le visage subit une transformation, notamment par l´apparition de rides et le changement de la forme des yeux et de la bouche. Dans le cas des expressions «*interrogation*» et «*surprise*», le changement des yeux est identique, aucune ride particulière n´apparaît, seule la bouche change (cf. figure 4.22). La projection de ces deux

expressions dans l´espace propre sont proches, impliquant parfois qu´elles correspondent à une même cellule de l´histogramme.

Une source d´erreur supplémentaire dans ces expériences est l´instabilité de la normalisation. Le module de localisation du visage utilisé produit un écart de quelques pixels. Une telle translation peut impliquer un changement important lors de la projection dans l´espace propre. Les trajectoires de deux exemples d´une même expression sont alors différentes et rendent difficile leur correspondance par mesure de la probabilité. HALL a démontré [Hal98] cet effet en effectuant manuellement la normalisation des images de visages. Le résultat donne des trajectoires d´expression plus proches dans l´espace propre. Le taux de reconnaissance est alors plus élevé.

4.6 Reconnaissance globale d´une trajectoire

Nous avons proposé, jusqu´à présent, une reconnaissance locale de la trajectoire d´un geste ou d´une expression de visage. Nous avons calculé la probabilité $p(g^n \mid \gamma_t)$. Il convient d´étendre ce calcul à la probabilité de

$$p(g^n \mid \mathcal{T}_\mathcal{M}) \quad = \quad p\Big(g^n \mid (\gamma_{t_s}, \ldots, \gamma_{t_T})\Big) \tag{4.43}$$

où $\mathcal{T}_\mathcal{M}$ est la signature de l´expression de visage dans l´espace $\mathcal{M}$ des expressions.
Si nous considérons l´indépendance des signatures de fenêtres locales :

$$\mathcal{T}_\mathcal{M} = \bigcup_{t=s}^{T} \gamma_t \tag{4.44}$$

Ceci revient à calculer :

$$p(g^n \mid \mathcal{T}_\mathcal{M}) \quad = \quad p(g^n \mid \bigcup_{t=s}^{T} \gamma_t) \tag{4.45}$$

$$= \quad p(g^n \mid \gamma_{t_s} \wedge \ldots \wedge \gamma_{t_T}) \tag{4.46}$$

Soit, en appliquant l´équation (4.32) :

$$p(g^n \mid \mathcal{T}_\mathcal{M}) \quad = \quad \frac{\prod_t p(\gamma_t \mid g^n) p(g^n)}{\sum_i \prod_t p(\gamma_t \mid g^i) p(g^i)} \tag{4.47}$$

Il est alors possible d´appliquer la reconnaissance probabiliste proposée par SCHIELE. La mesure de la signature de la fenêtre temporelle γ_t est utilisée pour calculer la probabilité $p(g^n \mid \gamma_t)$ et incrémenter la probabilité du geste g^n. L´algorithme est répété jusqu´à l´instant T de fin du geste ou bien jusqu´à l´obtention d´une probabilité suffisante pour l´un des gestes.

4.7 Conclusion

Nous avons proposé une nouvelle méthode de reconnaissance de trajectoires. La construction d´un sous–espace propre est basée sur les fenêtres temporelles contenant un paramètre et son historique. Une fenêtre temporelle est alors représentée par sa signature dans un tel espace. La statistique conjointe, représentée par un histogramme multidimensionnel de signatures, définit une méthode simple et puissante pour la reconnaissance locale de gestes. De bon résultats ont été obtenus pour les deux applications considérées : reconnaissance des gestes de dessins d´*Unistroke* et la reconnaissance d´expressions de visages. La projection des fenêtres temporelles dans un espace propre local fournit une représentation avec d´intéressantes propriétés :

1. Invariance au changement d´amplitude en normalisant l´énergie de projection ;
2. invariance en position, en supprimant le vecteur propre avec la plus grande valeur propre du sous–espace propre ;
3. robustesse au changement dans la vitesse d´exécution du geste par la création d´un espace propre avec des gestes effectués à différentes vitesses.

Algorithme 4.2 Algorithme d´apprentissage de la reconnaissance statistique de trajectoires

1. Extraction des fenêtres temporelles de taille s $\vec{w}_{t_i}$ pour toutes les trajectoires $T_{\mathcal{S}}$ des gestes de la base d´apprentissage. Les mesures de la trajectoire sont effectuées dans l´espace de configurations $\mathcal{S}$.
2. Calcul, par analyse en composantes principales sur l´ensemble de toutes les séquences de mesures locales $\vec{w}_{t_i}$, de l´espace $\mathcal{M}$.
3. Calcul des *signatures de mesures locales* γ_{t_i} correpondant à la projection de $\vec{w}_{t_i}$ sur l´espace $\mathcal{M}$.
4. Calcul des histogrammes h_i, pour toutes les classes de gestes considérées, et de l´histogramme h, défini par la somme de l´histogramme de toutes les classes.

Algorithme 4.3 Algorithme de la reconnaissance statistique de trajectoires

faire
  Calcul de la signature γ_t par projection de la fenêtre temporelle w_t dans l´espace $\mathcal{M}$;
  pour tout geste g^i **faire**
    Calcul de la probabilité $p(g^i \mid \gamma_t)$ à partir de l´équation (4.42) ;
    Mette à jour la probabilité $p(g^i \mid \mathcal{T}_{\mathcal{M}})$
  fin pour
jusqu'à $\exists i, p(g^i \mid \mathcal{T}_{\mathcal{M}}) > s$ ou

La reconnaissance globale du geste est effectuée par l´algorithme probabiliste de reconnaissance proposé par SCHIELE. Il fait l´hypothèse de l´indépendance des signatures de fenêtres locales. Cependant, cette hypothèse est une simplification, elle permet de facilement effectuer la reconnaissance. Il convient de calculer récursivement la probabilité $p(g^n \mid \mathcal{T}_t)$ en fonction de celle de $p(g^n \mid \mathcal{T}_{t-1})$. Les algorithmes 4.2 et 4.3 récapitulent les étapes d´apprentissage et de reconnaissance.

5 Synthèse du chapitre

Dans ce chapitre, nous avons considéré la classification des gestes dynamiques. Un geste est représenté par une trajectoire dans un espace de caractéristiques. Ces caractéristiques ont été définies au chapitre 2. Trois méthodes de reconnaissance ont été proposées : reconnaissance par automates d´états finis, par modèles de Markov cachés et par statistiques de trajectoires.

La technique de reconnaissance par automates d´états finis s´appuie sur une classification des caractéristiques telle qu´elle a été présentée au chapitre 3. Un geste est alors représenté par un automate d´états finis dans lequel les états représentent des caractéristiques particulières de la main. Cette méthode présente l´avantage d´être simple à mettre en place pour des gestes simples. De plus, la combinaison d´automates permet la reconnaissance de gestes plus complexes. Deux méthodes permettent d´améliorer cette technique : pondérer les transitions par des probabilités et construire automatiquent les automates à partir de base d´exemples. Ceci aboutit naturellement à la mise en place de chaînes de Markov cachées.

Les modèles de Markov cachés permettent la classification automatique de séquences d´observation dans laquelle chaque classe est représentée par un modèle. Ils sont entraînés à partir de bases d´apprentissage des gestes qu´ils doivent reconnaître. Nous nous sommes intéressés à trois choix critiques lors de la construction d´un modèle de Markov caché. L´étude a été effectuée sur la reconnaissance du geste de dessin de lettres du langage *Unistroke*.

Le premier choix concerne l´architecture du modèle. Deux architectures simples sont étudiées : l´architecture complète et celle gauche–droite. Les expérimentations réalisées semblent montrer que l´architecture complète est plus adaptée pour le problème étudié. Cette constatation est particulièrement vraie pour la lettre `Q`, beaucoup mieux reconnue avec l´architecture complète.

Le second choix considéré est le choix du nombre d´états. Aux classiques choix par méthode exhaustive et par méthode heuristique, nous proposons une méthode automatique basée sur le calcul du critère d´information bayésien, BIC. Ce critère cherche à maximiser l´adéquation des données et du modèle mais en pénalisant les modèles avec beaucoup de paramètres indépendants. Les expérimentations réalisées montrent que cette méthode

est un bon compromis entre le coût de calcul pour déterminer le nombre d´états et le résultat de reconnaissance. Il est intéressant de noter que, pour des cas simples, toutes les méthodes donnent le même nombre d´états.

Enfin, nous nous sommes intéressés à la nature des séquences d´observations : discrètes ou continues. Jusqu´à présent, nous avons considéré le cas discret, puisque, comme le souligne NAG *et al* [NWF86], ils sont plus faciles à utiliser. Ils ont expérimentalement montrés de bons résultats de reconnaissance. Cependant, la discrétisation de données continues en données discrètes implique toujours une perte d´information. L´utilisation de modèles continus semble donc plus adéquate. Nous avons montré au cours d´expérimentations réalisées sur les gestes d´*Unistroke* que la reconnaissance est meilleure en utilisant les positions (x, y) du doigt dans l´image dans le cas de données faiblement bruitées. Dans le cas de changement d´amplitude ou de vitesse, le taux de reconnaissance chûte. Les observations sont en effet dépendantes de la taille.

Nous avons enfin proposé une méthode originale de reconnaissance statistique de trajectoires. Cette méthode s´appuie sur les fenêtres temporelles de la trajectoire. Chaque fenêtre est utilisée pour construire un sous–espace propre dans lequel une fenêtre est transformée en signature du mouvement. Dans cet espace, la densité de probabilité des signatures est codée par un histogramme multidimensionnel, permettant la reconnaissance locale des signatures. Les expérimentations réalisées sur deux problèmes différents montrent l´utilisation générale de cette approche. La reconnaissance globale, c´est–à–dire pour toute la trajectoire et toutes les signatures extraites, est réalisée à l´aide d´un algorithme statistique. Cet algorithme fait l´hypothèse de l´indépendance des signatures. Cette hypothèse simplificatrice doit être étendue pour considérer la corrélation entre deux signatures successives. Cet algorithme n´a pas encore été testé. Il devra donc montrer son utilisation dans le cas de signatures d´un geste. De plus, la nature des observations nous fait penser aux modèles de Markov cachés. Il est en effet envisageable de considérer que les signatures des fenêtres temporelles soient utilisées comme observations pour un système de reconnaissance à base de modèles de Markov cachés.

Dans les deux chapitres suivants, nous proposons l´utilisation de la classification de gestes dynamiques pour la reconnaissance d´activités humaines et pour la reconnaissance de gestes dans un environnement intelligent et intéractif.

Le petit Nicolas en thèse [Pet]

La découverte (3/4)

«Moi, je trouve ça plutôt normal d´être content. D´ailleurs, quand mon patron trouve un théorême, il est super fier et ses copains (qui sont aussi des gens très, très forts) sont super contents de lui. Mais ça mes parents, ils ne le savent pas.»

Chapitre 

Application à la reconnaissance d´activités

Dans ce chapitre, nous présentons une application de l´étude des modèles de Markov cachés à la reconnaissance d´activités. L´approche proposée est la combinaison d´un capteur d´éléments d´activités et la reconnaissance par modèles de Markov cachés. Le capteur probabiliste, développé par Chomat *[CMC00, Cho00] utilise une description spatio–temporelle du mouvement. Le résultat de ce capteur est une carte de probabilités pour chaque classe d´activité considérée. Une règle de décision permet la transformation de ces cartes en un symbole correspondant à l´élément d´activité reconnu. Ce symbole sert d´entrée aux modèles de Markov cachés discrets. Chaque activité est représentée par un modèle, la reconnaissance d´une activité correspond à celui ayant donné la plus grande probabilité. Les résultats présentés dans ce chapitre sont préliminaires mais sont cependant encourageants.*

1 Introduction

La reconnaissance visuelle d´activités humaines présente de nombreuses applications dans le domaine de la surveillance visuelle, interaction homme–machine et la communication inter-personnelle. Elle offre également un moyen de communication avec un environnement interactif tel que celui présenté au chapitre 6. Considérant plusieurs classes d´activité humaines, l´ordinateur doit être capable de réagir à certaines de ces activités.

Dans ce chapitre, nous considérons la reconnaissance de neuf activités dans un bureau.

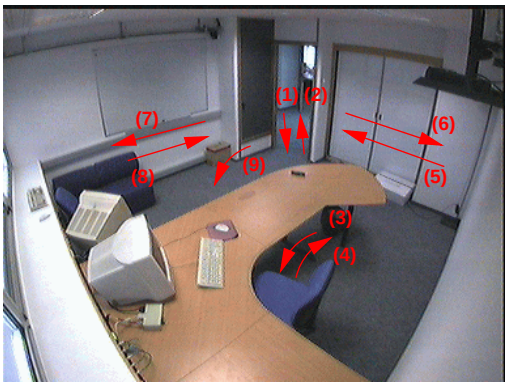


FIG. 5.1 – ***Activités considérées dans le bureau*** *Neuf activités sont considérées : (1) «entrer», (2) «sortir», (3) «s´asseoir», (4) «se lever», (5) «gauche1», (6) «droite1», (7) «gauche2», (8) «droite2» et (9) «tomber».*

Elles sont présentées dans la figure 5.1. Une activité est décrite par l´apparence de petits voisinages spatio–temporels sur une séquence. Une approche statistique est utilisée pour la reconnaîssance de motifs de mouvement alors que la reconnaîssance de l´activité complète utilise un modèle de Markov caché.

2 Capteur d´éléments d´activités par champs réceptifs spatio–temporels

L´apparence d´un objet est la composition de l´ensemble des images de l´objet lorsqu´il est observé sous tous les points de vue, sous différentes conditions d´illumination et de déformation. ADELSON et BERGEN [AB91] définissent l´espace d´apparence d´un objet dans une scène donnée comme une fonction à 7 dimensions locales :

$$I(x, y, \lambda, t, V_x, V_y, V_z) \tag{5.1}$$

Ces dimensions sont la position (x, y) de l´objet dans l´image, l´instant t, la longueur d´onde λ et la position du point de vue (V_x, V_y, V_z). Cette fonction porte le nom de «*fonction plénoptique*» du latin «*plenus*», signifiant «plein» et «*opticus*», «voir». L´apparence d´une scène peut être représentée comme un échantillonnage de la fonction plénoptique.

2.1 Champs réceptifs sensibles à l´énergie du mouvement

La fonction plénoptique donne un cadre puissant pour mesurer des structures locales spécifiques, comme par exemple les motifs spatio–temporels et, plus particulièrement, les motifs d´activités. Ces motifs sont caractérisés par la description de l´information locale visuelle en utilisant un ensemble de champs réceptifs spatio–temporels et en modélisant la réponse des descripteurs. Le résultat est un capteur capable de discriminer des motifs d´activités.

Considérons la fonction plénoptique $I(x, y, t)$ réduite aux seuls niveaux de gris et à un point de vue fixe. La description de I à l´aide de champs réceptifs spatio–temporels permet son analyse. Soit l´intensité lumineuse $I(x, y, t)$ au point (x, y) et à un instant t et sa transformée de Fourier $\hat{I}(u, v, z)$. Le mouvement horizontal r_x et vertical r_y d´une image en mouvement change sa transformée de Fourier :

$$I(x - r_x t, y - r_y t, t) \rightarrow \hat{I}(u, v, z + r_x u + r_y b) \tag{5.2}$$

Cette équation montre que les fréquences spatiales sont inchangées, mais toutes les fréquences temporelles sont translatées par le produit de la vitesse et des fréquences spatiales. Les champs réceptifs sensibles à l´énergie du mouvement sont définis en tenant compte du fait que, à une fréquence spatio–temporelle donnée, une mesure d´énergie dépend à la fois de la vitesse et du contraste du signal d´entrée. Ils sont définis de manière à échantillonner le spectre d´énergie d´une texture en mouvement. La structure des champs réceptifs utilisés fait référence au modèle d´énergie spatio–temporelle d´Adelson et Bergen [AB85], et de Heeger [Hee88]. Des mesures de l´énergie du mouvement sont calculées en sommant le carré des réponses de filtres de Gabor pairs et impairs de même bande passante et de même orientation, permettant une mesure indépendante de la phase du signal. Cette mesure est la suivante :

$$H(x, y, t) = \Big(I(x, y, t) * G_{\text{pair}}\Big)^2 + \Big(I(x, y, t) * G_{\text{impair}}\Big)^2 \tag{5.3}$$

Adelson et Bergen [AB85] proposent de combiner en opposition les réponses des capteurs de mouvements positifs et négatifs. La figure 5.2 présente un exemple de filtre d´énergie spatio–temporelle appliqué à un signal 2D. La sortie de tels filtres dépend à la fois de la vitesse et du contenu spatial du signal d´entrée $I(x, y, t)$. L´extraction d´une information de mouvement au sein d´une bande spectrale implique de normaliser l´énergie des réponses des filtres spatio–temporels par la réponse de filtres spatiaux de même orientation spatiale mais sans composante temporelle :

$$w(x, y, t) = \frac{H_{\text{Gauche}}(x, y, t) - H_{\text{Droite}}(x, y, t)}{H_{\text{statique}}(x, y, t)} \tag{5.4}$$

Un champ réceptif sensible à l´énergie du mouvement est défini par six filtres divisés en trois paires. Une paire est sensible aux mouvements positifs, une aux mouvements

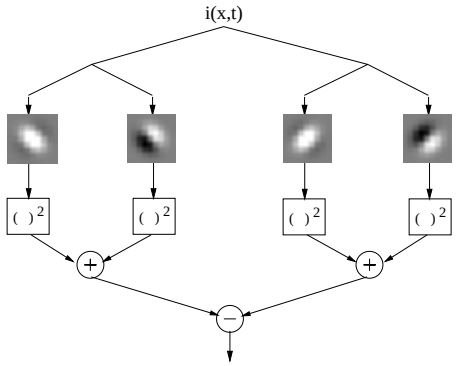


FIG. 5.2 – ***Exemple de filtre d´énergie spatio–temporelle appliqué à un signal 2D*** *(extrait de [CC99, Cho00])*

négatifs et la troisième au contenu spatial. Les filtres de chaque paire ont la même bande passante spatio–temporelle.

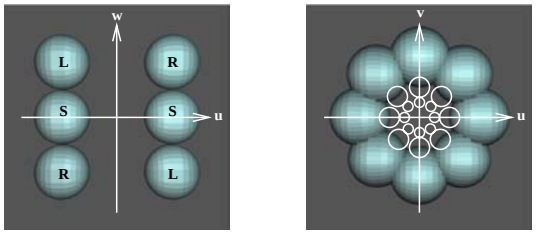
Un banc de 12 champs réceptifs est utilisé. Les filtres correspondent à 4 orientations spatiales et 3 gammes de vitesses. La figure 5.3 représente une carte des bandes passantes des champs réceptifs utilisés. Ce banc permet la description de l´apparence de mouvement.

2.2 Analyse probabiliste des caractéristiques spatio–temporelles

Soit le vecteur $\vec{w}(x,y,t)$ de taille 12 est la réponse du banc de filtres pour l´intensité lumineuse $I(x,y,t)$ où w_i est la réponse du i^{e} filtre. La sortie de l´ensemble des champs réceptifs fournit un vecteur de mesures, $\vec{w}(i,j,t)$ en chaque point (i,j,t). La statistique conjointe de ces vecteurs permet la perception statistique des activités. La probabilité $p\Big(a_k \mid \vec{w}(i,j,t)\Big)$, que le pixel à la position (i,j,t) appartienne à la classe d´activité a_k sachant que le vecteur de mesure $\vec{w}(i,j,t)$, est calculée à partir de la règle de BAYES :

$$p\Big(a_k \mid \vec{w}(i,j,t)\Big) = \frac{p\Big(\vec{w}(i,j,t) \mid a_k\Big)p(a_k)}{p\Big(\vec{w}(i,j,t)\Big)} = \frac{p\Big(\vec{w}(i,j,t) \mid a_k\Big)p(a_k)}{\sum_l p\Big(\vec{w}(i,j,t) \mid a_l\Big)p(a_l)} \tag{5.5}$$

Dans cette équation, $p(a_x)$ est la probabilité *a priori* de l´action a_x, $p(\vec{w} \mid a_x)$ est la probabilité du vecteur de mesure $\vec{w}$ sachant a_x et $p(\vec{w})$ est la probabilité du vecteur de mesure $\vec{w}$. La probabilité $p(a_x)$ de l´action a_x est estimée à partir du contexte. Sans connaissance *a priori*, celle–ci est considérée équi–probable :



(a) (b)

FIG. 5.3 – ***Banc de 12 champs réceptifs de l´énergie de mouvement*** (a) *champ réceptif sensible à l´énergie du mouvement défini par six filtres divisés en trois paires.* (b) *banc de* 12 *champs réceptifs. Les filtres correspondent à* 4 *orientations spatiales et* 3 *gammes de vitesses. (extrait de [Cho00])*

$$p(a_x) = \frac{1}{N} \qquad \forall a_x \tag{5.6}$$

où N est le nombre d´activités à reconnaître.

Pour chacune des classes d´activité a_k, un histogramme multi–dimensionnel des vecteurs de mesures est calculé. Il estime la probabilité $p\left(\vec{w} \mid a_k\right)$ pour la classe d´action a_k. L´espace des champs réceptifs présente un grand nombre de dimensions, 12 dans le cas qui nous intéresse. Afin de représenter cet histogramme, une extension de la technique de quad–tree a été proposée [Cho00, CC99].

La probabilité $p(a_k \mid \vec{w})$ permet seulement de prendre une décision locale en chaque point (x, y, t) de la séquence. Le résultat à un instant donné (t) est une carte de probabilité. Chaque pixel donne la probabilité d´appartenance à une activité de la base d´apprentissage. La figure 5.4 présente un exemple de cartes de probabilité pour un élément d´activité. L´image en haut à gauche est l´image d´origine. L´image encadrée correspond à l´élément d´activité reconnu selon une règle de décision (cf. 3.1).

3 Reconnaissance d´activités par Modèles de Markov cachés

La sortie du capteur probabiliste est une décomposition d´une activité en carte de probabilité d´éléments d´activité. Une activité complète peut être reconnue en utilisant

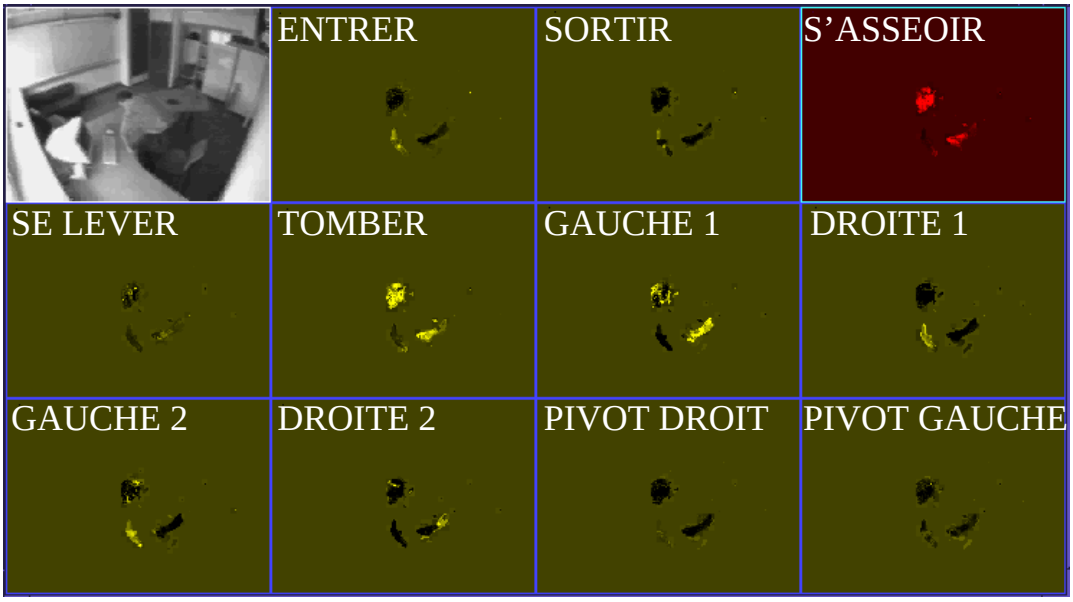


FIG. 5.4 – ***Exemple de cartes de probabilité d´un élément d´activité.*** *L´image d´origine se trouve dans la cellule en haut à gauche. Les autres cellules sont les cartes de probabilités pour les* 11 *éléments d´activité considérés. Les pixels clairs correspondent à de fortes probabilités et les foncés à de faibles probabilités. L´élément d´activité reconnu, ici «s´asseoir», est encadrée. (extrait de [CMC00] et [Cho00])*

les modèles de Markov cachés.

Les modèles de Markov cachés ont été présentés au chapitre 4. Ils sont composés de plusieurs paramètres : le type du modèle, continu ou discret, les séquences d´observations, l´architecture du modèle, complet ou gauche–droite, et le nombre d´états. Nous étudions ces paramètres dans les sections suivantes.

3.1 Type du modèle

Deux types de modèle de Markov cachés peuvent être considérés. Dans un modèle continu, les séquences d´observations sont à valeur dans un ensemble réel. Dans un modèle de Markov caché discret, les observations appartiennent à un vocabulaire. Par souci de facilité, nous considérons, dans un premier temps, les modèles discrets. Il convient donc d´opérer à une quantification des cartes de probabilités. Le vocabulaire utilisé contient les différentes classes d´éléments d´activité du capteur probabiliste. Celles–ci sont sélectionnées par une règle de décision. La classe d´activité ayant le plus grand nombre de fortes probabilités est sélectionnée. Dans la figure 5.4, l´élément d´activité reconnu par cette règle est encadré : il s´agit de l´élément d´activité «*s´asseoir*».

Modèles de Markov cachés ou chaînes simples de Markov ? Dans une chaîne de Markov, les états correspondent directement à l´observation. Dans le cas de la reconnaissance d´activités, le modèle est plus complexe. Une activité est composée de plusieurs éléments d´activités différents sans ordre prédéfini. Les modèles de Markov cachés permettent de représenter correctement ces compositions.

3.2 Architecture des modèles

Activités	*s´asseoir*	*se lever*	*droite1*	*gauche1*	*droite2*	*gauche2*	**Total**
Complet	100%	100%	83%	100%	100%	60%	85%
Gauche–droite	91%	94%	83%	83%	95%	85%	90%

TAB. 5.1 – ***Résultats de reconnaissance d´activités selon deux architectures de modèles de Markov cachés***

La nature des activités et la sortie du capteur probabiliste montrent une tendance à trouver une succession de différents éléments d´activité dans une activité complexe. Cette tendance nous oriente vers une architecture «gauche–droite». Cependant, afin de vérifier

cette hypothèse, nous avons entraîné des modèles «gauche–droite» et «complet» avec un nombre de 5 états choisis arbitrairement et avons essayé de reconnaître les séquences d´entrainement. La table 5.1 donne les résultats de cette comparaison et montre que l´architecture gauche–droite est effectivement plus adéquate.

3.3 Détermination du nombre d´états

Le nombre d´états peut être déterminé selon trois méthodes différentes. La méthode exhaustive ou *a priori* consiste à entraîner les modèles de Markov cachés pour toutes les combinaisons possibles, et à conserver la combinaison donnant le plus fort taux de reconnaissance. Dans la méthode heuristique ou *a posteriori*, la connaissance du problème et l´étude des séquences d´observation permettent la détermination d´un nombre d´états. La méthode automatique, que nous avons proposée au chapitre 4, s´appuie sur le critère d´information bayésien ou BIC. Ce critère mesure l´adéquation entre les données et le modèle en pénalisant les modèles ayant un grand nombre d´états.

Activités	*s´asseoir*	*se lever*	*droite1*	*gauche1*	*droite2*	*gauche2*
Heuristique	2	2	2	2	2	2
Automatique	1 ou 2	2	1 ou 2	1 ou 2	1 ou 2	1 ou 2

TAB. 5.2 – ***Nombre d´états estimés par les méthodes heuristique et automatique pour la reconnaissance d´activités***

La table 5.1 montre le nombre d´états pour chacune des activités et pour les méthodes heuristique et automatique. Les résultats pour la méthode automatique sont proposés dans le graphique 5.5. La méhode exhaustive, lourde en calculs, n´est pas considérée. Cette table montre l´adéquation entre l´estimateur BIC et la méthode heuristique. Dans la suite, le nombre d´états est donc 2 pour toutes les classes d´activités.

4 Résultats expérimentaux

Nous présentons, dans cette section, la reconnaissance d´activités dans un bureau. La caméra est fixe et observe tout le bureau : il n´y a donc pas de mouvement à compenser et l´utilisateur peut se déplacer partout. Les changements d´illumination de la scène ne sont pas contrôlés. Les images ont une taille de 192 par 144 pixels et l´acquisition est effectuée à 10 Hz.

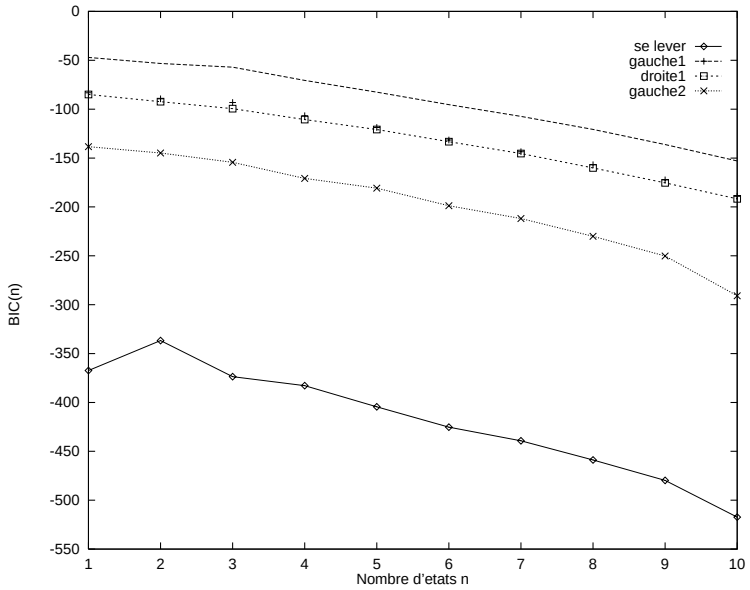


FIG. 5.5 – ***Critère d´information bayésien pour chaque activité considérée.*** *Seules les activités* «s´asseoir», «gauche1», «droite1» *et* «gauche2» *sont représentées. La base d´entraînement des autres activités sont trop petites pour l´estimation du critère* BIC.

Dans les expériences suivantes, nous considérons trois bases d´activités. La première permet la définition des histogrammes ; la seconde est utilisée pour tester la reconnaissance des éléments d´activités et l´apprentissage des modèles de Markov cachés. Enfin, la troisième base sert à tester la reconnaissance complète.

4.1 Reconnaissance d´éléments d´activités

	entrer	**sortir**	**s´asseoir**	**se lever**	**tomber**	**gauche1**	**droite1**	**gauche2**	**droite2**	**total**
entrer	**148**	22	8	4	0	43	39	85	8	423
sortir	28	**48**	3	25	0	71	107	8	27	394
s´asseoir	9	2	**306**	29	2	30	46	32	13	474
se lever	23	11	22	**520**	2	66	62	11	41	759
tomber	3	0	88	4	**14**	0	8	24	0	143
gauche1	1	1	1	1	0	**244**	9	53	60	370
droite1	1	0	5	7	0	0	**291**	16	0	320
gauche2	16	1	120	0	1	8	31	**336**	0	513
droite2	0	11	1	103	0	40	9	0	**618**	785

TAB. 5.3 – ***Matrice de confusion de reconnaissance d´activités pour la séquence de test.***

La table 5.3 résume les taux de reconnaissance pour chacune des classes d´éléments d´activités. Les éléments d´activités «*entrer*», «*sortir*» et «*tomber*» sont mal reconnus. Deux raisons expliquent que certains éléments ne peuvent être discriminés. La première est la vitesse d´acquisition de 10 Hz. Cette vitesse est trop petite pour saisir l´information de mouvement. La seconde est due à la règle de décision appliquée. Celle–ci est trop simple pour tenir compte de la complexité temporelle de chaque classe.

4.2 Reconnaissance d´activités

Les modèles de Markov cachés sont entraînés à l´aide de 130 séquences divisées dans les six classes d´activités. Cette base est trop petite pour estimer de manière efficace les paramètres des modèles. Elle permet néammoins de montrer des résultats préliminaires et estimer la faisabilité d´une telle reconnaissance. Les modèles de Markov cachés considérés

sont des modèles gauche–droite à deux états. Le nombre de paramètres à estimer est 16 : 2 pour la matrice de transition, 2 pour le vecteur de probabilités initiales et 6 pour chacun des deux vecteurs de probabilités des observations. En considérant entre 10 et 20 exemples par paramètre, nous avons besoin d´un ensemble d´apprentissage comportant entre 160 et 320 exemples.

Classes d´activités	*s´asseoir*	*se lever*	*gauche1*	*droite1*	*gauche2*	*droite2*	***Total***
Nombre de sequences	23	34	12	12	22	27	130

TAB. 5.4 – ***Nombre de séquences pour chaque classe d´activité.***

La table 5.4 donne le nombre de séquences pour chacune des classes d´activités considérées. L´expérimentation réalisée a utilisé une validation croisée. Parmi les 130 séquences, une a été extraite pour la reconnaissance, les 129 restantes sont utilisées pour l´apprentissage des six modèles de Markov cachés.

Classes d´activités	*s´asseoir*	*se lever*	*gauche1*	*droite1*	*gauche2*	*droite2*	***Total***
Taux de reconnaissance (%)	91%	88%	83%	92%	82%	85%	87%
Nombre d´activités non classifiées	2	4	2	1	4	5	18

TAB. 5.5 – ***Taux de reconnaissance pour les six activités considérées.***

La table 5.5 donne les résultats de reconnaissance. Nous avons obtenu un taux de reconnaissance de 87%, correspondant à 18 activités qui ont été mal classifiée. La mauvaise classification de ces activités est principalement due au manque de données d´apprentissage.

5 Conclusion

La reconnaissance d´activités humaines a de nombreuses applications dans la surveillance vidéo. Dans ce chapitre, une nouvelle approche pour la reconnaissance d´activité est proposée. Cette approche est la combinaison d´un capteur d´éléments d´activités et une reconnaissance par modèle de Markov cachés.

La sortie du capteur probabiliste est la décomposition temporelle d´activités en classes d´éléments d´activités. La fenêtre temporelle de cette description est relativement petite par rapport à la durée de l´activité. Des modèles de Markov cachés sont utilisés pour effectuer la reconnaissance des activités à partir des éléments d´activités. Les résultats présentés dans ce chapitre sont préliminaires et souffrent du manque de données nécessaire pour l´apprentissage efficace des modèles de Markov cachés. De plus nombreuses expérimentations sont à faire. L´utilisation faite dans ce chapitre des modèles de Markov cachés présentent deux problèmes.

Le premier problème concerne la cohérence des données. Dans le système proposé, le capteur probabiliste effectue une segmentation des activités en éléments d´activités, puis, les modèles de Markov cachés recomposent cette segmentation lors de la reconnaissance de l´activité. De plus, l´une des facultés des modèles de Markov cachés est la segmentation des données en les affectant à différents états. Il serait donc plus judicieux de modèliser les cartes probabilistes au cours de l´apprentissage des modèles de Markov cachés.

Le second est l´utilisation de modèles discrets. Ils présentent l´avantage d´être facilement mis en place. Il n´est en effet pas nécessaire de choisir une loi de probabilité. Cependant, ils impliquent une discrétisation des données. Dans ce chapitre, elle est effectuée par une règle de décision. Elle consiste à choisir l´élément d´activité dont la carte de probabilité contient le plus grand nombre de fortes probabilités. Ainsi, si deux activités ont un nombre proche, seule la première est considérée par le modèle de Markov caché. Une amélioration du système est donc de considérer des modèles vectoriels et continus. Les séquences d´observations sont alors définies à partir des cartes de probabilités du capteur.

Dans le chapitre suivant, la reconnaissance d´activité humaine est intégrée dans environnement intelligent. La connaissance de la position des utilisateurs permet l´estimation de la probabilité *a priori* de chaque classe d´activité. L´introduction de cette probabilité permet l´amélioration de la reconnaissance. Par exemple, si une personne se trouve au centre du bureau et proche d´une chaise, la probabilité d´occurence de l´activité «s´asseoir» et plus grande que celle de sortir de la pièce.

Le petit Nicolas en thèse [Pet]

La découverte (4/4)

«Des fois aussi, ça se passe mal, parce que je me trompe. Et quand je me trompe, avec mon patron, ça rigole pas, mais alors pas du tout. `Regardez–moi dans les yeux, Nicolas´, il me dit, pas content du tout. `Vous appelez ça du travail, peut–être?´ qu´il me demande. Eh ben, là ça à l´air d´une question, mais il ne faut pas répondre, parce que sinon, il se fâche tout rouge!»

Chapitre 

MONICA : *Un environnement de travail intelligent et interactif*

Dans un environnement intelligent, également nommé «SmartRoom» ou «SmartEnvironments», les ordinateurs participent aux activités de l´utilisateur en l´aidant dans ses tâches courantes. L´interaction avec le système se fait suivant les mêmes modes que les interactions humaines normales : la voix, le geste, le mouvement ... Dans ce chapitre, nous présentons MONICA[1]*, un projet visant à construire un bureau intelligent et interactif. Les environnements intelligents permettent l´application de la reconnaissance de gestes et d´activités humaines à de nouvelles formes d´interactions homme–machine. Dans ce chapitre, nous décrivons les composants matériels et logiciels du projet. Un ensemble d´applications liées à cet environnement est proposé, parmi lesquels le Tableau Magique, un tableau blanc augmenté.*

1 Motivations

L´arrivée de l´informatique a signifié pour beaucoup une réduction dans l´efficacité du travail. Pour ces personnes, l´informatisation s´est traduite par l´arrivée sur leur bureau

1. MONICA est l´acronyme recursif anglais «MONICA*: Office Network with Intelligent Computer Assistant*».

d´une grosse boîte envahissante transformant en calvaire des tâches simples. Les ordinateurs, même ceux qui se veulent proches des utilisateurs, sont intrusifs et pas vraiment conviviaux[2]. Les utilisateurs doivent engager un dialogue explicite.

Nous pensons que l´informatique ne doit pas impliquer une nouvelle manière de travailler mais simplement améliorer les habitudes. Pour WEISER [Wei91], l´informatique doit être invisible et ne doit pas demander d´adaptation de l´utilisateur tout en lui apportant les bénéfices de la puissance de calcul. L´idée de COEN [Coe98b] est de créer des interfaces de l´utilisateur pour l´ordinateur plutôt que des interfaces informatiques pour les utilisateurs.

L´objectif des environnements interatifs, également appelés «*SmartRoom*» [Pen96], «*Smart Environments*» [NLD99] ou encore «*Intelligent Environnement*» [Coe98a], est d´apporter l´informatique dans le monde réel et physique. Dans ce sens, ils sont considérés comme des réalités augmentées et permettent aux ordinateurs de participer à l´activité des utilisateurs. Ceux–ci interagissent avec le système selon les mêmes modes d´interactions humaines normales : la voix, le geste, le mouvement et le contexte. COEN considère qu´une «pièce intelligente est une pièce qui vous écoute et regarde ce que vous faites ; une pièce à laquelle vous pouvez parler ; avec laquelle vous pouvez interagir en utilisant d´autres modes complexes»[3]. Pour SHAFER *et al* [SKB+98], les environnements interactifs doivent être plus faciles à utiliser que les ordinateurs, ils veulent rendre l´«utilisation d´un ordinateur aussi naturelle que l´allumage d´une lumière»[4].

Contrairement à l´informatique intégrée[5] cherchant à informatiser tous les objets, les environnements interactifs cherchent à minimiser les modifications d´une pièce normale. Seuls des caméras et des microphones sont ajoutés. De plus, une fois que l´infrastructure est installée, il est possible d´augmenter les capacités de la pièce en ajoutant de nouveaux composants logiciels. Il est multimodal à travers des composants de vision et de reconnaissance vocale. Les environnements impliquent plusieurs techniques informatiques différentes [NDL98] :

interaction et contrôle Nous cherchons à interagir avec l´environnement de manière intuitive et appropriée. De nouvelles interactions multimodales doivent donc être créées : reconnaissance et synthèse de parole, interprétation visuelle de scène, reconnaissance de gestes, synthèse d´images.

apprentissage et interprétation L´environnement doit être capable d´apprendre par lui–même les évolutions en interprétant les situations et les contextes.

2. On parle d´interface conviviaux lorsqu´elle permet à n´importe qui de l´utiliser sans connaissance.

3. «Intelligent Rooms are rooms that listen to you and watch what you do; rooms you can speak with, gesture to, and interact with in other complex way.»

4. «computing as naturals lighting»

5. «ubiquitous computing»

robotique Les robots sont un moyen de plus pour aider l´utilisateur dans ses tâches. Ils utilisent avantageusement la connaissance et l´intelligence de l´environnement.

système et programmation Il est nécessaire de créer de nouveaux langages multimodaux permettant également l´intégration d´applications hétérogènes. Le pilotage d´autres appareils, tels que la chaîne hifi ou le micro–onde, sont également à considérer.

réseau Les composants logiciels sont hébergés sur plusieurs machines, ils doivent donc communiquer à travers le réseau. Des architectures client–serveur doivent être envisagées. Le réseau est hétérogène, mélant des réseaux cablés et des réseaux sans fils. Les interfaces utilisant le protocole WAP[6] sont également à considérer.

Une autre motivation est proposée dans la section suivante, elle s´appuie sur un exemple de scénario d´utilisation d´un environnement intéractif. Il s´agit d´un scénario prospectif auquel nous aimerions arrivé au terme de nos recherches. Dans ce scénario, MONICA est également le nom donné à l´environnement.

2 Exemple de scénario

Très tôt ce matin, Suzanne arrive à son bureau. Dehors, il fait encore noir. Elle entre dans le bureau, qui s´est automatiquement éclairé, et répond machinalement au courtois « Bonjour Suzanne » que lui adresse la pièce. Sur le mur situé en face de sa table de travail, Suzanne dispose d´une vue sur la cafétéria, une autre sur sa secrétaire virtuelle Monica, et une troisième sur le bureau de son plus proche collègue, Patrice.

« Monica, il y a-t-il des messages pour moi? interroge-t-elle.
— Vous avez eu un appel téléphonique hier à 19h05 : Suzanne? C´est Patrice. Rappelle-moi demain à la première heure. »

Patrice, adepte du télé–travail, a son bureau à domicile. Leurs réunions à distance sont donc fréquentes. Un « télé–coup d´oeil » via le MediaSpace[7] *confirme à Suzanne la présence de Patrice dans son bureau. En désignant d´un geste la vue sur le bureau de son interlocuteur, elle entre en communication audiovisuelle avec lui. Suzanne se déplace constamment en parlant, ce qui met le système de suivi automatique à rude épreuve.*

6. Le protocole WAP ou «Wireless Application Protocole» est le protocole de communication des appareils mobiles et, en particulier, des téléphones.

7. Le *MediaSpace* est un réseau audio–visuel piloté par des moyens informatiques favorisant les rencontres fortuites et informelles entre les personnes d´un même groupe. Nous nous basons sur le médiaspace `Comedi` developpé à l´IMAG par l´équipe IIHM–CLIPS.

FIG. 6.1 – ***Sur le mur, Suzanne dispose des vues virtuelles du*** **MediaSpace**

Elle s´approche du tableau et dessine quelques schémas. Pour les besoins de sa démonstration, elle pose quelques additions que Monica calcule pour elle. Patrice, qui n´a jamais le dernier mot, modifie bien entendu ces additions et ces schémas. Cette réunion étant importante, ils ne veulent pas être dérangés. Aussi, lorsque le téléphone sonne, c´est Monica qui répond sur un signe de Suzanne. Toutefois, peu avant 11 heures Monica interrompt : « Suzanne, je vous rappelle que vous avez un rendez–vous dans 5 minutes dans la salle de direction. »

Après avoir pris congé de Patrice, Suzanne sélectionne d´un geste les éléments intéressants du tableau et ordonne : « Monica, vous imprimerez deux versions de ce tableau et en faxerez une autre à Jacques. »

3 Description du système

Nous décrivons dans les sections suivantes les deux composants matériel et logiciel de notre prototype d´environnement intelligent : MONICA.

3.1 Architecture matérielle

MONICA est développé dans le batiment de l´INRIA Rhône–Alpes. La pièce mesure 4 mètres 65 par 4 mètres 10 et un large bureau se trouve au milieu. Six caméras sont installées dont cinq sont pilotables par logiciel. Quatre d´entre elles sont situées dans les coins de la pièce permettant de suivre une ou plusieurs personnes en considérant toujours

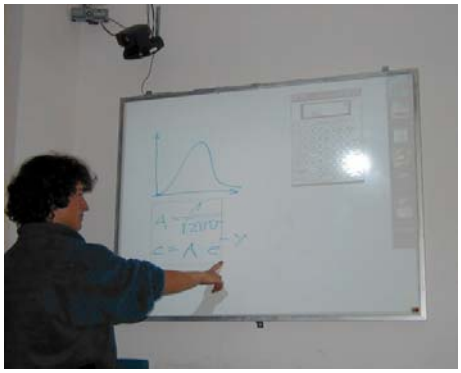


FIG. 6.2 – ***Au tableau, Suzanne manipule les objets réels et virtuels du*** **Tableau Magique**

le meilleur point de vue. Une caméra grand angle, située face à la porte, observe toute la pièce. Elle permet de diriger la stratégie de suivi des individus. Elle est aussi utilisée pour la reconnaissance d´activités. Un tableau blanc est suspendu sur le mur ouest, sur lequel un vidéo projecteur affiche des informations. Le tableau blanc est également observé par la cinquième caméra permettant de définir le «*Tableau magique*», un tableau blanc augmenté, décrit dans la section 4. Deux haut parleurs situés à coté du vidéo–projecteur permettent l´émission de son ou de parole synthétisée. La figure 6.3 montre l´installation de ces périphériques.

Sept ordinateurs PC Pentium II et III sous Linux sont connectés à un réseau Ethernet local. Ils apportent la puissance de calcul au bureau intelligent. Cinq ordinateurs sont utilisés pour les traitements de vision par ordinateur, ils sont connectés aux quatres caméras orientables de coins et à la caméra grand angle. Le cinquième ordinateur est dédié au *Tableau Magique*, il permet à la fois le traitement d´images et la projection des applications informatiques. Le dernier PC héberge le superviseur décrit à la section suivante.

3.2 Environnement logiciel

Les environnements logiciels, mentionne EZIONI [Etz93], offrent une bonne application des techniques développées dans le domaine de la robotique. Un environnement intelligent doit être considéré comme un robot, un robot non simulé précise LE GAL [LMD99], puisqu´il doit gérer à la fois des capteurs et des effecteurs mais aussi des applications logicielles évoluant dans un monde imprévisible. Ainsi, comme tous les robots, un envi-

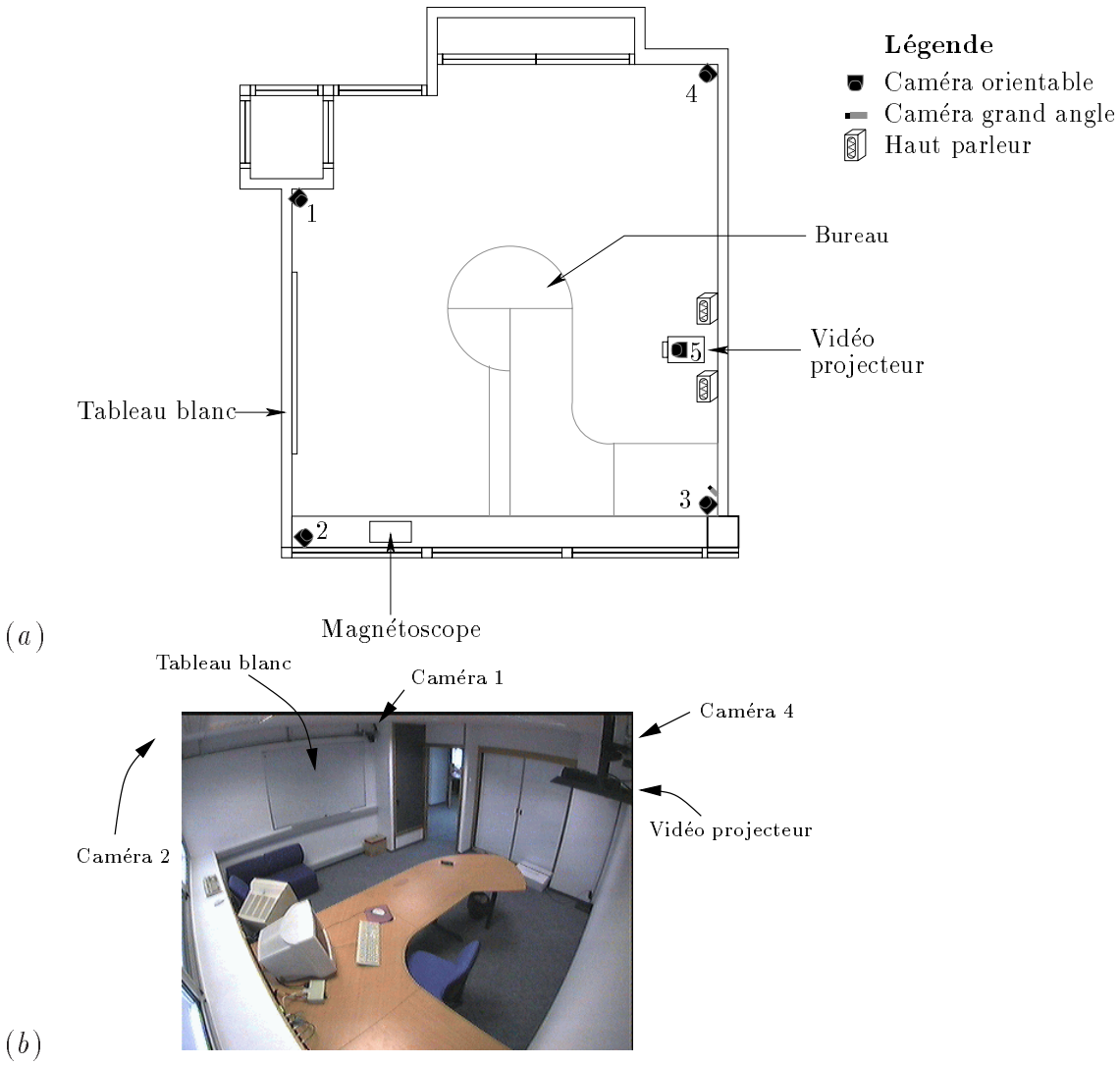


FIG. 6.3 – ***Le bureau intelligent*** MONICA. *Il est équipé de six caméras, cinq sont orientables et une sixième fournissant une image grand angle du bureau, d´un vidéo projecteur permettant l´affichage d´informations sur un tableau blanc et de deux haut parleurs. (*a*) plan d´installation (*b*) Vue du bureau depuis la caméra grand angle.*

ronnement intelligent nécessite un ensemble de modules de contrôle et un superviseur.

COEN [Coe97, Coe98b, CPW+99] propose une architecture multi–agents nommée `MetaGlue`. Chaque agent est responsable d´un équipement particulier de la pièce, comme par exemple d´une caméra ou du système de reconnaissance de la parole. Les 20 agents, répartis sur 10 machines, communiquent entre eux. Un agent peut, par exemple, demander à l´agent responsable de la reconnaissance de la parole de le prévenir à chaque apparition d´un mot particulier.

L´architecture, présentée ici, suit le même esprit. La différence principale est qu´un superviseur est responsable de la communication entre les différents modules ; il agit comme un serveur de ressources. Dans cette architecture, les agents ou modules ne sont pas conscients des ressources disponibles par les autres. Ils n´ont pas à s´adresser au module de localisation de personnes sous les termes : «Localisateur, donne–moi les informations que tu as» mais plutôt poser des questions au serveur de ressources : «Superviseur, donne–moi la position de l´utilisateur dans l´image de la caméra grand angle.». Ainsi, au lieu d´avoir une communication orientée «agents», nous proposons une communication orientée «ressources» [LMD99]. Nous définissons à présent le superviseur en détail et présentons ensuite les modules existants.

3.2.1 Superviseur

Les logiciels utilisés dans une telle application existent à l´intérieur comme à l´extérieur de notre équipe de recherche. Le système de synthèse de parole provient par exemple de l´université d´Edinbourg [BT97]. L´intégration de ces éléments hétérogènes à l´intérieur d´une application cohérente nécessite un superviseur très flexible. Le superviseur proposé est programmé à l´aide d´un langage à base de règles CLIPS[8]. Ainsi, l´addition ou la suppression de modules requère seulement l´addition ou la suppresion des règles correspondantes, sans aucune influence avec les autres règles et les autres modules. L´incrémentalité est alors garantie.

Le système est centralisé, tous les modules communiquent avec le superviseur par des sockets TCP/IP. La distribution des modules sur plusieurs machines est permise. Cependant, la distribution des modules est telle que seuls les flots de contrôle circulent sur le réseau ; les flots de données existent uniquement entre des modules hébergés sur une même machine. Ainsi, deux modules, effectuant un calcul sur une image, seront hébergés sur la même machine, celle sur laquelle est connectée la caméra.

8. CLIPS est l´acronyme de «C Language Intergrated Production System». il s´agit d´un système expert à base de règles développé par la NASA [Bra93]. Le système gère les règles en chaînage avant et permet l´intégration de code C.

a) Communication

Deux types de messages sont considérés : les «*messages poussés*»[9] et les «*messages tirés*»[10]. Lors d´un message poussé, le processus rédacteur est à l´initiative de la communication, il interrompt le lecteur pour lui donner de nouvelles informations et faits. Dans un message tiré, c´est le lecteur qui est à l´origine du message, il reçoit des données du rédacteur à l´aide d´une lecture non bloquante. Il s´agit alors d´une requête à laquelle les rédacteurs donnent une réponse. Cette requête peut être programmée par l´intermédiaire de règles particulières, comme nous le verrons plus loin.

Le protocole de communication est basé sur la technologie XML[11]. Elle permet de définir des valeurs associées à des attributs. Par exemple, le module de localisation, associé à la camera `1`, informe le superviseur qu´il vient de localiser la personne `1` à la position `(1.32, 2.5)` et quelle porte un T-shirt rouge en utilisant le message :

```
<monica_message>
    <head sender="tracker1"/>
    <push>
        <attr name="person">1</attr>
        <set name="position_x">1.32</set>
        <set name="position_y">2.5</set>
        <set name="shirt_color">red</set>
    </push>
</monica_message>
```

L´utilisation du langage XML permet de définir des messages non statiques dans lesquels des attributs peuvent être ajoutés ou supprimés. De plus, plusieurs messages `push` sont possibles dans le même message. Par exemple, le module peut également indiquer la position d´une seconde personne. Supposons qu´un nouveau module de reconnaissance de visage soit ajouté au système pendant l´exécution. Il peut identifier un utilisateur et avertir le superviseur en poussant le message :

```
<monica_message>
    <head sender="face_recognition"/>
    <push>
        <attr name="person">2</attr>
        <set name="name">Suzanne</set>
        <set name="username">suzy</set>
    </push>
</monica_message>
```

9. «*push messages*»
10. «*pull messages*»
11. eXtended Markup Language

A présent, si un module de connection automatique à un ordinateur est ajouté, le superviseur est capable de répondre à la question :

```
<monica_message>
    <head sender="automatic_login"/>
    <pull>
        <attr name="person">2</attr>
        <get name="username"/>
    </pull>
</monica_message>
```

Par ce message, le module nommé «`automatic_login`» demande le nom d´utilisateur de la personne idéntifiée par le numéro `2`. Ce module attend la réponse du superviseur. Pour répondre à la question, le superviseur n´a pas besoin de connaître la signification du terme «`username`».

b) Ajout de règles

Il est possible, en plus de lui fournir des informations, d´ajouter des règles au superviseur. Par exemple, le module de reconnaissance de personnes veut être averti dès qu´un nouvel utilisateur entre dans le bureau. Si, pour effectuer sa reconnaissance, il désire également la position de cet utilisateur dans le bureau, il peut envoyer la requète suivante au superviseur :

```
<monica_message>
    <head sender="face_recognition"/>
    <rule>
        <left_part>
            (0 < x[new_person] < 1) and (y[new_person] < 1)
        </left_part>
        <right_part>
            send face_recognition
                <message type="pull">
                    <set name="position_x">@x[new_person]</set>
                    <set name="position_y">@y[new_person]</set>
                </message>
       </right_part>
    </rule>
</monica_message>
```

Dans cette règle, la partie gauche, notée `<left_part>`, contient les conditions d´exécution de la règle où `new_person` est une valeur numérique donnant le numéro de la personne

entrante, `@x[new_person]` et `@y[new_person]` sont les coordonnées de cette personne (il ne s´agit alors pas d´un symbole mais de la valeur). La partie droite (`<right_part>`) est le programme à exécuter, il s´agit ici de l´envoi d´un message au module de reconnaissance de visage.

c) Redondance d´informations

L´architecture, présentée ici, autorise la redondance des informations permettant ainsi d´incrémenter la robustesse globale du système. Plusieurs modules peuvent fournir les mêmes informations, ces modules peuvent correspondre à des technologies différentes. Selon une heuristique, le superviseur fusionne ces données pour définir une valeur plus précise de cette information. Prenons l´exemple de la reconnaissance de visages, deux technologies principales existent [Bic95, Ess96, Mas98, INR00] : les réseaux de neurones et l´analyse en composantes principales. Il est possible de définir deux modules utilisant chacune des technologies. Lors de la reconnaissance, ils peuvent donner un facteur de confiance sur le résultat de la reconnaissance :

Le message du module à base de réseaux de neuronnes est :

```
<monica_message>
    <head sender="face_recognition.neuralnets"/>
    <push>
        <attr name="person">2</attr>
        <set name="name">Suzanne</set>
        <confident_factor>93\%</confident_factor>
    </push>
</monica_message>
```

Tandis que celui avec la technologie d´analyse en composantes principales est :

```
<monica_message>
    <head sender="face_recognition.pca"/>
    <push>
        <attr name="person">2</attr>
        <set name="name">Pierre</set>
        <confident_factor>96\%</confident_factor>
    </push>
</monica_message>
```

Le superviseur est alors capable, à partir de ces résultats et en utilisant une connaissance *a priori* des algorithmes utilisés, d´effectuer le choix entre Suzanne et Pierre. De plus, l´ordre d´exécution des règles peut être modifié en leur donnant une priorité. Un exemple est proposé dans la section suivante dans le cadre de la localisation des utilisateurs.

3.2.2 Exemple d´applications

Nous donnons dans cette section des exemples d´applications existant dans le projet MONICA. Certaines des applications ont été réalisées, d´autres sont en cours de réalisation. De plus, ces applications sont très liées ; comme nous le verrons, elles peuvent chacunes utiliser les résultats des autres.

a) Reconnaissance d´activités

L´application de reconnaissance d´activités s´appuie sur les résultats présentés au chapitre 5. Nous présentons ici l´architecture au sein de l´environnement MONICA.

Le capteur d´éléments d´activités utilise une base d´éléments à reconnaître, il s´agit des histogrammes multidimensionnels et de l´image provenant de la caméra. Il récupère la région de l´image contenant l´utilisateur, définie par (x, y) la position et (w, h) la hauteur et la largeur. Il pousse alors l´élément d´activité reconnu au superviseur. Dès qu´il reçoit un nouvel élément d´activité a, le superviseur le pousse vers le module de reconnaissance d´activités. Celui–ci, dès qu´il a reconnu l´activité complète à partir de la base des activités représentée par des modèles de Markov cachés, pousse l´activité A au superviseur qui le fait suivre au module de reconnaissance de scénari. Dès que le superviseur connaît l´activité ou le scénario reconnu, il les pousse à l´interface graphique pour affichage. La figure 6.4 représente graphiquement cette application.

b) Localisation des utilisateurs

Afin d´aider les utilisateurs, le système doit en permanence être au courant de la position de chacun d´eux. Lorsqu´une nouvelle personne entre dans le bureau, elle est référencée par un numéro et une carte d´identité lui est attachée. Cette carte sera mise à jour à partir des informations complémentaires obtenues sur la personne.

Lorsqu´il franchit le seuil de la porte, la taille de l´utilisateur est estimée en utilisant les montants de la porte. Un module de suivi, basé sur la couleur du visage, estime les coordonnées de l´utilisateur dans l´image de la caméra. Pendant que l´utilisateur se déplace dans le bureau, l´estimation est mise à jour. Un module est associé à chacune des quatre caméras de coin permettant de suivre plusieurs personnes ou bien de suivre une personne à l´aide de plusieurs caméras. Des estimations multiples de la position d´une personne sont fusionnées à l´aide d´un filtre de KALMAN [Kal60, WB97]. L´intégration d´un filtre de KALMAN dans un tel superviseur est inspirée du projet SAVA [CD95].

Les caméras sont calibrées en utilisant les meubles du bureau pour lesquels la taille est connue et fixe. Étant donné la taille et l´état, assis ou debout, de l´utilisateur, il est possible de calculer, par une simple trigonométrie, la position de la personne dans le bureau. L´état de l´utilisateur est mis à jour à partir du module de reconnaissance d´activités qui permet de détecter l´activité d´une personne s´asseyant ou se levant. Cet état est associé à un utilisateur par l´intermédaire de sa carte d´identité.

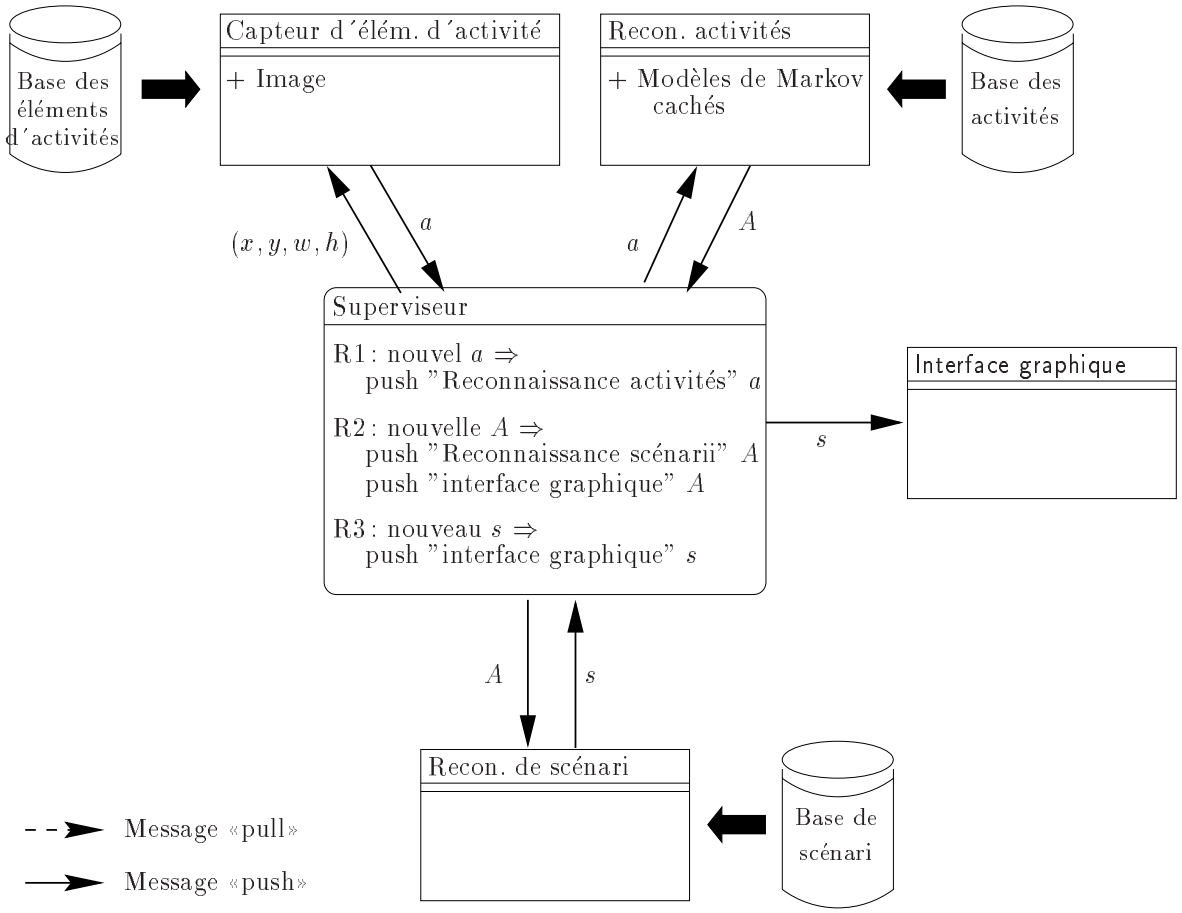


FIG. 6.4 – ***Architecture de la reconnaissance d´activités.*** *voir le texte pour les explications.*

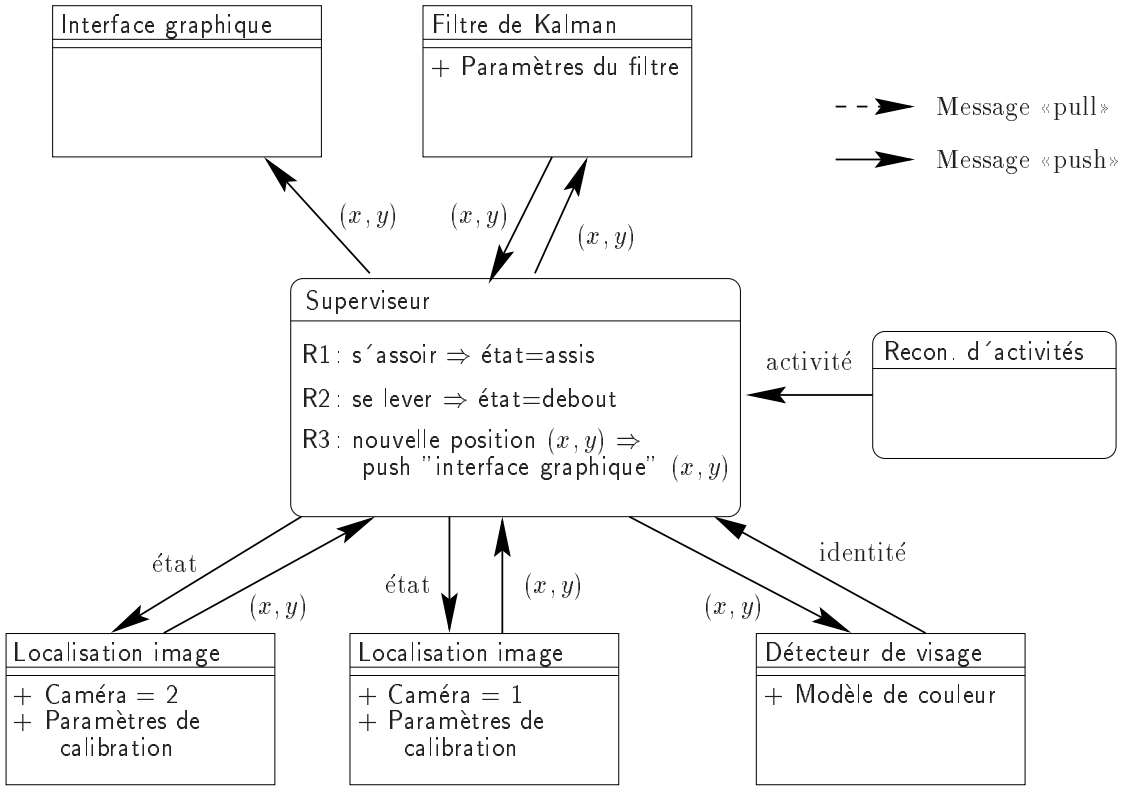


FIG. 6.5 – ***Architecture de la localisation des utilisateurs.*** *voir le texte pour les explications.*

La figure 6.5 montre un exemple d´architecture de l´application de localisation d´un utilisateur. Dans cet exemple, deux modules de localisation de l´utilisateur dans l´image estiment sa position à 25 Hz en utilisant son état (assis ou debout). Cette localisation est renforcée par un module de détection de visage fonctionnant à une vitesse plus lente, environ 5 Hz. Les positions estimées par chacun des modules sont poussées au superviseur. Le filtre de KALMAN récupère les positions de chacun des modules de localisation et pousse les positions fusionnées. Les changements de positions sont alors poussés à l´interface graphique. L´état de l´utilisateur est mis à jour par l´activité poussée par l´application de reconnaissance d´activités présentée précédemment.

c) Module de reconnaissance de gestes

Dans un environnement intératif, un utilisateur doit pouvoir dialoguer par le geste. Le geste le plus utilisé est certainement celui de désignation. Il permet principalement une interaction multimodale de type «Put that there»[12].

FREEMAN et WEISSMAN [FW95] utilisent des gestes simples pour le contrôle d´un poste de télévision. Dans cette application, deux gestes sont reconnus. La main est utilisée comme un pointeur : l´utilisateur voit son déplacement sur un moniteur de contrôle sur lequel sont dessinés les boutons de la télévision. Lorsque l´utilisateur ferme la main, le bouton est pressé. Dans le système ARGUS, KOHLER [Koh96, Koh97] propose également le controle d´appareils vidéo et audio par gestes. Après sélection d´un appareil en le pointant, des configurations particulières permettent de monter ou descendre le volume de l´appareil, le démarrer ou l´arrêter.

Dans `KidsRoom` [BDI97, BID$^+$97, BID$^+$], un environnement intératif et immersif développé au MIT, les enfants sont plongés dans une histoire dans laquelle ils interagissent. Le système est capable de reconnaître les gestes des enfants et modifie l´environnement en conséquence. La figure 6.6 montre un exemple de ces interactions. Dans la scène (a), l´enfant exécute un geste que le monstre reproduit. Dans la scène de la rivière (figure 6.6b), les enfants transforment leur lit en bateau et rament. La vitesse du geste change le défilement du décor. Il s´agit là d´une application ludique des environnements intelligents.

Nous avons également proposé [LM99] une démonstration de l´utilisation de gestes pour le jeu. `GesTris` et `GesBoing` sont deux jeux dans lesquels les pièces sont guidées par les gestes de l´utilisateur. La figure 6.7 montre les gestes effectués par le joueur pour déplacer la pièce vers la gauche ou vers la droite, changer son orientation et la faire tomber.

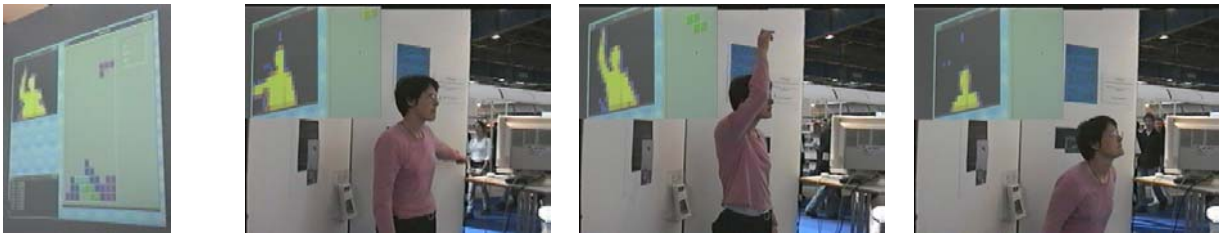
4 Le Tableau Magique

Le tableau magique est l´héritier direct du «`Bureau Numérique`» de WELLNER [Wel91a, Wel91b, NW92, Wel93b]. Le tableau magique est un tableau blanc augmenté par des fonc-

12. Ce principe est expliqué au chapitre 1

(*a*) (*b*)

FIG. 6.6 – ***Exemples d´interaction dans*** `KidsRoom` (a) *Le monstre reconnaît les gestes de l´enfant et les reproduit.* (b) *Les enfants rament sur le lit, transformé en bateau. La vitesse du geste est liée avec la vitesse de défilement du décor. (extrait de [BID+])*



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

FIG. 6.7 – ***Exemples de gestes utilisés dans*** `GesTris` (a) *Projection du jeu : en haut à gauche les mouvements de l´utilisateur et à droite le jeu.* (b) *Déplacement de la pièce vers la gauche.* (c) *Changement de l´orientation de la pièce.* (d) *Faire tomber la pièce. (extrait de [ML99])*

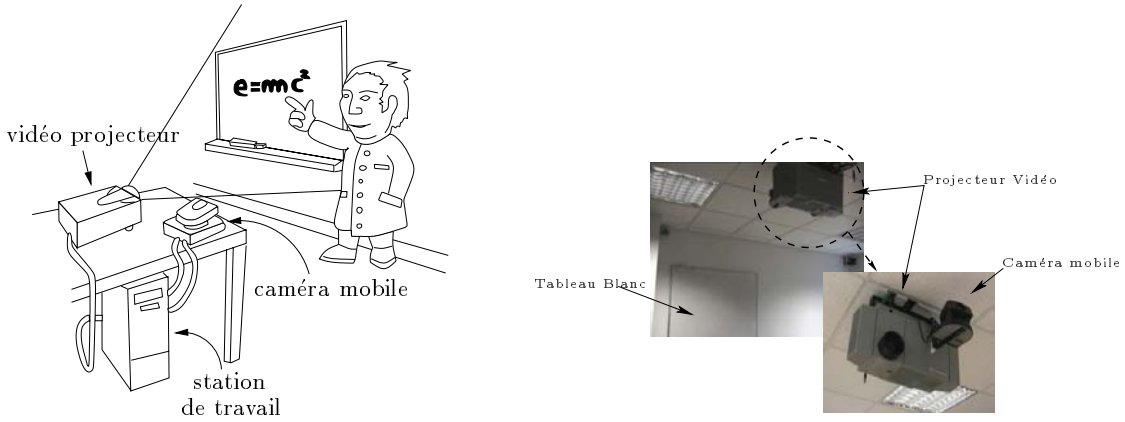


FIG. 6.8 – ***Appareillage du tableau magique***. *Le vidéo projecteur affiche les objets informatiques sur le tableau blanc, la caméra mobile permet l´acquisition des inscriptions et les gestes de l´utilisateur. ((*a*) extrait de [Bér00])*

tions informatiques. Un vidéo projecteur affiche des objets informatiques. Une caméra mobile permet l´acquisition des inscriptions physiques sur le tableau blanc ainsi que les gestes de l´utilisateur. La figure 6.8 montre l´appareillage du tableau magique. Il s´inscrit dans le domaine de *réalité augmentée* et offre quatre fonctionnalités :

1. utilisation classique du tableau blanc ; ceci s´inscrit directement dans la ligne des applications de réalité augmentée : ajouter des fonctions informatiques à l´environnement sans remplacer les fonctions existantes.
2. «capture a posteriori» des informations écrites de manière conventionnelle [Bér00]. Cette capture permet un stockage pour une séance de travail ultérieure, pour archivage ou pour envoi (par courrier électronique par exemple).
3. ajoût de fonctionnalités informatiques. Elles peuvent être exécutables à partir d´interfaces de type classiques, tels que des menus, ou des barres d´outils, ou par commandes gestuelles.
4. partage d´un espace de travail à distance. La connexion à internet, l´acquisition des inscriptions et la projection d´informations permet le partage du tableau sur des sites distants.

Les points 3 et 4 imposent un fonctionnement permanent du système. Dans le contexte de cette thèse, les fonctionnalités informatiques ajoutées au tableau magique sont de nature «commandes gestuelles».

4.1 Utilisation de gestes

Deux types de gestes peuvent être définis dans le contexte du *Tableau Magique* : les gestes de dessin et les gestes de manipulation.

4.1.1 Reconnaissance de gestes de dessin

Les gestes de dessin permettent deux types d´interaction. La première est la reconnaissance du dessin de caractères (lettres ou chiffres) comme nous l´avons présenté à la section 3.3. Elle est une alternative à l´utilisation du clavier. Il est alors possible de donner des paramètres à une application, un nom de fichier par exemple, en l´écrivant dans une zone donnée.

La seconde utilisation est la définition de commande par le dessin de symboles. Dans le `ZombieBoard`, développé à Xerox PARC, Saund [Sau, Sau97] propose un interface utilisateur diagrammatique[13]. Les symboles sont des formes géométriques simples composées des segments de droites. Dans ce système, il ne s´agit pas d´une reconnaissance du geste mais d´une reconnaissance du dessin *a posteriori*. Stafford–Fraser *et al* [SR96, Sta96] proposent également la reconnaissance de dessins.

13. Diagrammatic User Interface

BLACK *et al* utilisent des gestes pour l´exécution de commandes tels que la copie, l´impression, la sauvegarde, l´effacement du tableau, le démarrage et l´arrêt du système de reconnaissance de gestes. La figure 6.9 présente ces gestes.

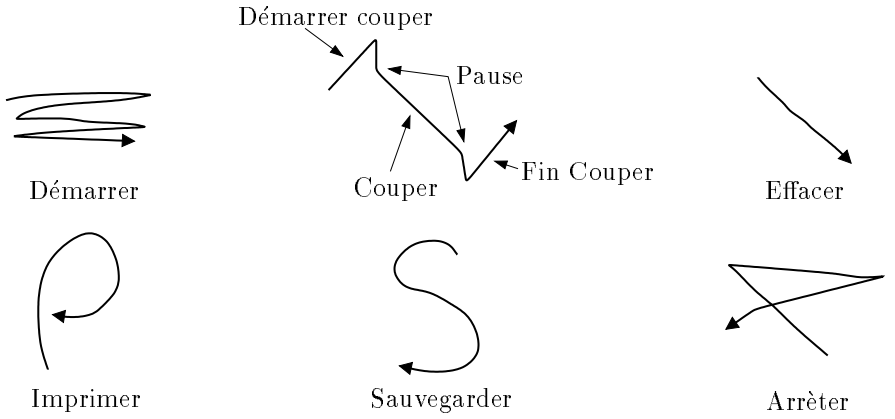


FIG. 6.9 – ***Exemple de gestes de dessins dans le*** `ZombieBoard`. *(d´après [BJ98])*

4.1.2 Reconnaissance de gestes de manipulation

Le *Tableau Magique* est un prototype idéal pour le développement de gestes de manipulation directe. Après avoir sélectionné un objet, nous proposons des gestes permettant des manipulations tel que agrandir ou réduire, tourner, effacer, saisir et relacher. La figure 6.10 propose des exemples de ces gestes inspirés par QUEK [Que94].

- La configuration «pointer» permet la sélection d´un objet ou d´une zone. Elle est alors associée, à la fois, au module de position du doigt et de détection de clic. Trois types de sélections sont possibles : sélection d´un objet électronique en «cliquant» dessus, sélection d´une zone rectangulaire ou sélection de type lasso.
- La configuration «stopper» permet d´arrêter une opération en cours. Il s´agit également de la configuration lorsqu´un objet est saisi.
- Le geste «effacer» permet d´effacer les objets électroniques du tableau. La taille du geste permet de distinguer l´effacement des objets sélectionnés ou l´effacement de tout le tableau.

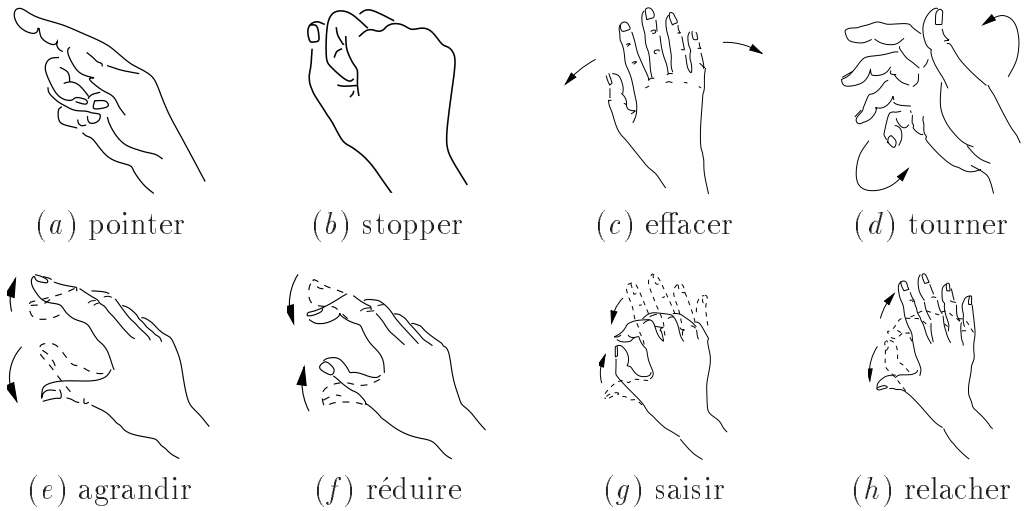


FIG. 6.10 – ***Exemple de gestes de manipulation sur le* Tableau Magique**. *Les gestes «pointer» et «stopper» sont des gestes statiques. Les autres sont dynamiques, les flèches symbolisent les mouvements, les dessins en traits pleins sont les configurations finales, les pointillés correspondent aux configurations initiales. (d´après [Que94])*

- Le geste «tourner» permet d´effectuer une rotation de l´objet. L´orientation de la main au début et à la fin du geste détermine l´angle de rotation.
- Les gestes «agrandir» et «réduire» permettent un changement de taille de l´objet sélectionné. L´opération est effectuée tant que la main n´a pas pris la configuration «stopper».
- Les gestes «saisir» et «relacher» permettent le déplacement d´objets électroniques ou la copie des inscriptions physiques. L´utilisateur saisit l´objet à un endroit pour le relâcher à un autre. Pendant le déplacement, la main est fermée comme dans la configuration «stopper».

4.2 Système supervisé

Les composants principaux du *Tableau Magique* sont un système de suivi du doigt, le système de fenêtrage XWINDOW et un ensemble d´applications. Les applications sont celles existantes sous le système XWINDOW, telles que la calculatrice ou l´application de dessin `XPaint`, ou des applications spécialement écrites pour le *Tableau Magique*, telles que les applications de reconnaissances de gestes et l´application de «copier–coller».

Les applications et modules présents dans le *Tableau Magique* sont les suivants :

Module XWINDOW Il effectue le lien entre le déplacement du doigt et le déplacement du pointeur de la souris, c´est–à–dire le système XWINDOW. Il remplace le pilote bas niveau du pointeur. À chaque nouvelle position ou clic, le superviseur lui pousse les coordonnées du doigt.

Système de suivi de doigt Il pousse la position du doigt dans les coordonnées du tableau. Ce système effectue un calcul rapide et peu précis. Lorsque la précision est nécessaire, lors du clic par exemple, un second module donne alors cette position avec précision.

Module de position de doigt Ce module permet une localisation précise au moment du clic du doigt utile pour certaines applications. Ces applications font alors la demande au superviseur qui, à son tour, tire les coordonnées auprès du module.

«**Copier–coller**» Il s´agit d´une application particulière permettant d´effectuer des copies de portions du tableau contenant du texte réel. Cette copie, après nettoyage, est alors reprojetée à un autre endroit. Cette opération est effectuée à l´aide de commandes gestuelles.

Nettoyage du tableau blanc Lors du «copier–coller» ou lors de la sauvegarde ou l´impression, il est nécessaire de nettoyer l´image. Ce nettoyage permet de classer chaque pixel de l´image dans les classes : «encre» ou

«fond». Le fond contient en particulier toutes les traces d´encre provenant d´un mauvais effaçage du tableau. La technique utilisée est celle proposée par WELLNER [Wel91b, Wel93a], également utilisée par BÉRARD [Bér00]. Le superviseur pousse au module les paramètres de la zone à nettoyer, c´est–à–dire la position (x, y) et la taille (w, h). Le module effectue le nettoyage directement dans l´image projettée.

Détection du clic Le *Tableau Magique* est confronté à l´absence de bouton de souris. Ainsi ce module est chargé de détecter une opération proposant une équivalence. Plusieurs solutions peuvent être retenues. BÉRARD [Bér00] propose l´utilisation d´une pause. Elle correspond à une statibilité de la position du doigt pendant un laps de temps donné. Une seconde solution consiste à effectuer un geste particulier, comme par exemple plier l´index ou décoller le pouce de la paume. Enfin, une solution proposée par WELLNER est l´utilisation d´un microphone derrière le tableau. Le clic est alors simulé par une tape sur le tableau. Chacune des trois solutions présente des inconvénients comme par exemple le ralentissement du système si le temps de pause est trop long ou, au contraire des clics intempestifs si elle est trop courte, la fatigue de l´utilisateur si le clic correspond à un geste de la main, ou bien la détection du clic si le niveau sonore vers le tableau est trop fort. Dans le système proposé, il n´est pas nécessaire de faire un choix entre les techniques. Un module implémentant chacune de ces solutions peut exister au sein du système, le superviseur s´occupe alors de les fusionner. Ceci permet, de plus, un choix possible pour l´utilisateur avec un «clic multimodal». La figure 6.11 donne l´exemple d´architecture avec un seul module de détection de clic basé sur le son.

Reconnaissance de gestes de dessin À partir des positions du doigt que lui pousse le superviseur, le module reconnaît le geste. Il utilise pour cela une base contenant la définition des gestes. Une fois la reconnaissance effectuée, il pousse le symbole d, représentant le geste, au superviseur. Les applications nécessitant des gestes tireront ce symbole du superviseur.

Reconnaissance de gestes de manipulation À partir de la base de connaissance des gestes, ce système reconnaît le geste en utilisant l´image provenant de la caméra face au tableau[14]. Lorsqu´un geste est reconnu, le système pousse le symbole g correspondant au superviseur.

La figure 6.11 représente graphiquement les connections entre ces modules. Ils ne sont que des exemples, par définition de l´architecture du système, il est possible d´ajouter de

14. Il est également possible de considérer que la reconnaissance des gestes soit à partir d´autres caméras du bureau.

nouveaux modules proposant de nouvelles fonctionnalités au système.

5 Synthèse du chapitre

La conception de l´environnement MONICA est en lien direct avec le nouveau domaine de recherche dans les environnements intelligents. Ceux–ci se trouvent à la frontière entre les interactions homme–machine multimodales, le concept de réalité augmentée et l´informatique intégrée. Dans ces systèmes, les interactions se font avec la parole, le mouvement, le geste et le contexte. Ils sont la combinaison de nombreux problèmes techniques : interaction et contrôle, apprentissage et interprétation, robotique, système et programmation, réseau.

Dans le cadre de cette conception, nous avons dû définir une architecture de type clients–serveur, basée sur un serveur de ressources. Il permet l´intégration de plusieurs composantes logicielles différentes et hétérogènes. Le superviseur gère l´ensemble des ressources, il reçoit des informations des modules et les envoie aux modules les demandant. Un ensemble de règles permet une gestion flexible des informations. La redondance des informations est possible et permet d´incrémenter la robustesse du système. Cette redondance est traitée par subsumption ou par fusion. Le dialogue entre les clients et le superviseur est réalisé à l´aide d´un protocole de communication basé sur la technologie XML. Elle permet la définition de messages dont les champs sont définis dynamiquement.

Dans cette thèse, l´environnement MONICA permet l´application des concepts, solutions et algorithmes de reconnaissance de gestes que nous avons proposés dans un domaine nouveau et motivant. Trois interactions sont considérées. La première est l´application des résultats du chapitre précédent dans le cadre d´un environnement intelligent, l´interaction par l´intermédiaire de gestes est également proposée. La définition du *Tableau Magique*, comme composante de l´environnement permet également la définition de gestes. Le *Tableau Magique* [Bér00] est un tableau blanc augmenté, descendant directement du «`Bureau Numérique`» de WELLNER [Wel91a, Wel91b, NW92, Wel93b]. Il permet la définition de commandes gestuelles et la manipulation directe et gestuelle.

Cet environnement est en cours de réalisation, ainsi l´intégration des modules de reconnaissance de gestes n´est pas achevée et aucune expérimentation pratique n´a pu être réalisée.

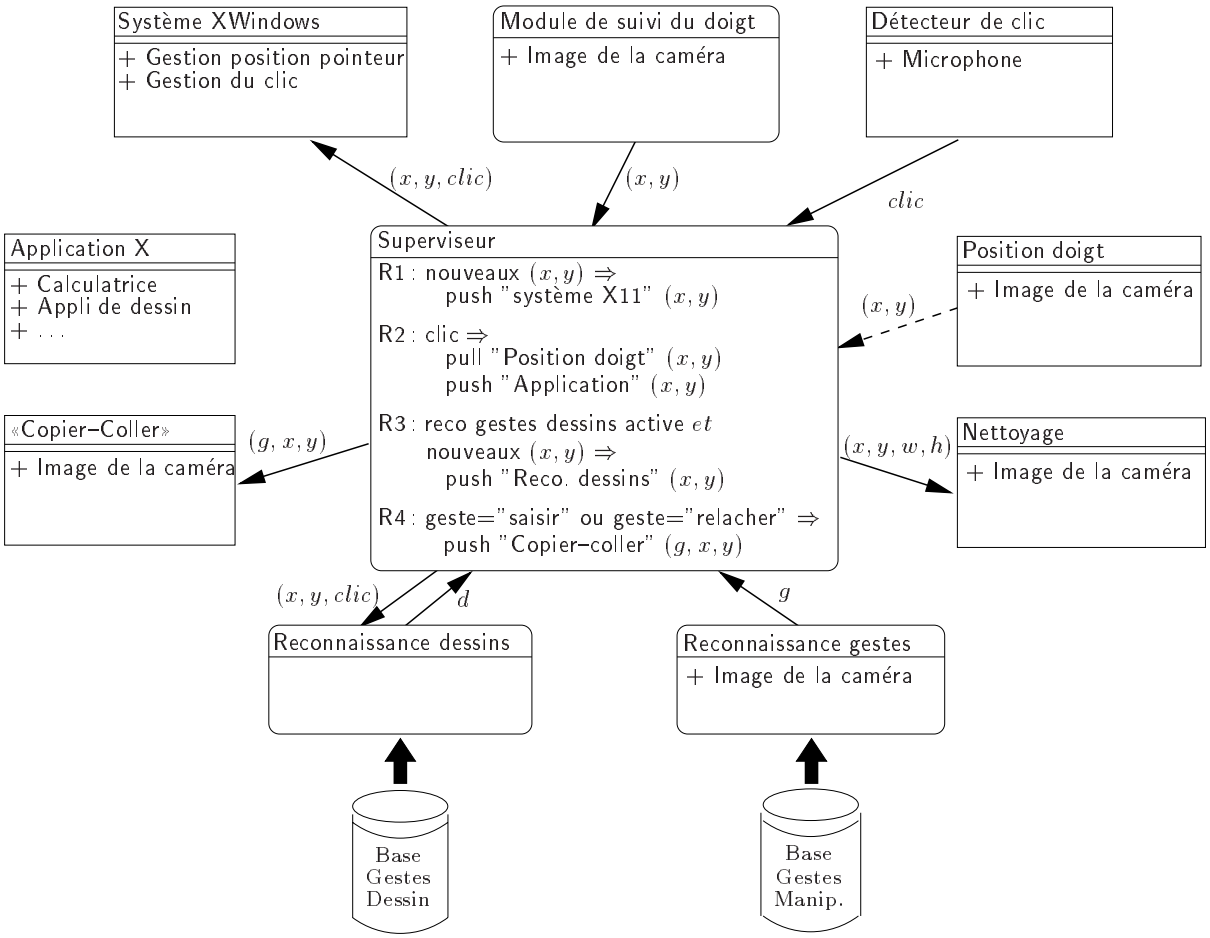


FIG. 6.11 – ***Exemple d´architecture logicielle du Tableau Magique***

Le petit Nicolas en thèse [Pet]

Les séminaires

«*De temps en temps, un monsieur très, très important et vachement fort (mais pas aussi fort que mon patron, quand même) vient nous parler de trucs super–compliqués. Ça s´appelle un séminaire, ça ne rigole pas non plus. Quand le monsieur a fini de parler, mon patron lui pose des tas de questions très compliquées, et il ne sait pas toujours répondre. Et là c´est pas juste, parce que lui, il ne se fait pas disputer !*»

Conclusion

1 Contributions

Dans cette thèse, nous avons présenté notre contribution aux recherches sur l´étude de la communication gestuelle pour les interactions homme–machine. Une première partie de ce manuscrit concerne l´étude de la communication gestuelle et son utilisation dans le domaine de l´interaction homme–machine. L´étude sur les gestes montre que sa fonction sémiotique est celle correspondant le mieux à une utilisation pour l´interaction. Elle a, du point de vue du concepteur du système, une logique de «faire–faire» [PRP94]. L´utilisateur montre au système ce qu´il veut que le système fasse. Les études et les exemples présentés montrent l´utilisabilité du geste pour l´interaction homme–machine. Les exemples sont regroupés autour de trois catégories principales : la reconnaissance de la langue des signes, la réalité virtuelle et la réalité augmentée.

Nous avons proposé le découpage d´un système de reconnaissance de gestes en trois étapes : analyse, reconnaissance et interprétation. Au cours de l´étape d´analyse, les paramètres de la main sont calculés. Il s´agit de déterminer un vecteur de mesures représentant la position et la configuration de la main. La dimension temporelle du geste définit une trajectoire dans l´espace des paramètres. La seconde étape est la reconnaissance du geste. L´analyse spatio–temporelle de la trajectoire permet sa classification. Un symbole représentant le geste reconnu est généré. Lors de l´étape d´interprétation, le symbole est utilisé pour effectuer les commandes correspondantes.

La contribution principale de cette thèse est la proposition de solutions techniques permettant la réalisation d´un système de reconnaissance de geste. Nous avons proposé un ensemble de méthodes permettant l´extraction de caractéristiques de position ou de configuration. La localisation de la main dans une image étant une opération difficile, nous avons proposé un système de coopération des techniques permettant de rendre cette

localisation plus robuste et plus fiable.

Deux approches pour l´extraction d´un vecteur représentant la configuration sont proposées. L´analyse en composantes principales d´images permet de réduire une image en un vecteur de petite dimension. Ce vecteur code les variations entre les images, c´est–à–dire principalement les changements de configuration par rapport à la moyenne. Nous avons également proposé d´étendre cette approche aux discriminants de FISHER. Ceux–ci proposent une réduction de l´espace en optimisant la discriminabilité contrairement à l´analyse en composantes principales optimisant la reconstruction. La deuxième méthode proposée est l´utilisation des invariants de HU. Ils sont calculés à partir des moments de la distribution des pixels de l´image de telle sorte qu´ils soient invariants en rotation et similitude. Les vecteurs de caractéristiques nous permettent, dans un premier temps, la reconnaissance des configurations. Cette reconnaissance s´appuie sur la classification euclidienne et bayésienne. Une classification basée sur la distance à l´espace de l´analyse en composantes principales est également proposée.

La reconnaissance de gestes est la seconde étape de notre schéma d´un système de reconnaissance et d´interprétation de gestes. Le problème est la classification consistant à associer, à une trajectoire de classe inconnue, la classe la plus probable ou représentant le mieux cette trajectoire. Cette décision est rendue difficile par la variabilité du geste rendant les méthodes de mise en correspondance directe impossible. Nous avons proposé trois méthodes pour résoudre ce problème. La première méthode est l´utilisation d´automates d´états finis. Un geste est représenté par un automate particulier dont les états représentent les configurations par lesquelles le geste passe. Une transition correspond à un changement de configuration. Cette méthode s´appuie sur la classification des vecteurs de caractéristiques en classes de configuration. Les modèles de Markov cachés constituent une extension naturelle de cette approche. Nous avons proposé leur utilisation en nous concentrant sur trois problèmes principaux : étude de l´architecture du modèle, détermination de la nature des séquences d´observations et détermination du nombre optimal d´états. Nous avons, en particulier, proposé une méthode automatique de sélection du nombre d´états. Cette sélection s´appuie sur le critère d´information bayésien permettant de déterminer le modèle le plus adéquat aux données et minimisant le nombre de paramètres, c´est–à–dire le nombre d´états.

Nous avons, enfin, proposé l´utilisation des solutions de cette thèse dans deux applications. La première est la reconnaissance d´activités humaines. Cette application présente la combinaison d´un capteur d´éléments d´activités et la reconnaissance par modèles de Markov cachés. L´étape d´analyse est ici réalisée par la définition d´un capteur probabiliste développé par CHOMAT [Cho00, CMC00]. Il utilise une description spatio–temporelle du mouvement donnant une carte de probabilité pour chaque classe considérée. Une règle de décision simple permet la transformation de ces cartes en symbole d´entrée aux modèles de Markov cachés discrets. Les premières expérimentations réalisées montrent des résultats encourageants. La seconde application est le bureau intelligent et interactif MONICA. Dans

cet environnement, les ordinateurs participent aux activités de l´utilisateur en l´aidant dans ses tâches quotidiennes. De plus, l´interaction avec l´environnement se fait de manière naturelle, en particulier en utilisant les gestes. Cette application représente donc un banc d´expérimentation intéressant pour la reconnaissance de gestes liée à de nouvelles formes d´interaction homme–machine.

2 Limites et perspectives

Nous reportons dans cette section les limitations de notre approche et des solutions proposées. Ces limites ouvrent naturellement des perspectives à court terme ou de nouvelles perspectives de recherche.

2.1 Limites et perspectives à court terme

Les limites de notre travail présentent deux aspects principaux : un aspect technique et un second applicatif.

Aspect technique Dans cette thèse, nous avons proposé un certain nombre de solutions permettant, dans un premier temps, d´extraire des caractéristiques spatiales et des configurations d´une image de mains ; de classifier la configuration ; puis, de reconnaître le geste. Certaines ont été peu ou pas expérimentées, il conviendrait donc dans un premier temps de pousser leur étude. En particulier, nous avons étendu l´extraction de caractéristiques de configuration par analyse en composantes principales vers l´utilisation des discriminants de FISHER. Bien que l´étude effectuée par BELHUMEUR *et al* semblent montrer une stabilité plus grande des discriminants de FISHER en présence de changements et pour la reconnaissance de visages, une étude centrée sur la reconnaissance sur la main en considérant les changements liés à nos applications est nécessaire.

La reconnaissance de gestes dynamiques en utilisant les modèles de Markov cachés ou l´algorithme de reconnaissance statistique de trajectoires s´est limité à la reconnaissance du geste de dessins de quelques lettres issues du langage *Unistroke*. Il conviendrait, dans un premier temps, de l´étendre cette reconnaissance à l´ensemble du langage, c´est–à–dire 78 signes supplémentaires. D´autre part, l´application de ces méthodesà des vecteurs de caractéristiques, contenant, en particulier, la configuration, reste à effectuer. Nous avons effectué un apprentissage simple des modèles. Chaque modèle est entraîné indépendamment des autres, avec sa base d´exemples. Pour un système de classification, il semble plus naturel de les entraîner de manière à ce qu´ils s´excluent mutuellement. Ce type d´apprentissage n´a pas été effectué dans cette thèse et, constitue une perspective logique à ce travail.

La reconnaissance statistique de trajectoires a montré des résultats intéressants. Cependant, l´algorithme de reconnaissance globale n´a pas été expérimenté. Il est donc

impossible d´affirmer que celui–ci est adéquat au problème et que l´approximation que nous avons adoptée, en considérant les signatures indépendantes, est valide. Nous pensons également utiliser les modèles de Markov cachés où les séquences d´observations sont les signatures des fenêtres temporelles.

Applicatif Nous avons proposé deux applications de reconnaissance d´activité humaines et de gestes. La reconnaissance d´activités humaines montre des premiers résultats prometteurs. Cependant, deux problèmes restent à résoudre. Le premier concerne la cohérence entre les résultats de la règle de décision et les états des modèles de Markov cachés. La règle de décision propose une segmentation de l´activité en éléments d´activité basée sur la réponse des capteurs probabilistes. Cette règle semble ne pas être adéquate, il serait plus judicieux d´effectuer cette segmentation directement par les modèles de Markov cachés. Dans un premier temps, une discrétisation des cartes de probabilité peut être effectuée par l´utilisation d´un dictionnaire. Une autre solution consiste à utiliser des modèles vectoriels et continus. Les séquences d´observations sont alors définies à partir des cartes de probabilités.

Pour l´environnement MONICA, l´application de la reconnaissance de gestes reste à faire. Les premiers composants logiciels sont en cours d´installation. L´intégration du module de reconnaissance d´activités humaines est également en cours d´installation. Cependant, une adaptation du logiciel de modèles de Markov cachés est nécessaire pour effectuer une reconnaissance «au fil de l´eau». L´interprétation des gestes nécessite dans un premier temps la définition de leur nature. Le geste de désignation est l´un des plus utiles, il requiert cependant l´achèvement du développement du module de localisation. Le *Tableau Magique*, bien qu´un prototype fonctionne au laboratoire CLIPS, n´est actuellement pas en fonction dans notre propre laboratoire. Un peu de développement est donc nécessaire pour son portage. Ce portage permet l´intégration du système de reconnaissance de gestes. L´ensemble de ces perspectives demandent d´avantage d´efforts d´ingénierie que de recherche. Ils sont cependant nécessaires pour la validation de l´utilisation des gestes dans ces applications.

2.2 Perspectives à long terme

Ce travail ouvre également de nombreuses perspectives de recherche à plus long terme. Nous présentons ici celles qui nous semblent les plus prometteuses.

Autres techniques de reconnaissance dynamique Dans cette thèse, nous avons étudié les modèles de Markov cachés simples. D´autres auteurs [Rig00, INR00] ont proposé d´utiliser des modèles plus complexes tels que des modèles hybrides. Ces modèles utilisent de réseaux de neurones comme estimateur *a posteriori* de la probabilité. Les modèles de Markov paramétriques [WB98] permettent d´estimer la probabilité d´un geste ainsi que

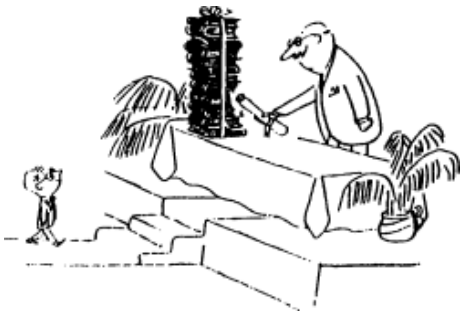
ces paramètres, comme, par exemple la direction pour le geste «*pointer*». Enfin, l´approche de `Condensation` proposée par BLACK et JEPSON [BJ98] semble également prometteuse. Elle permet la reconnaissance des gestes comme les modèles de Markov cachés mais aussi leurs paramètres.

Multimodalité L´objectif de cette thèse a été la reconnaissance de gestes pour l´interaction homme–machine. Comme nous l´avons présenté dans le premier chapitre, les nouvelles formes d´interactions sont multimodales or, nous sommes concentrés sur la reconnaissance des gestes indépendamment de celle de la parole. Une perspective de recherche est de considérer la reconnaissance des gestes conjointement avec la parole. Parole et geste étant souvent redondants ou complémentaires, la reconnaissance de l´un peut permettre de lever des ambiguïtés du second. Cette recherche implique également la synchronisation des reconnaissances, en particulier dans le cas des interactions de type «*Put that there*».

Le petit Nicolas en thèse [Pet]

La soutenance

«Quand j´aurai fini, il y aura une grande cérémonie avec plein de gens très, très forts (il y aura même d´autres patrons, c´est dire) et il y aura une gentille madame† très, très important qui me dira que c´est très bien, mon petit, les chemins de la Recherche me sont glorieusement ouverts et je suis l´honneur de mes parents et l´orgueil de mon pays, et tout le baratin. Et après, il y aura un super goûter avec tous mes amis. Génial!»

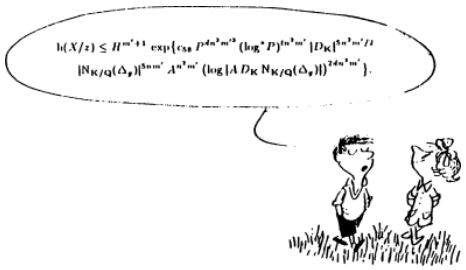


«Et quand il lira tout cela dans le journal, mon papa sera très fier et ma maman sera tellement contente qu´elle me servira deux fois de la crème renversée, mon dessert préféré. C´est vraiment super, une thèse, à la fin»

† changé par l´auteur de ce manuscrit

Après–propos

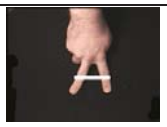
Le petit Nicolas en thèse [Pet]
La gloire



«D´ailleurs les filles, ça les impressionne drôlement de savoir qu´on a fait une thèse d´informatique† et qu´on a trouvé des tas de algorithmes† compliqués et tout, et tout. Même la maman de Marie–Edwige, elle me fait des grands sourires maintenant, alors qu´elle trouvait que j´étais un garçon très turbulent.»

† changé par l´auteur de ce manuscrit

Annexes

Annexe 

Analyse en Composantes Principales

1 Définitions.

Soit un ensemble de m images $I_i, i = 1 \ldots m$. L´image moyenne est définie par :

$$\bar{I} = \frac{1}{m} \sum_{i=1}^{m} I_i \tag{A.1}$$

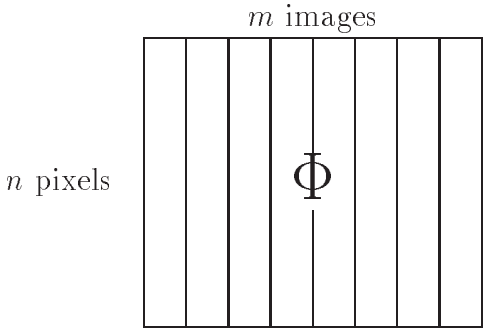
Toutes les images peuvent être normalisées par la soustraction de l´image moyenne :

$$\dot{I}_i = I_i - \bar{I} \tag{A.2}$$

Ces m images normalisées, $\dot{I}_i, i = 1 \ldots m$, peuvent être données par la matrice $\Phi_{[n,m]}$:

$$\Phi_{[n,m]} = [\dot{I}_1, \ldots, \dot{I}_m] \tag{A.3}$$

Cette matrice, de taille $n \times m$, contient la base d´entraînement.



2 Calcul des vecteurs propres.

La matrice de covariance $A_{[n,n]}$ des images normalisées résulte de la multiplication de $\Phi_{[n,m]}$ par la transposée $\Phi^T_{[m,n]}$ et divisé par le nombre m de données :

$$A_{[n,n]} = \frac{1}{m}\Phi.\Phi^T = \frac{1}{m}\sum_{n=1}^{m} \dot{I}_n \dot{I}_n^T \tag{A.4}$$

Le vecteur e, associé à la valeur propre λ, est le vecteur propre de la matrice $A_{[n,n]}$ si :

$$A_{[n,n]}.e = \lambda.e \tag{A.5}$$

Ce qui se traduit par le système linéaire :

$$A_{[n,n]}.E_{[n,m]} = E_{[n,m]}.\Lambda_{[m,m]} \tag{A.6}$$

Dans laquelle $E_{[n,m]}$ est la matrice contenant les m vecteurs propres e_i de taille n :

$$E_{[n,m]} = [e_1, \ldots, e_m] \tag{A.7}$$

et Λ la matrice diagonale contenant les valeurs propres λ

$$\Lambda_{[m,m]} = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix} \tag{A.8}$$

Les vecteurs propres e_i de la matrice E permettent la définition d´un sous–espace $\mathcal{E}$ des images. Les vecteurs propres de E sont orthonormés.

3 Calcul de l´espace pour un petit ensemble de données.

Soit $A'_{[m,m]}$ la matrice de covariance définie par :

$$A'_{[m,m]} = \frac{1}{m}\Phi^T.\Phi \tag{A.9}$$

Les matrices $E'_{[m,n]}$ et $\Lambda'_{[n,n]}$ sont les vecteurs propres et les valeurs propres de A' si elles repondent au système :

$$A'_{[m,m]}.E'_{[m,n]} = E'_{[m,n]}.\Lambda'_{[n,n]} \tag{A.10}$$

soit :

$$\frac{1}{m}\Phi^T.\Phi.E'_{[m,n]} = E'_{[m,n]}.\Lambda'_{[n,n]} \tag{A.11}$$

En multipliant par la gauche avec Φ, le système devient :

$$\Phi.\frac{1}{m}\Phi^T.\Phi.E' = \Phi.E'.\Lambda' \tag{A.12}$$

$$\frac{1}{m}(\Phi.\Phi^T).(\Phi.E') = (\Phi.E').\Lambda' \tag{A.13}$$

En appliquant l´équation A.4 :

$$A.(\Phi.E') = (\Phi.E').\Lambda' \tag{A.14}$$

Comme la solution du système linéaire est unique [Kre93], nous avons :

$$\Phi.E' = E \tag{A.15}$$

$$\Lambda' = \Lambda \tag{A.16}$$

Ainsi, en calculant les vecteurs propres pour la matrice de covariance A' de taille $m \times m$, il est possible de déduire ceux de la matrice A de taille $n \times n$. Cette méthode est très intéressante dans le cas où $m \ll n$.

4 Transformation vers le sous–espace propre.

4.1 La transformation $\mathcal{T}$

La transformation $\mathcal{T}$, permettant d´obtenir la représentation d´une image I_k dans l´espace propre $\mathcal{E}$ par les coefficients ω_k, est donnée par :

$$\omega_k = \mathcal{T}(I_k) = E^T_{[n,m]}.(I_k - \bar{I}) \tag{A.17}$$

La reconstruction (ou transformation inverse) $\mathcal{T}^{-1}$ est définie par :

$$E_{[n,m]}.\omega_k = I_k - \bar{I} \tag{A.18}$$

$$I_k = E_{[n,m]}.\omega_k + \bar{I} = \mathcal{T}^{-1}(\omega_k) \tag{A.19}$$

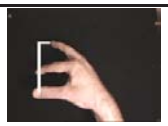
4.2 Erreur de reconstruction

Une erreur de reconstruction ϵ est mesurée entre l´image reconstruite $\hat{I}_k$ et l´image projetée I_k :

$$\epsilon_k = dist(\mathcal{T}^{-1}\Big(\mathcal{T}(I_k)\Big), I_k) \quad \text{(A.20)}$$

$$= dist(\hat{I}_k, I_k) \quad \text{(A.21)}$$

Cette erreur permet, entre autre, de déterminer si la base ayant été utilisée pour le calcul du sous–espace est suffisamment représentative ou bien de déterminer si l´objet contenu dans l´image correspond bien aux objets appris par l´intermédiaire de la base.

Annexe 

Résultats complémentaires

Cette annexe présente des résultats complémentaires. En particulier, nous donnons ici les matrices de confusion des expérimentations.

1 Reconnaissance de configurations

1.1 Classification des moments de Hu

1.1.1 Classification euclidienne

	pointer	*stop*	*gauche*	*droite*	*poing*	*main*	*V*	*L*	Total	%
pointer	**39**		6	23				12	39	48.8%
stop		**80**							80	100.0%
gauche			**0**	76			2	2	0	0.0%
droite				**73**				7	80	91.3%
poing	16	20	5	26	**1**		8	4	1	1.3%
main		30				**4**	46		4	5.0%
V		20			4		**59**	1	59	73.8%
L				28				**48**	48	60.0%
Total									311	47.5%

1.1.2 Classification gaussienne

	pointer	*stop*	*gauche*	*droite*	*poing*	*main*	*V*	*L*	Total	%
pointer	**80**								80	100.0%
stop		**80**							80	100.0%
gauche			**2**	42				36	2	2.5%
droite				**80**					80	100.0%
poing	67	1		2	**8**			2	8	10.0%
main		20				**50**		10	50	62.5%
V	7	14			4		**55**		55	68.8%
L				3				**77**	77	96.3%
Total									432	67.5%

1.2 Classification de configuration par distance à l´espace propre

1.2.1 Images normalisées à une taille 8×8

	pointer	*stop*	*gauche*	*droite*	*poing*	*main*	*V*	*L*	Total	%
pointer	**40**						25		40	100.0%
stop		**37**	1	40	40	5	11	4	37	92.5%
gauche			**39**						39	97.5%
droite		2		**0**		8		36	0	0.0%
poing					**0**				0	0.0%
main						**27**			27	67.5%
V		1					**4**		4	10.0%
L								**0**	0	0.0%
Total									147	45.9%

1.2.2 Images normalisées à une taille 16×16

	pointer	*stop*	*gauche*	*droite*	*poing*	*main*	*V*	*L*	Total	%
pointer	**40**								40	100.0%
stop		**36**							36	90.0%
gauche			**40**				1		40	100.0%
droite				**40**				2	40	100.0%
poing		4			**40**				40	100.0%
main						**40**			40	100.0%
V							**39**		39	97.5%
L								**38**	38	95.0%
Total									313	97.8%

1.2.3 Images normalisées à une taille 32×32

	pointer	*stop*	*gauche*	*droite*	*poing*	*main*	*V*	*L*	Total	%
pointer	**40**								40	100.0%
stop		**39**							39	97.5%
gauche			**40**						40	100.0%
droite				**40**					40	100.0%
poing		1			**40**				40	100.0%
main						**40**			40	100.0%
V							**40**		40	100.0%
L								**40**	40	100.0%
Total									319	99.7%

1.2.4 Images normalisées à une taille 64×64

	pointer	*stop*	*gauche*	*droite*	*poing*	*main*	*V*	*L*	Total	%
pointer	**40**								40	100.0%
stop		**38**							38	95.0%
gauche			**40**						40	100.0%
droite				**40**					40	100.0%
poing		2			**40**				40	100.0%
main						**40**			40	100.0%
V							**40**		40	100.0%
L								**40**	40	100.0%
Total									318	99.4%

2 Reconnaissance de gestes

2.1 Expérimentations sur Unistroke

Les tableaux suivants sont matrices de confusion entre les lettres pour chacune des méthodes.

2.1.1 Modèles de Markov discrets, architecture complète

a) Méthode directe

	Lettres à classer							
	A	E	H	L	O	Q	Total	
Nb états	5	5	5	5	5	5		
A	**23** **25**						48	96%
E		**0** **25**				1	25	50%
H	2		**19** **25**				44	88%
L				**21** 25			46	92%
O			7		**25** **20**	2	45	90%
Q		25		4	5	**23** **24**	47	94%
Total							255	85%

b) Méthode heuristique

	Lettres à classer							
	A	E	H	L	O	Q	Total	
Nb états	2	4	4	2	4	5		
A	**23** **25**						48	96%
E		**0** **25**				1	25	50%
H			**18** **25**				43	86%
L				**21** **25**			46	92%
O	2		7	4	**25** **20**	7	45	90%
Q		25			5	**18** **24**	42	84%
Total							249	83%

c) Méthode automatique

	Lettres à classer							
	A	E	H	L	O	Q	Total	
Nb états	2	2	2	2	2	2/3 [1]		
A	**23** **25**						48	96%
E		**25** **25**		2		1 1	50	100%
H			25 **25**	2			50	100%
L				**21** **25**			46	92%
O	2				**25** **21**	1	46	92%
Q					4	**23** **24**	47	94%
Total							297	96%

1. 2 ou 3 selon la serie

2.1.2 Modèles de Markov discrets, architecture gauche-droite

a) Méthode directe

	Lettres à classer							
	A	E	H	L	O	Q	Total	
Nb états	5	5	5	5	5	5		
A	**23** **25**						48	96%
E		**24** **25**		1		1 1	49	98%
H	2		**25** **24**	1			49	98%
L				**21** **25**			46	92%
O			1	2	**25** **25**	13 12	50	100%
Q		1				**11** **12**	23	46%
Total							265	88%

b) Méthode heuristique

	Lettres à classer							
	A	E	H	L	O	Q	Total	
Nb états	2	4	4	2	4	5		
A	**22** **25**						47	94%
E		**0** **23**					23	46%
H	1		**18** **23**				41	82%
L				**21** **25**			46	92%
O	2		7 2	4	**25** **25**	8 10	50	100%
Q		25 2				**17** **15**	32	64%
Total							239	80%

c) Méthode automatique

	Lettres à classer							
	A	E	H	L	O	Q	Total	
Nb états	2	2	2	2	2	3		
A	**23** **25**						48	96%
E		**24** **25**					49	96%
H	2		**25** **25**				50	100%
L				**21** **25**			46	92%
O					**25** **25**	13	50	100%
Q		1		4		**25** **12**	37	74%
Total							280	93%

2.1.3 Robustesse de la reconnaissance, modèles discrets

	Lettres à classer							
	A	E	H	L	O	Q	Total	
Nb états	2	2	2	2	2	3		
A	**1**						1	4%
E		**15**	2	10		6	15	60%
H			**15**	2			15	60%
L				**3**			3	12%
O			2		**2**		2	8%
Q	24	10	6	10	23	**19**	19	76%
Total							55	40%

Annexe C

Le Petit Prince

« Lorsque j´avais six ans j´ai vu, une fois, une magnifique image, dans un livre sur la Forêt Vierge qui s´appelait «Histoires Vécues». Ça représentait un serpent boa qui avalait un fauve. Voilà la copie du dessin.

On disait dans le livre : « Les serpents boas avalent leur proie tout entière, sans la mâcher. Ensuite ils ne peuvent plus bouger et ils dorment pendant les six mois de leur digestion. »

J´ai alors beaucoup réfléchi sur les aventures de la jungle et, à mon tour, j´ai réussi, avec un crayon de couleur, à tracer mon premier dessin. Mon dessin numéro 1. Il était comme ça :

J´ai montré mon chef-d´oeuvres aux grandes personnes et je leur ai demandé si mon dessin leur faisait peur.

Elles m´ont répondu : « Pourquoi un chapeau ferait–il peur ? »

Mon dessin ne représentait pas un chapeau. Il représentait un serpent boa qui digérait un éléphant. J´ai alors dessiné l´intérieur du serpent boa, afin que les grandes personnes puissent comprendre. Elles ont toujours besoin d´explications. Mon dessin numéro 2 était comme ça :

Les grandes personnes m´ont conseillé de laisser de côté les dessins de serpents boa ouverts ou fermés, et de m´intéresser plutôt à la géographie, à l´histoire, au calcul et à la grammaire. C´est ainsi que j´ai abandonné, à l´âge de six ans, une magnifique carrière de peintre. J´avais été découragé par l´insuccès de mon dessin numéro 1 et de mon dessin numéro 2. Les grandes personnes ne comprennent rien toutes seules, et c´est fatiguant, pour les enfants, de toujours et toujours leur donner des explications.

J´ai donc dû choisir un autre métier et j´ai appris à piloter les avions. J´ai volé un peu partout dans le monde. Et la géographie, c´est exact, m´a beaucoup servi. Je savais reconnaître, du premier coup d´oeil, la Chine de l´Arizona. C´est très utile, si l´on est égaré pendant la nuit.

J´ai ainsi eu, au cours de ma vie, des tas de contacts avec des tas de gens sérieux. J´ai beaucoup vécu chez les grandes personnes. Je les ai vues de très près. Ça n´a pas trop amélioré mon opinion.

Quand j´en rencontrais une qui me paraissait un peu lucide, je faisis l´expérience sur elle de mon dessin numéro 1 que j´avais toujours conservé. Je voulais savoir si elle était vraiment compréhensive. Mais toujours elle me répondait: « C´est un chapeau. » Alors je ne lui parlais ni de serpents boas, ni de forêts vierges, ni d´étoiles. Je me mettais à sa portée. Je lui parlais de bridge, de golf, de politique et de cravates. Et la grande personne était bien contente de connaître un homme aussi raisonnable. »

Le Petit Prince
De Saint–Exupéry
[dS43, Chap. 1]

Nombre de lettres

a	b	c	d	e	f	g	h	i	j
170	15	58	76	326	16	27	16	157	29

k	l	m	n	o	p	q	r	s	t
0	89	51	161	121	67	13	136	180	117

u	v	w	x	y	z
118	32	1	9	1	0

Bibliographie

[3Co] 3Com. « *PalmPilot, Users Manual* ».

[AB85] E. H. Adelson et J. R. Bergen. « Spatio–temporal energy models for the perception of motion ». *Optical Society of America*, 2(2):284–299, 1985.

[AB91] E. H. Adelson et J. R. Bergen. « *Computational Models of Visual Processing* », Chapitre The Plenoptic function and the lements of early vision. MIT Press, 1991.

[AG92] P. Aschwanden et A. Guggenbühl. « *Robust Computer Vision* », Chapitre Experimental Result from a Comparative Study on Correlation–Type Registration Algorithms, pages 268–189. Wichmann Publisher, 1992.

[AJC97] C. S. Andersen, S. D. Jones, et J. L. Crowley. « Appearance Based Processes for Visual Navigation ». Dans *Proceedings of the 5th International Symposium on Intelligent Robotic Systems (SIRS´97)*, pages 227–236, Royal Institue of Technology, Stockholm, Sweden, Juillet 1997.

[ATLC95] T. Ahmad, C. J. Taylor, A. Lanitis, et T. F. Cootes. « Tracking and Recognising Hand Gestures Using Statistical Shape Models ». Dans *British Machine Vision Conference*, 1995.

[Axt91] R. E. Axtell. *Gestures, The DO´s and TABOO´s of Body Language Around the Wolrd*. John Wiley & Sons, 1991.

[BBL93] T. Baudel et M. Beaudouin-Lafon. « CHARADE : Remote Control of Objects using Free-Hand Gestures ». *Communications of the ACM*, 36(7):29–35, Juillet 1993.

[BCG98] C. Biernacki, G. Celeux, et G. Govaert. « Assessing a Misxture Model for Clustering eith Integrated Classification Likelihood ». Rapport Technique, Institut National de Recherche en Informatique et Automatique, Octobre 1998.

[BD96] A. Bobick et J. Davis. « An Appearance–Based Representation of Action ». Dans *Proceedings of the 13th International Conference on Pattern Recognition (ICPR´96)*, volume I, pages 307–312, Vienna, Austria, Août 1996.

[BDI97] A. Bobick, J. Davis, et S. Intille. « The KidsRoom: An Example Application Using a Deep Perceptual Interface ». Dans M. Turk, éditeur, *Proceedings of Workshop on Perceptual User Interfaces (PUI´97)*, pages 1–4, Banff, Alberta, Canada, Octobre 1997.

[BGG+99] A. Braffort, R. Gherbi, S. Gibet, J. Richardson, et D. Teil, éditeurs. *Gesture–Based Communication in Human–Computer Interaction*, numéro 1739 dans Lecture Notes in Artificial Intelligent, Gif–sur–Yvette, France, Mars 1999.

[BHK96] P. N. Belhumeur, J. P. Hespanha, et D. J. Kriegman. « Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection ». Dans Springer Verlag, éditeur, *Proceedings of the 4th European Conference on Computer Vision (ECCV´96)*, numéro 1064 dans Lecture Notes in Computer Science, Cambridge, UK, Avril 1996.

[BHK97] P. N. Belhumeur, J. P. Hespanha, et D. J. Kriegman. « Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, Juillet 1997.

[BI98a] A. Blake et M. Isard. *Active Contours*. Springer Verlag, 1998.

[BI98b] A. Blake et M. Isard. « A mixed–state `Condensation` tracker with automatic model–switching ». Dans *Proceedings of the IEEE International Conference on Computer Vision (ICCV´98)*, pages 107–112, 1998.

[Bic95] M. Bichsel, éditeur. *Proceedings of the International Workshop on Automatic Face and Gesture Recognition (IWAFGR´95)*, Zurich, Switzerland, Juin 1995.

[BID+] A. Bobick, S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schütte, et A. Wilson. « KidsRoom ». `http://vismod.www.media.mit.edu/vismod/demos/kidsroom/kidsroom.html`.

[BID+97] A. Bobick, S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schütte, et A. Wilson. « The KidsRoom: A Perceptually–Based Interactive and Immersive Story Evironment ». Rapport Technique 398, Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, Septembre 1997.

[BJ96a] M. Black et A. Jepson. « EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View–Based Representation ». Rapport Technique RBCV–TR–96–50, Department of Computer Science, University of Toronto, Octobre 1996.

[BJ96b] M. J. Black et A. D. Jepson. « EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View–Based Representation ». Dans *Proceedings of the 4th European Conference on Computer Vision (ECCV´96)*, numéro 1064 dans Lecture Notes in Computer Science, pages 329–342, Cambridge, UK, Avril 1996. Springer Verlag.

[BJ98] M. J. Black et A. D Jepson. « Recognizing Temporal Trajectories using the Condensation Algorithm ». Dans *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition (FG´98)*, pages 16–21, Nara, Japan, Avril 1998.

[BL93] Y. Bar–Shalom et X-R. Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, 1993.

[BM96] H. Birk et T. B. Moeslund. « Recognizing Gestures From the Hand Alphabet Using Principal Component Analysis ». Rapport de DEA, Laboratory of Image Analysis, Aalborg University, Denmark, Octobre 1996.

[Bob96] A. F. Bobick. « Computers Seeing Action ». Rapport Technique 394, Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, Septembre 1996.

[Bol80] R. A. Bolt. « Put-That-There: Voice and Gesture at the Graphics Interface ». *ACM Computer Graphics*, 14(3):262–270, 1980.

[Bér94] F. Bérard. « Vision par Ordinateur pour la Réalité Augmentée : Application au Bureau Numérique ». Dea, Université Joseph Fourier, Grenoble, France — Institut National Polytechnique de Grenoble, Juin 1994.

[Bér99a] F. Bérard. « The Perceptual Window: Head Motion as a New Input Stream ». Dans *IFIP Conference on Human Computer Interaction (INTERACT´99)*, pages 238–244, 1999.

[Bér99b] F. Bérard. « The Perceptual Window Movies ». `http://iihm.imag.fr/demos/pwindow/`, 1999. Démonstration en ligne.

[Bér00] F. Bérard. « *Vision par Ordinateur pour l´iteraction homme–machine fortement couplée* ». Thèse de doctorat , Université Joseph Fourier, Grenoble, France, Janvier 2000.

[Bra93] Software Technology Branch. Clips Reference Manual. Lyndon B. Johnson Space Center, Juin 1993.

[Bra96] A. Braffort. « *Reconnaissance et Compréhension de gestes, application à la langue des signes* ». Thèse de doctorat , Université Paris–XI Orsay, Juin 1996.

[BT97] A. Black et P. Taylor. « Festival Speech Synthesis System: system documentation (1.1.1) ». Rapport Technique HCRC/TR–83, Human Communication Research Center, University of Edinburgh, 1997.

[BY92] A. Blake et A. Yuille, éditeurs. *Active Vision.* MIT Press, 1992.

[Cad94] C. Cadoz. « Le Geste Canal de Communication Homme/Machine. La Communication « Instrumentale » ». *Techniques et Sciences Informatiques*, 13(1):31–61, 1994.

[Cae91] J. Caelen. « Interaction multimodale dans ICPDraw ». Rapport Technique, ICP — Institut National Polytechnique de Grenoble, 1991.

[CB94] J. L. Crowley et J.-M. Bedrune. « Integration and Control of Reactive Visual Processes ». Dans J.-O. Eklundh, éditeur, *Proceedings of the 3th European Conference on Computer Vision (ECCV´94)*, volume 801 de *Lecture Notes in Computer Science*, Stockholm, Suède, Mai 1994. Springer Verlag.

[CB96] U. M. Chan von Seelen et R. Bajcsy. « Adaptative correlation tracking of targets with changing scale ». Rapport Technique 405, Departement of Computer and Information Science, University of Pennsylvania, Juin 1996.

[CB97] J. L. Crowley et F. Bérard. « Multi–Modal Tracking of Faces for Video Communications ». Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR´97)*, pages 640–645, San Juan, Puerto Rico, Juin 1997.

[CBA+96] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobich, et A. Pentland. « Invariant Features for 3–D Gesture Recognition ». Dans Irfan Essa, éditeur, *Proceedings of the Second IEEE International Conference on*

Automatic Face and Gesture Recognition (FG´96), pages 157–162, Kilington, Vermont, USA, Octobre 1996. Proceedings of IEEE.

[CC92] G. Celeux et J. Clairambault. « Estimation de chaînes de Markov cachées : méthodes et problèmes ». Dans *Journées thématiques* gdr *Traitement du Signal et Images : Approches Markoviennes en Signal et Images*, pages 5–19, Paris, France, Septembre 1992.

[CC95] J. L. Crowley et H. I. Christensen, éditeurs. *Vision As Process*. ESPRIT Basic Research Series. Springer Verlag, 1995.

[CC99] O. Chomat et J. L. Crowley. « Utilisation de Champs Receptifs Spatio-Temporels pour la Reconnaissance de l´Apparence Locale d´Activites ». Dans *Journée des Journées Francophones des Jeunes Chercheurs en Analyse d´Images et Perception Visuelle*, Aussois, France, Avril 1999.

[CD95] J. L. Crowley et C. Discours. « *Vision As Process* », Chapitre The Sava skeleton system, pages 23–45. ESPRIT Basic Research Series. Springer Verlag, 1995.

[Cho00] O. Chomat. « *Caractérisation d´éléments d´activités par la statistique conjointe de champs réceptifs* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, Juin 2000.

[CM97] J. L. Crowley et J. Martin. « Visual Processes for Tracking and Recognition of Hand Gestures ». Dans Matthew Turk, éditeur, *Proceedings of Workshop on Perceptual User Interfaces (PUI´97)*, Banff, Alberta, Canada, Octobre 1997.

[CMC00] O. Chomat, J. Martin, et J. L. Crowley. « A probabilistic Sensor for the perception and the recognition of activities ». Dans *Proceedings of the 7th European Conference on Computer Vision (ECCV´2000)*, Dublin, Ireland, Juin 2000.

[CNSD93] C. Cruz-Neira, D. J. Sandin, et T. A. DeFanti. « Surround–Screen Projection–Based Virtual Reality: The Design and Implementation of the CAVE ». Dans *SIGGRAPH*, pages 135–142, 1993.

[Coe97] M. Coen. « Building Brains for Rooms: Designing Dsitributed Software Agents ». Dans *Proceeding of the Ninth Conference on Innovative Applications of Artificial Intelligence (IAAI´97)*, 1997.

[Coe98a] M. Coen, éditeur. *Proceeding American Association for Artificial Intelligence 1998 Spring Symposium on Intelligent Environments*, numéro SS–98–02 dans

Rapport technique de Stanford University, Stanford University, California, Mars 1998. American Association for Artificial Intelligence.

[Coe98b] M. H. Coen. « Design Principals for Intelligent Environments ». Dans *Proceeding American Association for Artificial Intelligence 1998 Spring Symposium on Intelligent Environments*, Stanford, CA, USA, Mars 1998.

[Col96] V. Colin de Verdière. « Reconnaissance d´objets par leurs statistiques de couleurs ». Dea imagerie, vision et robotique, Institut National Polytechnique de Grenoble, GRAVIR – IMAG, Juin 1996.

[Col99] V. Colin de Verdière. « *Représentation et Reconnaissance d´Objets par Champs Réceptifs* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, Décembre 1999.

[CPW+99] M. Coen, B. Phillips, N. Warshawshy, L. Weisman, S. Peters, et P. Finin. « Meeting the Computational Needs of Intelligent Environments: The Metaglue Environment ». Dans *First International Workshop on Managing Interactions in Smart Environment (MANSE´99)*, pages 201–212, Trinity College, Dublin, Ireland, Décembre 1999.

[CT92] T. F. Cootes et C. J. Taylor. « Active Shape Models - ´Smart Snakes´ ». Dans David Hogg et Roger Boyle, éditeurs, *British Machine Vision Conference*, pages 266–275. Springer Verlag, 1992.

[CT98] R. Cutler et M. Turk. « View–Based Interpretation of Real–Time Optical Floaw fro Gesture–Recognition ». Dans *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition (FG´98)*, Nara, Japan, Avril 1998. IEEE Computer Society.

[CTCG92] T. F. Cootes, C. J. Taylor, D. H. Cooper, et J. Graham. « Training Models of Shape from Sets of Examples ». Dans David Hogg et Roger Boyle, éditeurs, *British Machine Vision Conference*, pages 9–18. Springer Verlag, 1992.

[Cux99] C. Cuxac. « French Sign Language: Proposition of a Structural Explanation by Iconicity ». Dans A. Braffort, R. Gherbi, S. Gibet, J. Richardson, et D. Teil, éditeurs, *Gesture–Based Communication in Human–Computer Interaction*, numéro 1739 dans Lecture Notes in Artificial Intelligent, pages 165–184, Gif–sur–Yvette, France, Mars 1999. Springer Verlag.

[CW96] Y. Cui et J. J. Weng. « Hand Sign Recognition from Intensity Image Sequences with Complex Backgrounds ». Dans Irfan Essa, éditeur, *Proceedings*

of the Second IEEE International Conference on Automatic Face and Gesture Recognition (FG´96), pages 259–264, Kilington, Vermont, USA, Octobre 1996. IEEE Computer Society.

[Dav96] J. W. Davis. « *Appearance–Based Motion Recognition of Human Actions* ». Thèse de doctorat , Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, Août 1996. Également référence comme rapport technique n° 387.

[DB97] J. W. Davis et A. F. Bobick. « The Representation and Recognition of Human Movement Using Temporal Templates ». Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR´97)*, pages 928–934, Puerto Rico, Juin 1997.

[DBM77] A. Dudani, K. J. Breeding, et R. B. McGhee. « Aircraft identification by moments invariants ». *IEEE Transations on Computers*, C(23):39–45, 1977.

[Dev98] V. Devin. « Techniques visuelles d´observation de gestes ergotiques ». dea Imagerie, Vision, Robotique, Institut National Polytechnique de Grenoble, Juin 1998.

[DH73] R. O. Duda et P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

[dM91] P. de Marconnay. « SAVOM : Système Artificielle de Vision et d´Observation des Mains ». Dea, Institut National Polytechnique de Grenoble, Juin 1991.

[DMV93] P. Dauchy, C. Mignot, et C. Valot. « Joint Speech and Gesture Analysis – Some Experimental Results on Multimodal Interface ». Rapport Technique CRIN 93–R–121, Centre de Recherche en Informatique de Nancy, Septembre 1993.

[DP92] T. J. Darrell et A. P. Pentland. « Recognition of Space–Time Gestures using a Distributed Representation ». Rapport Technique 197, Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, 1992.

[DP93] T. Darrell et A. Pentland. « Space–Time Gestures ». Dans *´Looking At People´ : Recognition and Interpretation of Human Action — 13th IJCAI*, Chambéry, France, 1993.

[dS43] A. de Saint–Exupéry. *Le Petit Prince*. Folio Junior, 1943.

[DS93] J. Davis et M. Shah. « Gesture Recognition ». Rapport Technique CS–TR–93–11, University of Central Florida, Orlando, 1993.

[DS94] J. Davis et M. Shah. « Recognizing Hand Gestures ». Dans J.-O. Eklundh, éditeur, *Proceedings of the 3th European Conference on Computer Vision (ECCV´94)*, volume 801 de *Lecture Notes in Computer Science*, pages 331–340, Stockholm, Sweden, Mai 1994. Springer Verlag.

[Dur99] J. B. Durand. « Reconnaissance Statistique de Trajectoires par Modèles de Markov Cachés ». Dea, École Nationale Supérieure d´Informatique et de Mathématiques Appliquées de Grenoble, Institut National Polytechnique de Grenoble, Juin 1999.

[Edw97] A. D. N. Edwards. « Progress in Sign Language ». Dans M. Fröhlich et I. Wachsmuth, éditeurs, *Gesture and Sign Language in Human–Computer Interaction*, numéro 1371 dans Lecture Notes in Artificial Intelligent, pages 13–21, Belefield, Germany, Octobre 1997. Springer Verlag.

[EF73] P. Ekman et W. V. Friesen. « Hand Movements ». *The Journal of Communication*, pages 353–374, Décembre 1973.

[Ess96] I. Essa, éditeur. *Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition (FG´96)*, Killington, Vermont, USA, Octobre 14–16 1996. Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, IEEE Computer Society.

[Etz93] O. Etzioni. « Intelligence without robots: A reply to Brooks ». *AI Magazine*, 14(4):7–13, 1993.

[EW97] D. Ergo et X. Wielemans. « Reconnaissance du Langage des Signes pour Malentendants par Extraction Automatique de Caractéristiques Discriminantes ». Grade d´ingénieur civil en informatique, Université Catholique de Louvain, Faculté des Sciences Appliquées, Département d´Ingénierie Informatique, 1997.

[FAB+98] W. T Freeman, D. B. Anderson, P. A. Beardsley, C. N. Dodge, M. Roth, C. D. Weissman, W. S. Yerazunis, H. Kage, K. Kyuma, Y. Miyake, et K Tanaka. « Computer Vision for Interactive Computer Graphics ». *IEEE Computer Graphics and Applications*, 18:42–53, Mai 1998.

[Fau93] O. D. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Boston, 1993.

[Fey87] P. Feyereisen. « Gestures and Speech Interaction and Separations : A Reply To McNeil ». *Psychological Review*, 94(4):493–498, Avril 1987.

[FH93] S. S. Fels et G. E. Hinton. « Glove–Talk: a Neural Network Interface between a Data–Glove and a Speech Synthesizer ». IEEE *Transactions on Neural Networks*, 4:2–8, Janvier 1993.

[Fit53] P. M. Fitts. « The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement ». *Journal of Experimental Psychology*, 47(6):381–391, 1953.

[FMS93] S. Feiner, B. MacIntyre, et D. Seligmann. « Knowledge–Based Augmented Reality ». *Communications of the ACM*, 36(7):53–62, Juillet 1993.

[FR95] W. T. Freeman et M. Roth. « Orientation Histograms for Hand Gesture Recognition ». Dans *Proceedings of the International Workshop on Automatic Face and Gesture Recognition (IWAFGR´95)*, pages 296–301, Zurich, Switzerland, 1995.

[FTOK96] W. T. Freeman, K. Tanaka, J. Ohta, et K. Kyuma. « Computer Vision for Computer Games ». Dans Irfan Essa, éditeur, *Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition (FG´96)*, pages 100–105, Kilington, Vermont, USA, Octobre 1996. IEEE Computer Society.

[FW95] W. T. Freeman et C. D Weisman. « Television Control by Hand Gestures ». Dans *Proceedings of the International Workshop on Automatic Face and Gesture Recognition (IWAFGR´95)*, Zurich, Switzerland, Juin 1995.

[Gon97] S. Gong. Présentation, Réunion SMART, Décembre 1997.

[Gua98] A. Guarda. « *Apprentissage génétique de règles de reconnaissance visuelle. Application à la reconnaissance d´éléments du visage* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, 1998.

[Hac98] Hachette. « Encyclopédie Multimédia Hachette », 1998.

[HAJ90] X. D. Huang, Y. Ariki, et M. A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[Hal98] D. Hall. « Recognition of Facial Expressions with Principle Components Analysis ». Rapport de DEA, Universität Karlsruhe — Institut National Polytechnique de Grenoble, 1998.

[Hau89] A. G. Hauptmann. « Speech and Gestures for Graphic Image Manipulation ». Dans *ACM Conference on Human Factors in Computing Systems*, pages 241–245, Mai 1989.

[HB98] A. J. Howell et H. Buxton. « Towards Visually Mediated Interaction using Appearance–BasedModels ». Dans *Proceedings ECCV´98 Workshop on Perception of Human Action*, University of Freiburg, Germany, Juin 1998.

[HB99] A. J. Howell et H. Buxton. « Gesture Recognition for Visually Mediated Interaction ». Dans A. Braffort, R. Gherbi, S. Gibet, J. Richardson, et D. Teil, éditeurs, *Gesture–Based Communication in Human–Computer Interaction*, numéro 1739 dans Lecture Notes in Artificial Intelligent, Gif–sur–Yvette, France, Mars 1999. Springer Verlag.

[HE96] P. A. Harling et A. D. N. Edwards. « Hand Tension as a Gesture Segmentation Cue ». Dans P. A. Harling et A. D. N. Edwards, éditeurs, *Proceeding of Gesture Workshop (GW´96)*, pages 75–88. Springer, 1996.

[Hee88] D. J. Heeger. « Optical Flow Using Spatio–Temporal Filters ». *International Journal of Computer Vision*, pages 279–302, 1988.

[HH96a] T. Heap et D. Hogg. « 3D Deformable Hand Models », 1996.

[HH96b] T. Heap et D. Hogg. « Towards 3D Hand Tracking using a Deformable Model ». Dans Irfan Essa, éditeur, *Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition (FG´96)*, pages 140–145, Killington, Vermont, USA, 14–16 Octobre 1996. IEEE Computer Society.

[HHH97] F. G. Hofmann, P. Heyer, et G. Hommel. « Velocity Profile Based Recognition of Dynamic Gestures with Discrete Hidden Markov Models ». Dans M. Fröhlich et I. Wachsmuth, éditeurs, *Gesture and Sign Language in Human–Computer Interaction*, numéro 1371 dans Lecture Notes in Artificial Intelligent, pages 81–96. University of Bielefeld, Springer Verlag, Octobre 1997.

[HM83] R. Horaud et O. Monga. *Vision par ordinateur, outils fondamentaix*. Traité des Nouvelles Technologies, série Informatique. Hermes, Paris, 1883.

[HMS88] A. G. Hauptmann, P. McAvinney, et S. R. Shepard. « Gesture Analysis for Graphic Manipulation ». Rapport Technique CMU–CS–88–198, Carnegie Mellon University, Pittsburgh, USA, Novembre 1988.

[Hol97] E-J. Holden. « *Visual Recognition of Hand Motion* ». Thèse de doctorat , University of Western Australia, Janvier 1997.

[HP95] T. S. Huand et V. I. Pavlović. « Hand Gesture Modelling, Analysis and Synthesis ». Dans *Proceedings of the International Workshop on Automatic Face and Gesture Recognition (IWAFGR´95)*, pages 73–79, Zurich, Juin 1995.

[Hu62] M.-K. Hu. « Visual Pattern Recognition by Moment Invariants ». *IRE Transaction on Information Theory*, IT–8:179–187, 1962.

[IB96] M. Isard et A. Blake. « Contour Tracking bu Stochastic Propagation of Conditional Density ». Dans *Proceedings of the 4th European Conference on Computer Vision (ECCV´96)*, numéro 1064 dans Lecture Notes in Computer Science, pages 343–356, Cambridge, UK, Avril 1996. Springer Verlag.

[INR00] INRIA Rhône–Alpes. *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG´2000)*, Grenoble, France, Mars 2000. IEEE Computer Society.

[IWYY96] Y. Iwai, K. Watanabe, Y. Yagi, et M. Yachida. « Gesture Recogntion Using Colored Gloves ». Dans *13th International Conference on Pattern Recognition (ICPR´96)*, volume 1, pages 662–666, Vienna, Austria, Août 1996.

[Jon97] S. D. Jones. « *Robust Task Achievement* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, Mai 1997.

[Kal60] R. Kalman. « A new approach to linear filtering and prediction problems ». *Transaction of the ASME — Journal of Basic Engineering*, 82:35–45, Mars 1960.

[Koh96] M. Kohler. « Vision Based Remote Control in Intelligent Home Environments ». Dans B. Girod, H. Niemann, et H.-P. Seidel, éditeurs, *3D Image Analysis and Synthesis´96*, pages 147–154, University of Erlangen-Nuremberg/Germany, Novembre 1996. Infix–Verlag.

[Koh97] M. Kohler. « Technical Details and Ergonomical Aspects of Gesture Recognition applied in Intelligent Home Environments ». Rapport Technique 638, Informatik VII, Universität Dortmund, Germany, Janvier 1997.

[Kre93] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 7th édition, 1993.

[Kru] P. Kruizinga. « Face Recognition Homepage ». `http://www.cs.rug.nl/~peterkr/FACE/face.html`.

[Kru91] M. W. Krueger. *Artificial Reality II*. Addison–Wesley Publishing, 1991.

[KS90] M. Kirby et L. Sirovich. « Application of the Karhumen–Loève Procedure for the Characterization of Human Faces ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, Janvier 1990.

[KS98] M. Kohler et S. Schröter. « A Survey of Video-based Gesture Recognition — Stereo and Mono Systems — ». Rapport Technique 693, Informatik VII, Universität Dortmund, Germany, Août 1998.

[Le] C. Le Gal. Communications personnelles.

[LM99] C. Le Gal et J. Martin. « GesTris et GesBoing : jeux commandés par geste ». Prototype de démonstration, Salon TEC, 1999. Octobre.

[LMD99] C. Le Gal, J. Martin, et G. Durand. « SmartOffice : An intelligent and Interactif Environment ». Dans *First International Workshop on Managing Interactions in Smart Environment (MANSE´99)*, Trinity College, Dublin, Ireland, Décembre 1999.

[LP99] D. M. Lyons et D. L. Pelletier. « A Line–Scan Computer Vision Algorithm for Indetifying Human Body Features ». Dans A. Braffort, R. Gherbi, S. Gibet, J. Richardson, et D. Teil, éditeurs, *Gesture–Based Communication in Human–Computer Interaction*, numéro 1739 dans Lecture Notes in Artificial Intelligent, Gif–sur–Yvette, France, Mars 1999. Springer Verlag.

[LX96] C. Lee et Y. Xu. « Online, Interactive Learning of Gestures for Human/Robot Interfaces ». Dans *Internationnal Conference on Robotics Applications (ICRA´96)*, 1996.

[LZ97] A. Lux et B. Zoppis. « An Experimental Multi-language Environment for the Development of Intelligent Robot Systems ». Dans *Proceedings of the 5th International Symposium on Intelligent Robotic Systems (SIRS´97)*, pages 169–174, Royal Institue of Technology, Stockholm, Sweden, 1997. Informations complémentaires à `http://www-prima.imag.fr/Ravi/`.

[LZG98] M. Lucente, G.-J. Zwart, et A. George. « Visualization Space: A Testbed for Deviceless Multimodal User Interface ». Dans *Proceeding American Association for Artificial Intelligence 1998 Spring Symposium on Intelligent Environments*, pages 87–92, Stanford University, CA, USA, Mars 1998.

[LZPL] A. Lux, B. Zoppis, C. Poizat, et C. Le Gal. « The Ravi Homepage ». `http://www-prima.imag.fr/Ravi/`.

[Mac96] W. E. Mackay. « Réalité Augmentée : Le Meilleur des Deux Mondes ». *La Recherche*, 285:32–37, Mars 1996.

[Mac98] W. E. MACKAY. « Augmented Reality: Linking Real and Virtual Worlds – A New Paradigm for Interacting with Computers ». Dans *Proceeding of ACM Conference on Advanced Visual Interfaces*, 1998.

[Mar82] D. MARR. *Vision.* W. H. Freeman, San Francisco, USA, 1982.

[Mar94] J. MARTIN. « Techniques Visuelles de Détection et de Suivi de Mouvements ». Magistère informatique, Université Joseph Fourier, Septembre 1994.

[Mar95a] J. MARTIN. « Interprétation de Gestes par Modèle de Distribution de Points ». Magistère informatique, Université Joseph Fourier, Septembre 1995.

[Mar95b] J. MARTIN. « Suivi et Interprétation de Geste: Application de la Vision par Ordinateur à l´Interaction Homme–Machine ». DEA, Université Joseph Fourier – Institut National Polytechnique de Grenoble, 1995.

[Mar00] J. MARTIN. « Introduction à la théorie des modèles de Markov cachés ». Rapport Technique, PRIMA—IMAG, INRIA Rhône–Alpes, 655 Av. de l´Europe, 38330 Montbonnot Saint Martin, FRANCE, 2000.

[Mas98] K. MASE, éditeur. *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition (FG´98)*, Nara, Japan, Avril 1998. ATR, IEEE Computer Society.

[MC95] J. MARTIN et J. L. CROWLEY. « Comparison of Correlation Techniques ». Dans U. Rembold et AL., éditeur, *Intelligent Autonomous Systems – IAS–4*, pages 86–93, Karlsruhe, Germany, 27–30 Mars 1995.

[MC97] J. MARTIN et J. L. CROWLEY. « An Appearance–Based Approach to Gesture–Recognition ». Dans A. Del BIMBO, éditeur, *Proceddings of the 9th International Conference on Image Analysis and Processing (ICIAP´97)*, numéro 1311 dans Lecture Notes in Computer Science, Florence, Italia, Septembre 1997. Springer Verlag.

[McN85] D. MCNEIL. « So You Think Gesture Are Nonverbal? ». *Psychological Review*, 92:350–371, 1985.

[McN87] D. MCNEIL. « So You Do Think Gesture Are Nonverbal! ». *Psychological Review*, 94(4):499–504, Avril 1987.

[MD00] J. MARTIN et J. B. DURAND. « Automatic Handwriting Gestures Recognition using Hidden Markov Models ». Dans *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG´2000)*, pages 403–409, Grenoble, France, Mars 2000. INRIA Rhône–Alpes, IEEE Press.

[MDC98] J. Martin, V. Devin, et J. L. Crowley. « Active Hand Tracking ». Dans *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition (FG´98)*, Nara, Japan, Avril 1998. IEEE Computer Society.

[MFFM98] W. E. Mackay, A.-L. Fayard, L. Frobert, et L. Médini. « Reinventing the Familiar: Exploring au Augmented Reality Design Space for Air Traffic Control ». Dans *Proceedings of Conference on Human Factors in Computing Systems (CHI´98)*, Los Angeles, USA, 1998.

[MHC98] J. Martin, D. Hall, et J. L. Crowley. « Statistical Recognition of Parameter Trajectories for Hand Gestures and Face Expressions ». Dans *Proceedings ECCV´98 Workshop on Perception of Human Action*, Freiburg, Germany, Juin 1998.

[MHC99] J. Martin, D. Hall, et J. L. Crowley. « Statistical Gesture Recognition through Modelling of Parameter Trajectories ». Dans *Gesture Workshop (GW´99)*, Lecture Notes in Artificial Intelligent, Gif–sur–Yvette, France, Mars 1999. Springer Verlag.

[MHO91] H. Morita, S. Hashimoto, et S. Ohteru. « A Computer Music System that Follows Human Conductor ». *IEEE Computers*, pages 44–53, Juillet 1991.

[Mig95] C. Mignot. « *Usage de la parole et du geste dans les interfaces multimodales – Étude expérimentale et modélisation* ». Thèse de doctorat , Universite Henri Poincarré – Nancy I, 1995.

[ML99] J. Martin et C. Le Gal. « `GesTris` et `GesBoing` : jeux commandés par geste ». Prototype de démonstration, Salon TEC : Vidéo, 1999. Octobre.

[Moz] M. Mozer. « The Neural Network House ». `http://www.cs.colorado.edu/~mozer/nnh/index.html`.

[Moz98] M. Mozer. « The Neural Network House: An Environment that Adapts to its Inhabitants ». Dans M. Coen, éditeur, *Proceeding American Association for Artificial Intelligence 1998 Spring Symposium on Intelligent Environments*, pages 110–114, Stanford University, California, USA, Mars 1998. AAAI Symposium, American Association for Artificial Intelligence. Référencé comme rapport technique SS–98–02.

[MP94] W. E. Mackay et D. S Pagani. « Video Mosaic: Laying Out Time in a Physical Space ». Dans *Proceeding of Multimedia´94*, San Francisco, USA, 1994.

[MP95a] B. MOGHADDAM et A. PENTLAND. « Probabilistic Visual Learning for Object Detection ». Dans *Proceedings of the 5th IEEE International Conference on Computer Vision (ICCV´95)*, Cambridge, UK, Juin 1995.

[MP95b] B. MOGHADDAM et A. PENTLAND. « A Subspace Method for Maximum Likelihood Target Detection ». Dans *IEEE International Conference on Image Processing*, Washington D.C., USA, Octobre 1995.

[MS99] E. Granum M. STÖRRING, H. J. Andersen. « Skin colour detection under changing lighting conditions ». Dans H. ARÁUJO et J. DIAS, éditeurs, *Proceedings of the 7th International Symposium on Intelligent Robotic Systems (SIRS´99)*, pages 187–195, The University of Coimbra, Portugal, Juillet 1999.

[MT91] K. MURAKAMI et H. TAGUCHI. « Gestures Recognition Using Recurrent Neural Networks ». Dans *Proceedings of Conference on Human Factors in Computing Systems (CHI´91)*, pages 237–241, 1991.

[MYD96] C. MORIMOTO, Y. YACOOB, et L. DAVIS. « Recognition of Head Gestures Using Hidden Markov Models ». Dans *Proceedings of the 13th International Conference on Pattern Recognition (ICPR´96)*, volume III, pages 461–465, Vienna, Austria, Août 1996.

[NDL98] P. NIXON, S. DOBSON, et G. LACEY. « Smart Environments: some challenges for the computing community ». Dans P. NIXON, S. DOBSON, et G. LACEY, éditeurs, *First International Workshop on Managing Interactions in Smart Environment (MANSE´99)*, pages 1–4, Dublin, Irelande, Décembre 1998. Department of Computer Science, University of Dublin, Trinity College, Springer Verlag.

[NLD99] P. NIXON, G. LACEY, et S. DOBSON, éditeurs. *First International Workshop on Managing Interactions in Smart Environment (MANSE´99)*. Department of Computer Science, University of Dublin, Trinity College, Springer Verlag, Décembre 1999.

[NNM96] S. A. NENE, S. K. NAYAR, et H. MURASE. « Columbia Object Image Library (COIL–100) ». Rapport Technique, Department of Computer Science, Columbia University, 1996.

[NR97] C. NÖLKER et H. RITTER. « Detection of fingertips in human hand movement sequences ». Dans M. FRÖHLICH et I. WACHSMUTH, éditeurs, *Gesture and Sign Language in Human–Computer Interaction*, numéro 1371 dans Lecture Notes in Artificial Intelligent, Belefield, Germany, Septembre 1997. University of Bielefeld, Springer Verlag.

[NR99] C. Nölker et H. Ritter. « GREFIT: Visual Recognition of Hand Postures ». Dans A. Braffort, R. Gherbi, S. Gibet, J. Richardson, et D. Teil, éditeurs, *Gesture–Based Communication in Human–Computer Interaction*, numéro 1739 dans Lecture Notes in Artificial Intelligent, Gif–sur–Yvette, France, Mars 1999. Springer Verlag.

[NSMO94] K. Nirei, H. Saito, M. Mochimaru, et S. Ozawa. « Model–Based Hand Tracking Using Stochastic Optimization ». Référence inconnue, 1994.

[NSO96] S. Nagaya, S. Seki, et R. Oka. « A Theoritical Consideration of Pattern Space Trajectory for Gesture Spotting Recognition ». Dans *Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition (FG´96)*, pages 72–76, Killington, Vermont, USA, Octobre 1996. IEEE Computer Society Press.

[NW92] W. Newman et P. Wellner. « A Desk Supporting Computer–Based Interaction with Paper Document ». Dans *ACM Conference on Human Factors in Computing Systems*, Mai 1992.

[NWF86] R. Nag, K. H. Wong, et F. Fallsize. « Script Recognition Using Hidden Markov Models ». Dans *International Conference on Signal and Speech Processing (ICASSP´86)*, pages 2071–2074, 1986.

[Ohk95] M. Ohki. « The Sign Language Telephone ». Dans *Telecommunication Forum (Telecom´95)*, pages 391–395, 1995.

[OMG98] E.-J. Ong, S. McKenna, et S. Gong. « Tracking Head Pose for Inferring Intention ». Dans *Proceedings ECCV´98 Workshop on Perception of Human Action*, University of Freiburg, Germnay, Juin 1998.

[OTK93] T. Onishi, H. Takemura, et F. Kishino. « The Manipulation of Graphics Object by Using Hand Gestures ». Dans *Imagina´93*, pages 129–134, 1993.

[PC98] F. Pourraz et J. L. Crowley. « Use of Eigen Space Techniques for Position Estimation ». Dans *5th International Workshop on Advanced Motion Control (AMC´98)*, Coimbra, Portugal, Juin 1998.

[Pen96] A. Pentland. « Smart Rooms ». *Scientific American*, pages 54–62, Avril 1996.

[Pet] « Le Petit Nicolas en thèse ». `http://saturn.umh.ac.be/ maesa/nicolas.htm`. Dessins: J. J. Sempé, Formules : Y. Burgeaud, M. Mignotte et F. Normandin, Texte : G. Taviot.

[Pou98] F. Pourraz. « Estimation de position d´un robot mobile par projection dans un espace de composantes principales ». dea Imagerie, Vision et Robotique, Institut National Polytechnique de Grenoble, Juin 1998.

[PRP94] X. Pouteau, L. Romary, et J-M. Pierrel. « Voix, geste et multimodalité : quand dire c´est faire–faire ». Dans IDLS, éditeur, *Ergonomie et Informatique Avancée (Ergo–IA´94)*, pages 491–500, Biarritz, France, Mars 1994.

[PSH96] V. I. Pvalović, R. Sharma, et T. S. Huang. « Gesture Interface to a Visual Computing Environment for Molecular Biologist ». Dans *Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition (FG´96)*, pages 30–35, Killington, Vermont, USA, Octobre 1996. Proceedings of IEEE.

[PSH97] V. I. Pavlović, R. Sharma, et T. S. Huang. « Visual Interpretation of Hand Gestures for Human–Computer Interaction: A Review ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, Juillet 1997.

[PSOS98] I. Poddar, Y. Sethi, E. Ozyildiz, et R. Sharma. « Toward Natural Gesture/Speech HCI: A Case Study of Weather Narration ». Dans *Proceedings of Workshop on Perceptual User Interfaces (PUI´98)*, 1998.

[PTVF92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, et B. P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing.* Cambridge University Press, second edition édition, 1992.

[Que94] F. K. H. Quek. « Toward a Vision–Based Hand Gesture Interface ». Dans *Virtual Reality Software and Technology*, Août 1994.

[Rhe91] H. Rheingold. *Virtual Reality.* ???, 1991.

[Rig00] G. Rigoll. Hidden Markov Models in Computer Vision and Pattern Recognition. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG´2000) Tutorial, Grenoble, France, Mars 2000.

[RJ86] L. R. Rabiner et B. H. Juang. « An Introduction to Hidden Markov Models ». *IEEE ASSP Magazine*, pages 4–16, Janvier 1986.

[RK93] J. M. Rehg et T. Kanade. « DigitEyes:Vision-Based Human Hand Tracking ». Rapport Technique CMU-CS-93-220, Carnegie Mellon University, Pittsburgh, USA, 1993.

[Roh94] K. Rohr. « Towards Model–Based Recognition of Human Movements in Image Sequences ». *CVGIP: Image Understanding*, 59(1):94–115, Janvier 1994.

[Rub91] D. H. Rubine. « *The Automatic Recognition of Gestures* ». Thèse de doctorat , Carnegie Mellon University, Pittsburgh, USA, Décembre 1991.

[Rub92] D. Rubine. « Combining Gestures and Direct Manipulation ». Dans *Proceedings of Conference on Human Factors in Computing Systems (CHI´92)*, pages 659–660, 1992. (Vidéo).

[Sau] E. Saund. « Image Mosaicing and a Diagrammatic User Interface for an Office Whiteboard Scanner ». http://www.parc.xerox.com/spl/members/saund/.

[Sau97] E. Saund. « Machine Perception in Support of Instrumented Office Whiteboards ». Dans M. Turk, éditeur, *Proceedings of Workshop on Perceptual User Interfaces (PUI´97)*, pages 33–25, Banff, Alberta, Canada, Octobre 1997.

[SB91] M. J. Swain et D. H. Ballard. « Color Indexing ». *International Journal of Computer Vision*, 7(1):11–32, 1991.

[SC98] B. Schiele et J. L. Crowley. « Recognition without Correspondence using Multidimensional Receptive Field Histograms ». Dans *International Journal of Computer Vision*, 1998.

[Sch96] C. Schmid. « *Appariement d´images par invariants locaux de niveaux gris* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, 1996.

[Sch97] B. Schiele. « *Reconnaissance d´Objets utilisant des Histogrammes Multidimentsionnels de Champs Réceptifs* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, Juillet 1997.

[Sch00] K. Schwerdt. « *Computer Vision for Computer Assisted Video–Communication* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, 2000. A paraître.

[SHP+96] R. Sharma, T. S. Huang, V. I. Pavlović, Y. Zhao, Z. lo, S. Chu, K. Schlten, A. Dalke, J. Philips, M. Zeller, et W. Humphrey. « Speech/Gesture Interface to a Visual Computing Environment for Molecular Biologists ». Dans *Proceedings of the 13th International Conference on Pattern Recognition (ICPR´96)*, Août 1996.

[SK87] L. Sirovich et M. Kirby. « Low–Dimensional Procedure for the Characterization of Human Faces ». *Journal of Optical Society of America A*, 4(3):519–524, Mars 1987.

[SKB+98] S. Shafer, J. Krumm, B. Brumitt, B. Meyers, M. Czerwinski, et D. Robbins. « The New EasyLiving Project at Microsoft Research ». Dans *Joint DARPA/NIST SmartSpaces Workshop*, Gaithersburg, Maryland, USA, Juillet 1998.

[SP95] T. Starner et A. Pentland. « Real–Time Amercian Sign Language Recognition form Video Using Hidden Markov Model ». Rapport Technique 375, Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, 1995.

[SR96] Q. Stafford–Fraser et P. Robinson. « Brightboard: A Video–Augmented Environment ». Dans *Proceedings of Conference on Human Factors in Computing Systems (CHI´96)*, 1996.

[Sta95] T. E. Starner. « *Visual Recognition of American Sign Language Using Hidden Markov Model* ». Thèse de doctorat , Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, Février 1995.

[Sta96] Q. Stafford–Fraser. « *Video–Augmented Environments* ». Thèse de doctorat , Gonville & Caius College, University of Cambridge, Février 1996.

[Stu92] D. J. Sturman. « *Whole hand input* ». Thèse de doctorat , Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, 1992.

[SW95] B. Schiele et A. Waibel. « Estimation of the Head Orientation based on a Face–Color–Intensifier ». Dans *Proceedings of the 3rd International Symposium on Intelligent Robotic Systems (SIRS´95)*, 10–14 Juillet 1995.

[Tar98] L. Tarassenko. *A Guide To Neural Computing Applications*. Arnold, 1998.

[Tea80] M. R. Teague. « Image Analysis via the General Theory of Moments ». *Journal of Optical Society of America*, 70(8):920–930, Août 1980.

[TK95] C. P. Tung et A. C. Kak. « Automatic Learning of Assembly Tasks using a Dataglove System ». Dans *Proceedings of the IEEE/RSJ Conference on Intelligent Robots Systems*, volume 1, pages 1–8, 1995.

[TP90] M. Turk et A. Pentland. « Face Recognition without Features ». Dans *IAPR Workshop on Machine Vision Applications*, pages 267–270, Tockyo, Japan, Novembre 1990.

[TP91] M. Turk et A. Pentland. « Eigenfaces for Recognition ». *Journal of Neuroscience*, 3(1):71–86, 1991.

[VIS] VISIONICS Corp. « FaceIt ». `http://www.FaceIt.com`.

[VSC99] W. E. Vieux, K. Schwerdt, et J. L. Crowley. « Face–tracking and Coding for Video–Compression ». Dans *Proceeding of International Conference on Vision System (ICVS´99)*, Gran Canarai Espagne, Janvier 1999.

[Wal97] F. Wallner. « *Estimation de position d´un robot mobile par utilisation des composantes principales des données d´un capteur télémétrique laser* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, Octobre 1997.

[WB97] G. Welch et G. Bishop. « An Introduction to kalman Filter ». Rapport Technique 95—041, Departement of Computer Science, University of North California at Chapel Hill, 1997.

[WB98] A. D. Wilson et A. F. Bobick. « Recognition and Interpretation of Parametric Gesture ». Rapport Technique 421, Massachusset Institute of Technology, Media Laboratory, Perceptual Computing Section, 1998.

[WC96] J. J. Weng et Y. Cui. « Recognition of Hand Signs from Complex Backgrounds ». Dans R. Cipolla et A. Pentland, éditeurs, *Computer Vision for Human–Machine Interaction*, pages 235–266, Cambridge, UK, 1996. Cambridge University Press.

[Wei91] M. Weiser. « The Computer for the 21st Century ». *Scientific American*, pages 66–71, Septembre 1991.

[Wel91a] P. Wellner. « The DigitalDesk Calculator : Tactile Manipulation on a Desktop ». Dans *ACM Symposium on User Interface Software and Terchnology*, pages 27–33, Novembre 1991.

[Wel91b] P. Wellner. « Interacting with Paper on the DigitalDesk ». Vidéo, Juin 1991.

[Wel93a] P. Wellner. « Adaptive Thresholding for DigitalDesk ». Rapport Technique EPC–93–110, EuroPARC, Xerox Center, 1993.

[Wel93b] P. Wellner. « Interacting with Paper on the DigitalDesk ». Dans *Communications of the ACM*, Juillet 1993.

[Wex97] A. Wexelblat. « Research Challenges in Gesture: Open Issues and Unsolved Problems ». Dans M. Fröhlich et I. Wachsmuth, éditeurs, *Gesture and Sign Language in Human–Computer Interaction*, numéro 1371 dans Lecture Notes in Artificial Intelligent, pages 1–12, Belefield, Germany, Octobre 1997. University of Bielefeld, Springer Verlag.

[WMG93] P. Wellner, W. E. Mackay, et R. Gold, éditeurs. *Computer Augmented Environments: Back to the Real World.*, volume 36(7). Communications of the ACM, 1993. Special Issue.

[XTS98] Fact Sheet XRCE TeXnology Showroom. « LightWorks: A Digital Desk ». `http://www.xrce.xerox.com/showroom/techno/lightworks.html`, 1998.

[YI96] M. Yachida et Y. Iwai. « Looking at Human Gestures ». Dans R. Cipolla et A. Pentland, éditeurs, *Computer Vision for Human–Machine Interaction*, pages 291–311, Cambridge, UK, 1996. Cambridge University Press.

[YOI92] J. Yamato, J. Ohya, et K. Ishii. « Recognizing Human Action in Time–Sequential Images using Hidden Markov Model ». Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ´92)*, pages 379–385, 1992.

[You93] S. J. Young. *HTK: Hidden Markov Model Toolkit V1.5*. Entropic Research Laboratories Inc, Décembre 1993.

[YX94] J. Yang et Y. Xu. « Hidden Markov Model for Gesture Recognition ». Rapport Technique CMU–RI–TR–94–10, Carnegie Mellon University, Pittsburgh, USA, Mai 1994.

[Zop97] B. Zoppis. « *Outils pour l´Intégration et le Contrôle en Vision et Robotique Mobile* ». Thèse de doctorat , Institut National Polytechnique de Grenoble, Juin 1997.

Index des auteurs et applications cités

Résumé
Cette thèse se place dans le domaine de la reconnaissance de gestes dans le cadre d´interactions homme–machine. L´objectif est la conception de systèmes de reconnaissance et de compréhension adaptés au canal gestuel et son intégration dans de nouvelles interactions entre un utilisateur et un système informatique. Une revue de la communication gestuelle et de l´interaction homme–machine nous permet de nous interroger sur leurs applications pour une nouvelle interaction naturelle et la définition d´un geste du point de vue d´un concepteur d´interaction gestuelle. Nous définissons un geste par une trajectoire, c´est–à–dire une courbe paramètrée par le temps dans un espace de caractéristiques. Ces caractéristiques sont la position spatiale de la main et sa configuration.
Nous proposons de décomposer la reconnaissance en trois étapes : analyse, reconnaissance et interprétation. L´étape d´analyse calcule les caractéristiques de la main dans chaque image de la séquence, créant ainsi la trajectoire du geste. Son analyse spatio–temporelle lors de l´étape de reconnaissance permet de la classifier parmi l´ensemble des gestes connus, spécifiques à l´application. Enfin, l´étape d´interprétation effectue la correspondance entre le geste reconnu et l´action à réaliser. Cette étape est dépendante de l´application visée. Dans cette thèse, nous nous intéressons aux gestes réalisés dans le cadre d´un environnement intelligent. Nous considérons ainsi des gestes de manipulations d´objets en réalité augmentée et des gestes de dessins. Nous présentons enfin une application de reconnaissance d´activités humaines se basant sur le mouvement d´un individu dans cet environnement.

Mots–clés
Vision par ordinateur, communication homme–machine, interactions gestuelles, modèles statistiques de reconnaissance

---

TITLE

GESTURES RECOGNITION IN COMPUTER VISION

Abstract
This thesis lies in the field of gesture recognition in the context of Human Computer Interaction (HCI). The goal was the design of recognition and understanding systems dedicated to the gestual channel, and their integration into new forms of interaction between users and computerized systems. A review of gestual communication and the domain of HCI allows a reflection of their use for new, natural interaction and the definition of gestures from a designer´s point of view. We define a gesture by the concept of its trajectory, which is a parametrized curve over time in characteristics space. Characteristics of a hand are, e.g., its location and its posture.
We propose to decompose the recognition problem into three stages: analysis, recognition and interpretation. During the analysis stage we compute the characteristics of a hand for each image of a sequence and create the gesture´s trajectory. Spatio–temporal analysis during the recognition step then classifies the gesture between a set of known gestures specific to the application. The interpretation stage finally finds the correspondence between the recognized gesture and the action to be taken by the system. This stage is application dependent. In this thesis we focus on gestures in an intelligent environment, considering gestures for the manipulation of computerized objects and for drawing. We also present an application for human activities recognition based on the movement of individuals in that environment.

Key words
Computer vision, computer–human communication, gestural interactions, statistical models of recognition

---