Fusion de données audio-visuelles pour l'interaction Homme-Robot

Brice Burger

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Discipline ou spécialité :

Systèmes Embarqués

Présentée et soutenue par :
Brice Burger

le : 29 janvier 2010

Titre :

Fusion de données audio-visuelles pour l'interaction Homme-Robot

---

JURY
Isabelle FERRANÉ (MCF Univ. Toulouse, IRIT), Frédéric LERASLE (MCF Univ. Toulouse, LAAS)
Rapporteurs : Olivier BERNIER (MCF HDR, France Telecom R&D),
Laurent BESACIER (Pr Univ. Grenoble, CLIPS)
Examinateurs : Olivier COLOT (Pr Univ. Lille, LAGIS), Michel DEVY (DR LAAS),
Philippe JOLY (Pr Univ. Toulouse, IRIT), Patrick SAYD (Dr CEA Saclay)

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
IRIT et LAAS-CNRS

Directeur(s) de Thèse :

Isabelle Ferrané et Frédéric Lerasle

Rapporteurs :

Olivier Bernier et Laurent Besacier

MANUSCRIT DE THÈSE

Fusion de données audio-visuelles pour l'interaction Homme-Robot

Brice BURGER

Thèse de l'UPS effectuée du 01/09/06 au 31/12/09 en co-tutelle au LAAS-CNRS et à l'IRIT de Toulouse

Directeur de thèse LAAS : **Frédéric LERASLE**
Directeur de thèse IRIT : **Isabelle FERRANÉ**

Résumé

Dans le cadre de la robotique d'assistance, cette thèse a pour but de fusionner deux canaux d'informations (visuelles et auditives) dont peut disposer un robot afin de compléter et/ou confirmer les données qu'un seul canal aurait pu fournir, et ce, en vue d'une interaction avancée entre homme et robot. Pour ce faire, nos travaux proposent une interface perceptuelle pour l'interaction multimodale ayant vocation à interpréter conjointement parole et geste, notamment pour le traitement des références spatiales.

Nous décrivons dans un premier temps la composante parole de nos travaux qui consiste en un système embarqué de reconnaissance et d'interprétation de la parole continue. Nous détaillons ensuite la partie vision composée d'un traqueur visuel multi-cibles chargé du suivi en 3D de la tête et des deux mains, ainsi que d'un second traqueur chargé du suivi de l'orientation du visage. Ces derniers alimentent un système de reconnaissance de gestes par DBNs décrit par la suite. Nous poursuivons par la description d'un module chargé de la fusion des données issues de ces sources d'informations dans un cadre probabiliste. Enfin, nous démontrons l'intérêt et la faisabilité d'une telle interface multimodale à travers un certains nombre de démonstrations sur les robots du LAAS-CNRS. L'ensemble de ces travaux est fonctionnel en quasi-temps réel sur ces plateformes robotiques réelles.

Abstract

In the framework of assistance robotics, this PHD aims at merging two channels of information (visual and auditive) potentially available on a robot. The goal is to complete and/or confirm data that an only channel could have supplied in order to perform advanced interaction between a human and a robot. To do so, we propose a perceptual interface for multimodal interaction which goal is to interpret jointly speech and gesture, in particular for the use of spatial references.

In this thesis, we first describe the speech part of this work which consists in an embedded recognition and interpretation system for continuous speech. Then comes the vision part which is composed of a visual multi-target tracker that tracks, in 3D, the head and the two hands of a human in front of the robot, and a second tracker for the head orientation. The outputs of these trackers are used to feed the gesture recognition system described later. We continue with the description of a module dedicated to the fusion of the data outputs of these information sources in a probabilistic framework. Last, we demonstrate the interest and feasibility of such a multimodal interface through some demonstrations on the LAAS-CNRS robots. All the modules described in this thesis are working in quasi-real time on these real robotic platforms.

Avant-propos

Ce document présente le travail accompli au cours de la thèse que j'ai eu l'honneur d'effectuer en co-tutelle au LAAS-CNRS et l'IRIT de Toulouse, au sein du groupe RAP et de l'équipe SAMOVA, respectivement. Ce travail a été sanctionné par un doctorat de l'Université Paul Sabatier (UPS) de Toulouse.

Il convient tout d'abord de remercier mes laboratoires d'attache, LAAS-CNRS et IRIT, ainsi que leurs directeurs, Raja Chatila et Luis Fariñas del Cerro, d'avoir bien voulu m'accueillir dans leurs locaux et de m'avoir fourni les moyens humains et matériels indispensable pour mener à bien mes travaux. Je remercie de même les groupes, RAP et SAMOVA, et leurs directeurs, Michel Devy et Philippe Joly, pour m'avoir accueilli en leur sein.

Je remercie également Olivier Bernier et Laurent Besacier pour le courage nécessaire et le temps consacré à lire puis rapporter ce manuscrit, ainsi que pour leur participation à mon jury de thèse. Je remercie de même mes examinateurs, Olivier Colot, Michel Devy, Philippe Joly et Patrick Sayd, pour avoir accepté d'assister à ma soutenance et de faire partie de mon jury de thèse.

Il serait bien sûr injuste de ne pas remercier ici Matthieu Herrb, Sara Fleury et Antony Mallet, ingénieurs de recherche au LAAS durant ma thèse, sans l'aide desquels il aurait été difficile de décoller dans ce travail et sans qui les robots sur lesquels j'ai eu le plaisir de travailler ne seraient que des coquilles vides. De même, je ne peux pas oublier ici mes compagnons de thèse du LAAS comme de l'IRIT sans qui ces trois années auraient été bien longues, pour leur aide souvent indispensable ou tout simplement pour leur présence amicale : Akin, Alireza, Anh, Ariane, Aurélie, Benjamin, Christine, David, Élie, Eduardo, Hélène, Hervé, Jérôme, Julien, José, Luis, Mathias, Mathieu, Michel, Minh, Mokhtar, Oussama, Patrick, Philippe, Régine, Sébastien, Thierry, Xavier, thésards, post-docs et permanents, difficile d'être exhaustif, merci !

Je remercie également Michel Taïx sans le soutient duquel je n'aurais pu obtenir l'opportunité de cette thèse, ni même acquérir les bases qui m'ont permis de la mener à bien.

Je remercie enfin grandement, ceux sans qui rien n'aurait été possible. Mes directeurs de thèse tout d'abord, Isabelle Ferrané et Frédéric Lerasle, pour leur suivi, leurs encouragements, leurs conseils, leur patience et pour tout le savoir et savoir-faire qu'il m'ont transmis durant ces trois années. Mes parents et toute ma famille pour m'avoir fait tel que je suis, et bien entendu Claire qui va encore devoir me supporter de longues années.

Sommaire

Introduction générale

Les trois lois de la robotique :

1 . *Un robot ne peut porter atteinte à un être humain, ni, restant passif, permettre qu'un être humain soit exposé au danger.*

2 . *Un robot doit obéir aux ordres que lui donne un être humain, sauf si de tels ordres entrent en conflit avec la première loi.*

3 . *Un robot doit protéger son existence tant que cette protection n'entre pas en conflit avec la première ou la seconde loi.*

Isaac Asimov

À l'époque où Isaac Asimov invente le mot « robotique » cette dernière n'existe pas encore à proprement parler. Mais les développements de la mécanique, de l'automatisme, de l'électronique, puis de l'informatique ont permis de donner une forme bien réelle à un nombre grandissant des idées de l'écrivain. L'histoire de la robotique réelle commence avec les robots industriels qui permettent d'accélérer les cadences de production des usines et d'éviter aux ouvriers les tâches les plus répétitives, pénibles ou dangereuses. Elle se poursuit aujourd'hui par la robotique d'assistance qui est devenue un domaine de recherche extrêmement riche et donne peu à peu lieu à des applications réelles voire commercialisables.

Les robots personnels tel qu'évoqués par Asimov dans ses romans peuvent être divisés en deux catégories dont le dénominateur commun est d'être des machines au service de leur propriétaire respectant les trois lois citées plus haut. La première catégorie concerne les *robots d'assistance* à proprement parler. Il s'agit de robots, chargés d'assurer des tâches ménagères (telles que le ménage ou la cuisine), souvent nombreux et spécialisés. La seconde catégorie est composée de *robots personnels* bien plus complexes puisque doués de réflexion. Leur rôle est alors celui d'un assistant presque humain. Dans les deux cas, les robots doivent être mobiles et autonomes afin de s'acquitter de leurs tâches, c'est-à-dire être capables de se mouvoir dans leur environnement sans aucune intervention extérieure. La première loi impose de plus qu'ils soient dotés de fonctions de perception de l'homme, fonctions sans lesquelles il est impossible d'assurer la sécurité de ce dernier. La seconde loi entraîne également le besoin d'une perception de l'homme, mais aussi plus largement, de la capacité à interagir avec lui à différents niveaux.

La première catégorie est l'objectif affiché de nombre de recherches et projets actuels et passés. Dans une terminologie plus scientifique, on parle de « robots assistants » ou auxiliaires

de services. Leurs capacités sont définies *a priori* et liées au service à assurer dans certains lieux publics. Différentes applications commerciales ont été développées ces dernières années, telles des machines de service capables d'aspirer les pièces d'un appartement ou de tondre le gazon de manière quasi-automatique. Mais celles-ci restent extrêmement basiques puisque douées d'une autonomie toute relative et surtout n'ont qu'une conception très simpliste de leur environnement physique et souvent aucune prise en compte de leur environnement humain. Dans un cadre académique, des robots plus avancés émergent à travers divers projets, tels COMMROB ou AMORCES, et certains ont donné lieu à des applications telles que des robots guide de musée ou des trolley intelligents.

La seconde catégorie définie précédemment fait d'avantage penser aux recherches sur les robots dits cognitifs, c'est-à-dire capables d'apprendre et donc de s'adapter. Dans la terminologie scientifique, on parle plus généralement de robots personnels ou compagnons, qui sont assimilables à des robots assistants de seconde génération, destinés à des interactions avec l'homme et des tâches plus personnelles. À l'instar des ordinateurs personnels, la finalité des robots personnels est d'acquérir de nouvelles capacités et connaissances à l'aide d'un apprentissage ouvert et actif et d'évoluer en constante interaction et coopération avec l'homme. Cette dernière catégorie devient également un champ de recherche actif et a notamment fait l'objet du projet européen COGNIRON dans lequel le LAAS-CNRS est impliqué.

1 Contexte et objectifs de nos travaux

1.1 Notre contexte robotique

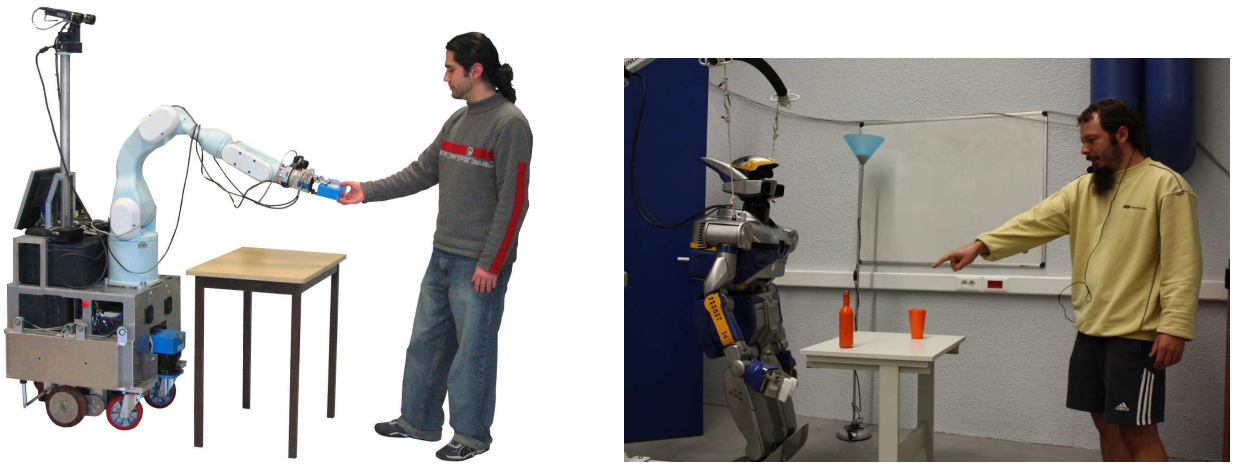


FIG. 1: Nos robots *Jido* (à gauche) et *HRP-2* (à droite) lors d'une interaction.

Les robots du LAAS-CNRS sont pour la plupart des robots mobiles que l'on cherche à doter d'un maximum d'autonomie. Des robots tels que *Jido* ou *HRP-2* (figure 1) ont été équipés pour la navigation en environnement intérieur ainsi que pour la manipulation d'objets. Ces robots évoluent dans le cadre de la robotique d'assistance, c'est-à-dire que leurs applications potentielles consistent à être capables d'aider un être humain, par exemple une personne à mobilité réduite (personne âgée, handicapé moteur), dans ses tâches quotidiennes. On cherche, par conséquent, à les doter de capacités leur permettant d'interagir avec l'homme et son environnement, et ce, de la manière la plus autonome et naturelle possible.

1.2 Objectifs de nos travaux

Dans ce contexte, nous nous intéressons à l'interaction multimodale homme-robot. Il s'agit de permettre à l'homme d'interagir avec un robot grâce à différentes modalités, et ce, en cherchant à s'inspirer de la communication homme-homme. En effet, lors d'une conversation entre interlocuteurs humains, les omissions, abréviations ou sous-entendus sont des phénomènes extrêmement fréquents et naturels, la plupart du temps pour la simple raison qu'ils permettent d'alléger le discours. Pour se comprendre, deux personnes doivent par conséquent être capable d'interpréter ce langage dit naturel, mais également de le compléter au besoin par des informations non-verbales, notamment des expressions gestuelles. C'est vers une telle perception de l'homme par le robot que nous souhaitons tendre dans nos travaux.

La fusion de données audio-visuelles prend tout son sens dans un tel contexte. En effet, les principales modalités de communication utilisées par l'homme sont la parole et le geste. La compréhension de l'homme par le robot lors d'une interaction avec ce dernier passe donc par une interprétation et une mise en commun de ces deux types d'informations, celles-ci pouvant être acquises par les capteurs audio et vidéo dont sont équipés nos robots. Ces deux canaux (audio et vidéo) donnent des informations qui peuvent se révéler, suivant le cas, complémentaires ou redondantes. Il s'agira alors, dans le premier cas, de déterminer les informations incomplètes d'un canal grâce à celles de l'autre. Par exemple, dans le cas d'ordres du type « viens vers moi » ou « donnes-moi cet objet », la perception de l'homme, la détermination de sa position ou la détection d'un geste déictique (voir chapitre III pour une description de la catégorisation des gestes) permettent la construction d'un ordre complet exécutable par le robot. Dans le second cas, qui s'applique par exemple lorsque l'utilisateur du robot lui dit « bonjour » tout en le saluant de la main, il s'agit plutôt de renforcer l'information et de rendre le système plus robuste.

Il est à noter que, quel que soit le geste destiné au robot, mais en particulier pour des gestes de pointage, exploiter l'information apportée par la direction du regard peut être d'une grande importance. En effet, l'homme a naturellement tendance à regarder dans la direction de ce qui l'intéresse, typiquement son interlocuteur, l'objet sur lequel il se focalise ou un endroit qu'il souhaite désigner.

1.3 Contraintes liées à notre application

L'objectif final de nos travaux étant l'intégration complète d'une telle interface multimodale sur les plateformes robotiques, il est important de mentionner ici les contraintes auxquelles nous devons faire face dans ce cadre applicatif.

Les premières sont communes à l'ensemble des travaux appliqués à la robotique. En effet, nos robots devant être autonomes et par conséquent ne pas faire appel à des calculateurs extérieurs, les ressources (processeur et mémoire) sont entièrement embarquées et donc limitées. De plus, ces mêmes ressources doivent être partagées entre l'ensemble des modules devant fonctionner en parallèle sur le robot (navigation, manipulation, etc, voir figure IV.3 pour une idée du nombre de modules nécessaires au fonctionnement d'un robot). Ces limites sont d'autant plus contraignantes que la réactivité du robot est importante pour son acceptabilité par l'homme. Un utilisateur pourra en effet être irrité si le temps de réaction d'un robot dépasse trop fortement des normes sociales. Dans ce contexte, les temps de calculs des algorithmes utilisés sont par conséquent d'une importance fondamentale. Ces limitations démarquent notamment les travaux du domaine de l'interaction homme-robot (*IHR*) de ceux menés dans le domaine de l'interaction homme-machine (*IHM*).

Le second type de limitation est lié aux capteurs (caméras et microphone) dont nous avons besoin pour atteindre nos objectifs. En effet, et contrairement aux applications IHM, le cadre de la robotique mobile sous-entend un environnent non contrôlé, ce qui engendre une variabilité importante de l'environnement dans lequel évolue le robot. Ainsi, l'environnement, tant visuel que sonore, est potentiellement bruité et encombré, tandis que la perception peut également être altérée par les mouvements du robot.

2 État de l'art et positionnement de nos travaux

La robotique interactive est un défi majeur et relativement récent [Fong et al., 2003]. Ce type de robots, avant de sortir des laboratoires, doivent gagner en sociabilité afin de permettre une interaction directe avec un utilisateur non expert, que ce soit dans le champ domestique, public ou industriel. Un tel robot assistant requiert une intelligence dite « spatiale » aussi bien que « transactionnelle » :

- L'intelligence spatiale est basée sur les capacités de perception de son environnement par le robot. Il s'agit pour lui non seulement de comprendre et de naviguer dans cet espace mais aussi de manipuler des objets.
- L'intelligence transactionnelle est pour sa part basée sur les capacités du robot à percevoir l'homme et à communiquer avec lui.

Tandis que la première a fait l'objet de nombre de travaux par le passé, relativement peu de systèmes robotiques sont aujourd'hui équipés d'une interface multimodale permettant de contrôler un robot par des moyens de communication naturels pour l'homme, tels que des sens tactiles (non abordés dans nos travaux), de la parole ou des gestes humains.

2.1 Reconnaissance et compréhension de la parole

La reconnaissance de parole est incontournable pour tout robot interactif. Citons les robots *Godot* [Theobalt et al., 2002], *Coyote* [Skubic et al., 2004] et *Maggie* [Gorostiza et al., 2006] qui ne disposent que de cette composante pour communiquer avec leurs utilisateurs. Pour sa part, *BIRON* [Maas et al., 2006] utilise un système de reconnaissance de parole performant, mais déporté, et détecte des personnes grâce à différentes modalités (détection sonore, visuelle et laser). L'importante présence de cette modalité s'explique d'une part par l'importance que nous, humains, accordons à ce moyen de communication, mais également par l'ancienneté du domaine. En effet, cette dernière entraîne un grand nombre de solutions logicielles (commerciales ou libres) permettant de mettre sur pied relativement facilement un système de reconnaissance fonctionnel, bien qu'il faille aller plus loin pour le rendre performant et adapté à un contexte tel que celui de la robotique. Il est cependant à noter que, si la reconnaissance est très répandue parmi les plateformes robotiques, la compréhension de la parole y reste relativement marginale et peu décrite (un état de l'art détaillé est accessible dans le chapitre I), mais celle-ci sera abordée dans nos travaux.

2.2 Analyse et interprétation des mouvements de l'homme

L'interprétation des mouvements de l'homme à partir de capteurs embarqués est un point essentiel en IHR qu'on retrouve de plus en plus dans la littérature associée. D'une part, tout robot interactif doit maintenir une estimation de la cinématique de son utilisateur humain (et par conséquent son état) afin de prendre des décisions durant l'interaction. D'autre part, les mouvements du corps sont d'une importance fondamentale puisque 65% de l'information lors d'une interaction entre un homme et un robot est un acte non verbal [Davis, 1971], c'est-à-dire une position globale, un mouvement, un geste, etc.

Ces mouvements peuvent faire l'objet d'un suivi visuel dans le plan (2D) pour des robots équipés d'un système de vision monoculaire embarqué. C'est le cas des robots *ALBERT* [Rogalla et al., 2004] et *Pioneer* [Yoshizaki et al., 2002], qui suivent une main dans le but d'en extraire la trajectoire. Mais, d'une manière générale, les approches 3D sont plus adaptées à l'estimation des mouvements humains, puisque ceux-ci restent rarement fronto-parallèles. [Stiefelhagen et al., 2004] (*ARMAR*), mais aussi [Hanafiah et al., 2004] ont ainsi équipé leurs robots respectifs d'un système de suivi visuel 3D de la tête et des deux mains. Ces derniers utilisent également un système de suivi de l'orientation de la tête, montrant l'intérêt d'une telle démarche pour l'interprétation de gestes déictiques.

Dans tous les cas, qu'il s'agisse d'une approche 2D ou 3D, le suivi du mouvement d'une ou de plusieurs extrémités du corps (et notamment des mains) ouvre la voix à une forme de reconnaissance de gestes. Cette dernière va d'une simple détection de position maintenue durant un certains laps de temps [Hanafiah et al., 2004], à la reconnaissance de gestes proprement dite pour *ARMAR* et *ALBERT*, en passant par la mise en correspondance de modèles statiques de

forme (en anglais, "template matching") pour *Pioneer* (en l'occurrence, une main). La reconnaissance de gestes est aujourd'hui un enjeu majeur dans la communauté Robotique. Les gestes visuels traduisent les pensées humaines, et complètent, accentuent et ajustent les informations verbales. L'interprétation visuelle de gestes est particulièrement adaptée à un environnement dans lequel la communication verbale peut être confuse ou noyée dans le bruit ambiant.

Une dernière observation concerne les hypothèses sous-jacentes communes à certains travaux. D'une part, les gestes sont souvent supposés mono-manuel [Corradini and Gross, 2000, Siegwart et al., 2003, Skubic et al., 2004, Stiefelhagen et al., 2004, Yoshizaki et al., 2002] et/ou les extrémités du haut du corps sont souvent suivies séparément [Hasanuzzaman et al., 2007, Nickel and Stiefelhagen, 2006, Park et al., 2005], ce qui induit inévitablement des erreurs de suivi et donc de reconnaissance lorsqu'elles s'occultent. À notre connaissance, peu d'analyse simultanée du mouvement de toutes les extrémités du haut du corps humain n'ont encore été intégrées sur un robot mobile alors qu'un suivi de gestes efficace est essentiel à une reconnaissance de gestes ultérieure. Ceci ouvrirait en effet un nombre grandissant de possibilité d'interactions, en particulier par la reconnaissance de gestes bi-manuels. Ces derniers font partie de nos travaux et nous décrivons une approche originale permettant de traiter ces problèmes de suivi multi-cibles.

2.3 Multimodalité pour une interaction homme-robot plus avancée

L'assistance mutuelle entre les capacités visuelles et sonores d'un robot permet à un utilisateur d'introduire de manière robuste des références spatiales dans ses déclarations verbales. Combinée à un geste de pointage, ce type de commande ouvre la possibilité de désigner des objets ou des endroits de manière naturelle, par exemple faire changer le robot de position ou de direction, ou désigner un objet. Les techniques visuelles pour la perception de l'homme et le traitement du langage naturel ont été principalement étudiées indépendamment du fait qu'ils constituent chacun un domaine de recherche spécifique [Prodanov and Drygajlo, 2003b, Skubic et al., 2004, Triesch and Von der Malsburg, 2001, Waldherr et al., 2000]. Différents travaux visent à coupler ces deux canaux de communication et plusieurs robots sont aujourd'hui équipés d'interfaces multimodales combinant le geste et la parole à différents niveaux et suivant diverses stratégies. Dans ces travaux, la parole est le canal principal de la communication.

Ainsi, la stratégie la plus simple est celle développée par [Hanafiah et al., 2004] qui, n'ayant pas à sa disposition une véritable reconnaissance de gestes, part du principe que paroles et gestes sont parfaitement corrélés. [Yoshizaki et al., 2002] préfère pour sa part n'utiliser la vision qu'après que le besoin en soit exprimé par la parole, décorrélant de cette manière les deux canaux. [Rogalla et al., 2004] réalise la fusion d'événements (en provenance d'un canal ou de l'autre) associés pour définir les bonnes actions à mener, mais le système est handicapé par des aspects visuels trop peu avancés (suivi simpliste, reconnaissance de gestes 2D). Enfin, [Stiefelhagen et al., 2004] définit certainement l'interface la plus évoluée en fusionnant parole et gestes dans un cadre probabiliste. Néanmoins, leur interface multimodale n'a pas donné lieu, à

notre connaissance, à une intégration et des évaluations poussées sur une plateforme robotique, contrairement aux objectifs de nos travaux.

Le tableau 1 synthétise nos propos en décrivant les capacités d'interaction des principaux robots de la littérature. La dernière ligne a pour but de situer les robots sur lesquels portent nos travaux parmi ces derniers. La figure 2 montre l'aspect physique de trois de ces robots.

Robot	localisation seule	suivi des membres	orientation visage	gestes	parole	fusion
Biron		2D mono-manuel			X	X
Pioneer		2D mono-manuel		symbolique	X	X
Saitama		3D bi-manuel	X		X	X
ARMAR		3D bi-manuel	X	déictique	X	X
ALBERT		2D mono-manuel		déictique	X	X
RoboX		2D mono-manuel			X	
Godot					X	
ALPHA	X					
Coyote					X	
Maggie	X				X	
JIDO / HRP2		**3D bi-manuel**	**X**	**déictique / symbolique**	**X**	**X**

TAB. 1: Les principaux robots capables d'interaction complexe. En dernière ligne, les robots sur lesquels portent nos travaux et les attributs dont on veut les doter.

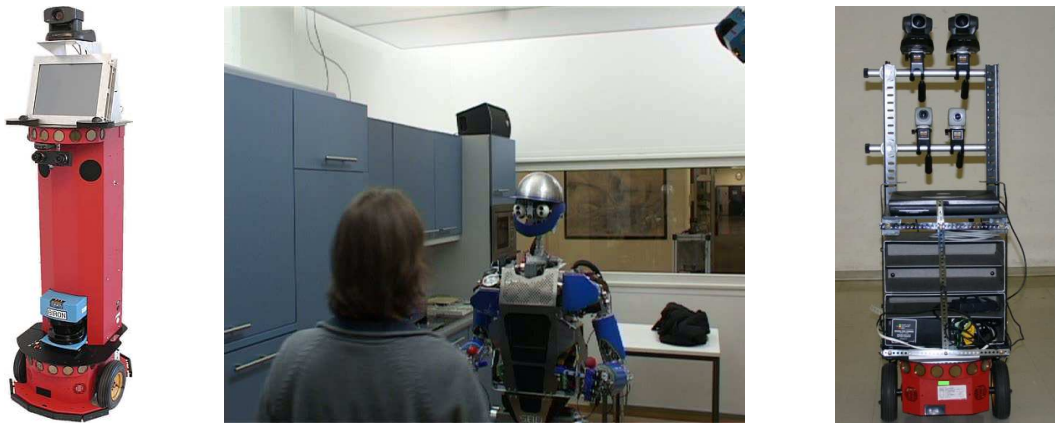


FIG. 2: Les robots, BIRON de l'université de Bielefeld, ARMAR du laboratoire CV-HCI de l'université de Karlsruhe, et celui de l'université de Saitama.

3 Articulation et spécificités de nos travaux

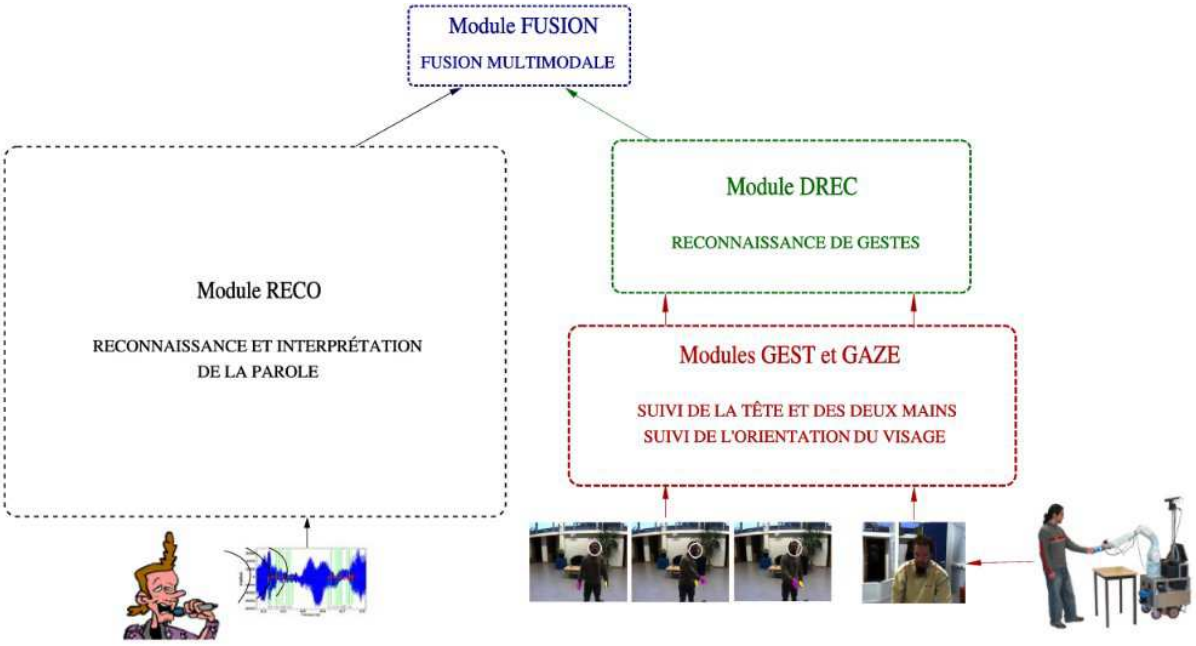


FIG. 3: Synoptique de notre interface multimodale homme-robot.

Suivant les objectifs exposés précédemment, nos travaux proposent une interface perceptuelle pour l'interaction multimodale ayant vocation à interpréter conjointement parole et geste, notamment pour le traitement des références spatiales. Celle-ci peut être schématisée par le synoptique de la figure 3 qui jalonne le plan de ce manuscrit. Listons les différentes composantes de notre interface, détaillées dans les prochains chapitres, ainsi que leurs spécificités :

1 . Un module de reconnaissance et d'interprétation de la parole continue, adapté à notre contexte applicatif, est proposé et représenté en noir sur ce schéma. Une interprétation de la parole générique et totalement embarquée, peu décrite dans la littérature, en est un point clef.

2 . En parallèle, un traqueur multi-cible chargé du suivi visuel 3D conjoint de la tête et des deux mains de l'utilisateur du robot a été développé. Celui-ci permet la gestion des interactions entre cibles, donc des occultations mutuelles lors de l'exécution de gestes. De même, un système de suivi de l'orientation du regard a été développé et ces deux composantes sont représentées en rouge sur le schéma.

3 . Les résultats des modules de suivis peuvent alors être couplés afin d'alimenter un système de reconnaissance de gestes par DBNs, formalisme peu utilisé dans la littérature, représenté en vert sur le schéma.

4 . Un dernier module (en bleu) vise à effectuer une fusion tardive des données issues de nos deux canaux d'information audio et vidéo.

L'intégration sur nos plateformes robotiques est également un point clef de nos travaux.

Le large spectre des travaux réalisés nous a permis de participer à différents projets :

Le projet européen COGNIRON (2004-2008) : L'objectif scientifique du projet « Cognitive Robot Companion » est de conférer des capacités cognitives aux robots à travers l'étude et le développement de méthodes et de technologies pour la perception, l'interprétation, le raisonnement et l'apprentissage en interaction avec l'homme. L'apport de notre système de suivi visuel à ce projet a notamment permis une interaction physique plus respectueuse des normes sociales entre l'utilisateur et le robot *Jido*.

Le projet européen COMMROB (2007-2009) : La finalité de ce projet « Advanced Robot Behaviour and hight-level Multimodal Communication » est la conception et la mise en œuvre de chariots mobiles autonomes évoluant dans un lieu public type supermarché. Le système de suivi couplé à la reconnaissance de gestes que nous avons apporté à ce projet a eu pour but de permettre le fonctionnement d'une interface multimodale proche de celle décrite dans ce manuscrit sur le trolley *Inbot*.

Le projet national ANR AMORCES (2007-...) : L'objectif de ce projet « Algorithmes et MOdèles pour un Robot Collaboratif Éloquent et Social » est d'étudier les interactions décisionnelles et opérationnelles entre homme et robot, et plus particulièrement l'impact de la communication verbale et non-verbale sur l'exécution d'une tâche collaborative entre un robot et un partenaire humain. Nos travaux sont utilisés dans ce projet sur le robot humanoïde *HRP-2*.

4 Annonce du plan

L'organisation de ce manuscrit suit dans un premier temps l'articulation de nos travaux en décrivant successivement dans les chapitres I, II, III et IV les modules développés dans le cadre de cette thèse afin de construire l'architecture précédemment décrite. Chacun de ces chapitres présente tout d'abord l'état de l'art associé, avant de formaliser nos travaux. Ils se poursuivent par la description de nos choix et méthodes et se concluent par des résultats propres à chacun des modules. Dans un second temps, le chapitre IV porte sur l'intégration de notre interface sur plusieurs plateformes robotiques et la réalisation de scénarios interactifs mettant en jeu un homme et un robot. Enfin, le dernier chapitre résume nos contributions et présente quelques extensions envisagées.

Chapitre I

Composante parole pour l'IHR en langage naturel

La parole étant le mode de communication privilégié de l'homme, il est indispensable d'équiper un robot d'assistance d'un système lui permettant d'effectuer des tâches simples ou complexes en suivant les instructions orales de son utilisateur. Ce chapitre présente la composante parole développée durant cette thèse dans le but de permettre le traitement du langage naturel dans notre contexte robotique. Les besoins d'un tel système sont dictées par son utilisation :

- les utilisateurs étant multiples et inconnus *a priori*, le système doit être indépendant du locuteur,
- pour que le système ne soit pas trop contraignant, les temps de réponse doivent être réduits,
- afin de permettre une utilisation réaliste, les taux de reconnaissance doivent également être acceptables.

De plus, certaines contraintes matérielles sont à prendre en compte. En effet, le contexte acoustique est évolutif et l'environnement potentiellement bruyant, mais surtout, la puissance de calcul disponible est limitée. Enfin, l'un des objectifs de cette thèse est également de favoriser une interaction naturelle entre homme et robot.

D'une manière générale, un système de traitement du langage peut se décomposer en trois parties :

- le traitement du signal capté par un microphone,
- la reconnaissance de la parole à partir des données acquises et de ressources phonétiques, lexicales et grammaticales,
- l'interprétation des données reconnues afin qu'elles puissent être utilisées par une machine.

La figure I.1 illustre le fonctionnement d'un tel système. Quelques précisions sont de plus données en bleu sur notre implémentation (ressources ou logiciels tiers utilisés). Les traits verticaux

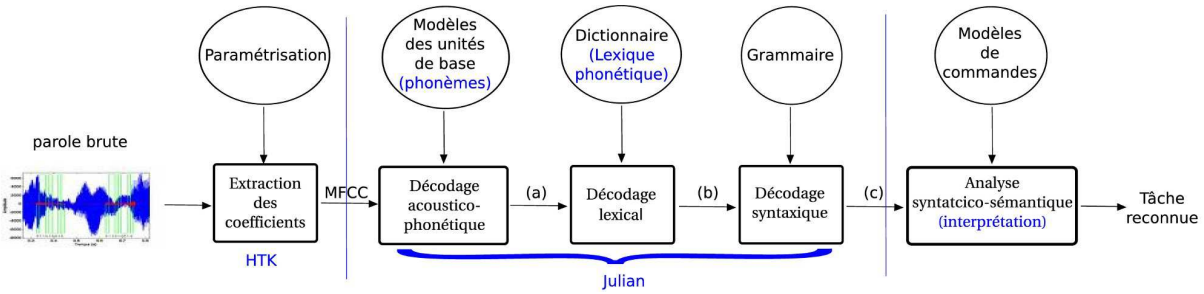


FIG. I.1: Synoptique d'un module de traitement de la parole. En bleu, des précisions sur notre implémentation. Les lettres entre parenthèses correspondent aux sorties des différentes boites du schéma : (a) suite de phonèmes, (b) suite de mots, (c) hypothèses de phrases.

représentent le découpage du module dans les explications qui vont suivre tout au long de ce chapitre.

Dans ce chapitre, nous commencerons par présenter un état de l'art du domaine du traitement du langage en mettant l'accent sur les réalisations en communication homme-machine et surtout homme-robot (section I.1). Un grand nombre d'outils, tant théoriques ([Rabiner, 1989],etc) que logiciels (HTK, Julius, etc), étant à notre disposition, nous évoquerons ensuite assez rapidement les aspects de traitement du signal et les bases de la reconnaissance de parole, ainsi que les applications qui en découlent, avant de préciser notre implémentation de ce système (section I.2). La section I.3 abordera alors le problème de la compréhension du langage. Les différentes améliorations apportées à ce système de base sont ensuite explicitées en section I.4. Enfin, quelques résultats de nos évaluations validant ces travaux concluront ce chapitre.

I.1 État de l'art

I.1.1 Reconnaissance de parole en robotique

Le traitement automatique du langage naturel est un vaste domaine de recherche qui a donné lieu à nombre d'applications. Ainsi, les premières applications réelles ont eu pour but la dictées vocales, puis, les performances matérielles évoluant, des composantes parole ont pu être intégrés dans des dispositifs plus complexes en IHM [Potamianos et al., 2004, Potamianos et al., 2009] et plus récemment en IHR.

Ces dernières, se différencient notamment par l'application qu'elles visent (guide de musée, robot assistant, etc) qui entraînent des contraintes et des conditions d'utilisation différentes :

– certains systèmes se veulent totalement embarqués, tandis que d'autres permettent le calcul déporté pour les opérations lourdes en terme de capacité de calcul,

– l'environnement peut être plus ou moins bruyant,
– le système n'est pas forcément indépendant du locuteur,
– etc.

Mais elles se différencient également par des choix logiciels, certains [Bischoff and Graefe, 2004, Gorostiza et al., 2006, Skubic et al., 2004] utilisant des logiciels commerciaux, d'autres des moteurs de recherche adaptés comme *Janus* de [Stiefelhagen et al., 2004], *Sphinx* du CMU ou *Julius* de [Lee et al., 2001]. Il est également à noter que si tous ces systèmes permettent la reconnaissance de parole, la compréhension n'en fait pas toujours partie. Voici quelques exemples de systèmes existants.

Ainsi, dans une application de robot-guide comme celle de [Prodanov and Drygajlo, 2003a], le robot prend l'initiative de l'interaction vocale en posant des questions fermées à l'utilisateur, le but étant de savoir si celui-ci souhaite se rendre ou pas à un stand donné. Le fait d'avoir l'initiative lui permet de mieux contrôler l'interaction, compte tenu d'un environnement fortement bruité. Le système de reconnaissance est cependant limité : il reconnait deux mots clefs (« oui » et « non »), le reste étant considéré comme des « mots poubelle » (grâce à un modèle "garbage"). Dans [Ghidary et al., 2002], il s'agit plutôt d'un robot domestique. La reconnaissance de parole y est alors utilisée pour générer une carte de navigation grâce aux indications de l'utilisateur, mais là encore, le vocabulaire est extrêmement restreint afin de n'avoir aucune ambiguïté possible et de pouvoir les interpréter directement (à chaque phrase est associée une et une seule interprétation).

D'autres systèmes permettent une interaction plus avancée grâce à une utilisation plus extensive d'un système commercial de reconnaissance. Ainsi, *Maggie* [Gorostiza et al., 2006] intègre le logiciel *Dragon Naturally Speaking* dans un système de gestion de dialogue se basant sur la transcription de l'énoncé de l'utilisateur. Mais ce système est purement réactif, ce qui réduit fortement l'initiative du robot. De même, *Hermes* [Bischoff and Graefe, 2004] combine une reconnaissance commerciale avec une gestion du dialogue, mais sous la forme d'un système plus directif. L'ensemble n'est malheureusement pas intégré sur le robot, mais fait appel à un traitement distant et, dans les deux cas, la reconnaissance est dépendante du locuteur. En réalité, ces systèmes sont d'avantages dédiés à la communication homme-machine qu'à l'interaction homme-robot, puisqu'aucun ne prend en compte la problématique robotique.

Si l'on s'oriente vers des systèmes réellement intégrés sur une plateforme robotique, [Hanafiah et al., 2004] et [Yoshizaki et al., 2002] utilisent tout deux le logiciel commercial *ViaVoice* afin d'effectuer une reconnaissance de parole limitée (les phrases sont très courtes et simples). La sortie de cette phase de reconnaissance est alors analysée, *via* une division en morphèmes suivie d'une catégorisation pour le premier, tandis que le second utilise des informations contextuelles pour l'enrichir. [Skubic et al., 2004], avec leur robot *Coyote*, vont plus loin en définissant un langage spatial, ainsi que la notion de référent pouvant être l'utilisateur, mais aussi le robot ou un objet. La compréhension est également abordée à travers la description d'un interpréteur de commandes mettant successivement en jeu une analyse lexicale, syntaxique et sémantique. Cependant, dans les trois cas, aucune réelle expérimentation n'est menée, ce qui laisse à penser que la prise en compte des contraintes robotiques (définies en introduction de ce document) n'est pas totale.

Enfin, [Stiefelhagen et al., 2004] décrit sans doute l'intégration la plus aboutie et la plus

recherchée d'une composante parole sur un robot assistant. La reconnaissance de parole y est effectuée grâce à leur moteur de reconnaissance nommé *Janus*, qui est basé sur une grammaire hors contexte, partant du principe que ce type d'approche est plus adaptée à système robotique. Ayant choisi pour les mêmes raisons une approche analogue, nous reviendrons sur cette dernière affirmation dans la section I.2. L'utilisation d'une grammaire évolutive, la gestion de phénomènes extra-linguistiques (notamment des bruits de cuisine), des hésitations, mais également du contexte en font un système de reconnaissance très évolué. L'interprétation est effectuée en utilisant une nouvelle grammaire hors contexte ainsi qu'une ontologie, permettant ainsi la gestion de dialogue. Néanmoins, si le système est extrêmement complet, le manque d'expérimentations rapportées dans l'état de l'art reste un point faible.

I.1.2 Compréhension du langage

La compréhension de la parole a donné lieu à une riche littérature, notamment dans des communautés telles que celle de l'indexation de documents, de la traduction ou de la transcription automatique[Linarès et al., 2007]. On trouve par ailleurs nombre d'applications basées sur ces techniques qui vont de l'application de téléphonie [Baggia et al., 1992] à des systèmes beaucoup plus complets et généraux tels que *ROMUS*. [J. Goulian, 2003] décrit ce dernier système tout en expliquant les bases du traitement automatique du langage dit robuste. Le principe, en trois étapes est le suivant.

1 . L'étiquetage, consiste à associer à chaque mot ou expression une catégorie syntaxique (qui peut être ambigüe).

2 . La segmentation permet, à partir des symboles définis dans la première phase, de découper la phrase en « chunks », c'est-à-dire en unités sémantiques minimales.

3 . Enfin, la dernière phase du processus consiste à construire un graphe de dépendances entre ces « chunks » (dépendances sémantiques) afin d'en déduire la représentation finale de l'énoncé.

Dans un contexte robotique, la littérature est souvent plus évasive sur cette partie pourtant indispensable d'un système de reconnaissance et de compréhension de la parole. En voici tout de même quelques exemples. [Chong et al., 2000] utilise une catégorisation des mots permettant de transformer la séquence de mots issue de la reconnaissance en séquence de symboles qui constituent son interprétation. [Antoniol et al., 1993] utilise une technique proche, détectant des mots ou expressions clefs *via* une association de modèles (en anglais, "template matching"), mais précise qu'il permet une plus grande souplesse de modélisation en ne prenant en compte que les mots significatifs d'une phrase. Enfin, [Hüwel and Wrede, 2006] utilise un système plus complexe, et certainement le plus abouti parmi les systèmes robotiques, afin d'équiper le robot *BIRON* d'une couche de traitement dédié à la compréhension. Il s'agit ici, d'utiliser un ensemble de connaissances sémantiques, appelées « unités sémantiques situées » et qui construisent un réseau de relations entre concepts sémantiques, afin de produire une ontologie permettant l'analyse d'une phrase issue de leur système de reconnaissance de la parole spontanée et d'alimenter

un système de dialogue. L'ensemble de ces systèmes se situent dans la prolongation de systèmes de reconnaissance de la parole utilisant des modèles de langage statistiques de type N-grammes. Les premiers se veulent simples, partant du principe que leur contexte applicatif limite fortement le sens des phrases prononcées par un utilisateur, tandis que le dernier se veut plus général. Étant donné nos hypothèses de départ qui nous ont amenés à effectuer une reconnaissance par grammaire, une approche par association de modèles semble suffisante pour notre application.

I.2 Reconnaissance de la parole dans notre contexte robotique

Le but de notre système de reconnaissance de la parole est de pouvoir traiter des phrases prononcées en français par différents locuteurs. Nous ne pouvons par conséquent nous satisfaire d'une simple application de reconnaissance de mots isolés. Cette dernière technique est par exemple souvent utilisée par des opérateurs de téléphonie dans des applications où tout ce que l'on attend de l'utilisateur est un ordre prédéfini (« effacer ») ou une réponse binaire (« oui »/« non »). Dans notre cas, le système de reconnaissance doit être indépendant du locuteur et capable de reconnaître des ordres plus ou moins complexes sous forme de phrases dont le tableau I.1 donne quelques exemples.

Début/clôture d'interaction et présentation au robot (+ geste symbolique)	« Bonjour Jido, je m'appelle Paul. » , « Salut, c'est Jean. » , « Au revoir Jido. » , etc
Ordres de mouvements basiques	« Tourne à gauche. » , « S'il te plaît, fait demi-tour. » , « Tourne de dix degrés sur la droite. » , etc
Requête de guidage	« Jido, nous allons voir Rackham. » , « Emmène-moi à la salle robotique. » , etc
Accord / désaccord / remerciement	« Oui. » / « Non. » / « Merci. » , « D'accord. » / « Non, c'est pas ça. » / « Merci beaucoup. » , « C'est bien Jido. » / « Non merci. » / « Merci Jido. » , etc
Requête de guidage dans l'environnement humain (+ geste symbolique + résolution de références spatiales)	« Viens à ma droite. » , « Approche-toi de moi. » , « Suis-moi. » , etc
Ordres plus avancés impliquant un geste déictique (+ geste déictique)	« Viens ici. » , « Prend cette bouteille. » , « Pose la tasse à cet endroit. » , etc
Interaction avec échange d'objet (+ résolution de références spatiales)	« Donne-moi cette bouteille. » , « Apporte le verre orange à Jacques. » , etc

TAB. I.1: Exemples de requêtes reconnues par notre module (classiques et nécessitant une perception plus avancée de l'homme). Entre parenthèses, les références (intéressantes ou obligatoires) extérieures au module de traitement de la parole.

Des logiciels clef en main pouvant effectuer ce type de reconnaissance sont disponibles sur le marché et sont utilisés dans divers travaux cités en section précédente. Dans notre cadre, leur utilisation se heurte à plusieurs obstacles, notamment :

- peu sont utilisables sous linux,
- à notre connaissance, aucun ne permet la reconnaissance de la langue française (mis à part les systèmes de type dictée vocale qui sont dépendants du locuteur),
- peu sont libres, ils n'ont donc pas la flexibilité requise pour notre utilisation (modifications interdites, accès impossible aux données internes des algorithmes,etc).

C'est pourquoi nous utilisons un moteur de reconnaissance pour effectuer cette tâche. Ceci nécessite donc de fournir un certain nombre de ressources linguistiques et grammaticales :

- des modèles acoustiques, pour nous des phonèmes modélisés par des HMMs,
- un lexique, c'est-à-dire un dictionnaire énumérant les différentes prononciations possibles des mots de notre vocabulaire,
- un modèle de langage, sous forme de grammaire ou de N-grammes, nous y reviendrons plus loin.

Les principes de la reconnaissance vocale sont présentés dans la première sous-section, tandis que le moteur de reconnaissance et les ressources utilisées font l'objet de la sous-section suivante.

I.2.1 Principes de la reconnaissance vocale

a) Prétraitements

Avant de pouvoir modéliser un signal, quel qu'il soit, une première étape indispensable consiste à effectuer divers prétraitements afin d'extraire du signal des vecteurs de données pertinentes capables d'alimenter un algorithme de reconnaissance. La reconnaissance de parole ne fait pas exception à la règle : ces traitements standards en traitement de signaux sonores (échantillonnage, transformée de Fourrier rapide), puis plus spécifiques à la parole sont effectués afin d'obtenir une séquence de vecteurs acoustiques qui forment l'entrée de l'algorithme de reconnaissance. Ces vecteurs sont appelés *MFCCs* (pour "Mel Frequency Cepstrum Coefficient") en référence à l'échelle de Mel qui est utilisée ici plutôt qu'une échelle fréquentielle classique car basée sur la perception humaine des sons (qui est non-linéaire) :

$mel(f) = 2595.log(1 + f/700)$, avec f la fréquence en Hz$[O'Shaughnessy, 1987]$.

De plus, dans le but de rendre la reconnaissance plus robuste, notamment au bruit, on rajoute souvent au vecteur la vitesse Δ voire l'accélération Δ^2 de ces MFCCs (c'est-à-dire les dérivées et dérivées seconde du vecteur). On peut également y rajouter un terme d'énergie E qui lui aussi sera dérivé autant que les MFCCs. Enfin, divers post-traitements des MFCCs sont possibles, tels une normalisation de l'énergie, avec là encore comme but de supprimer ou rendre négligeable des bruits de fond inintéressant pour la reconnaissance.

Il est à noter que, bien que le calcul des MFCCs soit la méthode la plus répandue dans la communauté parole, il existe d'autres méthodes comme le calcul du LPC (pour Linear Prediction Coefficients) qui ne seront pas abordés dans ce manuscrit. Le lecteur intéressé est invité

à consulter [Boite, 2000] pour de plus amples détails concernant l'ensemble de cette phase de prétraitements.

b) Modèles phonétiques et modélisation par HMM

Pour chaque signal traité, on obtient une séquence de vecteurs de paramètres. Imaginons que nous ayons deux jeux de donnés X et Y, composés chacun d'un certain nombre de ces séquences, et que nous voulions les classifier de manière automatique. Pour ce faire, il nous faut commencer par trouver un modèle dynamique (puisqu'il s'agit de séquences de données temporelles) adapté à cette tâche, c'est-à-dire capable de capter l'évolution des données d'une séquence de X, les similarités entre cette séquence et d'autres du même jeu, et capable de les distinguer des séquences de Y.

La modélisation la plus utilisée, car la plus efficace jusqu'à aujourd'hui, en reconnaissance de la parole est le modèle de Markov caché. Les HMMs sont utilisés pour modéliser les unités de base d'un système de reconnaissance. En reconnaissance de la parole [Jurafsky and Martin, 2000], ces unités peuvent être :

- des mots, pour des applications qui ne nécessitent qu'un vocabulaire très restreint (opérateurs de téléphonie par exemple),
- des phonèmes, pour des applications à vocabulaire de taille moyenne et au delà (dans la plupart des applications de communication homme-machine). Les phonèmes sont des unités phonétiques de base, la plus petite unité discrète que l'on puisse isoler par segmentation dans la chaîne parlée. Ils représentent les sons qui forment une langue.
- des N-phones, pour le même type d'applications que les phonèmes. Les N-phones (diphones ou triphones le plus souvent) sont en réalité des suites de N phonèmes qui sont modélisés par un unique HMM. L'utilisation de phonèmes est en fait synonyme d'utilisation de monophones. Une utilisation courante est celle de triphones car ceux-ci permettent la prise en compte du contexte gauche et droit, la prononciation d'un phonème étant différente suivant les phonèmes qui le précède et le succède. Elle nécessite en général un plus grand corpus d'apprentissage que pour des monophones car l'ensemble des triplets sonores doivent y être suffisamment représentés, bien qu'il soit possible de générer par simulation des triphones à partir de monophones précédemment appris.

Dans notre cas, l'IRIT, par des travaux antérieurs (voir sous-section I.2.2), dispose d'un jeu de modèles acoustico-phonétiques de phonèmes que nous avons pu utiliser. Ne disposant pas de N-phones à l'heure actuelle, et la construction de tels modèles sortant du cadre de nos travaux, nous utilisons uniquement des monophones.

➢ Définition

Un modèle de Markov caché (ou HMM, pour "Hidden Markov Model") [Rabiner, 1989] est un modèle temporel constitué de nœuds cachés S_k et de nœuds d'observation x_k. Nous détaillons ici à la fois le cas continu et discret, le premier ayant son application ici puisqu'on cherche à modéliser un signal continu, tandis que le second est souvent utilisé pour des applications de type décision ou reconnaissance d'activité, mais sera également utilisé pour notre système de reconnaissance de geste (voir chapitre III). La figure I.2 représente un HMM sous

sa forme déployée. Un nœud (ou variable) caché représente l'état interne du système à modéliser, il n'est pas observable. Un nœud d'observation est une variable observable conséquence de cet état interne. Les liens causaux (ici symbolisés par des flèches) sont de nature probabiliste et représentent la probabilité d'avoir les valeurs de la variable d'arrivée sachant la variable de départ.

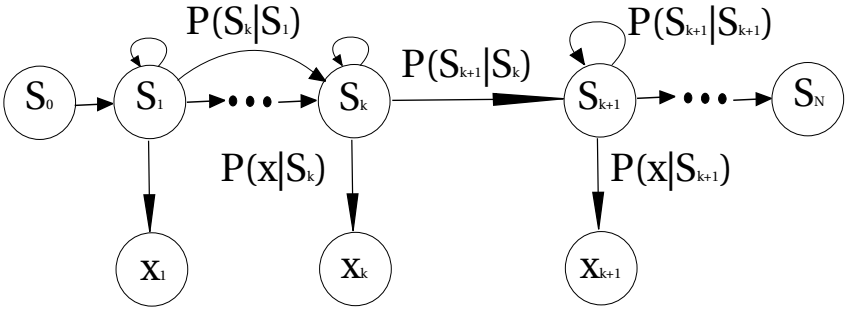


FIG. I.2: Schéma déployé d'un HMM.

Dans un formalisme plus mathématique, un modèle de Markov caché peut être représenté par son modèle $\lambda = (\pi, A, B)$, défini par :

N est le nombre d'états cachés possibles. Il correspond au nombre de nœuds S_k du modèle déployé. On note les différentes valeurs possibles de cette variable $S = S_1, S_2, ..., S_N$. On note également q_t et O_t respectivement l'état et l'observation au temps t.

M est le nombre de symboles d'observations possibles dans le cas discret, les symboles sont alors notés $V = v_1, v_2, ..., v_M$. Pour nous, dans le cas continu, M représente le nombre de fonctions de densité qui modélisent chaque nœud d'observation x_k.

π est la distribution initiale de probabilités : $\pi_i = P(q_1 = S_i)$, avec $1 \leq i \leq N$.

A est la matrice de transition. Elle contient la distribution de probabilité des transitions entre états : $A = \{a_{ij}\}$, avec

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i), \qquad 1 \leq i, j \leq N.$$

B est la matrice d'émission. Elle contient la distribution de probabilité des observations : $B = b_j(k)$, avec :

– dans le cas discret,

$$b_j(k) = P(O_t = v_k | q_t = S_j), \qquad 1 \leq j \leq N \qquad \text{et} \qquad 1 \leq k \leq M.$$

– dans le cas continu, B est une distribution de densité de probabilité et

$$b_j(O_t) = P(O_t | q_t = S_j) = \sum_{m=1}^{M} c_{jm} \mathfrak{N}(O_t, \mu_{jm}, \Sigma_{jm}), \qquad 1 \leq j \leq N,$$

avec $\mathfrak{N}$ une fonction de densité qui soit log-concave ou elliptiquement symétrique (le plus souvent une gaussienne) de moyenne μ_{jm} et de covariance Σ_{jm}.

➢ **Apprentissage et reconnaissance**

Si nous reprenons l'exemple des jeux de donnés X et Y donné au point précédent, une fois les modèles choisis, nous avons besoin d'une phase d'apprentissage pour leur donner forme. De même, une fois ces modèles appris, on peut les utiliser pour reconnaître une nouvelle séquence de données z, c'est-à-dire, pour déterminer automatiquement si z fait plutôt partie du jeu de données X ou du jeu Y.

Les HMMs peuvent être utilisés de plusieurs façons pour traiter des problèmes réels. Citons les trois problèmes dits « classiques » [Rabiner, 1989].

1 . Étant donnée une séquence d'observations $O = \{O_1, ..., O_T\}$ de taille T et un modèle $\lambda = (\pi, A, B)$, comment peut-on calculer efficacement la probabilité $P(O|\lambda)$ de l'apparition de cette séquence O connaissant le modèle λ ?

2 . Étant donnée une séquence d'observations $O = \{O_1, ..., O_T\}$ de taille T et un modèle $\lambda = (\pi, A, B)$, quelle est la séquence d'états $Q = \{q_1, ..., q_T\}$ qui explique le mieux l'observation ?

3 . Comment ajuster le modèle $\lambda = (\pi, A, B)$ afin qu'il explique le mieux une séquence d'observation O, c'est-à-dire qu'il maximise $P(O|\lambda)$?

L'apprentissage d'un HMM consiste donc à résoudre le problème 3 en modifiant itérativement π, A et B afin de maximiser $P(O|\lambda)$, et ce pour chaque séquence d'observations O que contiendra le corpus d'apprentissage. Mais pour ce faire, il faut résoudre également le problème 2, c'est-à-dire trouver le chemin dans le modèle qui explique le mieux l'observation afin de pouvoir le modifier si cela permet d'augmenter $P(O|\lambda)$. Et pour les deux derniers, la résolution du problème 1 est obligatoire puisqu'il est nécessaire d'évaluer le modèle pour pouvoir l'améliorer. Il existe plusieurs algorithmes pour faire tout cela : l'algorithme de Baum-Welch [Baum and Petrie, 1966], l'algorithme Expectation-Maximization (ou *EM*) [Baum, 1972] ou même simplement l'algorithme Viterbi [Forney, 1973].

La reconnaissance d'une séquence d'observations O par HMM consiste pour sa part à trouver parmi un ensemble de modèles $\{\lambda_1, ..., \lambda_n\}$ lequel maximise $P(O|\lambda_i)$. Il s'agit donc ici de résoudre uniquement le problème 1 pour chacun de ces modèles, puis de choisir celui qui obtiendra la meilleure probabilité. Cela est fait en général par un algorithme de type Viterbi [Forney, 1973]. Lorsque les observations données en entrée du système sont d'origine sonores, et qu'on cherche à les reconnaître *via* des HMMs modélisant des phonèmes, on parle de décodage acoustico-phonétique.

c) Lexique phonétique

Un lexique (aussi appelé dictionnaire) phonétique constitue un ensemble de données dont le but est de faire correspondre à chaque entrée (généralement un mot) la ou les prononciations correspondantes. Ces diverses prononciations (les notations utilisées ici sont décrites en annexe A) permettent de prendre en compte à la fois :

– les variantes locales de prononciation du français (un alsacien ou un chti prononceront par exemple systématiquement le /t/ de « vingt », tandis que certains toulousains prononceront le /s/ de « moins »),

– les raccourcis de langage (le /r/ n'est pas toujours prononcé dans « quatre »),
– ainsi que les liaisons entre mots (dans « est insuffisant », on prononce le /t/ de liaison).

Les systèmes de reconnaissance de la parole continue utilisent de tels lexiques pour construire des modèles pour chaque mot en concaténant les modèles phonétiques qui le composent. Ainsi, bien qu'il existe d'autres techniques, les algorithmes de ces systèmes construisent en général une structure arborescente basée sur ces modèles concaténés afin d'obtenir un gain en mémoire comme en temps de calcul lors du décodage. En effet, en parcourant un lexique, on se rend vite compte que bon nombre de prononciations ont des préfixes en commun, notamment les mots décomposés en variantes de prononciation. Certains systèmes proposent même une structuration en graphe permettant également une mise en commun des suffixes. La figure I.3 illustre cette concaténation de HMMs sous forme d'arbre pour modéliser une partie d'un lexique phonétique contenant les mots « Ceci », « Ça » et « est ». Afin de simplifier la lecture de la figure, les HMMs ne sont ici pas représentés directement mais *via* les phonèmes qu'ils modélisent.

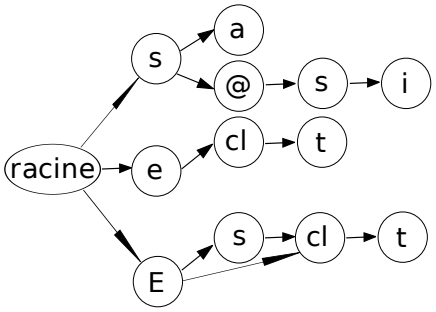


FIG. I.3: Représentation arborescente d'un (petit) lexique phonétique.

d) Modèle de langage

Dans le cadre d'un système entièrement probabiliste, la reconnaissance de la parole basée sur une phrase s'exprime sous la forme d'une équation bayésienne formulée par [Balh et al., 1983]. Le but du système est alors de trouver l'hypothèse W^* qui maximise, pour toutes les séquences de mots W possibles et pour une observation acoustique A, l'équation :

$$W^* = \arg\max_W P(W|A) = arg\max_W \frac{P(W).P(A|W)}{P(A)} \simeq \arg\max_W P(W).P(A|W). \qquad \text{(I.1)}$$

$P(A)$ est constante pour toutes les séquences d'observations, d'où l'approximation, et $P(W)$ est la probabilité *a priori* d'obtenir la séquence de mots W sans aucune notion acoustique. Cette dernière est générée par le modèle de langage pour lequel il existe deux approches : statistique (N-grammes) ou par règles (grammaires généralement hors contexte). Le choix entre ces deux types de modélisation dépend principalement de la taille du vocabulaire utilisé par l'application.

➢ **Approche statistique**

D'une manière générale, un N-gramme est une succession de N éléments. L'idée est qu'à partir d'une séquence d'éléments donnée, il est possible d'obtenir la fonction de vraisemblance

de l'apparition de l'élément suivant. Ainsi, à partir d'un corpus d'apprentissage, il est facile de construire une distribution de probabilité pour le prochain élément avec un historique de taille $N-1$. Cette modélisation correspond en fait à une chaîne de Markov d'ordre $N-1$ où seules les $N-1$ dernières observations sont utilisées pour la prédiction de la $N^{\text{ième}}$. En reconnaissance de la parole, on utilise classiquement cette modélisation pour construire un modèle de langage grand vocabulaire. Dans ce cas, chaque N-gramme représente une suite de N mots et l'ensemble des N-grammes doivent permettre de modéliser une langue dans le sens où il représente toutes les probabilités de retrouver chacune de ces suites de mots dans une phrase. Cette modélisation part donc du principe que connaissant une suite de $N-1$ mots, on est capable de prédire le mot suivant (ou plus exactement, on connait la probabilité d'apparition de chaque mot qui pourrait suivre cette séquence).

➢ **Approche par règle**

En théorie du langage, la notion de grammaire formelle précise les règles de syntaxe d'un langage, un langage étant un ensemble de *mots*. D'une manière générale, une grammaire formelle de type régulière ou hors contexte (que nous appellerons par la suite simplement grammaire) est constituée des quatre objets suivants :

- un ensemble fini de symboles terminaux, qui sont, d'une certaine façon, les unités de base d'une grammaire,
- un ensemble fini de symboles non-terminaux,
- un ensemble de règles de production, qui sont des paires formées d'un non-terminal et d'une suite de terminaux et de non-terminaux (à construire ou à analyser),
- un élément de l'ensemble des non-terminaux, appelé axiome.

En combinant ces objets, nous obtenons une représentation qui, partant de l'axiome et en se servant des règles de production, permet de reconstituer un certains nombre de suites de symboles terminaux. Dans une application de reconnaissance de la parole comme la notre, les symboles terminaux sont en réalité des mots (faisant partie d'un lexique) que notre grammaire permet de combiner afin d'obtenir un ensemble de phrases. Notre grammaire est par conséquent une description formelle de toutes les phrases connues du système. Un moteur de reconnaissance utilisera celle-ci en la factorisant sous la forme d'un treillis de mots. La figure I.4 donne un exemple du treillis de mots construit à partir de la grammaire figure I.2.

e) Méthodes d'évaluation des systèmes de reconnaissance

Un système de reconnaissance de la parole a pour but de faire correspondre une suite de mots (phrase) compatible avec ses ressources grammaticales à un signal d'entrée, et ce grâce à ses ressources linguistiques. Cette phrase peut-être assimilée à une suite de HMMs ayant été combinés dans un premier temps afin de former des mots *via* le lexique. Le processus de reconnaissance produit par conséquent, et pour une telle phrase, un score de reconnaissance qui n'est autre que la combinaison des scores (en réalité des probabilités) obtenus par les différents HMMs parcourus au cours du processus pour les faire correspondre au signal. En pratique, les probabilités en jeu étant extrêmement faibles, on se sert du logarithme de ces dernières, on parle alors de « log-vraisemblance ». Le rôle d'un système de reconnaissance de parole consiste

donc à calculer ces scores pour les phrases compatibles avec les ressources grammaticales utilisées. La phrase considérée comme reconnue est alors celle qui obtient le meilleur score (en l'occurrence, la log-vraisemblance la plus proche de zéro). C'est cette hypothèse, la plus vraisemblable, qui est utilisée pour évaluer un système de reconnaissance, mais il est à noter qu'en réalité nous obtenons bien une liste de phrases accompagnées de leurs scores.

Il existe diverses mesures de performances permettant d'évaluer des systèmes de reconnaissance de parole. Les deux plus répandues dans la communauté sont basées sur une mesures de taux d'erreur sur les mots : il s'agit du WER (pour "Word Error Rate") et de l'*accuracy* (précision, taux de mots bien reconnus). Leurs définitions sont respectivement :

$$\begin{cases} WER & = & 1 - \frac{N-D-S}{N} \\ Accuracy & = & \frac{N-D-S-I}{N} \end{cases}$$

avec N le nombre total de mots dans la phrase, D (*Deletion errors*) le nombre de mots manquant dans la reconnaissance, S (*Substitution errors*) le nombre de mots remplacés par un autre et I (*Insertion errors*) le nombre de mots insérés (*i.e.* qui n'auraient pas dû faire partie du résultat). Une différence est faite entre *accuracy* et WER car les mots insérés ne sont pas toujours considérés comme des erreurs « répréhensibles ». Ces deux mesures sont les plus répandues pour l'évaluation des systèmes à grand vocabulaire. Mais lorsqu'il s'agit de systèmes à vocabulaire plus restreint et dont le but est plus souvent de reconnaître plus spécifiquement des phrases, une autre mesure est souvent utilisée en complément des précédentes : il s'agit du SER, pour "Sentence Error Rate" ou taux d'erreur sur les phrases. Il est à noter que WER et SER ne sont liés qu'en partie : une amélioration significative du WER ne se traduira pas forcément par une amélioration similaire du SER car une phrase n'est considérée comme juste que si elle correspond parfaitement à ce qui est recherché.

I.2.2 Implémentation sur nos plateformes

Nous allons maintenant décrire le système qui constitue la composante parole dédiée à notre application d'interaction homme-robot.

a) Paramétrisation et ressources linguistiques

La phase de traitement du signal et d'extraction des coefficients pertinents est représentée dans la partie gauche du synoptique donné en figure I.1. Elle est effectuée dans notre module par l'intermédiaire d'un outil de HTK [Young et al., 2006], une « boîte à outils » développée à l'université de Cambridge et consacrée à la construction et à l'utilisation des HMMs. Cette boîte à outils se veut générique, mais est tout de même très orientée vers la reconnaissance de la parole et nous fournit par conséquent tous les outils nécessaires au prétraitement des données (extraction des MFCCs) en plus des outils pour HMMs (création, apprentissage de HMMs, puis reconnaissance).

➢ Paramétrisation

Les campagnes d'évaluation ESTER [Galliano et al., 2005], organisées conjointement par la Direction Générale à l'Armement (DGA), l'Association Francophone de la Communication Parlée (AFCP) et avec le concours de l'ELDA (Evaluations and langage resources Distribution Agency), visent à mesurer les performances actuelles de chacune des composantes d'un système d'indexation d'émissions radiophoniques. Dans ce but, des corpus sont construits sur la base de telles émissions et fournies aux laboratoires participants afin d'évaluer leurs systèmes de transcription automatique. Les transcriptions sont enrichies par un ensemble d'informations annexes, comme le découpage automatique en tours de paroles, le marquage des entités nommées, etc, qui permettent d'obtenir une transcription lisible d'une part et, d'autre part, une représentation structurée du document à des fins d'extraction d'informations. Les modèles phonétiques, sur lesquels est basé notre système de reconnaissance de la parole, ont été construits et appris lors de la participation de l'IRIT à l'une de ces campagnes. Les modèles phonétiques ont été construits suivant :

- un échantillonnage par des fenêtres de 16 ms avec un recouvrement 8 ms (avec fenêtrage de hamming),
- les vecteurs acoustiques sont constitués de 39 paramètres soit 12 MFCCs, l'énergie, leur vitesse et accélération,
- l'énergie est normalisée,
- les fréquences sont limitées à la bande de 300 à 8000 Hz.

Ces caractéristiques, utilisées lors de l'apprentissage doivent être les mêmes lors de la reconnaissance.

➢ Modèles phonétiques

La modélisation acoustique a été réalisée en utilisant des HMMs gauche-droite. Cela signifie que tous les liens de probabilité d'un HMM (voir la figure I.2) ne peuvent aller que dans un seul sens : la matrice de transition A est donc définie par $a_{ij} = P(q_{k+1} = S_j | q_k = S_i)$, avec $1 \leq i \leq N$ et $i \leq j \leq N$. Chaque état du HMM est décrit par un mélange de 32 gaussiennes (qui modélisent chacune des $b_j(k)$). Ces HMMs ont été appris sur les 31 heures d'enregistrement radiophonique de la phase 1 de la campagne ESTER et modélisent l'ensemble des phonèmes constituant la langue française, dont la définition est donnée en annexe A. En réalité, nous disposons au final de 39 HMMs :

- 35 HMMs modélisent les phonèmes français tels que définis dans l'annexe A, mis à part le modèle de /a/ qui est fusionné avec /ɑ/, et sont composés de 3 états chacun (sauf les consonnes plosives qui n'en ont que 2),
- 2 HMMs modélisent les silences, l'un court (3 états) et l'autre long (5 états), qui permettent notamment de modéliser les pauses entre deux mots, ainsi qu'en début et fin de phrase,
- 2 HMMs modélisent les pseudo-phonèmes (quasi-silences très courts) précédents les consonnes plosives (l'un pour les /b/ et /d/, l'autre pour les /k/, /p/ et /t/) et sont composés de 2 états chacun.

➢ **Lexique phonétique**

Le lexique phonétique est un extrait de la base de données lexicale française BDLEX [Pérennou and de Calmès, 2000]. Cette base a été développée dans le cadre du groupe de recherche GDR-PRC (Communication Homme-Machine) à l'IRIT et contient environ 440 000 formes fléchies (issues de 50 000 formes canoniques). Les informations associées qui nous intéressent sont la graphie accentuée (c'est à dire l'orthographe des mots) et leurs prononciations, mais cette base en contient d'autres comme des attributs morphosyntaxiques (catégorie syntaxique, accords, ...), la graphie du mot canonique et un indicateur de fréquence. Pour sa part, notre lexique contient les mots de notre vocabulaire en lien avec leurs prononciations (un exemple en est donné à travers le tableau I.2a). Il est à noter que la diversité des prononciations possibles dans notre cadre est grand, étant donné que notre système doit être indépendant du locuteur.

b) Choix d'une modélisation

Au vu de ces descriptions, force est de constater qu'il est quasiment impossible et extrêmement lourd de représenter l'intégralité d'une langue par des règles, et ce d'autant plus que notre but est de comprendre le langage spontané, qui peut comprendre des hésitations, des répétitions ou des abus de langage. De plus, une représentation probabiliste du langage peut beaucoup plus facilement couvrir une grande partie des possibilités d'une langue sous réserve de disposer d'un corpus d'apprentissage adapté.

Mais dans le même temps, si l'on ne dispose pas d'un corpus suffisant et adapté à notre contexte, il sera impossible de construire une représentation assez satisfaisante pour faire de la reconnaissance. En effet, certains N-grammes peuvent ne pas apparaître ou peu dans le corpus d'apprentissage. Leurs probabilités après apprentissage seraient alors biaisées : nulle dans le premier cas, alors que le symbole peut apparaître lors d'une reconnaissance, ou trop approximative dans le second cas. Dans tous les cas, sur un corpus d'apprentissage de taille trop faible, certains triplets seront sous- ou sur-représentés, biaisant leurs probabilités et par conséquent risquant d'entraîner de mauvaises reconnaissances. Bien qu'il existe des méthodes de lissage et de redistribution de probabilité, elles deviennent inutiles pour des corpus trop petits. D'autre part, une modélisation statistiques du langage permet de générer, dans certaines conditions, des suites de mots complètement incohérentes avec une forte probabilité, alors qu'une grammaire ne peut produire que des phrases correctes (ou du moins faisant partie de cette grammaire). Enfin, une modélisation probabiliste génère des phrases qui ne peuvent être prévues à l'avance, ce qui est une excellente chose pour la reconnaissance de parole, mais rend l'interprétation de ces phrases plus complexe que pour celles générées par une grammaire vu leur variabilité accrue.

Cette énumération des principaux avantages et désavantages de chacune de ces méthodes de modélisation du langage expliquent les cadres dans lesquels elles sont utilisées. Les modélisations probabilistes sont utilisées dans toutes les applications à grand vocabulaire (LVCSR, pour "Large Vocabulary Continuous Speech Recognition") où elles ont prouvé leur grande efficacité (mesurée en taux d'erreur sur les mots (WER)). Les grammaires sont plus souvent utilisées dans des applications à petit vocabulaire où il est facile de modéliser l'ensemble des phrases possibles. Dans notre cas, bien que notre volonté de traiter la parole la plus naturelle possible puisse nous attirer vers les modèles probabilistes, les restrictions de notre cadre applicatif nous

font choisir la modélisation par grammaire. Les raisons de notre choix sont les suivantes :

1 . Nous ne disposions au départ d'aucun corpus de textes dans le contexte de l'interaction homme-robot ou même homme-machine et la construction d'un tel corpus est assez longue. Les systèmes à grand vocabulaire utilisent souvent des tri-grammes appris à partir de journaux (écrits) pour des tâches de reconnaissance sur des journaux ou émissions télévisés ou radiophoniques. Notre contexte est très différent et de tels modèles n'auraient donc pas été adaptés et auraient mal orienté la reconnaissance. À long terme, nos corpus grandissant pourront peut-être permettre un apprentissage de tels modèles en remplacement de nos grammaires.

2 . Notre but étant uniquement de contrôler un robot et le nombre d'actions exécutables par le robot étant limité, il est possible de construire une grammaire assez complète limité à ce contexte.

3 . L'analyse sémantique de phrases issues d'une grammaire, et donc prévisibles, est plus aisée que l'analyse de phrases issues de N-grammes. Cela nous permet dans un premier temps de construire rapidement notre premier système de compréhension.

4 . Le temps de calcul, comme la place prise en mémoire, consommées par un système basé sur des N-grammes est en général bien plus important que pour un système basé sur une grammaire. Néanmoins, les optimisations utilisées par un moteur de reconnaissance tel que Julius (voir point suivant) ajoutées à une modélisation moins précise (et donc moins performante, mais plus rapide) pourraient venir à bout de cet argument.

Enfin, les tâches que nous envisageons sont suffisamment précises pour envisager de définir une grammaire pour chacune d'elles (salutations, guidage, manipulation d'objets, etc).

c) Moteur de reconnaissance

Afin de satisfaire aux exigences de notre plateforme robotique (ressources mémoire et processeur limités et partagés avec d'autres modules, quasi temps réel obligatoire), nous avons choisi d'utiliser un moteur de reconnaissance nommé Julius. Julius est un logiciel libre développé par le "Continuous Speech Recognition Consortium" [Lee et al., 2001] au Japon. Il est décliné en deux versions :

– Julius, qui utilise des N-grammes (représentation probabiliste) pour modèle de langage,
– Julian, qui utilise des grammaires (représentation par règles).

Julian, la version de Julius utilisant une reconnaissance par grammaire, utilise les ressources lexicales et grammaticales sous forme de deux fichiers dont la syntaxe suit l'exemple donné dans le tableau I.2. La grammaire (partie I.2b du tableau) respecte la définition d'une grammaire hors contexte : « `S` » représente l'axiome, les expressions séparées par des « **:** » sont les règles de production et tous les mots ici en majuscules sont des symboles non-terminaux, également appelés catégories. Le lexique phonétique (partie I.2a du tableau) contient le vocabulaire utilisé par le moteur de reconnaissance et constitue la deuxième partie de la grammaire : les mots précédés par des « `%` » sont les catégories de plus bas niveau de la grammaire, les mots qui les suivent sont les symboles terminaux et les symboles qui suivent chaque mot forment leurs prononciations. Cette grammaire permet de générer les phrases suivantes : « Ceci est un bol. »,

« Ceci n'est pas un bol. », « Ça c'est pas de bol. », plus un certain nombre de phrases « parasites » comme « Ça est un bol. ». Il est à noter que dans la réalité, les « c' » et « n' » ne seraient jamais insérés de cette manière dans un lexique : on préfère toujours utiliser des expressions entières (ici « c'est » et « n'est ») car les mots trop courts se laissent trop facilement insérer et génèrent trop d'incertitudes lors de la recherche dans le treillis de mots. De même, pour obtenir un système de reconnaissance efficace, on évite les phrases parasites, quitte à complexifier l'écriture de la grammaire. Mais dans cet exemple, nous avons cherché à montrer comment une grammaire permet de combiner des mots, puis des expressions, même avec un vocabulaire et un nombre de phrases cibles très réduites, d'où ces incohérences.

```
%PAUSE_D              %EST              %DE
<S>       pause       est   E           de      vcl d @
%PAUSE_F              est   E cl t      de      vcl d
<\S>      pause       est   e           %BOL
%PAS                  est   e cl t      bol     vcl b O l
pas       cl p a      %UN               %SUJET
pas       cl p a z    un    U~          Ceci    s @ s i
%_C_                  %_N_              Ça      s a
c'        s           n'    n
```

(a) Exemple de vocabulaire pour Julian.

```
S :              PAUSE_D SUJET GROUPE_VERBAL OBJET PAUSE_F
GROUPE_VERBAL :  EST
GROUPE_VERBAL :  _N_ EST PAS
GROUPE_VERBAL :  _C_ EST PAS
OBJET :          UN BOL
OBJET :          DE BOL
```

(b) Exemple de grammaire pour Julian.

TAB. I.2: Exemple de ressources lexicales et grammaticales construites pour Julian.

Après avoir réorganisé ce type de grammaire sous forme d'automate à état fini, Julian utilise ces données en deux passes. La première passe consiste en une recherche approchée (*via* une méthode d'élagage d'arbre, ou "beam search") gauche-droite basée sur des contraintes faibles : dans cette phase, seule les contraintes inter-catégories sont prises en compte. Cette première passe a pour but d'éliminer rapidement les hypothèses les moins probables afin de gagner du temps pour la seconde passe, plus précise. Le fait que seules les contraintes inter-catégories soient prises en compte signifie que la seule contrainte extraite de la grammaire et utilisée ici est qu'une catégorie qui n'en suit pas une autre dans la grammaire ne fera pas partie de l'arbre de recherche. La seconde passe remonte le graphe construit par la première de droite à gauche en recalculant les scores de manière plus précise et en utilisant cette fois toutes les contraintes

de la grammaire. La solution optimale est trouvée grâce à un algorithme de type A*. Les scores de chaque hypothèse, qui permettent cette recherche, sont calculés grâce à une combinaison des scores de chaque HMM (obtenu par un algorithme de type Viterbi) et des pénalités d'insertion de mots. La figure I.4 donne un exemple du treillis de mots qui aurait pu être construit par Julian à partir de la grammaire I.2. Au final, la sortie de cette phase de reconnaissance est une liste des N phrases les plus probables d'après le moteur (souvent appelée *N-best*), ainsi que leurs scores.

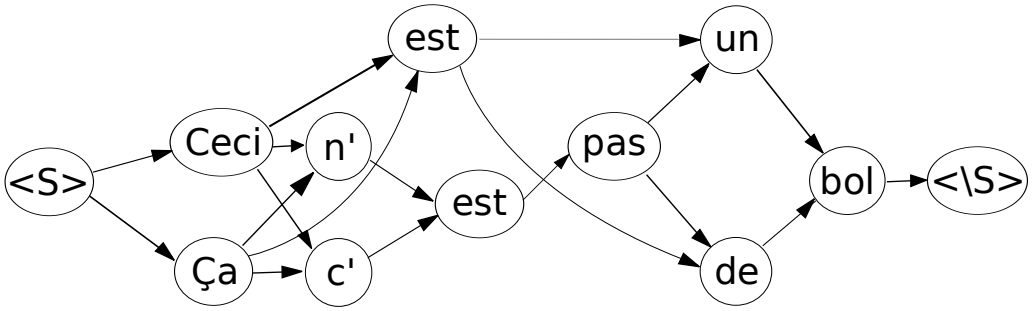


FIG. I.4: Exemple de treillis de mots.

L'ensemble de la phase de reconnaissance, utilisant comme ressources nos modèles de phonèmes, notre lexique phonétique et notre grammaire, est représentée dans le second tiers de la figure I.1. Elle est effectuée dans notre module par le moteur de reconnaissance Julian. Les modèles de phonèmes basiques qui ont été présentés ici, la manière d'utiliser la grammaire, ainsi que Julian, ont été modifiés durant cette thèse dans divers but. C'est ce que présentent le point et la section suivante.

I.3 Compréhension de la parole dans le contexte IHR

Les systèmes LVCSR ont pour tâche la transcription automatique de grandes quantités de données par exemple à des fins d'indexation de contenus audio ou audio-visuels, mais leur tâche s'arrête souvent là car le but de l'utilisateur final sera la recherche par mots clefs dans le texte transcrit (pour un moteur de recherche par exemple) ou tout simplement la lecture du texte transcrit (pour un mal-entendant). Dans notre cas, la transcription de la parole elle-même n'a pas d'intérêt direct pour l'utilisateur du robot : celui-ci veut que le robot obéisse à ses ordres. Nous ne pouvons donc pas nous arrêter à cette tâche de transcription, il nous faut maintenant interpréter les phrases reconnues, c'est-à-dire en extraire les informations pertinentes sous une forme exploitable par le robot, et plus précisément par un script ou par un programme de supervision (voir le chapitre IV pour une explication sur la supervision).

Cette section présente dans un premier temps les principes généraux de la compréhension de la parole avant de se focaliser sur notre application robotique.

I.3.1 Principe de la compréhension

D'une manière générale, en traitement du langage naturel, l'interprétation d'une phrase se divise en deux étapes : l'analyse syntaxique et/ou l'analyse sémantique. La première étape consiste à mettre en évidence la structure de la phrase à traiter. Cette opération suppose une formalisation des phrases qui sont alors définies par des règles de syntaxe formant une grammaire formelle. Connaître la structure syntaxique d'un énoncé permet d'expliciter les relations de dépendance (par exemple entre sujet et objet) entre les différents mots (ou groupes de mots), puis de construire une représentation du sens de cet énoncé. La seconde étape se sert alors de cette structure afin d'établir la signification de la phrase en utilisant le sens des éléments de la phrase dans leur contexte.

Pour un module de reconnaissance et de compréhension de la parole comme le nôtre, la phase d'interprétation représente la dernière partie du processus complet de traitement de la parole, et est représentée dans la dernière partie de la figure I.1. Dans ce cadre, il s'agit d'extraire les unités sémantiques significatives de phrases reconnues par le système, correctes ou non (c'est à dire correspondant ou non à la phrase réellement prononcée), avant de construire, *via* un nouveau modèle de langage, une interprétation compréhensible par la machine.

Dans notre cas, l'interpréteur fonctionne en deux passes, toutes deux basées sur des associations de modèles. Un lexique sémantique a été spécifiquement conçu pour associer des mots ou des expressions à leur interprétation. Certains mots sont liés aux actions qu'il est possible d'effectuer, tandis que d'autres sont liés aux objets, aux attributs d'objet comme la couleur, la taille ou l'emplacement, ou encore aux paramètres de configuration du robot (vitesse, rotation, distance). Le processus d'interprétation fonctionne alors, à partir d'une phrase reconnue, de la manière suivante :

1 . Une première passe recherche la tâche exprimée par la phrase. Elle n'a, en fait, pas pour but une désambiguïsation totale de la phrase, mais la détermination d'une liste de tâches possiblement exprimées par la phrase.

2 . La seconde passe se base alors sur ces possibilités et sur une grammaire liée au contexte (ici une tâche) afin d'en définir les paramètres. Si malgré l'application de ces règles locales, des ambiguïtés restent possibles, l'ensemble [tâche+paramètres] contenant le plus grand nombre de paramètres définis sera choisi.

Enfin, l'ensemble de ces données sémantiques récoltées sont fusionnées en une interprétation globale conforme à l'un de nos modèles d'interprétation défini par les besoins de la supervision.

I.3.2 De l'interprétation d'un énoncé à la commande destinée au robot

Voyons maintenant comment ce principe s'applique dans notre contexte robotique.

a) Énoncés classique, énoncés déictiques

Dans notre contexte robotique, notre interpréteur doit être capable d'interpréter l'ensemble des tâches qu'un robot est susceptible de se voir assigner. Les tâches les plus classiques consistent en des déplacements du robot, des demandes d'informations, et autres actes de communication n'impliquant aucune connaissance autre que celles dont dispose le robot en interne ou celles données dans la phrase elle-même. Quelques exemples de telles phrases sont données dans la première partie du tableau I.1.

Mais notre robot doit également être capable de répondre à des demandes plus complexes, notamment celles contenant une référence à l'homme. Il s'agit alors de tâches nécessitant un langage spatial, c'est-à-dire des tâches telles que : un mouvement par rapport à la position de l'utilisateur, une action basée sur un geste de pointage, ou encore une interaction physique avec ce dernier (par exemple un échange d'objet). La deuxième partie du tableau I.1 donne quelques exemples.

b) Implémentation

En pratique, notre interpréteur commence par rechercher une action (viens, tourne, avance, recule, cherche, conduit, guide, etc) ou une expression d'interaction (bonjour, au-revoir, oui, non, etc) qui lui permettra d'associer à la phrase cible un modèle d'interprétation. Puis, il recherche les paramètres de l'action trouvée (ou des actions s'il y a ambiguïté) : *3 mètres*, *20 degrés*, *un peu*, *le verre*, etc.

```
Quel est l'état de tes batteries ?    Conduis-nous à la salle de robotique.
Donne moi l'état de tes batteries.    Guide-nous jusqu'à la salle Gérard Bauzil.
```

(a) Exemple de phrases à interpréter.

```
INFO = V_INFO + O_INFO
V_INFO : l'état
O_INFO : batteries
```

```
ACTION = V_ACTION + O_ACTION
V_ACTION : Conduit-nous
V_ACTION : Guide-nous
O_ACTION : salle de robotique
O_ACTION : salle Gérard Bauzil
```

(b) Exemple de grammaire sémantique modélisant les phrases précédentes.

TAB. I.3: Exemple de ressources lexicales et grammaticales construites pour Julian.

Afin d'effectuer cette tâche, le système a donc besoin d'une grammaire qui lui permette de faire correspondre des mots ou groupes de mots d'une phrase avec des unités sémantiques. Illustrons ce processus d'interprétation par un exemple simple. Le tableau I.3 prend pour exemple quatre phrases à interpréter. Les grammaires sont ici composées des axiomes « `INFO` » et « `ACTION` », de règles de production (« `:` » et « = », ici « + » permet de séparer plus clairement les symboles non-terminaux entre eux) et de symboles terminaux (en minuscules) et

non-terminaux (en majuscules). Afin d'interpréter une phrase, l'interpréteur commencerait donc ici par rechercher `V_INFO` et `V_ACTION` dans la phrase, car il s'agit du symbole le plus discriminant et qui permet de définir le type de tâche auquel la machine aura à faire face. Ensuite on recherchera les `OBJET` qui permettront de définir l'objet de la tâche.

c) Limites de cette approche

L'exemple I.3 montre un cas simple : sans ambiguïté possible entre les deux modèles d'interprétation et sans besoin d'une gestion quelconque de l'ordre des mots dans la phrase. Des ambiguïtés pourraient être générées par la présence dans la grammaire de deux phrases telles que « Jido, avance moins vite. » et « Jido, avance ton bras. ». En effet, leurs modèles d'action (« avance ») sont identiques, mais leurs paramètres ([« moins »,« vite »] et « bras ») sont différents en nombre et en qualité, ce qui permet à la seconde passe de notre algorithme de lever l'ambiguïté. De même, si l'on ajoute dans les possibilités une phrase telle que « Jido, avance vers moi. », il est possible de supprimer l'ambiguïté créée en spécialisant le modèle d'action (« avance vers »). Force est donc de constater que s'il est possible, dans la plupart des cas, de lever toute ambiguïté pour la phase d'interprétation, cela se fait au dépend de la commodité de la grammaire. En effet, l'augmentation du nombre de phrases reconnaissables par le système entraîne un supplément d'ambiguïtés qui devront se traduire par une plus grande complexité de la grammaire d'interprétation.

D'autre part, la gestion de l'ordre des mots modélisant dans une phrase est difficile avec une telle approche. Par exemple, si l'on ne prend pas en compte l'ordre des mots dans une phrase telle que « Pose le cube bleu sur le cube rouge. », il est impossible d'être sûr que les couleurs seront correctes dans son interprétation. Mais une telle prise en compte n'est possible qu'au prix d'un accroissement de la complexité et de la rigidité de la grammaire d'interprétation.

Ainsi, dans tous les cas, lorsque la grammaire qui a généré ces phrases s'étend, nous nous apercevons que la grammaire qui doit les interpréter devient de plus en plus lourde et figée. De plus, chaque modification de la grammaire de reconnaissance doit s'accompagner d'une modification, parfois conséquente à cause de nouvelles ambiguïtés, de la grammaire d'interprétation.

Partant de ce constat, cette approche, suffisante dans un premier temps afin de simplifier le problème, ne peut pas être conservée à long terme. C'est pourquoi nous avons décidé de la faire évoluer, ce qui fera l'objet de la sous-section I.4.2.

d) Évaluation de la compréhension

La sortie de notre système de reconnaissance et de compréhension de la parole doit alimenter un module de fusion ou un superviseur (voir chapitre IV). Le système ne pourra donc être évalué que sur la qualité des interprétations qu'il fournira à partir des phrases prononcées par l'utilisateur. Il convient par conséquent de définir une nouvelle mesure : le taux d'erreur sur les interprétations (ou IER, pour "Interpretation Error Rate"). Ce taux ne représente pas, contrairement par exemple à [J. Goulian, 2003], un taux d'échec de l'interpréteur après reconnaissance, mais le taux d'interprétations erronées par rapport à l'interprétation voulue lors de la prononciation d'une phrase. En effet, nous modélisons dans notre système l'ensemble des phrases de

la grammaire de reconnaissance, ce qui signifie que l'interpréteur ne connait aucun échec après reconnaissance (sauf erreur dans la conception de la grammaire d'interprétation). Une conséquence est que, comme nous le verrons dans la section I.5, cette mesure annoncera toujours un taux d'échec inférieur à celui du SER, deux phrases pouvant avoir la même interprétation.

I.4 Intégration et améliorations

Nous sommes ici dans le cadre d'un module destiné à être utilisé sur nos robots. Cette section détaille les améliorations apportées au module de base, décrit précédemment, afin de l'adapter aux objectifs et contraintes de notre contexte.

I.4.1 Adaptation des modèles acoustiques

Les premiers résultats obtenus par notre module de traitement de la parole nous ont poussé à explorer un certains nombre de pistes afin d'améliorer nos modèles acoustiques.

a) Adaptation au contexte sonore

Comme nous l'avons vu précédemment, nos modèles de phonèmes ont été appris sur des enregistrements d'émissions radiophoniques. Ils ne sont, par conséquent, pas très adaptés à notre utilisation puisque ni l'ambiance sonore ni la façon de parler ne sont les mêmes que dans notre contexte de robotique mobile. En effet, les bruits de fond sont totalement différents tout comme la qualité de l'enregistrement : nous utilisons pour communiquer avec le robot un microphone sans fil monté sur serre-tête qui n'a ni les mêmes bandes passantes, ni les mêmes caractéristiques que les microphones d'un studio de radiophonie.

Afin d'améliorer les performances de notre système de reconnaissance de la parole (voir la section I.5), nous avons donc utilisé notre premier corpus de test pour en faire un corpus d'apprentissage à partir duquel nous avons réestimés les modèles phonétiques de bases dont nous disposions. Cette réestimation a été effectuée *via* l'implémentation de la boîte à outils HTK de l'algorithme de Baum-Welch [Baum and Petrie, 1966] sur des données qui sont donc indépendantes du locuteur et désormais adaptées à notre contexte.

b) Modélisation de mots critiques

Après le dépouillement des premières évaluations effectuées, il est apparu qu'au delà des problèmes de reconnaissance purement liés à la qualité des phonèmes appris les phrases les plus courtes, et notamment les mots seuls, constituaient le plus grand nombre d'échecs. Or, ces mots seuls s'avèrent parfois les plus importants, comme par exemple le mot « Stop ! » qui doit être un ordre disponible pour les utilisateurs du robot de la même manière qu'une expression plus longue comme « Arrête-toi Jido. ». Partant de ce constat, nous avons choisi divers mots-clefs

(« stop », « oui », « non », etc), avec les mots pouvant être utilisés seuls en priorité, et les avons modélisés chacun par un unique HMM.

c) Les difficultés de la parole spontanée

Dans le contexte de la parole spontanée, tout système de reconnaissance est confronté aux problèmes des disfluences, c'est à dire d'inconsistances par rapport à une grammaire spécifique. Ces disfluences se présentent notamment sous la forme de répétitions ou d'hésitations et peuvent représenter jusqu'à 20% des mots lors d'une conversation [Eklund, 2000]. Dans le cas d'hésitations, des pauses plaines sont intercalés dans la phrase prononcée afin de combler les silences créés par le temps de réflexions, on parle alors de mots de remplissage (en anglais, "fillers") : [Siegel, 2002] étudie par exemple l'utilisation du mot « like » en langue anglaise. Pour donner un exemple, lorsqu'il hésite ou réfléchi à ce qu'il veut faire faire au robot, il n'est pas rare pour un utilisateur d'insérer des « euh » dans ses phrases ou de bégayer légèrement en répétant des syllabes voire des mots. De même, il arrive à chacun de nous d'ajouter des mots comme « ben » ou « bon » à nos phrases, même si cela est totalement inutile. Aucune de ces hésitations ou répétitions n'étant modélisées, notre système ne peut les gérer correctement et aura tendance à essayer de les remplacer par d'autres mots, ce qui est évidemment néfaste pour la reconnaissance (le lecteur intéressé pourra consulter la thèse de [Eklund, 2004] pour de plus amples explications sur les disfluences et leurs implications).

d) Modélisation de disfluences

Pour un système dont la reconnaissance est basée sur des N-grammes, ces phénomènes ne posent aucun problème lors de cette phase, mais doivent être prises en compte lors de la phase d'interprétation. Ainsi, [J. Goulian, 2003] construit des graphes de dépendances dont le coût permet de hiérarchiser les solutions, permettant ainsi de gérer des mots ou groupes de mots insérés dans une phrase. Mais dans le cadre d'une reconnaissance par grammaire, et étant donné notre système plus simple d'interprétation, il faut envisager un autre type de solution.

Dans le cas d'hésitations, une proposition est de créer des modèles de mots comme « euh ». Pour les répétitions, deux solutions s'offrent à nous : la création de modèles de mots spécifiques, ou la création de nouvelles entrées dans le lexique prenant en compte ces répétitions comme des prononciations supplémentaires en utilisant les phonèmes existants. Nous avons tenté d'évaluer cette alternative (voir la section I.5), mais ces tentatives restent peu concluante. En effet, le principal problème ici n'est pas la manière de modéliser ces disfluences, mais de les insérer dans le treillis de mots. Pour nos évaluations, tous les modèles créés ont été insérés à la main dans nos grammaires. Or, non seulement une telle tâche est rébarbative, mais pour une grammaire grandissante et évoluant souvent, cela devient impossible à gérer. Il faudrait par conséquent que ces insertions puissent se faire automatiquement soit dans la grammaire (en modifiant le lexique par exemple), soit directement dans le treillis de mot durant la reconnaissance (de la même manière que sont insérés automatiquement des silences entre les mots). La seconde solution est bien entendu préférable puisqu'elle économise de la mémoire et du temps de calcul. Malheureusement, nous n'avons pas eu le temps, durant cette thèse, de mener ces investigations à leur terme.

I.4.2 Généralisation de l'écriture des grammaires

a) Motivations

Comme nous l'avons vu dans la sous-section I.3.2, les principaux problèmes de l'interprétation telle qu'elle a été définie précédemment (et telle qu'elle est utilisée dans nombre de systèmes) sont sa lourdeur et sa rigidité. En effet, en se remettant dans notre contexte robotique, quand une nouvelle utilisation du robot nécessite de comprendre de nouveaux ordres, nous écrivons une nouvelle grammaire afin de permettre au système de reconnaissance de les reconnaître, puis nous écrivons une nouvelle grammaire afin de pouvoir extraire les informations de ces nouvelles reconnaissances. Or, conceptuellement, les phrases à reconnaître, et donc la grammaire de reconnaissance, découlent des ordres que l'on veut voir exécutés par le robot, c'est-à-dire de la grammaire d'interprétation. Partant de ce constat, nous avons modifié notre approche afin d'atteindre différents objectifs :

1 . une plus grande simplicité d'utilisation des grammaires,

2 . un système plus générique, tant au niveau de l'indépendance des plateformes robotiques que de la conception des grammaires.

b) Principe

Afin d'atteindre ce nouvel objectif, nous définissons une grammaire, que nous appellerons de haut niveau, décrivant à la fois le niveau syntaxique et sémantique des ordres à traiter. Nous disposons également d'un lexique contenant le vocabulaire lié à cette grammaire. Cette nouvelle grammaire contenant l'ensemble des informations syntaxique, elle permet à notre système, lors de son initialisation, de générer un ensemble de grammaires sous une forme adaptée à notre moteur de reconnaissance (en l'occurrence Julian). Ces grammaires générées sont alors utilisées de manière standard durant la phase de reconnaissance. Mais elles permettent également, une fois la reconnaissance effectuée, de remonter à travers l'arbre de reconnaissance construit par Julian durant cette phase afin d'identifier pour chaque phrase reconnue, la grammaire dont elle est issue. Cette méthode nous permet donc, en générant une grammaire différente pour chaque tâche ou modèle d'interprétation, de retrouver le modèle lié à une phrase sans post-traitement et sans ambiguïté possible. De plus, les informations syntaxico-sémantiques contenues dans notre grammaire de haut niveau permettent d'extraire les paramètres de l'action, sans erreur possible sur l'ordre des mots, leur position étant connue.

Ainsi, en plus de faire disparaître les principaux problèmes dont souffrent les méthodes inspirées du *template matching*, cette méthode enlève beaucoup de lourdeur dans le processus d'écriture de grammaires : pour ajouter une nouvelle action, on ne complète plus qu'une seule grammaire (au lieu des deux syntaxique et sémantique) et il n'est plus nécessaire de se soucier des ambiguïtés que pourrait générer ce rajout. Il est cependant à noter que cette méthode n'est évidemment pas applicable à un système de reconnaissance basé sur une modélisation statistique.

c) Exemple

Afin d'illustrer de manière pratique l'utilisation de cette grammaire, prenons comme exemple le tableau I.4, dont les règles sont les suivantes :

1 . Les différents types d'actions sont indiqués en tête de chaque modèle d'interprétation (balise `<action>`).

2 . Les paramètres liés à ces actions y sont encapsulés (balise `<params>`).

3 . Les phrases génériques sont encapsulées dans les paramètres auxquels ils correspondent (balise `<phrase>`).

```
<action> T-GET_OBJECT
    <cap> bras
    <params> NIL
        <phrase> PRENDS_CA </phrase>
    </params>
    <params> object
        <phrase> PRENDS LE $object/M </phrase>
        <phrase> PRENDS LA $object/F </phrase>
    </params>
    <params> object color
        <phrase> PRENDS LA $object/F $color/F </phrase>
        <phrase> PRENDS LE $object/M $color/M </phrase>
    </params>    </cap>
    <cap> GEST
    <params> object this=THIS
        <geste type="POINTING"/>
        <phrase> PRENDS CE $object/M </phrase>
        <phrase> PRENDS CETTE $object/F </phrase>
    </params>
    </cap>
</action>
```

TAB. I.4: Exemple d'action dans notre grammaire de haut niveau.

La première tâche de notre module de reconnaissance et de compréhension de la parole est de générer des grammaires à partir des modèles décrits dans ce fichier. `NIL` signifiant ici l'absence de paramètre, la première grammaire générée contiendra simplement un axiome produisant la catégorie « `PRENDS_CA` ». La seconde grammaire contiendra les deux « phrases génériques » de la seconde sous-action, plus les règles de production de chaque objet formant le paramètre de cette sous-action. Ici, par exemple, « `$object/M` » désigne tous les objets masculins du vocabulaire. Les deux grammaires suivantes seront générées de la même manière.

Ces grammaires étant générées, la seconde tâche de notre module est de traiter et de reconnaître une phrase prononcée par un utilisateur. Une fois cette phrase traitée, nous retrouvons

la liste des N meilleurs résultats qu'il nous faut interpréter. Le système remonte alors dans les treillis de mots afin de retrouver la grammaire originelle de chacun de ces résultats. Si par exemple la phrase « Prends la bouteille verte. » a été reconnue, on sait que l'action qui lui correspond est `T-GET_OBJECT` et que les paramètres qui la constituent sont `object` et `color`. Après avoir extrait les paramètres de la phrase et mis en forme le résultat, nous obtenons une interprétation de la forme :

```
(DIALOG-TASK-INPUT(task-type T-GET_OBJECT)(params (object BOTTLE)(color GREEN)))
```

Au final, la sortie du module complet est constituée non plus des phrases reconnues, mais de la liste de leurs N interprétations accompagnées des scores calculés lors de la phase de reconnaissance. On pourra alors envoyer le premier membre de cette liste (c'est-à-dire la meilleure interprétation) au superviseur, ou envoyer l'ensemble de la liste au module de fusion de données audio-visuelles qui sera présenté dans le chapitre IV.

d) Vers une plus grande généricité et pour un système multi-plateforme

L'un des objectifs de cette thèse est de développer des outils génériques, qui puissent fonctionner sur tout robot en nécessitant le moins de modification possible. Or, chaque robot a des capacités (bras articulé, capacités sensorielles,etc) et des tâches à effectuer différentes. Il est donc nécessaire de contextualiser le traitement de la parole en fonction de ces particularités afin de ne pas avoir à créer des systèmes dédiés pour chacun. C'est le sens de la balise `<cap>` que nous retrouvons dans le tableau I.4 : ils précisent ici que les trois premières actions nécessitent un bras, tandis que la dernière nécessite en plus le module ***GEST*** (c'est-à-dire le suivi et la reconnaissante de gestes, voir chapitres II et III). Ainsi, si un robot ne possédant pas de bras entame un processus de compréhension de la parole basé sur une grammaire de haut niveau contenant cette description, aucune grammaire ne sera générée pour cette action : celle-ci étant impossible, il est peu probable qu'un utilisateur essaye de la faire exécuter par le robot.

De la même manière, afin de rendre notre grammaire multimodale et ainsi qu'elle serve à la compréhension de la parole, mais également au module de fusion, nous y avons ajouté la balise `<geste>` qui permet de préciser qu'un certain geste ou type de geste peut ou doit accompagner la parole prononcée (pour de plus amples explications, voir chapitre IV). Au final nous obtenons une grammaire permettant une maintenance et une évolution aisée, et adaptative aux robots sur lesquels elle est implantée et à leurs capacités.

I.4.3 Calcul de scores de confiance en vue de la fusion

a) Problème et stratégies de calcul d'un score de confiance

Le score de sortie du moteur de reconnaissance est une log-vraisemblance concaténant les scores de chaque HMM cumulant chaque mot de la phrase et les diverses pénalités utilisées. Mais dans une application réelle, comme la notre, différents problèmes empêchent une reconnaissance de qualité. D'une part, le bruit ambiant, la diversité des locuteurs, les distorsions

matérielles, etc, doivent être gérés par le moteur de reconnaissance. C'est ce que l'on appelle la reconnaissance de parole robuste et nous avons abordé cette problématique dans la sous-section I.4.1. D'autre part, il est inévitable que le processus de reconnaissance soit entaché d'erreurs, il est donc important de pouvoir mesurer la fiabilité des sorties d'un système de reconnaissance, c'est à dire de calculer un score de confiance pour chacune de ses sorties. Dans notre cas, ce besoin d'un score fiable est d'autant plus important que celui-ci doit pouvoir être utilisé lors d'une fusion probabiliste avec la reconnaissance gestuelle (voir chapitre IV).

D'une manière générale, on peut classer les méthodes de calcul d'un score de confiance en trois catégories, comme cela est fait dans [Jiang, 2005].

1 . Le type de méthodes le plus répandue [Schaaf and Kemp, 1997, Chase, 1997, Benitez et al., 2000, San-Segundo et al., 2001] est basé sur une combinaison de données collectées durant la phase de décodage. Ces données sont extrêmement variées et vont d'informations acoustiques à des informations de bout de chaîne comme une log-vraisemblance normalisée.

2 . Comme nous l'avons vu dans la section I.2, les systèmes de reconnaissance vocale s'appuient sur le calcul du maximum a posteriori exprimé par l'équation (I.1). Dans ce cadre, la probabilité $P(A)$ d'observer A est souvent ignorée en pratique car constante sur l'ensemble des séquences W. Or, ce n'est qu'une fois normalisé par cette probabilité que le maximum a posteriori est une mesure quantitative absolue de la correspondance entre A et W. Malheureusement, il est impossible de le calculer de manière exacte, c'est pourquoi diverses méthodes d'approximation ont été imaginées [Wessel et al., 2001, Young, 1994, Kamppari and Hazen, 2000].

3 . Il est également possible d'utiliser un post-traitement afin de déterminer la fiabilité de l'hypothèse de décision choisie par le processus de reconnaissance. Il s'agit alors de calculer un ratio de vraisemblance sur cette hypothèse afin de déterminer si elle doit être acceptée ou refusée [Rose et al., 1995, Sukkar and Lee, 1996, Rahim et al., 1997].

Pour plus de détails sur ces méthodes et leur catégorisation, le lecteur pourra consulter l'étude approfondie menée par [Jiang, 2005].

b) Notre approche

Nous cherchons ici à obtenir un score de confiance plus significatif que les log-vraisemblances dont nous disposons. Nous utilisons pour cela une méthode faisant partie de la première catégorie définie précédemment. Elle est inspirée d'une méthode de diminution du taux d'erreur sur les mots (WER) décrite dans [Kobayashi et al., 2007], sans prétendre aller aussi loin que ces auteurs. Pour ce faire, désignons $H = \{h_i\}_{i=1,\dots,N_s}$ comme étant la liste des N_s meilleures phrases (*N-best*). Nous définissons alors le score de confiance d'une hypothèse h_i comme étant :

$$S(h_i) = \frac{L(h_i)}{N_{W_{h_i}}} \cdot \frac{\sum_{w_i \in h_i} CS(w_i)}{\sum_{w_j \in H} CS(w_j)},$$

avec $N_{W_{h_i}}$ le nombre de mots dans l'hypothèse h_i, $L(h_i)$ la vraisemblance de h_i normalisée et $CS(w_j)$ une vraisemblance du mot w_j. En réalité, la fonction $CS(.)$ est un score par mot basé sur la liste des *N-best* : pour chaque mot w apparaissant dans la liste des *N-best* nous calculons $CS(w)$ en fonction de son taux d'apparition dans la liste, c'est à dire de la manière suivante :

$$CS(w) = \frac{\sum_{h_i \ni w} L(h_i)}{\sum_{h_j \in H} L(h_j)}.$$

Le calcul d'un tel score de confiance nous permet de réordonner légèrement la liste des N meilleures hypothèses. Mais son impact n'est pas réellement significatif, comme nous le verrons dans les évaluations de la section I.5 et comme [Stolcke et al., 1997] nous l'annonçait : non seulement le gain sur le taux d'erreur sur les mots est relativement faible, mais un tel gain ne se traduit pas forcément par une amélioration du taux de phrases bien reconnues. Malgré tout, cette méthode nous permet d'obtenir un score exploitable par la fusion de données (voir chapitre IV).

I.5 Évaluations

L'ensemble de ce système de traitement de la parole a été implémenté sous la forme d'un module Genom (voir chapitre IV) nommé ***RECO*** et fonctionne sur tous les robots du LAAS équipés d'un microphone. Rappelons que notre objectif est ici de montrer la faisabilité d'un système de traitement entièrement embarqué sur une plateforme robotique, c'est-à-dire avec ses contraintes matérielles et logicielles.

I.5.1 Recueil de corpus

Afin, d'évaluer quantitativement notre système de reconnaissance vocale et de compréhension, nous avons construit un corpus sur lequel mener des évaluations. Ce corpus étant censé évaluer notre système dans les conditions les plus proches possible de son utilisation normale, il aurait été logique de construire notre corpus selon une procédure appelée « magicien d'Oz ». Cette procédure consiste à mettre en situation des utilisateurs naïfs (c'est-à-dire ne connaissant rien du système ni de la manière dont il fonctionne) et de simuler les réactions du système. Dans notre cas, cela aurait par exemple consisté à donner à un utilisateur un objectif à atteindre en donnant des ordres à l'un de nos robots. Nous aurions alors recueilli l'ensemble de ses ordres vocaux tout en faisant évoluer à son insu le robot comme s'il était réellement autonome, c'est-à-dire en obéissant à ses ordres, mais aussi en simulant des erreurs. Une telle construction aurait eu un intérêt double : la base de données aurait été à un haut degré de réalisme et nous aurions pu construire des grammaires plus adaptées et moins restrictives. Malheureusement, de

tels processus sont assez lourd à mettre en œuvre, et ce d'autant plus dans notre cadre. En effet, nos robots sont des plateformes de développement pour de nombreux autres domaines que la seule reconnaissance vocale, toutes les fonctionnalités que l'utilisateur attend du robot ne sont donc pas forcément disponibles et surtout risquent d'être lentes et difficiles à simuler discrètement. C'est pourquoi nous avons choisi de construire notre base de données à partir de phrases prédéterminées que l'utilisateur devra prononcer au fur et à mesure qu'il les découvre.

Le protocole de recueil de nos corpus a donc été le suivant.

1 . Choix de 50 phrases parmi les possibilités de notre grammaire et représentatives de la variété de nos requêtes.

2 . Les phrases apparaissent à l'écran et l'utilisateur doit les prononcer. Dès qu'il a terminé de prononcer une phrase, elle est enregistrée, traitée et la suivante apparait.

3 . Après avoir prononcé son jeu de phrase, l'utilisateur obtient son taux de reconnaissance par le système.

Ces recueils de données ont été effectués sur un ordinateur portable *via* son microphone intégré ou un microphone externe équivalent à ceux utilisés sur les robots. Ils ont été réalisés dans divers locaux, et donc environnements sonores, parfois silencieux, parfois encombrés de bruits de fonds (ordinateurs, autres personnes lointaines, claquement de porte). 16 personnes des deux sexes (dont 7 ne sont pas francophones natives et plusieurs autres ont un accent plus ou moins prononcé) se sont gentillement prêtées à l'expérience, répétant chacune le jeu de phrases une à quatre fois, durant trois campagnes de recueils : HRI_1 (1600 enregistrements, pour un total d'environ 35 minutes), HRI_2 (650 enregistrements, 14 minutes) et HRI_3 (550 enregistrements, 12 minutes).

I.5.2 Évaluations

Les résultats suivant ont été obtenus sur nos divers corpus en utilisant notre grammaire modélisant :

- 22 actions possibles,
- 54 sous-actions ou prototypes d'interprétation,
- un total de plus de 400 interprétations possibles,
- entre 2000 et 3000 phrases reconnaissables (suivant le robot),
- jusqu'à 500 mots.

Afin de simplifier la lecture des tableaux de résultats, ceux-ci sont tous affichés en terme de taux de reconnaissance (et non d'erreur). COR_W correspond ainsi à $1 - WER$, ACC à l'*accuracy*, COR_S à $1 - SER$ et COR_COM à $1 - IER$.

La première ligne du tableau I.5 présente les résultats globaux obtenus sur notre premier corpus en utilisant les modèles phonétiques de base (les modèles calculés durant la campagne ESTER, voir sous-section I.2). Comme nous l'avons vu au cours de la sous-section I.2, différentes investigations ont été menées afin d'améliorer ces résultats. Nous présentons dans le

reste du tableau leurs fruits en zoomant sur les mauvais résultats de ce premier corpus, c'est-à-dire sur les 451 fichiers sonores (représentant un peu plus de 10 minutes d'enregistrement) dont les phrases n'ont pas été correctement reconnues. $V1$ représente les résultats obtenus sur ce sous-corpus avec nos modèles phonétiques de base, tandis que les résultats $V2$ sont obtenus grâce aux modèles phonétiques réestimés sur un autre sous-corpus de HRI_1 formé des 1049 fichiers (environ 25 minutes) qui ont été correctement reconnus. Afin de prendre en compte les mots critiques et les disfluences qu'il est possible de rencontrer dans nos enregistrements, un corpus spécifique (que nous nommerons HRI_{DIS}) a été construit afin d'entraîner 6 modèles de disfluences (notés MD) et 26 modèles de mots (notés MM). Ces modèles sont toujours des HMMs, mais qui n'ont qu'une seule gaussienne par état car HRI_{DIS} est de trop petite taille pour apprendre un modèle plus complet. HRI_{DIS} est composé uniquement de phrases contenant les disfluences et les mots-clefs que nous cherchons ici à modéliser, le tout pour un total d'environ 8 minutes d'enregistrement.

version	COR_W	ACC	COR_S	COR_COM
HRI_1	82,06%	80,16%	61,31%	67,25%
$V1$	57,89%	53,87%	00,00%	17,60%
$V1 + MD$	64,19%	58,44%	09,44%	28,11%
$V1 + MM$	66,93%	63,40%	22,32%	33,48%
$V2$	90,60%	88,52%	71,24%	80,26%
$V2 + MD$	90,09%	87,84%	70,17%	80,47%
$V2 + MM$	89,95%	87,98%	72,75%	79,18%
$V2 + MD + MM$	90,00%	87,94%	72,75%	79,40%

TAB. I.5: Résultats sur HRI_1 et son sous-corpus de mauvaises phrases.

Comme nous pouvons le voir dans ce tableau, les améliorations les plus notables sont apportés par la réestimation des modèles. En effet, le contexte sonore dans lequel ces phonèmes ont été appris est véritablement différent du notre et cette grande différence implique des confusions entre phonèmes proches qui sont catastrophiques pour l'ensemble du processus de reconnaissance. Pour leur part, les modèles de mots, bien que très efficaces en eux-même comme le montre le tableau I.6, ne sont utilisés que dans une minorité de phrases. Leur impact est par conséquent moindre et on peut voir dans les résultats du tableau I.5 qu'ils peuvent même diminuer légèrement le taux de reconnaissance quand les modèles phonétiques deviennent bons : la multiplication des hypothèses entraîne alors de plus grandes pertes (due à l'insertion de modèles supplémentaires) que la précision de ces modèles supplémentaires n'apporte d'améliorations. Enfin, les modèles de disfluences sont ici les moins efficaces pour plusieurs raisons. Tout d'abord, comme nous l'avions expliqué dans la sous-section I.2, notre approche dans la construction comme dans l'utilisation de ces modèles est trop restrictive. Ensuite, la manière dont a été acquis notre corpus influe ici grandement : quand une personne lit des phrases, même inconnues au départ, elle hésite peu et en tout cas beaucoup moins que dans une situation plus réaliste dans laquelle elle devrait réfléchir à ce qu'elle doit dire. C'est pourquoi notre corpus contient trop peu de disfluences pour que l'impact de tels modèles soit significatif.

version	(a) « prends ça »	(b) « stop »
$V1$	28,12%	25,00%
$V1 + MM$	62,50%	93,75%
$V2$	90,63%	90,63%
$V2 + MM$	90,63%	93 .06%

TAB. I.6: Taux de phrases correctement reconnues pour deux phrases courtes.

Le tableau I.6, pointe l'influence des modèles de mots-clefs sur deux exemples de phrases courtes : (a) « prends ça » et (b) « stop ». Ces deux phrases se sont révélées avoir un taux de reconnaissance catastrophiques lors des premiers tests ($V1$). Les résultats affichés dans le tableau sont les taux de reconnaissance de phrases, qui sont ici équivalents au taux de bonnes interprétations. Les modèles d'hésitation utilisés modélisent directement les mots « stop » et « prends » et l'expression entière « prends_ça ». Même si, ici encore, la bonne influence de modèles phonétiques plus adaptés est criante, on distingue mieux l'apport de modèles de mots pour ce type de phrases.

Le tableau I.7 illustre les résultats globaux de nos différents corpus avec nos deux jeux de modèles phonétiques et en utilisant ou non des modèles de mots.

version	COR_W	ACC	COR_S	COR_COM
$HRI\_1(V1)$	80,16%	82,06%	61,31%	67,25%
$HRI\_2(V1)$	87,57%	88,09%	72 .00%	78,92%
$HRI\_3(V1)$	86,40%	87,36%	69,82%	76,00%
$HRI\_2(V1 + MM)$	88,84%	89,18%	73,85%	80,46%
$HRI\_3(V1 + MM)$	87,92%	88,56%	72,91%	79,09%
$HRI\_2(V2)$	92,81%	93,60%	82,62%	88,92%
$HRI\_3(V2)$	91,60%	92,34%	79,82%	84,91%
$HRI\_2(V2 + MM)$	93,20%	93,99%	83,23%	89,38%
$HRI\_3(V2 + MM)$	92,37%	92,98%	80,73%	86,00%

TAB. I.7: Résultats globaux sur l'ensemble de nos corpus.

Enfin, étant donné l'importance que prend la consommation, tant en temps processeur qu'en mémoire, d'un module fonctionnant sur des robots que l'on essaye de rendre les plus autonomes possible, il est important de préciser également ce type de performances. Notre module ***RECO*** fonctionne en moyenne avec un facteur temps réel de 0,23, c'est-à-dire que pour un signal ayant une durée d'une seconde, le module mettra 230 ms pour le traiter. De plus, le module ne nécessite pas plus de 15 Mo de mémoire vive pour fonctionner.

I.6 Conclusion

Nous avons présenté dans ce chapitre la partie de cette thèse concernant le traitement de la parole. Dans notre cadre robotique, nous avons développé un module nommé $RECO$ dédié à la reconnaissance et à l'interprétation de la parole continue. Ce travail est fonctionnel sur l'ensemble des robots du LAAS-CNRS équipés d'un microphone et permet à un utilisateur de donner au robot des ordres vocaux, en français. La figure I.5 rappelle notre architecture définie en introduction et complétée ici par le module $RECO$ décrit dans ce chapitre.

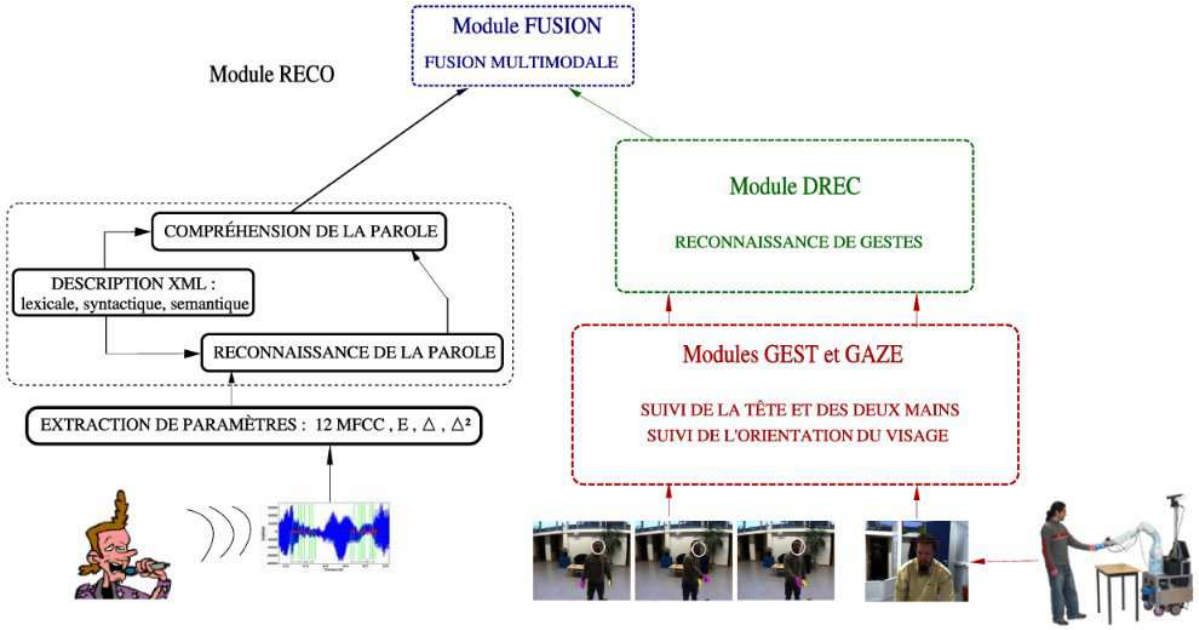


FIG. I.5: Architecture globale de notre interface homme-robot.

La première contribution apportée par ce module consiste en la capacité apportée aux robots de reconnaître de manière relativement efficace et rapide les paroles d'un utilisateur. Nous avons également développé un système d'interprétation de la parole générique *via* la génération de modèles de langage à partir d'une grammaire dite de haut niveau. Ces développements nous permettent de détecter le besoin d'un geste en complément de certaines paroles. Nous avons également ouvert différentes voix dans le post-traitement des résultats de la reconnaissance. Ces travaux ont contribué à la publication suivante : [Burger et al., 2009c].

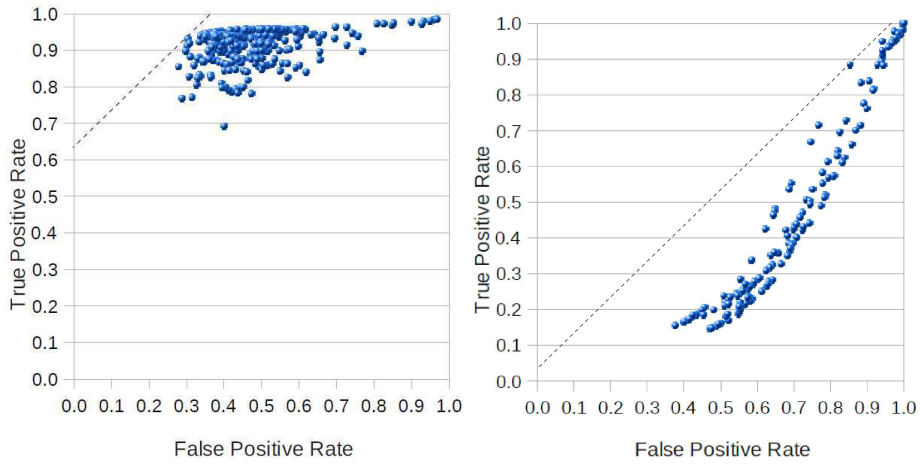
a) Perspectives

Malgré ces avancées, et le fait que ce système soit parfaitement fonctionnel sur nos robots, des investigations restent à mener. Nous pensons notamment à une prise en compte automatique des modèles d'hésitation et de répétition qui permettrait à ces derniers de développer tout leur potentiel. Dans la même lignée, il serait intéressant d'étendre ces modèles à des bruits courants (claquement de porte, etc).

D'autres développements plus périphériques sont également envisageables, comme par exemple emmener notre système vers une gestion de dialogue, fonction qui manque encore cruellement à nos robots et qui permettrait une prise en compte du contexte. Il serait également possible d'envisager d'y intégrer une reconnaissance de locuteur accompagnée d'une adaptation des phonèmes en direct pour les personnes les plus en contact avec le robot. Enfin, tenter d'intégrer un apprentissage en ligne de mots, basé par exemple sur une reconnaissance par phonème, et l'intégration de ces nouveaux mots dans la grammaire de façon dynamique serait un défi intéressant.

b) Travaux en cours

Nous inspirant des travaux de [Gabsdil and Lemon, 2004], et dans le but d'améliorer la fiabilité de notre système, nous cherchons à réordonner la liste des N meilleures interprétations grâce à une méthode d'apprentissage automatique en utilisant des données internes à notre moteur de reconnaissance. Nous utilisons un classificateur de type SVM, présenté en annexe B, qui est connu pour être un algorithme rapide et efficace, surtout quand il s'agit de traiter des dimensions élevées et des données hétérogènes.



(a) Avec utilisation de SVMs. (b) Avec utilisation d'un seuil.

FIG. I.6: Points ROC et lignes d'iso-coût optimal associées.

Dans l'état actuel de nos travaux, nous avons cherché à prouver que l'utilisation de SVMs, dans le but de déterminer si une interprétation est correcte ou non, est plus efficace qu'une méthode par seuillage basée sur les scores de confiance définis en sous-section I.4.3. La figure I.6 montre les points ROC (le lecteur ne connaissant pas cette méthode peut se référé à l'annexe C), et le front de Pareto associé, obtenus en faisant varier les paramètres libres de notre système pour ces deux méthodes. Chaque point représente le résultat d'un processus de validation croisée à 3 parties sur l'ensemble de nos corpus. Le taux EER ("Equal Error Rate") est déterminé par la ligne d'iso-coût dont le coût représente la somme des coûts de mauvaises classification des exemples positifs et négatifs. Le meilleur jeu de paramètres libres est apporté par le front de

Pareto ayant le plus faible EER, c'est-à-dire 0,12 pour la courbe des SVMs.

La domination de la méthode par SVMs s'explique notamment par l'impossibilité de trouver une formule de calcul d'un score de confiance convenable à partir d'un grand nombre de données hétérogènes, alors que le SVM est fait pour gérer ce type de vecteurs de données. Il est également à noter que cette phase de classification est négligeable, en terme de temps de calcul (2 ms), par rapport aux étapes de pré-traitement et de reconnaissance (facteur temps réel d'environ 0,23).

Ces travaux restent préliminaires et doivent encore être étendus afin d'être utilisés de manière effective dans notre module $RECO$ et évalués en terme d'*IER*. Nous pensons notamment à l'utilisation d'une cascade de SVMs qui permettrait d'échelonner leurs décisions et de produire un nouveau score pour chaque membre de la nouvelle liste des N meilleures interprétations.

Chapitre II

Perception visuelle de l'homme : suivi de gestes et suivi du regard

Ce chapitre présente nos systèmes de suivi de gestes manuels et de regard que nous avons développés et qui sont respectivement encapsulés dans les modules *GEST* et *GAZE* de notre architecture multimodale. Plusieurs raisons ont motivées ces développements. Tout d'abord, le suivi 3D des mains et de la tête permet à chaque instant image de caractériser grossièrement la position de l'homme, sous l'hypothèse que celle-ci soit assimilée à la position de la tête, relativement au système de vision, c'est-à-dire au robot. Cette information est vitale dans toute tâche dynamique coordonnée entre homme et robot, ce dernier devant rester à une distance non seulement cohérente (vis à vis de la tâche à exécuter) mais aussi sociale (vis à vis de l'homme). De plus, le suivi des mains relativement à la tête permet, dans une approche ascendante allant du suivi de geste à sa reconnaissance, de caractériser des gestes destinés au robot, c'est-à-dire des gestes symboliques ou déictiques (c'est-à-dire de pointage) dans notre contexte (le chapitre III donne une définition générale des gestes, mais précise également ceux que nous utilisons).

Suivre le regard, assimilé ici au suivi du visage, est également pertinent dans notre contexte robotique. Rappelons que la direction du regard permet de vérifier l'intentionnalité de l'homme durant tout mécanisme d'interaction homme-robot. De plus, les mouvements de la tête sont souvent cohérents/coordonnés avec certains gestes manuels déictiques, voire symboliques. Il semble alors naturel de corroborer les gestes manuels par les mouvements associés du visage afin de vérifier l'occurrence d'un geste manuel et éventuellement affiner la direction de pointage lors de gestes déictiques. Ceci est fait par le biais d'un mécanisme de fusion qui sera induit uniquement lors d'une interaction proximale homme-robot $[1m; 2{,}5m]$, par exemple autour d'une table car :

- les mouvements de la tête sont de faibles amplitudes,
- les points anatomiques du visage doivent être caractérisables dans l'image.

Dans ce cadre, une stratégie de coopération entre les deux traqueurs sera envisagée (voir le chapitre III) puisqu'il s'agit notamment de caractériser, certes pour des fonctionnalités spécifiques,

les mouvements de la tête. L'exécution simultanée de ces deux traqueurs sera évidemment subordonnée à la tâche en cours.

Rappelons que la finalité est d'intégrer le système complet sur un robot mobile autonome dont les ressources CPU sont par définition limitées et dont la réactivité est vitale pour qu'il soit accepté par son interlocuteur humain. Lors de nos développements, nous avons logiquement porté une attention particulière sur les temps de traitement associés à chaque traqueur. Pour factoriser nos efforts, les deux traqueurs tirent partie d'un seul et unique formalisme de filtrage : le filtrage particulaire [Arulampalam et al., 2002, Doucet et al., 2001]. Ce formalisme, abondamment référencé dans la littérature, nous a semblé très adapté ici. Il permet, en effet, de s'affranchir de toute hypothèse restrictive quant aux distributions de probabilité entrant en jeu dans la caractérisation du problème. Ses nombreuses variantes permettent, comme nous le verrons, de coller aux spécificités de chaque traqueur. Enfin, ce formalisme permet de fusionner aisément différentes mesures image dans un cadre probabiliste justifié.

Ce chapitre propose tout d'abord un rapide état de l'art afin de positionner nos travaux sur le suivi par rapport à la littérature (section II.1). La section II.2 rappelle le formalisme bien connu du filtrage particulaire. Les section II.3 et II.4 décrivent respectivement nos deux traqueurs de gestes et de visage en termes de stratégie de filtrage, de mesures images, d'implémentation ainsi que de résultats qualitatifs et quantitatifs associés. Enfin, la section II.5 résume les contributions associées à ce chapitre et énonce quelques perspectives.

II.1 État de l'art et positionnement de nos travaux sur le suivi

II.1.1 Suivi de gestes

Le suivi des extrémités des membres corporels supérieurs fait l'objet de nombreuses investigations dans la communauté Vision. Le lecteur pourra se référer aux deux études [Erol et al., 2007, Murphy-Chutorian and Trivedi, 2008a] respectivement sur le suivi des mains et de la tête. Certaines approches dévolues à l'IHR suggèrent des hypothèses discutables dans notre contexte robotique et sortent donc du cadre de nos travaux :

- capteurs intrusifs [Fels and Hinton, 1997],
- arrière-plans statiques voire peu encombrés [Huang et al., 2002, Isard and Blake, 1998b, Just et al., 2004],
- connaissance *a priori* sur l'apparence vestimentaire du sujet [Azad et al., 2007, Waldherr et al., 2000], etc.

Il est possible de considérer le geste selon les approches 2D ou 3D. Les premières reposent sur une modélisation 2D et une inférence limitée au seul plan image afin de suivre des gestes mono- [Bretzner et al., 2002, Chen et al., 2003, Corradini and Gross, 2000, Rogalla et al., 2004,

Thayananthan et al., 2003] ou bi-manuels [Hasanuzzaman et al., 2004, Jeong et al., 2002, Park et al., 2005, Zieren et al., 2002]. Elles sont dédiées à l'analyse de gestes plutôt fronto-parallèles au plan image alors que nous visons l'interprétation de gestes déictiques référençant *a priori* l'espace partagé par l'homme et le robot mobile. Nous privilégions donc une approche 3D afin d'inférer les mouvements 3D des extrémités corporelles suivis. Parmi les approches de ce type, citons ici la capture de mouvements humain [Moeslund et al., 2006] qui vise à appréhender l'ensemble de la structure cinématique, c'est-à-dire les différents degrés de liberté de tout ou partie du corps humain à partir de flux vidéo mono- ou multi-oculaire. Ces stratégies requièrent classiquement un modèle géométrique 3D et cinématique du corps humain dont la configuration spatiale est inférée après projection image [Deutscher and Reid, 2005, Fontmarty et al., 2007, Sminchisescu and Triggs, 2003] ou reconstruction 3D [Ziegler et al., 2006]. Ces approches 3D, par leur estimation exhaustive des degrés de liberté, sont pertinentes, mais le nombre de degrés de liberté (environ 20 pour les membres corporels supérieurs), comme la gestion d'un modèle 3D complet mais frustre, induisent des extrema locaux dans la fonction de coût et restent peu compatibles avec les ressources CPU d'un robot autonome. De part ces limitations, les systèmes embarqués de capture visuelle de mouvement (fiables et quasi temp-réel) sont marginaux, voire inexistants dans la communauté IHR. Enfin, il est à noter que la reconstruction de l'ensemble de la chaîne cinématique n'est en rien essentiel pour la reconnaissance ultérieure de gestes.

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

FIG. II.1: Erreurs observées lors de suivi multi-cibles par traqueurs indépendants : cibles éloignées et fonctionnement correct (a), erreur de labellisation lors de leur rapprochement (b), fusion après leur occultation mutuelle (c), problème de ré-initialisation après perte d'observabilité.

Aussi, et à l'instar de [Bernier et al., 2009], nous privilégions un modèle 3D éparts des extrémités corporelles, c'est-à-dire ellipsoïdes déformables des mains et de la tête, qui semble plus compatible avec notre contexte applicatif. Celles-ci sont inter-connectées par des contraintes géométriques flexibles donc mieux adaptées à la morphologie du sujet observé. Les déformations permettent la gestion conjointe de l'orientation 3D et des variations dimensionnelles (induite typiquement par une paume ouverte ou fermée). Cette modélisation minimaliste ne présume en rien de la main (gauche ou droite) qui effectuera le geste, voire autorise la reconnaissance de gestes bi-manuels (voir chapitre III). Citons ici quelques travaux [Bernier et al., 2009, Hasanuzzaman et al., 2007, Nickel and Stiefelhagen, 2006, Park et al., 2005] sur le suivi 3D de gestes bi-manuels dans les communautés IHM ou IHR. Classiquement, le principe est d'instancier un filtre/traqueur par extrémité à suivre : on parle alors de suivi multi-cibles (ou MOT pour "Mutiple Object Tracking". Une gestion indépendante de ces filtres, à l'instar de la plupart des approches 3D mais aussi 2D induit de nombreux problèmes :

1 . la fusion des filtres lorsque les cibles se rapprochent, voire s'occultent lors de l'exécution de gestes (figure II.1-c),

2 . plus largement l'association de données [Bar-Shalom and Jaffer, 1998] pour les différentes cibles suivies induisant par exemple une erreur de labellisation de ces dernières (figure II.1-b),

3 . la (ré)-initialisation automatique des filtres en cas, par exemple, de sortie de champ de vue lors de l'exécution de gestes (figure II.1-d), typiquement par un sujet non-expert du robot.

Une alternative pour gérer ce dernier point est de considérer un seul filtre centralisant l'état de l'ensemble des cibles à l'instar de [Bardet et al., 2009, Isard and Blake, 2001, Khan et al., 2005] pour du suivi multi-personnes par filtrage particulaire. La principale réserve que nous pouvons émettre concerne le nombre de particules qui croît exponentiellement avec la dimension du vecteur d'état. Notre approche s'inspire de [Qu et al., 2007, Yu and Wu, 2004] (appliquée dans ces cas à du suivi multi-personnes) et considère des traqueurs indépendants mais interactifs (appelé IDMOT pour "Interactively Distributed MOT") pour répondre aux problèmes 1 et 2 évoqués ci-dessus dans une stratégie de filtrage particulaire ICONDENSATION qui diffère donc de la CONDENSATION [Isard and Blake, 1998a] afin de s'affranchir du problème #3. Cette variante de filtrage particulaire est initiée dans [Isard and Blake, 1998b, Pérez et al., 2004] pour du suivi 2D mono-cible. Clairement, elle est à nos yeux sous exploitée eu égard à ses nombreux atouts comme nous le verrons ci-après.

Notre stratégie de filtrage, qui constitue ici la spécificité de notre traqueur de gestes, sera appelée IIDMOT par la suite.

II.1.2 Suivi du regard

Pour estimer la direction du regard, dissocions les systèmes intrusifs [Morimoto and Mimica, 2005], certes précis et robustes mais très contraignants, des systèmes non intrusifs qui s'appuient sur des techniques de vision par ordinateur. La littérature propose ici de nombreux travaux où la direction du regard est assimilée à l'orientation de la tête. L'erreur sous-jacente, en moyenne 10° [Gee and Cipolla, 1994], est cohérente avec les précisions atteignables pour ce type d'approches [Brown and Tian, 2002].

Plusieurs classes d'approches sont alors proposées dans la communauté Vision [Murphy-Chutorian and Trivedi, 2008b]. Les approches 3D reposent sur une inférence dans l'espace à partir de modèle 3D fins (par exemple [Fidaleo and Medioni, 2007]) et/ou de systèmes multi-caméras (comme [Nickel and Stiefelhagen, 2006]). Elles sont hélas peu compatibles avec notre contexte applicatif en terme de ressources CPU embarquées. Les approches 2D et vision monoculaire, bien que moins précises, sont souvent privilégiées [Asteriadis et al., 2008, Benewitz et al., 2008, Brown and Tian, 2002, Cootes et al., 2000, Heinzmann and Zelinsky, 1999, Valenti et al., 2008] pour leur simplicité algorithmique et matérielle. Citons ici les méthodes de reconnaissance de postures à partir d'apprentissage hors-ligne et contraignant de modèles statistiques

de forme et d'apparence du visage selon différents points de vues [Benewitz et al., 2008, Brown and Tian, 2002, Cootes et al., 2000]. A l'instar de [Asteriadis et al., 2008, Heinzmann and Zelinsky, 1999], notre approche vise au suivi quasi temps réel image de quelques points anatomiques du visage (yeux, bouche, nez) pour intuiter la direction du regard dans l'espace afin de fusionner avec les gestes manuels déictiques.

Les spécificités de notre traqueur sont les suivantes :
- la modélisation des points anatomiques par un ensemble dédié de SURFs (pour "Speeded Up Robust Features") [Bay et al., 2006] avec apprentissage hors et en ligne permettant une initialisation automatique et une adaptation rapide au contexte courant (illumination, apparence du sujet observé),
- le suivi quasi temps réel de ces points par un filtre particulaire type ICONDENSATION et une fonction de vraisemblance combinant distances entre coordonnées image et entre descripteurs de distributions de SURFs dans la veine de [Zhou et al., 2009].

Précisons que Zhou *et al.* dans [Zhou et al., 2009] traitent du suivi d'objets à partir d'appariements par *mean shift* de distributions de SIFTs.

II.2 Formalisme du filtrage particulaire

Cette section rappelle le principe du filtrage particulaire à travers les stratégies SIR, CONDENSATION et ICONDENSATION. Pour plus de détails, le lecteur pourra se référer à la thèse de L. Brèthes [Brèthes, 2005] réalisée au LAAS-CNRS et dont les travaux ont portés sur le filtrage particulaire dans un contexte de suivi mono-cible de personnes par vision monoculaire embarquée.

II.2.1 Algorithme générique ou SIR

Les techniques de filtrage particulaire sont des méthodes de simulation séquentielles de type Monte Carlo permettant l'estimation du vecteur d'état d'un système Markovien non nécessairement linéaire soumis à des excitations aléatoires possiblement non Gaussiennes [Arulampalam et al., 2002, Doucet et al., 2001]. En tant qu'estimateurs Bayésiens, leur but est d'estimer récursivement la densité de probabilité *a posteriori* $p(x_t|z_{1:t})$ du vecteur d'état x_t à l'instant t conditionné sur l'ensemble des mesures $z_{1:t} = z_1, \ldots, z_t$, une connaissance *a priori* de la distribution du vecteur d'état initial x_0 pouvant être également prise en compte. À chaque instant image t, la densité $p(x_t|z_{1:t})$ est approximée au moyen de la distribution ponctuelle

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(x_t - x_t^i), \ \sum_{i=1}^{N} w_t^i = 1, \tag{II.1}$$

exprimant la sélection d'une valeur, ou « particule », x_t^i avec la probabilité, ou « poids », w_t^i, $i = 1, \ldots, N$ étant l'index de la particule. Les moments conditionnels de x_t, tels que l'estimateur du minimum d'erreur quadratique moyenne (ou MMSE, pour "Minimum Mean Square Error") $\mathrm{E}[x_t|z_{1:t}]$, peuvent alors être approchés par ceux de la variable aléatoire ponctuelle de densité de probabilité (II.1). Ainsi, nos différents filtres sont basés sur cet estimateur MMSE.

Les particules x_t^i évoluent stochastiquement dans le temps. Elles sont échantillonnées selon une fonction d'importance visant à explorer adaptativement les zones « pertinentes » de l'espace d'état.

L'algorithme générique de filtrage particulaire est présenté dans la table II.1. Son initiali-

---

1: **SI** $t = 0$ **(INITIALISATION) ALORS**
2: Échantillonner $x_0^1, \ldots, x_0^N$ i.i.d. selon $p(x_0)$, et poser $w_0^i = \frac{1}{N}$, $i = 1, \ldots, N$
3: **FIN SI**
4: **SI** $t \geq 1$ **ALORS**
5: **POUR** $i = 1, \ldots, N$, **FAIRE**
6: « Propager » la particule x_{t-1}^i en simulant de manière indépendante

$$x_t^i \sim q(x_t|x_{t-1}^i, z_t) \tag{II.2}$$

7: Mettre à jour le poids w_t^i selon l'équation

$$w_t^i \propto w_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, z_t)} \tag{II.3}$$

préalablement à une étape de normalisation assurant que $\sum_i w_t^i = 1$
8: **FIN POUR**
9: Rééchantillonner $\{x_t^i, w_t^i\}$ selon $P\left(\tilde{x}_k^i = x_k^j\right) = w_t^j$, ce qui conduit à un ensemble de particules pondérées $\{\tilde{x}_t^i, \frac{1}{N}\}$ tel que $\sum_{i=1}^N w_t^{(i)} \delta(x_t - x_t^i)$ et $\frac{1}{N}\sum_{i=1}^N \delta(x_t - \tilde{x}_t^i)$ approximent $p(x_t|z_{1:t})$; affecter x_t^i et w_t^i avec $\tilde{x}_t^i$ et $\frac{1}{N}$
10: **FIN SI**

TAB. II.1: Algorithme générique de filtrage particulaire (SIR).

sation consiste en la définition d'un ensemble de particules pondérées décrivant la distribution *a priori* $p(x_0)$, e.g. en affectant des poids identiques $w_0^i = \frac{1}{N}$ à des échantillons $x_0^1, \ldots, x_0^N$ indépendants identiquement distribués (i.i.d.) selon $p(x_0)$.

A chaque instant t, disposant de la mesure z_t et de la description particulaire $\{x_{t-1}^i, w_{t-1}^i\}$ de $p(x_{t-1}|z_{1:t-1})$, la détermination de l'ensemble de particules pondérées $\{x_t^i, w_t^i\}$ associé à la densité *a posteriori* $p(x_t|z_{1:t})$ se fait en deux étapes. Dans un premier temps, les x_t^i sont échantillonnés selon la fonction d'importance $q(x_t|x_{t-1}, z_t)$ évaluée en $x_{t-1} = x_{t-1}^i$, cf. l'équation (II.2). Les poids w_t^i sont ensuite mis à jour de façon à assurer la cohérence de l'approximation (II.1). Ce calcul obéit à (II.3), où $p(x_t|x_{t-1})$ rend compte de la dynamique du processus d'état sous-jacent, et la vraisemblance $p(z_t|x_t)$ d'un état possible x_t vis à vis de la mesure z_t est évaluée à partir de la densité de probabilité relative au lien état-observation.

Toute méthode de simulation séquentielle de type Monte Carlo souffre du phénomène de

dégénérescence, au sens où après quelques itérations, les poids non négligeables tendent à se concentrer sur une seule particule. Afin de limiter ce phénomène, une étape de rééchantillonnage peut être insérée en fin de chaque cycle, cf. l'item 9 de l'algorithme SIR (pour "Sampling Importance Resampling") table II.1. Ainsi, N nouvelles particules $\tilde{x}_t^i$ sont obtenues par rééchantillonnage avec remise dans l'ensemble $\{x_t^j\}$ selon la loi $P(\tilde{x}_t^i = x_t^j) = w_t^j$. Les particules associées à des poids w_t^j élevés sont dupliquées, au détriment de celles, faiblement pondérées, qui disparaissent, de sorte que la séquence $\tilde{x}_t^1, \ldots, \tilde{x}_t^N$ est i.i.d. selon $\sum_{i=1}^N w_t^i \delta(x_t - x_t^i)$.

Cette étape de redistribution peut soit être appliquée systématiquement, soit être déclenchée seulement lorsqu'un critère d'efficacité du filtre passe en deçà d'un certain seuil [Doucet et al., 2000, Arulampalam et al., 2002]. Le calcul des moments de (II.1) doit de préférence faire intervenir l'ensemble des particules pondérées avant rééchantillonnage.

Un dernier point concerne ici la fonction de vraisemblance où il est judicieux dans notre contexte robotique de fusionner plusieurs mesures de nature différentes. Sous hypothèse de leur indépendance conditionnellement à l'état, la vraisemblance unifiée de N_d mesures s'écrit alors :

$$p(z_t|x_t) = \prod_j^{N_d} p(z_t^j|x_k) \tag{II.4}$$

II.2.2 Échantillonnage guidé par la dynamique ou CONDENSATION

L'algorithme de CONDENSATION [Isard and Blake, 1998a] (pour "Conditional Density Propagation") peut être vu comme le cas particulier de l'algorithme SIR où la fonction d'importance est relative à la dynamique du processus d'état : $x_t^i \sim p(x_t^i|x_{t-1}^i)$. Ceci confère à la CONDENSATION une structure « prédiction / mise à jour » comparable à celle du filtre de Kalman. En effet, la densité ponctuelle $\sum_{i=1}^N w_{t-1}^i \delta(x_t - x_t^i)$ approxime la prédiction $p(x_t|z_{1:t-1})$. En outre, la mise à jour des poids selon $w_t^i \propto w_{t-1}^i p(z_t|x_t^i)$ rappelle la formule de Bayes sous-jacente à l'étape de mise à jour de l'estimé de Kalman.

Dans un contexte de suivi visuel, l'algorithme de CONDENSATION original définit les vraisemblances des particules à partir de primitives visuelles de type contour, mais d'autres primitives ont également été envisagées, par exemple des distributions de couleur [Nummiaro et al., 2003, Pérez et al., 2002].

II.2.3 Échantillonnage guidé par la mesure ou ICONDENSATION

Le rééchantillonnage utilisé seul ne suffit pas à limiter efficacement le phénomène de dégénérescence évoqué précédemment. En outre, il peut conduire à une perte de diversité dans

l'exploration de l'espace d'état, du fait que la description particulaire de la densité *a posteriori* risque de contenir de nombreuses particules identiques. La définition de la fonction d'importance $q(x_t|x_{t-1}, z_t)$ – selon laquelle les particules sont distribuées – doit donc également faire l'objet d'une attention particulière [Arulampalam et al., 2002].

En suivi visuel, les modes des fonctions de vraisemblance $p(z_t|x_t)$ relativement à x_t sont généralement très marqués. Il s'en suit que les performances de la CONDENSATION sont souvent assez médiocres. Du fait que les particules sont positionnées selon la dynamique du processus d'état et « en aveugle » par rapport à la mesure z_t, un sous-ensemble important d'entre elles peut être affecté d'une vraisemblance très faible par l'équation $w_t^i \propto w_{t-1}^i p(z_t|x_t^i)$, dégradant ainsi significativement les performances de l'estimateur.

Une alternative peut donc consister à échantillonner les particules à l'instant t – ou bien seulement certaines de leurs composantes – selon une fonction d'importance $q(x_t|z_t)$ définie à partir de l'image courante. Ainsi, l'exploration de l'espace d'état peut être guidée par des fonctionnalités de détection visuelle telles que les *blobs* peau ou bien toute autre primitive intermittente, qui, malgré un caractère sporadique, est très discriminante lorsqu'elle est présente : mouvement, son, etc. Cependant, rien n'empêche qu'une particule x_t^i, dont tout ou partie des composantes sont positionnées à partir de l'image courante, soit incompatible avec sa particule prédécesseur x_{t-1}^i du point de vue de la dynamique du processus d'état. Du fait que $p(x_t^i|x_{t-1}^i)$ prend de faibles valeurs, une telle particule est alors faiblement pondérée dans (II.3).

Une solution simple à ce problème (stratégie référencée ICONDENSATION) est de définir la fonction d'importance comme un mélange d'une fonction d'importance $\pi(.)$ basées sur des mesures, c'est-à-dire sur des détections visuelles et d'une fonction d'importance basée sur la dynamique [Pérez et al., 2004], soit :

$$q(x_t^i|x_{t-1}^i, z_t) = \alpha\pi(x_t^i|z_t) + (1-\alpha)p(x_t^i|x_{t-1}^i). \tag{II.5}$$

Ainsi, $\alpha\%$ des particules sont échantillonnées selon l'observation courante donc permettent une éventuelle (ré)-initialisation du filtre tandis les particules restantes suivent la dynamique du système. Ainsi, en cas de fausses mesures/détections ou de leur absence, $(1-\alpha)\%$ des particules continuent à évoluer selon la dynamique du système permettant alors de conserver une meilleure représentation de la distribution *a posteriori*.

II.3 Description de notre traqueur de gestes

Cette section présente ci-après notre algorithme de filtrage IIDMOT dédié au suivi multi-cibles 3D (dont les images sont acquises par un banc stéréo équipant nos robots, voir chapitre IV), puis les différentes mesures visuelles associées. L'implémentation du traqueur ainsi que les évaluations sont alors décrites.

II.3.1 Stratégie de filtrage

Notre stratégie IIDMOT, décrite table II.2, combine les formalismes de la stratégie ICONDENSATION (section II.2) et de la stratégie de filtres interactifs distribués IDMOT initiée par Qu *et al.* dans [Qu et al., 2007]. Cette démarche est motivée par le souci, dans notre contexte robotique, de pouvoir (ré)-initialiser notre banque de filtres dévolus aux mains et à la tête et de limiter autant que possible les erreurs de fusion/labellisation (lors de l'exécution de gestes) par une interaction entre ces filtres. Rappelons le principe de la stratégie IDMOT où la variable m ($m \in [1;3]$) indexe les cibles (en l'occurrence la tête et les deux mains).

A chaque instant image t, le principe est d'effectuer $k = 1, \ldots, K$ itérations. Après le calcul d'une première estimée $\{\hat{\mathbf{x}}_{t,1}^m\}_{m=1,2,3}$ pour chaque cible à $k = 1$, la distance 3D inter-cibles est estimée. Lorsque deux cibles sont distantes, c'est-à-dire que la distance entre centroïdes associés est supérieure à un seuil prédéfini d_{TH}, celles-ci sont sans interaction et les filtres associés sont logiquement indépendants. *A contrario*, si celles-ci sont proches, nous introduisons des fonctions d'« inertie », notée $\varphi_2(.)$, et de « répulsion magnétique », notée $\varphi_1(.)$, dans le calcul de leurs vraisemblances associées afin de limiter les erreurs précédemment mentionnées. Le principe est décrit par les étapes 9 à 22 de la table II.2.

La fonction de répulsion $\varphi_1(.)$ est calculée par :

$$\varphi_1(\mathbf{x}_t^{i,m}, z_t^m, z_t^n) \propto 1 - \frac{1}{\beta_1} \exp\left(-\frac{D_{i,m}^2}{\sigma_1^2}\right),$$

avec β_1 et σ_1 deux termes de normalisation déterminés *a priori*, $D_{i,m}$ est la distance euclidienne séparant la particule $\mathbf{x}_t^{i,m}$ de l'estimée temporaire $\mathbf{x}_{t,k}^n$. A l'instar des champs magnétiques, deux cibles m et n proches engendrent une force d'attraction entre elles pour leurs particules associées (du point de vue des mesures), phénomène alors compensé par cette fonction de répulsion. Le principe est extensible à trois cibles en possible interaction deux à deux sans ajout d'une combinatoire très élevée. L'équilibre entre répulsion magnétique et gravitation est trouvé après quelques oscillations k qui aboutissent à la stabilisation du nuage de particules. En pratique, quelques itérations ($K \in [4;6]$) assurent la convergence du processus.

La fonction $\varphi_2(.)$ considère la dynamique des cibles *via* une force d'inertie définie par :

$$\varphi_2(\mathbf{x}_t^{i,m}, \mathbf{x}_{t-1}^m, \mathbf{x}_{t-2}^m) \propto 1 + \frac{1}{\beta_2} \exp\left[-\frac{(|\overrightarrow{v_{t,i}}| - |\overrightarrow{v_{t-1}}|)^2}{\sigma_{22}^2}\right] \exp\left(-\frac{\theta_{i,m}^2}{\sigma_{21}^2} \cdot \frac{|\overrightarrow{v_{t,i}}|^2}{\sigma_{22}^2}\right),$$

avec $\overrightarrow{v_{t,i}}$ et $\overrightarrow{v_{t-1}}$ les vitesses de la cible en question modélisées respectivement par le mouvement de la $i^{ème}$ particule par rapport à l'estimée temporaire et par celui des estimées des deux pas précédents. Le principe est de supposer que $\overrightarrow{v_t}$ sera proche de $\overrightarrow{v_{t-1}}$, hypothèse réaliste de part la cadence de traitement des images. β_2 est un terme de normalisation, tandis que σ_{21} et σ_{22} caractérisent la variance des vecteurs. Enfin, $\theta_{i,m}$ représente l'angle entre $\overrightarrow{v_{t,i}}$ et $\overrightarrow{v_{t-1}}$. Le lecteur averti remarquera que $\varphi_2(.)$ diffère du principe original [Qu et al., 2007] par son terme angulaire. En effet, la vitesse est ici rajoutée afin d'éviter un effet collatéral de la méthode : dans le cas de vitesses faibles et donc pour une cible quasi statique, l'angle entre les vitesses n'a pas de

SI $t = 0$ (**INITIALISATION**) **ALORS**
  Échantillonner $\mathbf{x}_0^{1,m}, \ldots, \mathbf{x}_0^{i,m}, \ldots, \mathbf{x}_0^{N,m}$ i.i.d. selon $p(\mathbf{x}_0^m)$, et poser $\omega_0^{i,m} = \frac{1}{N}$
FIN SI
SI $t \geq 1$ **ALORS**
  —$[\{\mathbf{x}_{t-1}^{i,m}, \omega_{t-1}^{i,m}\}]_{i=1}^N$ étant la description d'une particule de $p(\mathbf{x}_{t-1}^m | z_{1:t-1}^m)$—
  « Propager » la particule $\{\mathbf{x}_{t-1}^{i,m}\}_{i=1}^N$ en simulant

$$\mathbf{x}_t^{i,m} \sim q(\mathbf{x}_t^m | x_{t-1}^{i,m}, z_t^m)$$

  Mettre à jour le poids $\{\omega_t^{i,m}\}_{i=1}^N$ associé à $\{\mathbf{x}_t^{i,m}\}_{i=1}^N$ selon l'équation

$$\omega_t^{i,m} \propto \omega_{t-1}^{i,m} \frac{p(z_t^m | x_t^{i,m}) p(x_t^{i,m} | x_{t-1}^{i,m})}{q(x_t^{i,m} | x_{t-1}^{i,m}, z_t^m)}$$

  préalablement à une étape de normalisation assurant que $\sum_{i=1}^N \omega_t^{i,m} = 1$
  Sélection de l'estimée temporaire $\hat{x}_{t,1}^m$, par exemple en calculant l'estimé du minimum d'erreur quadratique moyenne (MMSE), $\mathrm{E}_{p(\mathbf{x}_t^m | z_{1:t}^m)}[\mathbf{x}_t^m]$, afin d'approcher la loi de filtrage par

$$p(\mathbf{x}_t^m | z_{1:t}^m) = \sum_{i=1}^N \omega_t^{i,m} \delta(\mathbf{x}_t^m - \mathbf{x}_t^{i,m})$$

  POUR $n = 1 : m$, **FAIRE**
    SI $d_{mn}(\hat{\mathbf{x}}_{t,1}^m, \hat{\mathbf{x}}_{t,1}^n) < d_{TH}$ **ALORS**
      POUR k=1 :K iterations, **FAIRE**
        Calculer φ_1, φ_2
        Repondérer $\omega_t^{i,m} = \omega_t^{i,m}.\varphi_1.\varphi_2$
        Normaliser les $\{\omega_t^{i,m}\}_{i=1}^N$
        Sélectionner l'estimée temporaire $\hat{\mathbf{x}}_{t,k+1}^m$
        Calculer φ_1, φ_2
        Repondérer $\omega_t^{i,n} = \omega_t^{i,n}.\varphi_1.\varphi_2$
        Normaliser les $\{\omega_t^{i,n}\}_{i=1}^N$
        Sélectionner l'estimée temporaire $\hat{\mathbf{x}}_{t,k+1}^n$
      FIN POUR
    FIN SI
  FIN POUR
  De manière systématique ou en fonction d'un critère d'efficacité, rééchantillonner la description $[\{\mathbf{x}_t^{i,m}, \omega_t^{i,m}\}]_{i=1}^N$ de $p(\mathbf{x}_t^m | z_{1:t}^m)$ de façon à obtenir un ensemble de particules pondérées $[\{\mathbf{x}_t^{(s^{i,m})}, \frac{1}{N}\}]_{i=1}^N$ en échantillonnant dans $\{1, \ldots, N\}$ les index $s^{1,m}, \ldots, s^{N,m}$ tels que $P(s^{i,m} = j) = \omega_t^{j,m}$. Affecter $\mathbf{x}_t^{i,m}$ et $\omega_t^{i,m}$ avec $\mathbf{x}_t^{(s^{i,m})}$ et $\frac{1}{N}$
FIN SI

TAB. II.2: Notre algorithme de filtrage particulaire IIDMOT.

signification puisqu'il tient plus du hasard que de la réalité, il convient alors d'empêcher ce terme angulaire d'influencer trop fortement sur la fonction.

II.3.2 Modélisation

Nous considérons une sphère de rayon fixe et deux ellipsoïdes déformables (respectivement pour la tête et les deux mains), par le biais de l'estimation de leur position 3D $\mathcal{X} = (X, Y, Z)^T$,

ainsi que de l'orientation $\Theta = (\theta_x, \theta_y, \theta_z)^T$ et de la taille de leurs axes $A = (a_x, a_y, a_z)^T$ pour les ellipsoïdes (afin de tenir compte de l'orientation de la main en 3D). Chaque vecteur d'état $\mathbf{x}_t$ inclut ces données. Concernant la dynamique, et pour éviter d'augmenter la dimension du vecteur d'état, nous supposons que chaque membre du vecteur d'état évolue indépendamment des autres suivant une marche aléatoire gaussienne :

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t|\mathbf{x}_{t-1}, \Sigma),$$

où $\mathcal{N}(.|\mu, \Sigma)$ est une distribution gaussienne 3D de moyenne μ et de covariance Σ déterminée *a priori*. Le vecteur d'état à estimer est donc dans $\Re^9$ (respectivement $\Re^3$), soit $\mathbf{x}_t = [\mathcal{X}, \Theta, A]^T$ (respectivement $\mathbf{x}_t = [\mathcal{X}]^T$) pour les mains (respectivement la tête).

II.3.3 Mesures visuelles

Notre traqueur fusionne des mesures aux étapes #6 et #7 du filtre IIDMOT :
- des mesures intermittentes (issues de détecteurs) dans la fonction d'importance $q(\mathbf{x}_t|\mathbf{x}_{t-1}, z_t)$ (équation (II.5)), en l'occurrence $\pi(\mathbf{x}_t|z_t)$,
- des mesures persistantes dans la fonction de vraisemblance $p(z_t|\mathbf{x}_t)$.

Le suivi des mains s'appuie sur les mêmes mesures tandis que le suivi de la tête est spécifique en termes de mesures et de dynamique. Les mesures de la fonction de vraisemblance tirent partie de la projection préalable des ellipsoïdes dans chaque vue tandis que les mesures intégrées dans la fonction d'importance sont inférées en 3D, donc après triangulation sur les *blobs* détectés dans chaque paire d'images, afin de positionner les particules dans les zones pertinentes de l'espace recherche basé sur des grandeurs 3D. Notre approche combine les intérêts respectifs des approches par apparence et par reconstruction. Signalons enfin que les fonctions bas niveau de traitement des images sont issues de la librairie OpenCv [OpenCv, 2008].

a) Dans la fonction d'importance

Les mesures sont ici classiques. Il s'agit pour le suivi de tête du détecteur de visage de Viola *et al.* dans [Viola and Jones, 2001] qui permet d'extraire des *blobs* correspondant à des visages fronto-parallèles. Pour la tête et les mains, le détecteur repose sur une segmentation 2D des *blobs* peau, puis leur appariement dans la paire d'images à chaque instant. La classification des pixels s'appuie sur un histogramme dans l'espace colorimétrique CIE-Lab appris hors ligne et une règle de décision bayésienne [Schwerdt and Crowley, 2000].

Les *blobs* 3D associés sont obtenus après triangulation, « convertis » en ellipsoïdes, puis filtrés suivant leurs tailles volumiques. Le processus est formalisé en annexe D. La fonction $\pi(\mathbf{x}_t|z_t)$ associée à chaque détecteur est définie comme un mélange de gaussiennes associés à ces ellipsoïdes, ainsi pour une détection de L ellipsoïdes « peau » :

$$\pi(\mathbf{x}_t|z_t^{skin}) = \sum_{l=1}^{L} \mathcal{N}(\mu_l, \Sigma_{D_{skin}}),$$

où μ_l est le centroïde 3D de chaque ellipsoïde triangulée et $\Sigma_{D_{skin}}$ la covariance du décalage entre la position détectée et la position réelle de la cible déterminée *a priori*. En présence de plusieurs détecteurs comme pour la tête, la fonction unifiée $\pi(\mathbf{x}_t|z_t^{skin}, z_t^{face})$ est vue comme un mélange de gaussiennes des fonctions $\pi(\mathbf{x}_t|z_t^{skin})$ et $\pi(\mathbf{x}_t|z_t^{face})$.

b) Dans la fonction de vraisemblance

Les mesures sont ici plus ou moins classiques. Une mesure répandue pour le suivi de visage est la corrélation d'histogramme couleur [Brèthes, 2005, Nummiaro et al., 2003, Pérez et al., 2004]. Il s'agit de calculer l'histogramme de couleur dans la zone image englobée par la projection de l'ellipsoïde, en l'occurrence un cercle pour la tête. Cet histogramme $h_{x_t^i}$ associé à la particule $\mathbf{x}_t^i$ est corrélé avec un histogramme de référence h_{ref} (lié à la cible suivie) grâce à la distance de Bhattacharyya [Aherne et al., 1997]. La fonction de vraisemblance associée est :

$$p(z_t^{Mc}|x_t) = \exp^{-\frac{1-D_B(h_{x_t^i}, h_{ref})}{2.\sigma_{Mc}^2}}, \text{ avec } D(h_{x_t^i}, h_{ref}) = \sum_{b=1}^{N_{bin}} \sqrt{h_{b,x_t^i}.h_{b,ref}}.$$

N_{bin} est ici le nombre de cellules de chaque histogramme, dont h_{b,x_t^i} et $h_{b,ref}$ sont les $b^{ème}$ cellules, tels que $h_{b,(.)}$ est la fréquence d'occurrence de la cellule. L'histogramme de référence est initialisé automatiquement sur la première image puis ré-actualisé à chaque instant image [Nummiaro et al., 2003] :

$$h_{t,ref} = (1-\alpha).h_{t-1,ref} + \alpha.h_{t,x_t^i}.$$

Cette mise à jour est nécessaire pour des rotations spatiales de la tête.

Pour les mains, nous proposons une mesure inspirée de [Thayananthan et al., 2003]. Le principe vise à discrétiser uniformément en N_p pixels les contours des ellipsoïdes projetées, puis caractériser les normales en ces pixels. Deux groupes de pixels sont extraits sur ces orthogonales : $\mathcal{O}$ (respectivement $\mathcal{B}$) contient les pixels à l'intérieur (respectivement à l'extérieur) de la cible. La fonction de vraisemblance est alors :

$$p(z_t^{Mp}|\mathbf{x}_t) = \prod_{o\in\mathcal{O}} p_s(o|\mathbf{x}_t) \prod_{b\in\mathcal{B}} [1-p_s(b|\mathbf{x}_t)],$$

avec $p_s(j|\mathbf{x}_t)$ la probabilité peau du pixel j sachant $\mathbf{x}_t$ déterminée par l'image de probabilité obtenue lors de la segmentation peau. Cette double mesure est particulièrement discriminante lorsque les contours des ellipsoïdes projetés se superposent aux contours image de régions « peau » segmentées.

Une dernière mesure vise à favoriser les contours mobiles (s'il y en a) des cibles, c'est-à-dire

$$p(z_t^{Mm}|\mathbf{x}_t) \propto \exp\left(-D^2/2\sigma_s^2\right),\ D = \sum_{j=1}^{N_p} |x(j)-z(j)| + \rho\gamma(z(j)),$$

Symbole	Signification	Valeur
N	nombre de particules par filtre	100
α	taux de particules générées suivant les détections	0.4
K	nombre d'itérations du sous-algorithme IDMOT	4
d_{TH}	distance euclidienne d'interaction entre cibles	0.5
-	résolution des images	256×192
-	espace de couleur de travail	CIE Lab
N_p	nombre de points le long du contour des ellipsoïdes	20
(σ_1, β_1)	coefficients de la fonction de répulsion φ_1	$(0.12, 1.33)$
$(\sigma_{21}, \sigma_{22}, \beta_2)$	coefficients de la fonction d'inertie φ_2	$(1.57, 0.2, 2.0)$
Σ	écart-type des modèles de marche aléatoire	$\begin{pmatrix} 0.07 & 0.07 & 0.07 \\ 0.03 & 0.03 & 0.03 \\ 0.17 & 0.17 & 0.17 \end{pmatrix}$

TAB. II.3: Valeurs des paramètres principaux utilisés dans le traqueur de gestes.

qui dépend de la somme des distances au carré entre les N_p points précédents et les contours z les plus proches dans l'image. La variable σ_s est un écart-type déterminé *a priori*. $\gamma(z(j)) = 0$ si le pixel $z(j)$ est non nul, c'est à dire en mouvement, et $\gamma(z(j)) = 1$ dans le cas contraire. $\rho > 0$ détermine alors une pénalité. Notons que cette mesure reste effective en cas de mouvement du robot : un « bonus » est alors tout simplement donné à toutes les particules.

La fonction de vraisemblance globale est le produit des diverses fonctions de vraisemblance (voir équation (II.4)), par exemple pour les mains :

$$p(z_t|\mathbf{x}_t) = \prod_{c=1}^{2} p(z_{t,c}^{Mm}|\mathbf{x}_t).p(z_{t,c}^{Mp}|\mathbf{x}_t),$$

où l'index c réfère à l'image gauche ou droite de la paire stéréo.

II.3.4 Implémentation et évaluations associées

Ce traqueur est intégré dans l'architecture des robots du LAAS (les robots Jido et HRP-2 seront présentés dans le chapitre IV) permettant l'acquisition de séquences d'images stéréo afin de :

– régler les différents paramètres de notre traqueur,
– réaliser des évaluations qualitatives et quantitatives.

Ainsi, le tableau II.3 liste les valeurs des principaux paramètres utilisés pour notre traqueur ; les valeurs sont estimées, pour la plupart, empiriquement. Par exemple, les paramètres de la marche aléatoire sont déterminés par rapport à la distance maximum pouvant être parcourue par la cible en question entre deux pas de l'algorithme Il est à noter que, comme toute implémentation, notre approche a donné lieu à un certain nombre d'optimisations. Ces dernières permettent de rendre

Stratégie	MIPF	IDMOT	IIDMOT
FR_p	29%	18%	4%
FR_l	9%	1%	1%
Nombre d'images par seconde	15	12	10

TAB. II.4: Comparaison quantitative de performances et de vitesse.

nos mesures et détections plus robustes et plus rapides, mais concernent également l'algorithme général, celui-ci ne permettant pas directement une gestion des pertes de cible. Quelques détails sur ces optimisations sont données en annexes E.

La figure II.2 montre une réalisation de suivi sur une séquence impliquant des occultations et des sorties de champ de vue. Sur chaque image, les cercles et ellipses montrent la projection des estimées de chaque cible. Notre stratégie $IIDMOT$ permet une initialisation automatique et aide à la ré-initialisation après la perte d'une cible. La réalisation complète est accessible sur le lien URL `www.laas.fr/~bburger`.

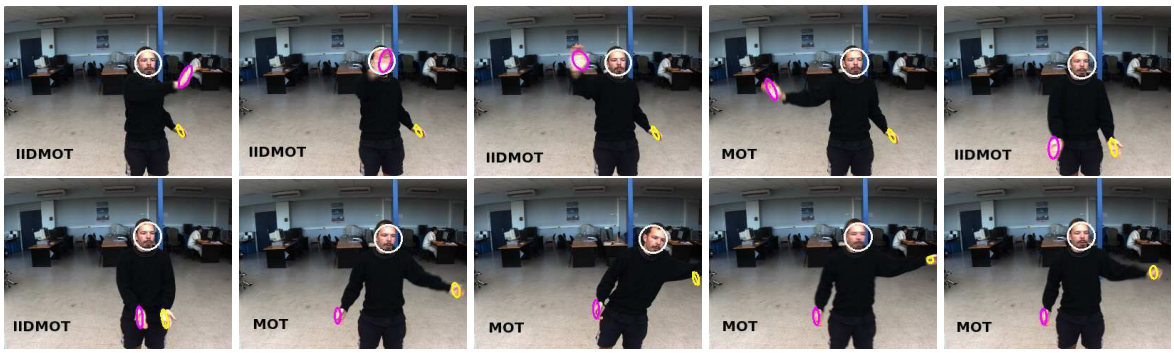


FIG. II.2: Scénario impliquant des occlusions et des sorties de champ de vue, suivi effectué par notre traqueur avec/sans interaction (c'est-à-dire IIDMOT/MOT) selon la distance inter-cibles.

Des évaluations quantitatives ont été réalisées. Une séquence vidéo de 1214 paires d'images a été acquise depuis le robot Jido. Ces séquences sont étiquetées manuellement afin de constituer une vérité terrain et ainsi évaluer les performances de notre traqueur IIDMOT. Celles-ci incluent diverses conditions d'illumination, différents sujets observés, des occultations, voire de sorties de champs de vue. Les performances comparées de notre stratégie IIDMOT, des stratégies IDMOT [Qu et al., 2007] et de filtres distribués indépendants (MIPF) [Isard and Blake, 1998b] portent sur le taux de mauvais positionnement FR_p et le taux de mauvaise labellisation FR_l. Ainsi, une cible n'ayant aucun filtre associé dans une image correspond à une erreur en position FR_p, tandis qu'un filtre associé à la mauvaise cible sera considéré comme une erreur de labellisation FR_l. Le tableau II.4 compare les performances des stratégies pré-citées.

La stratégie IIDMOT est supérieure aux approches conventionnelles pour des temps de calcul équivalents. Les filtres indépendants souffrent logiquement du problème de labellisation de part la non modélisation des interactions entre filtres distribués. La stratégie IDMOT est ici plus performante, mais ne gère pas la ré-initialisation des filtres après pertes de cibles.

(a) Suivi des gestes « salut » et « pointage à gauche » sans interaction des filtres.

(b) Suivi des gestes « salut » et « pointage à gauche » avec interaction des filtres.

FIG. II.3: Exemple de suivi avec ou sans interaction des filtres distribués.

La figure II.3 illustre nos propos. Elle montre une réalisation du suivi de deux gestes, l'un symbolique (« salut ») et l'autre déictique (« pointage à gauche »). Sans interaction entre les filtres, le suivi perd facilement la cible si celle-ci s'approche trop d'une autre cible de teinte peau. De plus, le traqueur ne peut raccrocher la cible après sa perte tandis que la stratégie IIDMOT est robuste à ces divers artefacts.

II.4 Description de notre traqueur de visage

La stratégie de filtrage ICONDENSATION (voir section II.2) est privilégiée pour sa capacité à (ré)-initialiser automatiquement le traqueur. Le modèle de visage considéré ainsi que la caractérisation associée de la direction approximative du regard sont décrits ci-après. Cette section énumère ensuite les mesures visuelles puis ouvre sur l'implémentation du traqueur et ses performances.

II.4.1 Modélisation

Notre approche vise à inférer la direction du regard à partir de données image donc sans aucune information 3D. L'idée est d'utiliser les propriétés locales du visage, c'est-à-dire de s'appuyer sur quelques invariants géométriques associés. Ainsi, Gee *et al.* dans [Gee and Cipolla, 1994] propose les trois grandeurs fixes suivantes pour paramétriser un visage :

$$R_e = \frac{l_e}{l_f},\ R_m = \frac{l_m}{l_f},\ R_n = \frac{l_n}{l_f},$$

où l_e est la distance entre les yeux, l_m la distance entre la base du nez et la bouche, l_n la distance du bout du nez à sa base, l_f celle de la ligne des yeux à la bouche. Les relations entre

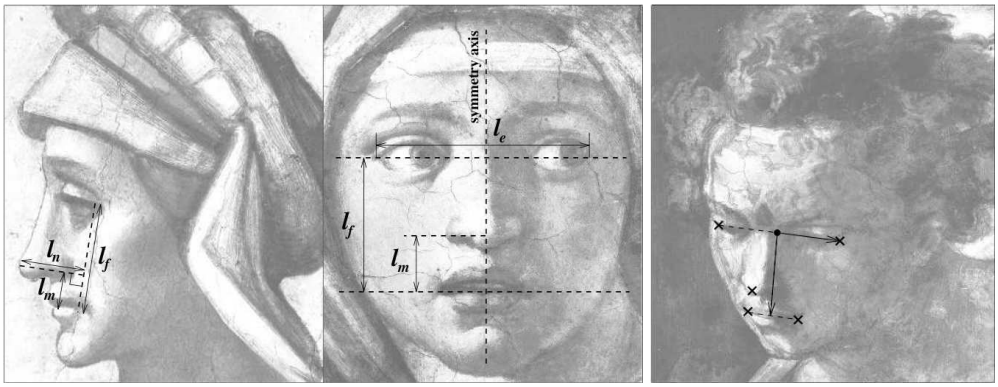


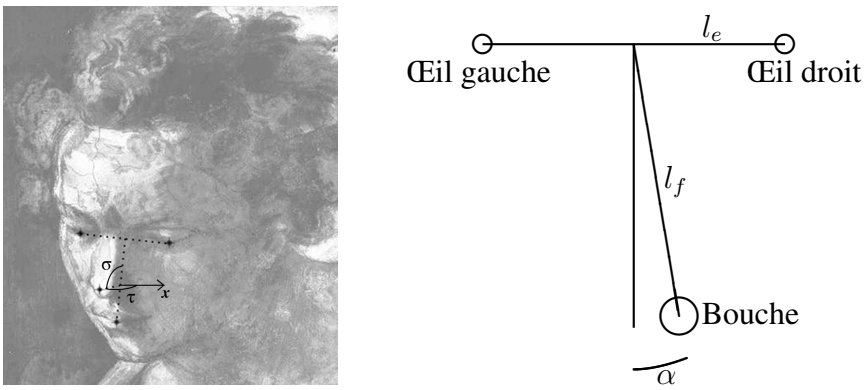
FIG. II.4: Modèle de visage de référence [Gee and Cipolla, 1994].

ces paramètres et leurs valeurs mesurées sur l'image vont permettre d'estimer la direction du regard. Ce modèle requiert la connaissance de la position dans le plan image des yeux, du nez et de la bouche.

Gee *et al.* dans [Gee and Cipolla, 1994] proposent plusieurs heuristiques pour déduire les angles à partir des grandeurs précédentes. Ainsi, nous exploitons la grandeur R_e, grandeur s'appuyant sur la distance entre les yeux l_e, la distance de la bouche aux yeux l_f et la position gauche/droite du nez pour déterminer une estimation de l'orientation du visage. Cette méthode ne fonctionne correctement que de face (de profil elle ne peut pas être précise puisque la distance ente les yeux n'est plus mesurable). Elle a l'avantage de ne pas nécessiter de mesure précise de la position du nez, mesure qui est difficile à réaliser de façon robuste en raison du faible nombre de caractéristiques du nez constants sous tous les angles et éclairages.

En utilisant le modèle de la figure II.5 et en faisant intervenir les paramètres rigides introduits précédemment, on peut calculer l'orientation du visage en résolvant le système suivant :

$$\begin{cases} \tan\sigma \sin\tau &= \tan\alpha \\ \sin\sigma \tan\tau &= \frac{R_e l_f \sin\alpha}{l_e} \end{cases}$$



(a) Angles mesurés (b) Modèle géométrique

FIG. II.5: Définition des angles et distances.

On trouve τ (*pan*) et σ (*tilt*) comme les racines positives de deux polynômes de degré 2 en $\tan^2 \tau$ et $\tan^2 \sigma$ au signe près. L'ambiguïté sur le signe est levée en utilisant la position du nez (gauche/droite).

II.4.2 Mesures visuelles

Différents indices visuels sont utilisés par le traqueur. Nous les présentons ici, puis précisons dans la section suivante comment ils sont intégrés dans le filtre.

a) Détecteurs de points anatomiques

FIG. II.6: De gauche à droite : image initiale, image après changement d'espace chromatique, détection de bouche obtenue par convolution avec masques de Haar, détection de *blobs* circulaires.

Nous cherchons à caractériser les points anatomiques du visage (bouche, yeux) qui vont nous permettre de déterminer l'orientation du visage. Ces détecteurs sont exploités dans la fonction d'importance (II.5) du filtrage. Le visage est tout d'abord segmenté grâce au détecteur multi-échelle de visages de Viola *et al.* [Viola and Jones, 2001] de sorte que les détecteurs

suivants seront appliqués à cette sous-image. Même si ce détecteur est limité aux rotations azimut (τ) entre $\pm 45°$, notre traqueur fonctionne au delà de cette plage angulaire, les particules étant alors positionnées selon la seule dynamique (équation II.5) et pondérées grâce aux distributions de SURFs. Ce détecteur repose sur des masques de Haar mesurant des contrastes relatifs entre les yeux, le nez, la bouche, les joues, etc. Nous avons proposé ainsi un détecteur de bouche, basé sur ce même principe, mais appliqué sur l'image préalablement transformée dans l'espace colorimétrique $\frac{rouge}{rouge+vert}$ afin d'augmenter son contraste (figure II.6).

Un second type de détecteur repose sur le détecteur multi-échelle de *blobs* introduits par Lindeberg dans [Lindeberg, 1998]. Nous nous focalisons sur les seules régions circulaires pour détecter les yeux (figure II.6), le détecteur de régions elliptiques (pour la bouche) n'ayant pas encore été implémenté. Le principe repose sur des invariants différentiels normalisés qui consiste à :

1 . passer dans l'espace colorimétrique *Irg*,

2 . convoluer chaque canal $c \in \{Irg\}$ avec un noyau gaussien $\mathcal{N}(.;\Sigma)$ de covariance Σ dont le résultat est noté $L_c(.;\Sigma)$,

3 . sélectionner les extrema locaux maximisant la relation

$$B^c_{norm} = \sum_{c\in\{Irg\}} t^2(\partial_{xx}L_c + \partial_{yy}L_c)^2.$$

b) Descripteurs locaux

Des descripteurs locaux pour caractériser des parties de l'image seront utilisés dans la fonction de vraisemblance du filtrage particulaire.

Les SURFs (pour "Speeded Up Robust Features") [Bay et al., 2006] sont des points caractéristiques sur une image. Ils sont censés être invariants par changement d'échelle, rotation, translation, variation de luminosité et de contraste et avoir un taux de reproductibilité (taux de SURFs que l'on peut apparier sur deux images similaires) important. Chaque SURF comprend un descripteur qui décrit la structure de l'image autour du point repéré (figure II.7). L'objectif est le même que pour les SIFTs [Lowe, 2004] (pour "Scale Invariant Feature Transform"), mais les SURFs utilisent des approximations basées sur des masques de Haar pour accélérer le traitement.

Un SURF est caractérisé par :

1 . ses coordonnées sur l'image d'où il a été extrait,

2 . son échelle (il peut s'agir du bord d'un livre ou du livre entier par exemple),

3 . une orientation générale du motif local de l'image (c'est un angle qui permet de rendre le descripteur invariant par rotation),

4 . un descripteur relativement invariant par rotation, changement d'échelle, de contraste ou de luminosité (dans les limites de l'information disponible dans l'image bien entendu) de dimension 64,

5 . d'autres paramètres non exploités ici.

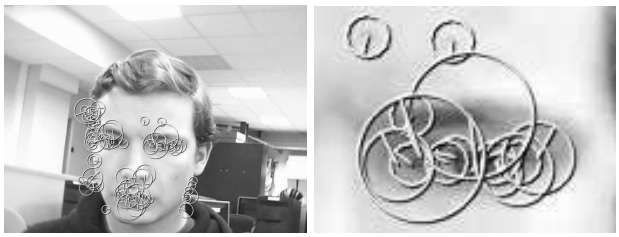


FIG. II.7: Exemple de SURFs sur un visage - Zoom sur un œil - Les cercles représentant les SURFs sont centrés à l'emplacement de chaque point d'intérêt, le rayon correspond à l'échelle du SURF et le trait matérialise son orientation.

Ces descripteurs permettent par exemple d'apparier des points entre deux images en appariant les SURF qui leur correspondent. On ne peut pas caractériser un élément du visage en utilisant un unique SURF à l'échelle à laquelle on travaille. En effet, un SURF étant un point caractéristique de l'image, il va représenter un bord, un coin ou un autre élément local de l'image. Or on trouve de tels éléments un peu partout autour des yeux, du nez et de la bouche, ils ne sont donc pas assez caractéristiques d'un objet particulier. Ceci est remarquable pour les yeux à cause de leurs propriétés de symétrie. De plus, il est bien connu qu'une image photographique a des propriétés d'auto-similarité, un seul SURF est donc très insuffisant pour ce que l'on souhaite faire. Ici les SURFs présents dans une fenêtre sont extraits autour d'un point anatomique et ne sont pas comparés un à un, mais des groupes de SURFs sont comparés entre eux en utilisant une mesure de similarité qui permet de caractériser la ressemblance avec une résolution spatiale ajustables. Cette méthode exploite une idée développée dans [Zhou et al., 2009] qui consiste à associer un ensemble de points caractéristiques à un objet de façon à pouvoir le suivre grâce à ces points.

c) Mesure de similarité

La mesure de similarité utilisée dans la fonction de vraisemblance (étape 7 table II.1) est inspirée de [Zhou et al., 2009] mais elle a été simplifiée afin de réduire le coût CPU (voir ci-après). Elle mesure la similarité $\mathcal{S}$ entre deux groupes de SURFs S_0 et S_1 :

$$\begin{aligned}
\mathcal{S}(S_0, S_1) &= \sum_{P \in S_0} \sum_{Q \in S_1} \mathcal{F}(P,Q) \times \mathcal{G}(P,Q) \text{ , avec} \qquad \text{(II.6)} \\
\mathcal{F}(P,Q) &= \exp\left(-\frac{\|P-Q\|_{2D}^2}{2\sigma_{2D}^2}\right) \\
\mathcal{G}(P,Q) &= \exp\left(-\frac{\|P-Q\|_{desc}^2}{2\sigma_{desc}^2} - k \times angleDist(P,Q)^2\right)
\end{aligned}$$

avec $\|.\|_{2D}$ et $\|.\|_{desc}$ les normes euclidiennes, respectivement, dans le plan et l'espace des descripteurs. La fonction $angleDist(a,b)$ renvoie la distance angulaire entre deux SURF a et b , k est une constante qui mesure l'importance donnée à la correspondance angulaire entre deux SURFs, les paramètres σ_{desc} définit la tolérance dans l'espace des descripteurs tandis que

σ_{2D} définit la tolérance spatiale. Cette mesure de similarité permet d'identifier deux nuages de SURFs similaires si leurs localisations sont très proches. Comme il n'y a qu'un nuage de SURFs correspondant à chaque point anatomique du visage (une seule bouche, un seul œil gauche...), le maximum de cette fonction va correspondre à l'élément recherché.

II.4.3 Implémentation du traqueur

a) Généralités

Le but est de recaler notre modèle 2D de visage (figure II.4) sur chaque image dans le flot vidéo et donc d'estimer :

- ses coordonnées image (x, y) (c'est-à-dire la position du point équidistant des deux yeux),
- son facteur d'échelle s (c'est-à-dire la taille du modèle par rapport à l'image),
- l'angle α et l'orientation globale β suivant l'axe optique,
- la distance inter-yeux l_e, la distance l_f bouche-yeux.

Concernant le modèle de dynamique $p(\mathbf{x_t}|\mathbf{x_{t-1}})$ du vecteur d'état $\mathbf{x_t}$ à l'instant t, les mouvements image des personnes observées sont difficiles à caractériser. Ici encore, nous supposons que les composantes de l'état évoluent suivant des modèles de marche aléatoire gaussienne indépendants, soit

$$p(\mathbf{x_t}|\mathbf{x_{t-1}}) = \mathcal{N}(\mathbf{x_t}|\mathbf{x_{t-1}}, \Sigma),$$

où $\mathcal{N}(.|\mu, \Sigma)$ est une distribution gaussienne de moyenne μ et covariance $\Sigma = diag(\sigma_x^2, \sigma_y^2, \ldots)$. Le vecteur d'état estimé dans le filtre est dans $\Re^6$: $\mathbf{x_t} = [x_t, y_t, \alpha_t, \beta_t, l_{e_t}, l_{f_t}]^T$.

Avec les notations introduites en section II.2, nous définissons la fonction $\pi(\mathbf{x}_t(.)|z_t)$ (voir équation II.5) pour chaque paramètre $(.)$ de $\mathbf{x_t}$ comme un mélange de gaussiennes estimé à partir des B associations possibles yeux-bouche détectés, par exemple pour $\{l_{e_t}^j\}_{j=1,\ldots,B}$, nous avons :

$$\pi(l_{e_t}|z_t) = \sum_{j=1}^{B} \mathcal{N}(l_{e_t}|l_{e_t}^j, \Sigma).$$

Concernant la fonction de vraisemblance $p(z_t|\mathbf{x_t})$, nous faisons l'hypothèse simplificatrice que les mesures de similarité sur les distributions de SURFs sont indépendantes entre elles conditionnellement à l'état $\mathbf{x_t}$, de sorte que pour $(S_0, S_1) = (f(\mathbf{x_t}), z_t)$, on a :

$$p(z_t|\mathbf{x_t}) = \prod_{f \in \mathcal{F}} \mathcal{S}_f(S_0, S_1),$$

avec $\mathcal{F} = \{left\ eye, right\ eye, mouth\}$.

Afin de rendre la méthode plus robuste, on réalise un apprentissage en ligne des SURF qui ont été détectés correctement. De ce fait, les SURF ne permettent que de détecter la personne

Symbole	Signification	Valeur
N	nombre de particules par filtre	250
α	taux de particules générées suivant les détections	0.4
-	résolution des images	640×480
-	espace de couleur de travail	Irg
Σ	écart-type des modèles de marche aléatoire	$\begin{pmatrix} 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.07 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.020 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.002 \end{pmatrix}$

TAB. II.5: Valeurs des paramètres principaux du traqueur de regard.

suivie. Cette détection spécifique permet de corriger l'apprentissage hors-ligne lorsque celui-ci fait défaut. Le tableau II.5 énumère les paramètres de notre traqueur dont les valeurs associées sont estimées empiriquement.

b) Optimisation de la fonction de similarité

La comparaison de deux SURFs est coûteuse en calcul, et celle de deux groupes de SURFs encore plus. La complexité de la fonction de similarité σ (equation (II.7)) est très importante. Il a donc été nécessaire de réduire le coût en calcul de cette fonction de similarité afin de garantir l'exécution du suivi en temps réel. Nous avons pour cela choisi d'utiliser des tables pré-calculées afin d'éviter de re-effectuer des calculs similaires, notamment pour approximer la fonction exponentielle qui est la plus coûteuse en terme de calculs dans l'équation (II.7). De cette façon le coût calcul de la fonction exponentielle a été réduit de 80%.

Considérons deux groupes de SURFs S_0 et S_1 et une translation possible $X = (x, y)$ appliquée à S_0 (ou de façon équivalente, une translation de $-X$ appliquée à S_1). Détecter un objet défini par S_0 revient à chercher le maximum de la fonction :

$$f(x,y) \;=\; \mathcal{S}(S_0, S_1 - X) \tag{II.7}$$

L'équation (II.7) devient :

$$\begin{aligned} f(x,y) &= \sum_{P \in S_0} \sum_{Q \in (S_1 - X)} \mathcal{F}(P,Q) \times \mathcal{G}(P,Q) \\ &= \sum_{(P,Q) \in S_0 \times S_1} \mathcal{G}\,(P,Q) \exp\left(-\frac{\|(P-Q)+X\|^2_{2D}}{2\sigma^2_{2D}} \right) \\ &= \sum_{Y \in \Gamma = S_0 - S_1} \alpha_Y \exp\left(-\frac{\|Y+X\|^2_{2D}}{2\sigma^2_{2D}} \right) \\ \text{avec } \alpha_Y &= \sum_{(P,Q) \in S_0 \times S_1, P-Q=Y} \mathcal{G}\,(P,Q) \end{aligned} \tag{II.8}$$

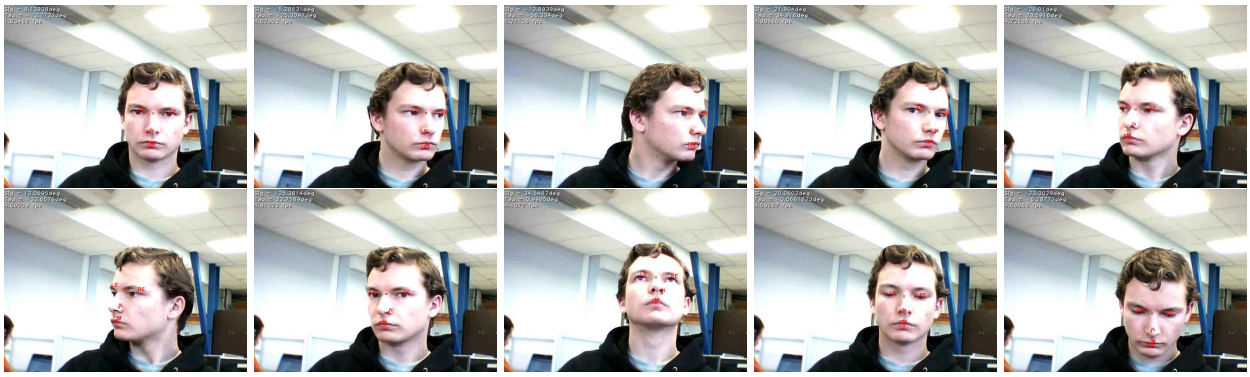


FIG. II.8: Exemple de réalisation : suivi sur une séquence prise par webcam. Les croix rouges représentent les centroïdes estimés des régions d'intérêt.

En effet, pour deux groupes de descripteurs, la fonction de similarité $\mathcal{S}$ se réduit à une somme de gaussiennes dans l'espace image. Ces gaussiennes ont le même écart-type σ_{2D} et les coordonnées de leurs centres correspondent à des pixels entiers. Nous recherchons le maximum de la fonction de similarité. Pour cela, une méthode très rapide consiste à appliquer un simple filtre de flou gaussien de rayon σ_{2D} à une image de valeurs nulles, dont les seules valeurs non nulles sont aux coordonnées des centres des gaussiennes Y et dont les valeurs sont les coefficients α_Y. La complexité est alors linéaire par rapport à la surface analysée. L'algorithme de recherche de maximum de la fonction σ dans une zone de l'image est donc le suivant :

1 . récupérer l'ensemble de SURF servant de modèle,
2 . extraire les SURFs de la zone du visage segmenté,
3 . calculer les termes exponentiels dépendant des descripteurs (les α_Y),
4 . dans une image I en niveau de gris (avec suffisamment de précision), pour chaque $Y \in \Gamma$ (Γ étant l'ensemble des centres des gaussiennes) ajouter α_Y à $I(Y)$,
5 . appliquer un flou gaussien de rayon σ_{2D} à I,
6 . rechercher le maximum de I.

La pertinence du maximum trouvé est ensuite évaluée en fonction de la valeur moyenne du bruit ambiant. Si cette valeur est très supérieure au bruit, alors il ne s'agit pas d'une fausse détection mais bien d'un ensemble de descripteurs très semblable au modèle.

II.4.4 Expérimentations et résultats associés

Le traqueur a été tout d'abord testé sur des séquences vidéo acquises depuis une *webcam* lors de son prototypage. La figure II.8 présente les résultats obtenus sur une séquence comportant une grande variabilité d'angle en azimut et en élévation. Malgré des rotations bien supérieures à ± 45, notamment en azimut, le suivi se fait correctement tout au long de la séquence. Les vidéos associées aux différents extraits présentés et quelques vidéos complémentaires sont accessibles à l'URL `www.laas.fr/~bburger/`.

FIG. II.9: Exemple de réalisation : situation H/R (haut gauche) et suivi sur une séquence acquise depuis le robot. Les croix rouges représentent les centroïdes estimés des régions d'intérêt.

Le traqueur a ensuite été intégré dans l'architecture du robot humanoïde HRP2 (voir chapitre IV) puis testé hors ligne sur une base de 2500 images acquises depuis le robot dans un scénario d'interaction H/R proximale ($[0.5m; 2.5m]$). Cette base, notée Seq_HRP2, inclut cinq sujets, des sauts dans la dynamique de la cible, voire des variations de distance H/R ou d'illumination. La figure II.9 illustre le comportement qualitatif de notre filtre sur une de ces séquences.

Le traqueur opère à une fréquence de $5Hz$ sur le Pentium $2.3Ghz$ du robot. L'échantillonnage, visant à explorer adaptativement les zones pertinentes de l'espace d'état par la fonction d'importance ((II.5)), permet de réduire le nombre de particules à $N = 200$ tandis que le nombre de SURFs varie de 50 à 100 durant le processus de suivi. Le traqueur fonctionne logiquement tant que les deux yeux restent visibles dans l'image donc environ sur $\pm 75°$.

Des évaluations quantitatives en terme de robustesse et de précision ont été menées. Le caractère aléatoire du filtrage particulaire ne permettant pas de baser celles-ci sur une seule réalisation de suivi, une étude statistique du comportement moyen du filtre est effectuée sur la base de 10 réalisations appliquées sur chaque séquence de la base de tests. Nous avons évalué le taux d'échec (en $\%$) sur la base Seq_HRP2. Ce dernier est quantifié par le nombre de décrochages observés, chacun étant notifié lorsque la distance entre la position image estimée et la vraie position est supérieure à un seuil préalablement fixé.

Ce taux est de 75% sur les séquences de la base Seq_HRP2 incluant une seule personne et 66% sur les séquences incluant jusqu'à 5 personnes se présentant successivement face au robot HRP2. Ce dernier résultat tend à démontrer l'adaptabilité rapide du traqueur grâce à l'apprentissage en ligne. Les écart-types associés à ces deux statistiques (chacune ayant été répétée, rappelons-le, 10 fois) sont de 3.5% attestant ici de la bonne répétabilité du traqueur malgré sa nature stochastique.

Concernant la précision, une vérité terrain sur les angles estimés du regard était nécessaire. Nous avons ici exploité la base public d'images *Yale* [Georghiades et al., 2001] pour laquelle chaque image est étiquetée, permettant notamment de connaître les angles associés τ (*pan*) et le σ (*tilt*) du regard. La précision obtenue est de l'ordre de $10°$ pour ces angles pour un écart-type de $1°$.

II.5 Conclusion

Ce chapitre présente nos deux systèmes de suivi, l'un dédié aux gestes bi-manuels et l'autre au regard. Ceux-ci sont intégrés dans les architectures logicielles de nos plateformes *via* les modules *GEST* et *GAZE*. Nous listons ci-après les spécificités de chaque système ainsi que leurs extensions possibles, tandis que la figure II.10 rappelle notre architecture définie en introduction et complétée ici par les modules $GEST$ et $GAZE$ décrits dans ce chapitre.

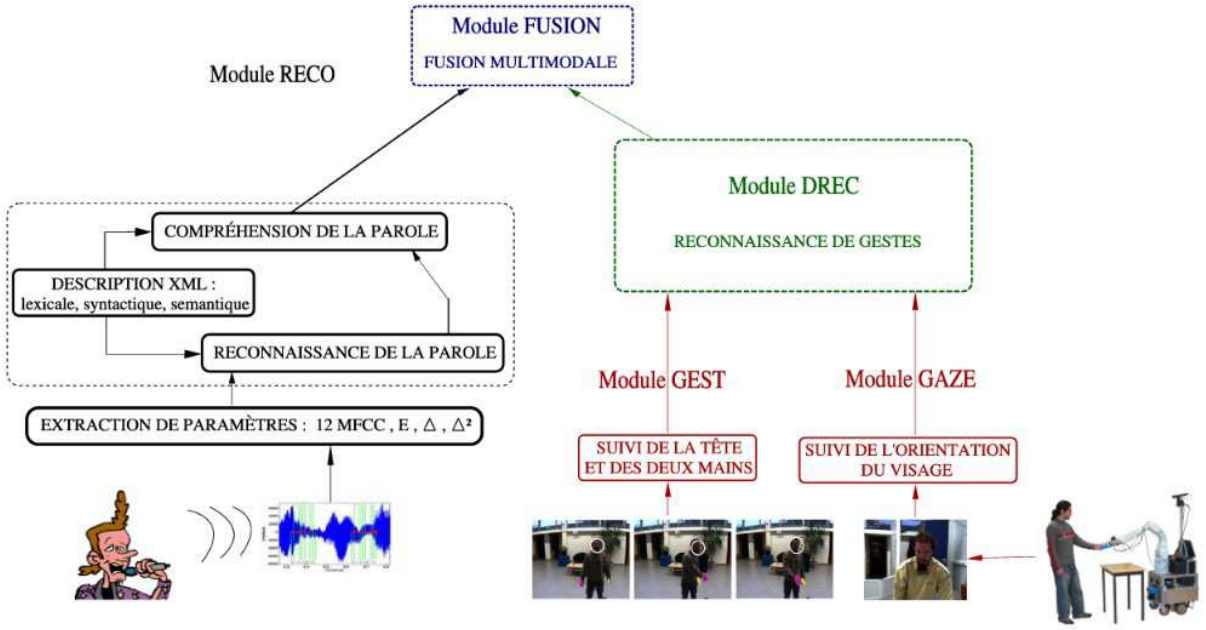


FIG. II.10: Architecture globale de notre interface homme-robot.

a) Traqueur de gestes 3D

Nous avons adapté la stratégie IDMOT, proposée par [Qu et al., 2007] sur du suivi multi-personnes et basée sur des filtres distribués et interactifs, à notre contexte de suivi des extrémités corporelles (mains+tête) d'une personne. Cette stratégie est connue pour limiter les erreurs de labellisation et de fusion des cibles. Notre stratégie IIDMOT, en s'appuyant sur le formalisme de filtrage particulaire ICONDENSATION, permet initialisation et ré-initialisation automatique, lors d'occultations ou de sortie du champs de vue. Ces situations sont courantes dans notre contexte lorsqu'un utilisateur non-expert exécute un geste. De plus, notre traqueur, par une modélisation minimaliste des membres corporels, est indépendant de l'homme observé et offre des temps de calcul compatibles avec notre contexte applicatif où un des challenges est la réactivité du robot pour se voir accepté par l'homme. Enfin, la stratégie de fusion de données hybrides, car basée sur des informations d'apparence mais aussi de reconstruction, se démarque des approches existantes dans la littérature. Ces travaux ont contribué aux publications [Burger et al., 2008a, Burger et al., 2008b, Burger et al., 2010].

Ce traqueur est assez mature et robuste car largement utilisé sur nos plateformes. Quelques extensions pourraient néanmoins améliorer ses performances et surtout son applicabilité. Une hypothèse forte est le port de manches longues, nos mesures reposant majoritairement sur la teinte chair. Nos investigations actuelles se focalisent sur l'ajout de carte de disparité stéréo (figure II.11) qui permettra l'exploitation de contraintes 3D supplémentaires qui sont par définition plus discriminantes. L'objectif est aussi de robustifier le traqueur lors de déplacements du robot. A moyen terme, nous visons à développer une stratégie de fusion auto-adaptative qui vise à choisir/pondérer les mesures les plus pertinentes en fonction du contexte courant donc de la situation du robot dans son environnement.

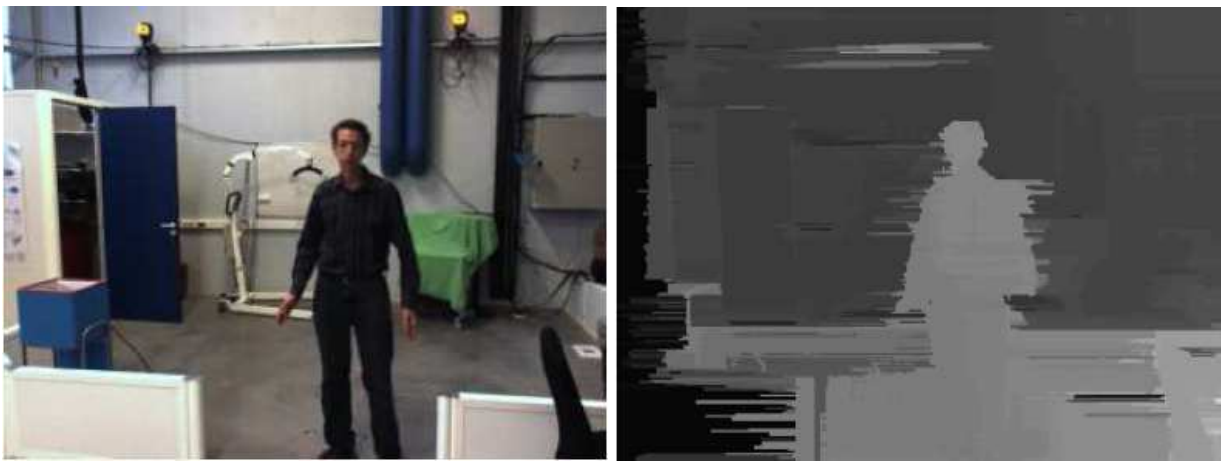


FIG. II.11: Image acquise depuis Jido et carte de disparité associée.

b) Traqueur de regard

Le traqueur de regard est moins abouti que le précédent car développé plus récemment. Son objectif est d'estimer la direction du regard à partir du suivi image de régions d'intérêt dans un contexte d'interaction proximale homme-robot. Notre traqueur par filtrage particulaire se démarque des approches existantes par :

1 . Une fonction d'importance dans une stratégie de filtrage ICONDENSATION basée sur la dynamique et le résultat de détecteurs permettant la (ré)-initialisation du filtre. L'absence courante de détections permet de faire évoluer le nuage de particules selon la dynamique et donc de gérer des rotations en azimut bien supérieures à ± 45, c'est-à-dire tant que les deux yeux restent visibles. Nous nous démarquons ici des approches qui se limitent à la caractérisation du regard fronto-parallèle [Asteriadis et al., 2008, Gee and Cipolla, 1994, Heinzmann and Zelinsky, 1999, Valenti et al., 2008].

2 . Une fonction de vraisemblance basée sur la similarité de distributions de SURFs associés à ces régions d'intérêt avec apprentissage conjoint hors et en ligne. La configuration du nuage de SURFs, ainsi que leurs descripteurs, sont ici considérés afin de pallier l'instabilité ponctuelle de l'un des deux critères.

3 . L'optimisation de son calcul par quelques approximations et pré-calculs effectués hors-ligne afin de répondre aux contraintes temporelles de notre contexte applicatif.

Ces travaux ont permis la rédaction de la communication [Brochard et al., 2009].

Nos investigations actuelles visent à améliorer la robustesse du traqueur dont le concept, basé sur des SURFs, nous semble prometteur. Une extension à moyen terme serait de considérer davantage de points anatomiques (oreilles, barbe, moustache, lunettes, structure de la chevelure, etc). Le défi sera alors de permettre la combinaison, mais aussi la disparition et la réapparition de ces divers points, ce qui impose l'utilisation d'une stratégie de filtrage plus souple que la simple ICONDENSATION, comme une adaptation de notre stratégie IIDMOT.

Chapitre III

Reconnaissance de gestes

Dans notre approche ascendante allant du suivi de geste (qui à fait l'objet du chapitre précédent) à sa reconnaissance exposée dans le présent chapitre, nous avons développé un nouveau module chargé de reconnaître un ensemble de gestes, symboliques ou déictiques. Ces gestes peuvent être réalisés par l'utilisateur lors de commandes multimodales, ces dernières pouvant alors nécessiter de fusionner les informations transmises par le geste avec le message verbal qui l'accompagne (cette fusion fera l'objet du prochain chapitre). Notre objectif est ici de reconnaître ces gestes de manière la plus robuste possible et ce malgré les contraintes inhérentes à notre contexte applicatif. Une première contrainte provient des entrées de ce système, puisque nous nous appuyons ici sur des données potentiellement imparfaites. En effet, malgré sa robustesse, le suivi de geste peut décrocher temporairement durant l'exécution d'une commande. Le suivi peut également s'avérer d'une précision insuffisante dans son calcul de la profondeur, notamment dans des situations de sous- ou sur-éclairage qui ne permettent pas de distinguer correctement les mains de l'utilisateur. De même, le suivi de l'orientation du visage, investigation récente dans cette thèse, ne donne pas toujours satisfaction. La seconde contrainte est la même que pour tous les modules précédemment décrits : les limitations en puissance de calcul imposent que les gestes soient reconnus en un temps acceptable et sans accaparer les ressources nécessaires au fonctionnement des autres modules embarqués.

Dans ce chapitre, nous présentons tout d'abord un état de l'art non exhaustif sur la reconnaissance de gestes, notamment dans le cadre de l'interaction homme-robot. Les HMMs, et à moindre degré les DBNs, sont largement exploités dans ce contexte. La section III.2 rappelle les formalismes du HMM, puis du DBN, appliqués à notre problématique. Nous détaillons ensuite notre système de reconnaissance basé sur une modélisation par HMM ou DBN, ainsi que les divers prétraitements utilisés afin de rendre ce dernier plus robuste. Nous expliquons également comment nous nous sommes servis des propriétés des DBNs afin de permettre une segmentation automatique des gestes. Enfin, la section III.4 montre, outre une comparaison des formalismes HMM et DBN pour la reconnaissance de gestes dans notre contexte, l'intérêt et les performances de notre système de reconnaissance hors-ligne, mais également sur notre robot.

III.1 État de l'art

Dissertant sur le canal gestuel associé à la main, [Cadoz, 1994] considère trois fonctions distinctes, mais complémentaires, intervenant à des degrés différents dans chacune des deux autres :

1. une fonction d'action matérielle, de modification et de transformation de l'environnement (typiquement, prendre un verre), nommée fonction *ergotique*,
2. une fonction *épistémique* de perception de l'environnement (par exemple lorsque l'on tâte un objet),
3. une fonction d'émission d'information à destination de l'environnement dite fonction *sémiotique*.

Dans le cadre de cette thèse, et par conséquent dans la suite de cet état de l'art, nous ne nous intéressons qu'à la dernière fonction qui correspond en réalité aux gestes dits *communicatifs*. Ces derniers ont été déclinés dans une taxonomie des gestes par [Quek, 1994], que l'on peut résumer par le schéma III.1. Nous nous focalisons ici sur les gestes de commande, c'est-à-dire des gestes déictiques (typiquement des gestes de pointage) et des gestes symboliques modélisant (c'est-à-dire en lien avec la parole et renforçant le sens de ce mode de communication). Nous commettrons par la suite l'abus de confondre ces derniers avec l'ensemble des gestes symboliques.

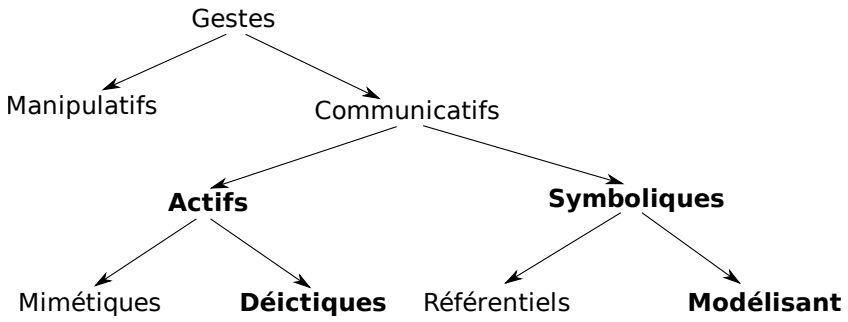


FIG. III.1: Taxonomie des gestes proposée par [Quek, 1994].

La reconnaissance de gestes basée sur des données visuelles a été utilisée pour nombre de tâches et compte aujourd'hui quelques applications bien établies dans les communautés IHM et IHR [Derpanis, 2004, Ong and Ranganath, 2005, Pavlovic et al., 1997]. La littérature associée divise la reconnaissance de gestes selon deux approches principales : la mise en correspondance de modèles statiques (en anglais, « template matching ») [Park et al., 2005, Triesch and Von der Malsburg, 2001, Waldherr et al., 2000, Yoshizaki et al., 2002] et celle basée sur des modèles dynamiques.

La première consiste à utiliser des modèles statiques basés sur les données images [Bretzner et al., 2002, Huang et al., 2002, Rogalla et al., 2004, Thayananthan et al., 2003]. Dans de tels cas, on ne peut pas réellement modéliser et reconnaître de gestes (dynamiques), mais on parle

de reconnaissance de configuration, ou parfois de « gestes statiques » ([Bretzner et al., 2002] reconnaît par exemple certaines configurations d'une main). Selon [Pavlovic et al., 1997], les gestes se divisent en trois étapes :

1 . la *préparation*, durant laquelle la main est déplacée vers la position de départ du geste,

2 . le *noyau*, qui correspond à la phase de réalisation effective du geste,

3 . la *rétraction*, où la main revient à sa position de départ avant l'exécution éventuelle du geste suivant.

Cette approche peut donc être pertinente pour certains gestes dont le noyau est statique, comme par exemple les gestes de pointage ou un geste tel que « stop ». Mais, contrairement à la seconde approche (dynamique), elle ne permet pas de modéliser des gestes tels que « salut » ou « viens vers moi » dont le noyau est actif (ces gestes et quelques autres sont exposés en annexe G). C'est pourquoi nous nous intéressons davantage à cette modélisation des mouvements dynamiques pour laquelle les réseaux de neurones [Stiefelhagen et al., 2004, Waldherr et al., 2000] ainsi que les modèles de Markov cachés (HMMs) [Chen et al., 2003, Nickel and Stiefelhagen, 2006, Yang et al., 2007] et leurs variantes : IOHMM [Just et al., 2004], Coupled-HMM [Oliver et al., 2000], S-HSMM [Duong et al., 2005], etc ont été largement exploités dans la communauté. Les HMMs conventionnels sont notamment utilisés pour leur simplicité et leur fiabilité.

Un grand nombre d'interfaces d'IHM existent en environnement contrôlé (par exemple [Chen et al., 2003, Huang et al., 2002, Just et al., 2004, Ong and Ranganath, 2005, Zieren et al., 2002]). De même, quelques interfaces d'IHR ont été incorporées et évaluées sur des plateformes réelles. Toutefois, presque aucune ne satisfait à toutes les exigences d'une plateforme mobile autonome. En plus des interfaces multimodales mentionnées en introduction de ce mémoire, citons : [Yoshizaki et al., 2002] qui reconnaissent simplement une forme construite par les positions successives d'un doigt, [Rogalla et al., 2004] qui reconnaissent des postures 2D d'une seule main et [Stiefelhagen et al., 2004] qui reconnaissent des gestes de pointages 3D en les découpant en trois phases, chacune modélisée par un HMM ; nous donnons ici quelques exemples d'interfaces intéressantes de reconnaissance de gestes trouvées dans la littérature IHR. [Waldherr et al., 2000] introduit ainsi une interface de reconnaissance de gestes dynamiques à partir de données issues d'une caméra couleur dans le but de contrôler un robot mobile, mais seuls quatre gestes déictiques mono-manuels ont été expérimentés. [Yang et al., 2007] montrent un module avancé de reconnaissance par HMM de quatorze mouvements du corps entier, mais la capacité du système à fonctionner en présence d'un environnement encombré et à satisfaire des critères temps réel n'est pas clair. [Triesch and Von der Malsburg, 2001] reconnaît douze configurations de la main grâce à une mise en correspondance de graphes élastiques dans un scénario de manipulation (dit « pick-and-place ») utilisant un vrai robot. [Richarz et al., 2006] utilisent un filtre de Gabor pour extraire un vecteur de données d'un système monoculaire et une cascade de perceptrons multi-couches comme estimateur, le tout afin de caractériser des gestes déictiques basés sur un bras statique et une détection de visages frontaux.

Enfin, il est à noter que la plupart des approches existantes dans la littérature IHR [Corradini and Gross, 2000, Park et al., 2005, Shimizu et al., 2006, Stiefelhagen et al., 2004, Triesch and Von der Malsburg, 2001, Yoshizaki et al., 2002] supposent un utilisateur droitier et ne se déplaçant pas pendant qu'il effectue un geste quasi-fronto-parallèle et mono-manuel. En suivant

en 3D la tête en plus des deux mains de l'utilisateur, notre système peut reconnaître des gestes même en présence de mouvements du sujet. Nous nous intéressons donc ici à des gestes mono-, mais également bi-manuels 3D, car les mouvements naturels se font dans l'espace et la plupart des être humains, gauchers ou droitiers, ne connaissant pas le système doivent être capables de commander le robot. De plus, [Just et al., 2004] ont montré que les gestes bi-manuels offrent des signatures plus discriminantes donc aboutissant à des taux de classification plus élevés.

[Quek, 1994], reprenant une étude anthropologique de [Kendon, 1980], montre qu'un humain se sert d'un certain nombre d'observations pour définir qu'un geste est expressif et en déterminer la segmentation :

- le mouvement brusque d'une extrémité du corps (typiquement une main) en s'éloignant de ce dernier, puis retournant à sa position d'origine,
- les mouvements de la tête qui retournent à leur position de départ,
- les mouvements du corps entier qui retournent à leur position de départ.

Les méthodes décrites dans la littérature afin de permettre une telle segmentation de manière automatique sont relativement rares. [Aggarwal and Cai, 1999] modélisent le début et la fin d'un geste par deux HMMs. Pour sa part, [Zhao, 2001] s'inspirent de la reconnaissance de parole en développant une méthode de segmentation par *zero-crossing*. Il s'agit alors de calculer différents paramètres du mouvement et d'en déduire des seuils d'activité. [Wang et al., 2001] s'inspire de cette dernière méthode et se basent sur les minimums locaux en accélération afin de déterminer les limites d'un geste. Enfin, [Kahol and Kahol, 2003] utilisent un classifieur bayésien appris sur des segmentations définies par plusieurs utilisateurs, et dont le vecteur d'état représente l'activité de segment corporels connectés. Globalement, la segmentation automatique de gestes reste marginale dans la communauté IHR.

Les principales spécificités de notre système de reconnaissance de gestes, consistent donc en :

- la modélisation des gestes par DBN, marginalement exploité dans la littérature et particulièrement en IHR,
- la prise en compte de gestes bi-manuels, là encore souvent absente de la littérature IHR,
- l'intégration de données issues du suivi de l'orientation de la tête,
- la segmentation automatique de gestes.

III.2 Méthodes utilisées pour la reconnaissance de gestes

Comme nous l'avons vu, les modèles de Markov cachés (HMMs) sont largement répandus dans la communauté parole, mais sont également populaires, avec leurs variantes, dans d'autres domaines tels que la reconnaissance de gestes qui nous intéresse ici. Des modèles à dépendances temporelles plus généraux tels que les réseaux bayésiens dynamiques (ou DBNs pour "Dynamic Bayesian Network") ont été utilisés pour la modélisation et la reconnaissance d'activités

humaines [Du et al., 2006], mais également dans quelques applications à la reconnaissance de gestes comme [Suk et al., 2008] et [Arriaga et al., 2003] qui reconnaissent ainsi des gestes 2D. Mais cette représentation générique reste marginale et n'a pas encore, à notre connaissance, été utilisée dans le cadre de IHR. C'est pourquoi, après avoir créé un système de reconnaissance par HMMs du niveau de l'état de l'art et comme nous le verrons dans cette section, nous avons exploré le formalisme associé au DBN. Les principales motivations de ces investigations sont :

- les avantages avérés du DBN par rapport aux HMMs dans notre contexte applicatif,
- l'expertise dont nous disposons en reconnaissance de parole (voir chapitre I), en filtrage particulaire (voir chapitre II) et sur le formalisme DBN exploité dans un contexte de navigation robotique [Infantes, 2006].

III.2.1 Formalisme HMM

a) Généralités

La reconnaissance par modèles de Markov cachés ayant déjà été abordée dans le chapitre I, nous ne reviendrons pas ici sur la définition de ces derniers. Nous ne reviendrons pas non plus sur la façon de mener leur apprentissage ou d'effectuer une reconnaissance grâce à ses derniers, puisque que la démarche est similaire à celle de la reconnaissance de mots isolés par des modèles de mots. Cette sous-section a plutôt pour but de présenter brièvement leur adaptation dans le contexte de la reconnaissance de gestes et par là-même d'introduire quelques concepts utiles à la compréhension des DBNs qui seront présentés dans la sous-section suivante.

Dans notre application de reconnaissance de gestes, nous avons choisi d'utiliser un espace discret des observations. Ce choix est lié en premier lieu à des raisons de simplicité et d'efficacité. En effet, un espace continu, et par conséquent une modélisation par gaussiennes, impose une quantité de données d'apprentissage conséquente afin d'obtenir un modèle représentatif. Un espace discret, dont on pourra adapter le nombre et la taille des cellules du tableau de probabilités (voir sous-section III.3.1) permet une modélisation moins précise mais plus aisée et pouvant se contenter d'un corpus modeste. Ceci est d'une grande importance dans notre domaine qui ne dispose d'aucun corpus publique (à cause des difficultés d'acquisition, mais également de l'absence de standardisation de ces données, contrairement aux données sonores) et pour lequel la construction de telles bases de données se révèle bien plus longue et fastidieuse que pour la reconnaissance de parole.

La figure III.2 montre la signification d'une représentation factorisée (ou compacte) (à droite) au vu d'une représentation déployée (à gauche) d'un HMM. Rappelons que cette dernière a été utilisée tout le long du chapitre I, qui présentait les bases du formalisme des HMMs. Ces deux représentations (compacte ou déployée) sont strictement équivalentes, mais suggèrent deux conceptions différentes, la première privilégiant les relations temporelles, tandis que la seconde privilégie la structure. Dans la représentation factorisée, les N nœuds sont fusionnés en un seul nœud de N valeurs. Le lien horizontal reliant q_{t-1} à q_t représente alors à lui seul la matrice de transition A, bien que cela sous-entende les sous-liens $a_{ij} = P(q_t = S_j | q_{t-1} =$

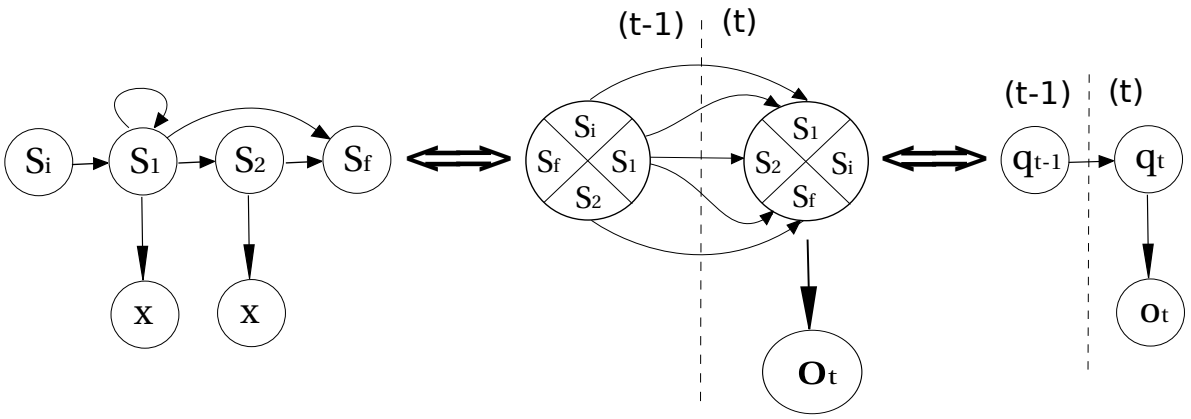


FIG. III.2: Représentations déployée et factorisée d'un HMM.

S_i), $1 \leq i, j \leq N$ (comme le montre le pseudo modèle du milieu de la figure III.2). De même, le lien vertical représente la matrice d'émission B.

b) Apprentissage dans le cadre HMM

L'apprentissage de nos modèles HMMs est assuré par un algorithme type : Expectation-Maximization (*EM*). Cet algorithme consiste à améliorer itérativement le modèle λ en calculant à chaque pas de temps la probabilité d'être en chaque état. Il utilise pour cela l'algorithme forward-backward [Baum, 1972] qui calcule les messages avant $\alpha_t(i)$ et arrière $\beta_t(i)$, c'est-à-dire les probabilités que la variable cachée ait pour valeur S_i à l'instant t connaissant le modèle et, respectivement, le début et la fin de la séquence d'observation O_T servant à l'apprentissage. Le lecteur pourra se référer à [Rabiner, 1989] pour plus de détails sur ces algorithmes.

III.2.2 Formalisme DBN

Un réseau bayésien dynamique (ou DBN pour "Dynamic Bayesian Network") ou réseau probabiliste dynamique [Dean and Kanazawa, 1990] est constitué d'un ensemble de variables cachées, de variables observables, ainsi que d'un ensemble de liens causaux entre ces variables. Il s'agit d'un réseau bayésien dans lequel les variables sont indicées par le temps (chaque variable a donc une instance à chaque pas de temps) et où les liens causaux sont les mêmes quel que soit le pas de temps. Ainsi, on peut voir le DBN comme une généralisation du HMM, qui sous-entend la gestion d'une seule variable cachée et un seul nœud observable lié par une structure simple et figée. En réalité, nous avons alors N_c variables cachées et N_o variables observables liés entre elles par quasiment n'importe quel type de structure causale. Les seules contraintes sont de n'avoir aucun lien causal cyclique et que l'hypothèse de Markov reste applicable, c'est-à-dire que les liens causaux influençant une variable à l'instant t ne peuvent provenir que de variables à l'instant t ou à l'instant $t-1$. Ceci permet une représentation compacte et

intuitive d'un DBN, puisqu'il suffit de représenter les variables et leurs liens causaux aux instants t et $t-1$ pour visualiser l'ensemble de la structure, c'est pourquoi on utilise très souvent la représentation compacte présentée précédemment. Enfin, comme pour les HMMs, les liens causaux sont constants au cours du temps tant en structure qu'en quantité (les probabilités sont indépendantes du temps).

a) Définition

Afin de formaliser cette représentation, basons nous sur les notations utilisées pour les HMMs dans le chapitre I et adaptons les à notre cadre DBN. Nous notons ainsi N le nombre de variables q^i (cachées ou non) et q_t^i la valeur de ces variables à l'instant t ($0 < i \leq N$). Chacune de ces variables peut prendre l'une des valeurs S_j^i, avec $0 < j \leq M^i$, où M^i est le nombre de valeurs possibles pour la variable q^i et est appelée l'arité de celle-ci. O_t représente alors la variable composite de toutes les variables observables à l'instant t (qui, rappelons le, étaient concaténées en un seul vecteur d'observation représenté par un unique nœud dans le cadre HMM). Nous employons également l'expression de « parents d'une variable » afin de désigner l'ensemble des variables créant un lien causal entrant dans la variable en question. On note alors $pa(q_t^i)$ les parents d'une variable q^i à l'instant t.

Nous pouvons maintenant définir la structure d'un DBN (en suivant la représentation concaténée, c'est à dire ne contenant que les deux tranches temporelles $t-1$ et t) de la manière suivante :

- un ensemble de variables $Q = \{q^1, ..., q^N\}$,
- un graphe direct acyclique $G = (\mathbf{So}, \mathbf{A})$ dont les sommets ($\in \mathbf{So}$) sont organisés en 2 tranches repérées par leur indice temporel ($t-1$ et t), $\mathbf{A}$ définissant l'ensemble des arêtes du graphe et représentants les liens causaux. $\mathbf{So}$ est tel qu'à chaque variable $q^i \in Q$ correspond un unique sommet dans la tranche temporelle t et au plus un sommet dans la tranche $t-1$ (dont les valeurs sont respectivement q_t^i et q_{t-1}^i). Les arêtes du graphe ($\in \mathbf{A}$) relient les sommets de manière acyclique et de telle façon que les sommets de la tranche $t-1$ ne soient reliés qu'avec des sommets de la tranche t.
- une paramétrisation Θ, qui associe à chaque sommet de la tranche t une distribution conditionnelle de probabilités $P(q_t^i|pa(q_t^i))$, où les parents de q_t^i sont définis par $\mathbf{A}$. Une distribution *a priori* est définie pour les variables non déductibles (sans lien causal entrant) en $t = 0$.

Partant de là, nous constatons que les probabilités de transition ne se calculent plus de même manière que pour un HMM. En effet, la probabilité d'une variable à l'instant t ne dépend plus uniquement de sa valeur à l'instant $t-1$, mais de l'ensemble de ses parents à t et $t-1$. Ainsi, la matrice A contient les

$$a_{\sigma_k i} = P(q_t^i = S_j^i | pa(q_t^i) = \sigma_k),$$

avec $\sigma_k \in EPa(q^i)$, $EPa(q^i)$ représentant ici l'ensemble des instanciations des parents d'une variable q^i et σ_k une instance de cette variable qui représente l'ensemble des valeurs prises par ses parents.

Il est à noter que, pour un réseau bayésien (dynamique ou non), le sens des flèches représentant les liens causaux est souvent interprété intuitivement comme la causalité au sens commun

du terme. Mais le sens de la flèche indique en fait simplement quelle est la probabilité conditionnelle qui est stockée dans les données quantitatives attachées à ce lien. Or, il n'y a pas de causalité dans une probabilité conditionnelle, puisqu'on peut très bien retourner ces probabilités en utilisant la règle de Bayes. L'utilisation de ce sens des flèches pour définir un lien de cause à effet est un abus par rapport à sa définition, bien que certains travaux essaient de tenir compte de cette sémantique. Dans le cadre des DBNs, la seule exception consiste en les liens causaux temporels (allants de $t-1$ à t), puisqu'il est évident qu'on ne peut influencer le passé, mais tous les autres liens sont parfaitement inversables.

b) Inférence exacte versus inférence approchée

➢ Inférence exacte

Comme nous l'avons vu, la représentation explicite de tous les liens causaux dans un DBN nous permet de faire apparaître des indépendances entre variables. Malheureusement, une représentation factorisée en de multiples variables pose un problème fondamental qui va rendre l'inférence exacte extrêmement complexe. En effet, si les interactions entre variables sont très structurées et locales, les corrélations vont se propager à travers le modèle et, en général, toutes les variables sont en réalité corrélées à toutes les autres. Dans ce cas, si l'on appelle « état courant cru » la distribution de probabilités sur les variables à un instant t, alors celui-ci n'est pas représentable de façon factorisée, puisqu'il doit contenir toutes les corrélations entre variables pour donner une image exacte du processus. Or, suivant ces structures, ces corrélations ne sont pas markoviennes d'ordre 1 :

- afin de tenir compte des corrélations entre un ensemble v_t de variables à l'instant t, il faut se souvenir des variables qui influencent cet ensemble à l'instant $t-1$ (v'_{t-1}),
- mais il est rare que ces variables ne soient pas elles-mêmes corrélées suivant des dépendances dues à un ensemble de variables v''_{t-2},
- etc.

Comme l'inférence est, par définition, la propagation dans le temps de cet état courant cru, dont la taille peut augmenter exponentiellement avec t, elle devient impossible à réaliser de manière exacte.

➢ Inférence approchée

Pour les raisons avancées précédemment, différentes approches ont été proposées afin de permettre une inférence approchée. Celle-ci est chargée d'approximer les corrélations entre variables et ainsi d'approximer l'état courant cru de façon à en tirer une représentation factorisée dont la taille n'augmente pas exponentiellement avec le temps t. Certaines visent à décomposer l'état courant cru en groupes (« clusters ») de variables faiblement corrélées, puis à approximer la corrélation entre ces différents clusters [Boyen and Koller, 1998, Boyen and Koller, 1999], mais ces approches sont fortement dépendantes du processus à modéliser, même si des variations les rendant un peu plus générales ont été proposées [Koller and Fratkina, 1998]. Une méthode plus générale est d'estimer l'état courant cru grâce à un filtre particulaire (voir chapitre II). Chaque particule y représente alors une hypothèse qui est réévaluée au fur et à mesure

du processus et garde l'ensemble des corrélations intrinsèques à chaque hypothèse.

Cette dernière méthode est celle que nous utilisons ici. Il est très important, notamment pour l'apprentissage du modèle, de maintenir une bonne approximation des corrélations entre variables et de leurs valeurs possibles. Mais il est dans le même temps indispensable d'avoir une représentation compacte pour que les algorithmes puissent traiter ce type de modèles. Cette application du filtrage particulaire permet de maintenir des hypothèses sur plusieurs pas de temps et de supprimer, lors du processus de reconnaissance, les particules les moins probables grâce au rééchantillonnage.

En pratique, nous utilisons un filtre de type CONDENSATION (voir chapitre II) pour modéliser l'état courant cru du système. Une particule représente alors une hypothèse sur ce dernier et chaque membre de son vecteur d'état contient une valeur possible d'une variable. Une particule est, par conséquent, une affectation complète des variables sur deux pas de temps consécutifs $t-1$ et t. Ces particules sont propagées à chaque pas selon les probabilités de transition du modèle λ (qui remplace ici la dynamique) :

$$q(x_t|x_{t-1}^n, O_t) = p(x_t|x_{t-1}^n, \lambda),$$

avec x_t l'état courant cru et x_{t-1}^n la l'état de la $n^{\text{ième}}$ particule du filtre. La pondération $\omega_t^n \propto p(O_t|x_t^n)$ des particules reflète alors la probabilité d'apparition des observations sachant l'affectation complète représentée par la particule. Une phase de rééchantillonnage est également utilisée (étape 9 de l'algorithme II.1).

c) Apprentissage dans le cadre DBN

Grâce à l'approximation des états courants crus, il devient possible d'aménager l'algorithme Expectation-Maximization (*EM*), classiquement utilisé pour les HMMs, pour permettre l'apprentissage d'un DBN. Une adaptation implicite de cet algorithme consiste à déduire directement les $\alpha_t(.)$ (message avant) du filtre particulaire. Mais, parce qu'elle permet une plus grande stabilité des calculs (notamment dans le cas d'un nombre de particules trop faible) et par là même de meilleurs résultats, nous utilisons la méthode décrites dans [Infantes et al., 2006]. Celle-ci consiste à n'approximer par le filtrage particulaire que les facteurs d'échelles nécessaires au calcul des $\alpha_t(.)$, rendant ce dernier calcul moins dépendant du filtrage.

d) Application à la reconnaissance de gestes

Il est possible d'utiliser les DBNs pour notre application en modélisant chaque geste par un DBN. Ainsi, durant la phase de reconnaissance, on teste la séquence de données à reconnaître sur chacun des modèles appris, c'est-à-dire qu'on calcule la probabilité de voir cette séquence suivant chaque modèle. Ainsi, celui menant au score le plus élevé est choisi comme étant le geste reconnu. Cependant, ce genre d'approche oblige à construire plusieurs modèles différents et par conséquent à dupliquer des informations qui peuvent être communes à plusieurs d'entre eux. Or, grâce à sa liberté de modélisation, il est possible de modéliser l'ensemble des gestes par un seul DBN dit « hiérarchique ». Il s'agit alors de grouper les différents modèles du départ en un seul modèle contenant une variable « de haut niveau » qui permettra par la suite de les différencier.

Par exemple dans notre application, il suffit d'ajouter un nœud G (à la façon de la variable **d** dans la figure III.4 de droite) qui influencera l'état interne du système et/ou ses autres variables observables. La méthode consiste alors, durant la phase d'apprentissage, à utiliser ce nouveau nœud comme une variable observable normale (le geste courant étant alors connu). Pour la phase de reconnaissance, deux solutions s'offrent à nous :

1 . Nous pouvons successivement tester la séquence à reconnaître sur cet unique DBN pour chaque geste, en imposant à chacune de ces itérations la valeur du nœud G (inconnu à ce moment) de manière à tester toutes ses valeurs. Mais cette méthode oblige malgré tout à effectuer autant de tests sur une séquence à reconnaître que de gestes.

2 . Nous pouvons également considérer le nœud G comme une variable cachée et nous chercherons alors à inférer sa valeur grâce à un filtre particulaire, comme nous l'avons décrit précédemment. Le résultat de la reconnaissance est ensuite déduit en marginalisant sur cette variable.

C'est cette seconde méthode que nous avons donc choisie d'utiliser.

Le lecteur intéressé par le formalisme détaillé des DBNs, certes dans un cadre applicatif différent, est invité à consulter le mémoire de thèse de Guillaume Infantes [Infantes, 2006]. Celle-ci a été réalisée au LAAS-CNRS et portait sur les aspects décisionnels de la navigation de robots à travers la modélisation de comportements par DBN.

III.2.3 DBN *versus* HMM : avantages et inconvénients respectifs

a) Sur la structure

Le lien entre HMM et DBN a été clairement explicité dans [Smyth et al., 1997], ainsi, un HMM peut être assimilé à un cas particulier de DBN. La figure III.3 montre un HMM (à gauche) et son équivalent sous forme de DBN (à droite), les nœuds grisés représentants les nœuds cachés. La figure de gauche exhibe une complétude des liens causaux, c'est-à-dire qu'il n'y a aucune indépendance entre les variables. Ainsi, même si l'on constate dans la réalité que l'observation **a** est indépendante de la classe $\mathbf{c_1}$, on est obligé, dans la représentation HMM, d'exprimer les probabilités conditionnelles correspondantes. Ceci n'étant pas le cas pour les DBNs, nous pouvons exprimer une telle indépendance par le retrait du lien causal en question, comme dans la figure III.4 (gauche), ce qui nous permet de rendre la représentation plus compacte.

Un deuxième avantage des DBNs porte sur la représentation des variables cachées. En effet, la représentation de l'état interne du système par une unique variable cachée n'est pas forcément réaliste. On peut par exemple vouloir séparer les sous-systèmes qui composent le système global afin de modéliser certaines indépendances d'un sous-système à l'autre ou de certaines variables d'observation relatives à l'un des sous-systèmes. Un état au sens classique du terme est alors

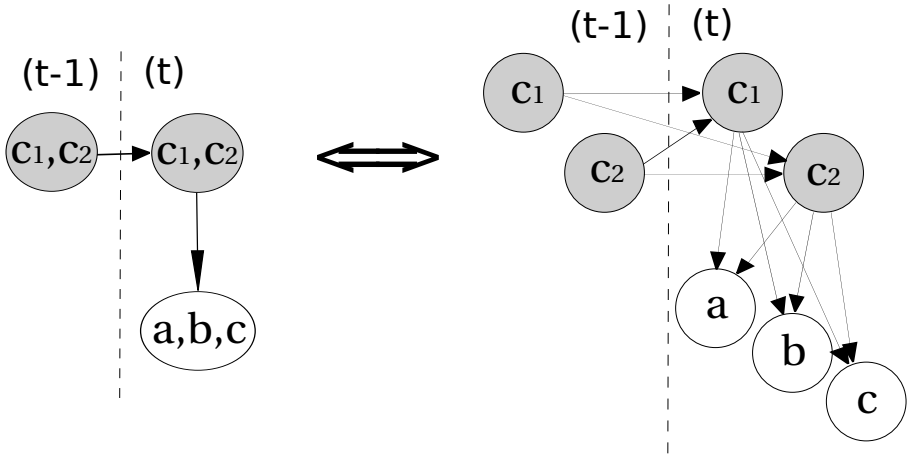


FIG. III.3: Équivalence HMM/DBN.

représenté par une instanciation de l'ensemble des différentes variables, des différents facteurs qui constituent l'état. On passe donc d'un état global à un ensemble de variables qui peuvent être de sémantiques différentes. La figure III.4 (gauche) illustre là encore la prise en compte de telles indépendances *via* le retrait de certains liens causaux. Il est également possible qu'un paramètre extérieur au système et observable influence celui-ci. Dans une structure de type HMM, on ne peut pas représenter ce type de variable simplement, mais il faudrait l'intégrer dans l'état caché, et donc se priver des observations correspondantes. Au contraire, dans un DBN, toutes les composantes peuvent être explicitées et prises en compte de la même manière et un lien causal entre une variable observable notée **d** et un nœud caché peut tout à fait être inséré dans la structure, à l'instar de la figure III.4 (droite).

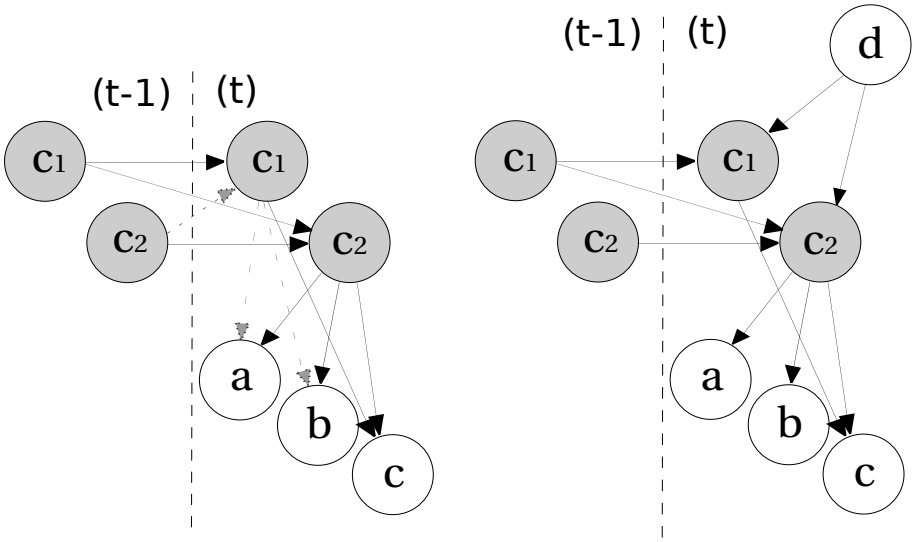


FIG. III.4: Avantages de la modélisation par DBN.

Il est donc clair que la représentation DBN est plus générique que la représentation HMMs. Le pouvoir expressif d'un DBN est plus élevé et sa structure graphique est relativement intuitive. Cette flexibilité permet une modélisation plus fine qui peut améliorer les taux de reconnaissance si le modèle est pertinent. Cette souplesse dans la modélisation est d'autant plus intéressante que les variables à considérer sont hétérogènes. De plus, des algorithmes bien adaptés permettent de gagner en complexité, et par conséquent en temps de calcul, du fait de l'utilisation des indépendances entre variables. Mais un désavantage qui en découle est sa plus grande complexité en terme de difficulté de choix de la structure. En effet, lors de la création du modèle, le nombre de possibilités de représentation est important et grandit avec le nombre de variables sous-jacentes. De plus, il n'existe aucun outil permettant une modélisation automatique, tout comme il n'existe aucun outil permettant de déterminer le nombre d'états à utiliser pour une modélisation par HMM.

b) Sur l'utilisation du filtrage particulaire

L'utilisation du filtrage particulaire dans le cadre d'un DBN hiérarchique afin d'inférer la valeur d'un geste à reconnaître, nous permet d'obtenir à chaque pas de temps la valeur du nœud de haut niveau la plus probable. Il est certes possible d'attendre la fin de la séquence d'observations afin de désigner le geste le plus probable sur l'ensemble de la séquence. Mais il est également possible de « deviner » le geste à reconnaître avant même la fin de la séquence, puisqu'un résultat est disponible à chaque pas de temps. Dans notre application, si un même geste se trouve être le plus probable durant un certain nombre de pas consécutifs et avec un score (relatif ou non) supérieur à un seuil prédéfini, l'algorithme se termine sans attendre la fin de la séquence et désigne ce geste comme reconnu, nous permettant ainsi d'économiser en temps CPU et d'accroître la réactivité du robot.

Il est à noter qu'un autre avantage de l'utilisation du filtrage particulaire, plutôt que des algorithmes de recherche plus classiques, est de permettre une répartition aisée de la charge de calcul à chaque pas du suivi de gestes et donc de la reconnaissance, au lieu de concentrer toute cette charge en fin de séquence. En effet, la plupart des algorithmes classiques, utilisés notamment dans le cadre HMM, nécessitent de connaître l'intégralité d'une séquence de données avant de pouvoir effectuer leurs calculs. Une conséquence en est que durant l'exécution d'un geste par une personne, un module de reconnaissance ne peut qu'acquérir les données et doit attendre la fin du geste pour calculer les probabilités liées. En utilisant un filtre particulaire, un module de reconnaissance peut effectuer ses calculs à chaque acquisition et ne requiert donc pas de supplément de CPU en fin de geste. Ce dernier avantage est un argument de poids en robotique où tout système qui doit s'intégrer dans une architecture temps réel se doit de répartir le plus uniformément possible sa charge de calcul lors de l'exécution du processus.

III.3 Implémentation

Nous allons maintenant détailler notre système de reconnaissance de gestes tel qu'il a été implémenté dans le cadre de l'interaction homme-robot.

III.3.1 Modélisation et prétraitements

a) Modélisation

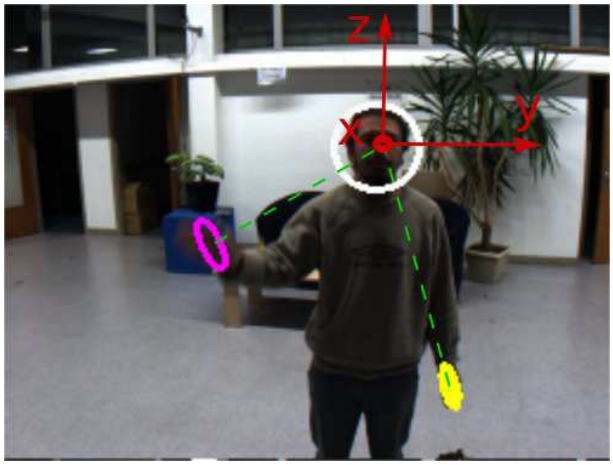


FIG. III.5: Notre système de coordonnées et la modélisation liée.

Afin de rendre notre système de reconnaissance indépendant de la position de l'utilisateur par rapport au robot, nous avons défini notre vecteur d'observation comme étant la position des mains relativement à la tête dans un repère sphérique centré sur la tête et orienté suivant ce que nous appellerons « le plan de l'homme » : les axes $\overrightarrow{y}$ et $\overrightarrow{z}$ de notre système de coordonnées forment ce plan, tandis que l'axe $\overrightarrow{x}$ est orienté vers l'avant de l'homme, orthogonalement au plan. Nous obtenons alors à chaque instant t le vecteur d'observation dans $\mathbb{R}^7$ suivant :

$$O_t = \{\rho(R), \theta(R), \varphi(R), \rho(L), \theta(L), \varphi(L), D_{H_R - H_L}\},$$

où $\rho(.), \theta(.), \varphi(.)$ sont les coordonnées sphériques des mains droite (R) et gauche (L), tandis que $D_{H_R - H_L}$ est la distance entre les deux mains. Il est à noter que l'orientation et la forme des mains n'est pas considérée ici. La raison est que ces données, bien que très utiles lors du suivi, ne sont pas assez fiables pour la reconnaissance, surtout dans des cas extrêmes d'éclairage (trop sombre ou trop clair). Le plan de l'homme est défini de la manière suivante :

- il est perpendiculaire au sol, c'est-à-dire que l'axe $\overrightarrow{z}$ est orthogonal au sol,
- il passe par le centre de la tête,
- il est coplanaire à la droite formée par le couple main gauche - main droite durant la position de repos (ceci définit l'angle θ_H qui représente son orientation par rapport au repère monde, c'est-à-dire par rapport à un repère fixe indépendant du robot et de l'homme).

Il est à noter que, l'orientation de ce plan (θ_H) étant définie durant la position de repos (c'est-à-dire, en réalité, à la première observation d'une séquence), l'homme n'est plus censé bouger durant l'exécution de son geste (ou uniquement en translation). Cette limitation est imposée par le manque de données auquel doit faire face cette modélisation : si nous disposions, par exemple,

de la position des jambes de l'utilisateur ou d'un suivi de son buste, elle n'aurait pas lieu d'être et l'orientation du plan de l'homme pourrait être recalculée à chaque pas de la reconnaissance. Malheureusement, si une telle détection de jambes est possible sur certains robots (comme Jido) grâce à un balayage laser, elle ne l'est pas sur d'autres et notamment sur un robot humanoïde. Mais cette limitation ne restreint que les mouvements de l'homme et il est tout à fait possible au robot de bouger durant l'exécution d'un geste sans perturber la reconnaissance (tant que le suivi réussit à conserver ses cibles et que le robot conserve une bonne approximation de sa position dans l'environnement). La figure III.5 schématise ce système de coordonnées, les ellipses et le cercle représentant la projection du résultat du suivi.

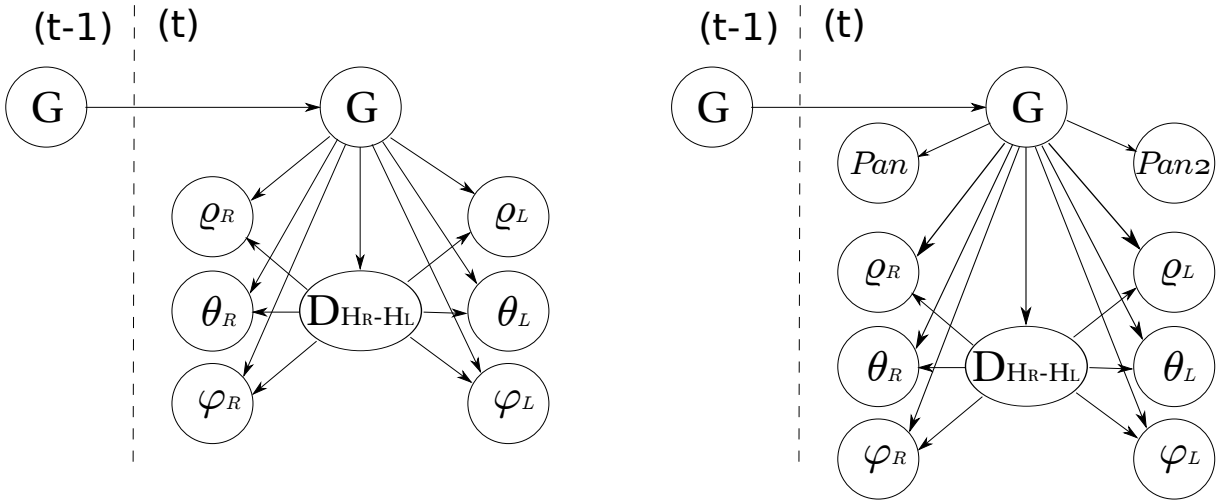


FIG. III.6: Structures utilisées pour la reconnaissance par DBN : modèles $MG1$ utilisant uniquement les données en provenance du module *GEST* - gauche - et $MG2$ utilisant également les données issues du module *GAZE* - droite -.

La figure III.6 montre la structure des DBNs utilisés actuellement pour la reconnaissance de gestes sur nos robots. Le modèle de gauche modélise la reconnaissance utilisant les données en provenance du module *GEST* seul, tandis que celui de droite utilise en plus les données issues du module *GAZE*, c'est-à-dire l'orientation du visage. Dans ce second modèle, les deux variables supplémentaires, Pan et $Pan2$, représentent deux mesures de l'orientation panoramique du visage. Pan correspond ainsi à la variable τ définie dans la sous-section II.4.1 du chapitre précédent. La seconde mesure, $Pan2$, est le résultat d'une heuristique exploitant la correspondance des modèles utilisées dans les deux modules de suivi (*GEST* et *GAZE*) afin d'en extraire une orientation horizontale grossière du visage. Ces modèles ont été prototypés empiriquement.

b) Discrétisation

Indépendamment de la représentation (HMM ou DBN), rappelons que nous utilisons des espaces d'observations discrets. Étant donné que nos données en entrée sont pour leur part continues, il nous reste à choisir un algorithme chargé d'effectuer une discrétisation la plus

efficace possible, c'est-à-dire un classifieur. En effet, déterminer simplement les classes *via* des bornes réparties uniformément impliquerait soit un nombre de classes très élevé et peu représentatives, soit un nombre de classes plus raisonnable, mais peu discriminantes et mal réparties.

L'un des algorithmes les plus simples est celui des k-moyennes (en anglais "k-means") [Kanungo et al., 2002]. Étant donné les valeurs minimales, maximales et un nombre de classes (en anglais "clusters") fixés *a priori*, cet algorithme de classification supervisée permet de définir les bornes entre ces classes. Malheureusement, le processus à modéliser est souvent complexe et on ne connaît pas précisément le nombre de classes. On a alors seulement un ordre de grandeur de celui-ci, à moins de se fixer une limite afin de permettre un apprentissage du modèle et une reconnaissance en temps raisonnable. Dans ce cas, il faut recourir à des algorithmes de classification non-supervisées.

Nous avons exploité ici les cartes de Kohonen [Kohonen, 1984] dont l'utilisation dans un contexte proche a été proposé par G. Infantes durant sa thèse. Cet algorithme permet une discrétisation efficace en déterminant la taille de l'espace des observations et leur géométrie, c'est-à-dire respectivement la taille de chacune de ces classes et le nombre de classes par variable observable, à travers une carte auto-organisée. Nous utilisons pour notre part une grille bidimensionnelle, ce qui est l'utilisation courante et la plus adaptée dans notre cas étant donné notre nombre de classes limité et la taille de nos corpus (à noter que la dimensionnalité de la grille n'a rien à voir avec celle des vecteurs à classer). Dans ce cas, l'algorithme consiste, dans un premier temps, à remplir cette grille de taille prédéterminée ($s \times s$) avec des vecteurs (ou valeurs dans le cas d'une observation unidimensionnelle) aléatoires, chaque case correspondant au final à un vecteur. En considérant que deux vecteurs sont voisins s'ils sont dans deux cases contigües, la phase d'auto-organisation de l'algorithme se décompose comme suit :

1 . pour chaque vecteur v_i à classifier, on recherche la case c_j correspondant au vecteur le plus proche (par exemple selon une distance euclidienne) de v_i,

2 . ce vecteur se rapproche alors de v_i selon la formule $c_j \leftarrow c_j + \eta v_i$, avec $\eta < 1$,

3 . chaque vecteur c_k du voisinage se rapproche également (mais dans une moindre mesure) de v_i : $c_k \leftarrow c_k + \eta' v_i$, avec $\eta' < \eta < 1$.

Ainsi, chaque vecteur à classifier réorganise la carte, puis l'ensemble du processus est réitéré (le nombre de ces itérations est fonction du nombre de vecteurs v_i) avant la stabilisation des valeurs des cases : la carte est alors dite organisée. Les vecteurs de cases proches sont alors proches entre eux, et on obtient donc une relative continuité lorsque l'on suit un voisinage.

Une fois la carte organisée, on parcourt une dernière fois les vecteurs v_i à classifier afin que chacun d'eux « vote » pour la case qui contient le vecteur le plus proche. On peut alors construire le paysage correspondant aux scores obtenus pour chaque case en se servant de ce score comme altitude. Les centres des classes seront alors les sommets de ce paysage (certains sommets étant éliminés s'ils sont trop proches d'un autre plus haut qu'eux-mêmes) et chaque classe est ainsi définie par le vecteur sommet qui est appelé vecteur caractérisant de la classe. Ainsi, lors d'une reconnaissance, un nouveau vecteur sera classifié *via* cette carte grâce à un simple calcul de distance aux différents sommets sélectionnés : il fera alors partie de la classe dont le vecteur caractérisant est le plus proche. Il est à noter que la taille de la grille est un

facteur déterminant de l'obtention de bon résultats. En effet, si certains scores sont nuls, le paysage construit perd tout son sens. D'après [Kohonen, 1984], la taille de la grille doit être au moins 500 fois plus petite que $card(v_i)$. Dans notre application, nous aboutissons généralement à des espaces de dimension 4 à 8. Le tableau III.1 énumère les paramètres de notre système de reconnaissance de gestes par DBN dont les valeurs associées ont été estimées empiriquement pour les unes et obtenus après optimisation (de plus amples explications sur ce processus seront données en sous-section III.4.4) pour les autres.

Symbole	Signification	Valeur
N	nombre de particules du filtre	675
-	taille de la grille de Kohonen	5×5
$(\eta; \eta')$	taux d'adaptation de Kohonen	$(0{,}146; 0{,}105)$
DM	nombre de gestes identiques successifs nécessaires à la validation d'une reconnaissance	6
$(\omega_{Nmin}; \omega_{Tmin})$	poids normalisés et cumulés minimum pour valider une reconnaissance	$(0{,}8; 0{,}01)$

TAB. III.1: Valeurs des paramètres principaux de la reconnaissance de gestes par DBN.

III.3.2 Segmentation automatique des gestes

a) Motivations

Nous souhaitons mettre en place un système de reconnaissance de gestes réaliste sur notre plateforme robotique. Le formalisme DBN, ses avantages énumérés précédemment, ainsi que la possibilité donnée par le filtrage particulaire de mettre fin à une reconnaissance avant même la fin de la séquence à reconnaître, nous permet de nous approcher de cet objectif. Mais, si nos premiers tests ont été effectués sur des séquences enregistrées sur notre plateforme, celles-ci ont ensuite été étiquetées à la main. Cet étiquetage nous permet de bénéficier d'une vérité de terrain pour nos évaluations, mais nous fournit également la segmentation de nos séquences pour la reconnaissance. De même, lors de la plupart de nos démonstrations, la segmentation des gestes reposait sur une heuristique : une nouvelle reconnaissance était lancée (et par conséquent le début du geste était déterminé) par la détection d'une activité vocale de l'utilisateur et sa fin était décidée par une temporisation. Charge était alors à l'utilisateur d'être en position de repos au début de son énoncé et d'avoir terminé l'exécution de son geste avant la fin de la temporisation.

Un tel procédé impose donc une contrainte forte d'occurrence conjointe de geste et de parole, ce qui empêche un comportement réellement naturel de l'utilisateur. C'est ce dernier point qui est ici le plus dérangeant puisque l'ensemble de cette thèse tente de rendre le système le plus

accessible possible à un utilisateur quelconque. Or, il n'est pas évident que le geste démarre en même temps (ou même après) la parole qui l'accompagne [Wu et al., 1999]. C'est pourquoi, nous pensons qu'un système de reconnaissance de gestes réaliste dans notre contexte doit être capable de segmenter automatiquement ces derniers.

b) Notre approche

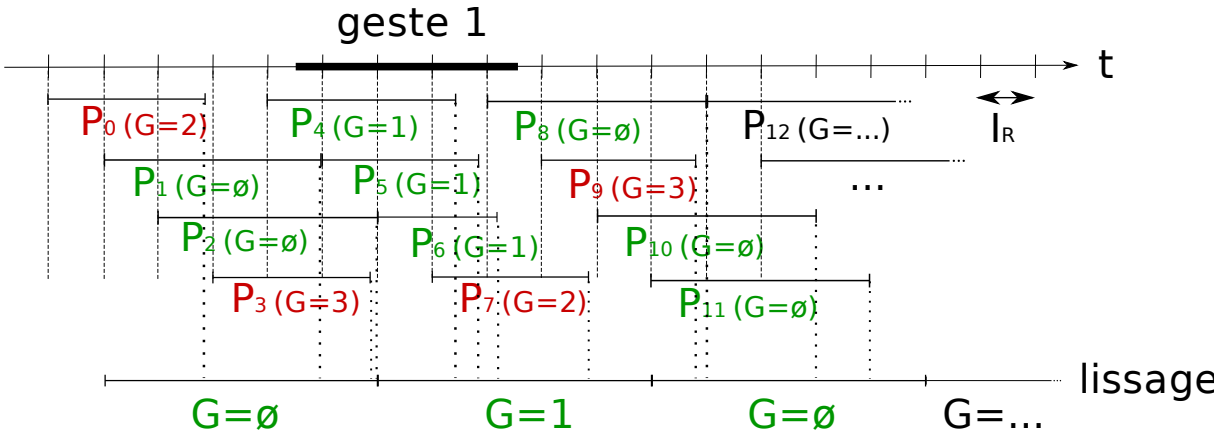


FIG. III.7: Procédure pour la segmentation automatique de gestes.

Dans notre cadre, et étant donné notre formalisme, la méthode la plus simple pour une segmentation automatique serait de segmenter les gestes en se basant uniquement sur la position de repos. Mais cela rendrait les gestes extrêmement statiques. Nous avons donc développé une technique simple, basée sur les avantages combinés de l'utilisation du filtrage particulaire pour la reconnaissance par DBN. Cette méthode a pour but de calculer

$$[G_{k-n:k}]_{MAP} = \arg\max_{g_i} card_{k-n<t<k}[P_t(G = g_i)],$$

c'est-à-dire le geste le mieux reconnu sur une fenêtre temporelle de taille n allant jusqu'à l'instant k, avec $[g_i]$ l'ensemble des gestes modélisées et $P_t(G = g_i)$ le processus de reconnaissance aboutissant à l'instant t tel que le geste reconnu soit g_i.

Pour ce faire, il s'agit de lancer, dans un premier temps, de nouveaux processus de reconnaissance P_i à différents instants image et de les laisser évoluer puis s'arrêter en parallèle. Nous pouvons alors, dans un second temps, lisser et filtrer ces résultats afin d'obtenir des résultats de reconnaissance plus significatifs : lors d'un geste effectué, la plupart des processus donneront une réponse identique qui sera certainement le geste en question. Lorsqu'aucun geste n'est effectué, cette seconde passe doit permettre d'éliminer un maximum de faux positifs. Pour cela nous disposons de la capacité de la reconnaissance de geste elle-même à détecter les faux positifs (*via* une modélisation des « non-gestes » ou *via* nos différentes heuristiques et seuils), mais nous pouvons également nous servir du lissage de la seconde passe pour ne sortir aucun résultat tant que les reconnaissances ne sont pas suffisamment cohérentes.

La figure III.7 schématise ce procédé : le trait épais sur l'axe des temps symbolise un geste effectué (ici nommé « 1 »). Les P_i sont les processus de reconnaissance lancés à intervalles

réguliers et dont le résultat est donné entre parenthèses. Ces derniers sont verts quand ils sont corrects et rouges quand il s'agit d'un faux positif. Enfin la procédure de lissage et ses résultats sont illustrés sur le bas du schéma.

III.4 Mise en œuvre et expérimentations

III.4.1 Recueil de données

Afin de pouvoir évaluer notre système, mais aussi d'apprendre un modèle apte à être utilisé pour effectuer des reconnaissances en direct, il nous faut acquérir des données et constituer un corpus. Et pour que ces dernières soient utilisables, nous avons dû définir un protocole pour leurs constructions. La première contrainte est que tous les gestes s'effectuent debout devant le robot de manière à ce qu'aucun geste ne fasse sortir les mains ou la tête du champ de vue des caméras du robot. La seconde est que nous supposons que tous les gestes commencent et s'arrêtent dans la même position, c'est-à-dire dans la position naturelle de repos de l'homme debout : les bras le long du corps.

Ce premier corpus a été construit *via* les acquisitions d'un système commercial de capture de mouvement (MOCAP pour "MOtion CAPture") afin d'effectuer quelques évaluations préliminaires non détaillées ici. Des exemples de ce type d'acquisition sont visibles en annexe G. Mis à part ce premier corpus, toutes les acquisitions de données ont été et sont effectuées sur le robot en enregistrant les sorties des modules de suivi de gestes et de suivi de regard embarqués. Si l'on utilise le suivi de l'orientation du visage, les résultats de ce dernier sont également recueillis dans le même temps. Ces enregistrements sont alors étiquetés (c'est-à-dire segmentés et labellisés) manuellement afin de fournir une vérité de terrain pour l'apprentissage des modèles et les évaluations hors-ligne. Notre système de reconnaissance extrait alors de ces vecteurs d'état ses propres vecteurs d'observation selon la modélisation décrite dans la sous-section III.3.1.

III.4.2 Méthode d'évaluation

Ce système de reconnaissance de geste a été implémenté sur le robot Jido sous la forme d'un module Genom nommé ***DREC***. Bien que ce module ait été rendu le plus générique possible dans le but de pouvoir utiliser ces méthodes pour n'importe quelle modélisation et reconnaissance à base de DBN, nous ne nous intéressons ici qu'à la partie appliquée à la reconnaissance de gestes. Dans ce cadre, le module est intégré et utilisable sur les robots décrits dans le chapitre IV, mais il peut en réalité être fonctionnel sur n'importe quel robot capable de fournir les vecteurs de

données nécessaires, c'est-à-dire sur lesquels le module *GEST* (et éventuellement *GAZE*) est fonctionnel.

Afin d'obtenir les résultats les plus significatifs possibles et étant donné la taille de nos corpus qui n'est pas forcément représentative, tout nos résultats présentés dans cette section ont été obtenus suivant une méthode de validation croisée en trois partie (en anglais, "3-fold cross-validation"). Cette méthode consiste à diviser notre corpus de gestes en trois sous-corpus : deux d'entre elles sont alors utilisées pour l'apprentissage, tandis que la dernière est utilisée comme ensemble de validation. Ce processus est répété trois fois de façon à ce que chacune des trois sous-corpus ait été utilisé une et une seule fois comme corpus de validation. Enfin, les résultats obtenus à chaque validation sont moyennés afin d'obtenir une évaluation globale.

III.4.3 Expérimentations préliminaires

Les premiers résultats obtenus par notre module de reconnaissance de geste en utilisant une modélisation par HMMs sont exposés dans le tableau III.2.

Type	**Geste à reconnaître**		**Geste reconnu**								
			1	2	3	4	5	6	7	8	sensibilité
(S)	« stop »	(1)	67	0	20	0	7	0	0	7	67
(S)	« viens vers moi (une main) »	(2)	0	93	0	0	0	7	0	0	93
(S)	« viens vers moi (deux mains) »	(3)	0	0	100	0	0	0	0	0	100
(D)	« pointage bas droite »	(4)	0	7	0	73	0	0	7	7	79
(D)	« pointage bas gauche »	(5)	0	0	0	0	100	0	0	0	100
(D)	« pointage devant »	(6)	0	0	0	0	0	93	0	0	100
(D)	« pointage haut droite »	(7)	0	0	0	0	0	0	100	0	100
(D)	« pointage haut gauche »	(8)	0	7	0	0	0	0	0	93	93
sélectivité		100	88	83	100	94	93	94	88		

TAB. III.2: Matrice de confusion de tests préliminaires obtenus avec des HMMs (en %). La première colonne indique le type de geste (symbolique ou déictique).

Nous rappelons que la sensibilité mesure la proportion de gestes correctement identifiés, tandis que la sélectivité mesure la proportion de gestes correctement rejetés, c'est-à-dire qu'ils mesurent respectivement la capacité du système à détecter les vrais positifs (il s'agit donc du taux de reconnaissance) et à rejeter les faux positifs. Les résultats exposés ici portent sur un corpus de 8 gestes représentants 118 séquences d'observations acquises à partir du système commercial de capture de mouvement décrit précédemment. Le lecteur trouvera le prototype des huit gestes en annexe G. Il est à noter que seul un vecteur acquis sur 10 a été pris en compte ici. En effet, l'acquisition se déroulant à 100 Hz, les modèles construits par ce biais n'auraient pas été utilisables pour la reconnaissance de données acquises par le module *GEST*, puisque celui-ci ne fonctionne qu'à 10 Hz et la taille des séquences d'observations à traiter auraient donc été totalement différentes. Ces premiers résultats sont excellents : 91% de reconnaissance pour une sélectivité de 92% ; mais ne doivent pas faire oublier que :

– ils ont été acquis de manière « parfaite » (aucune perte de cible, précision au millimètre),
– le corpus est petit (15 mouvements effectués par modèle de geste) et ne contient les mouvements que d'une seule personne.

Néanmoins, ces résultats encourageants nous ont poussé à créer un corpus acquis *via* le module *GEST* dont la taille et la composition ont rapidement rendu ce premier corpus caduque. C'est ce que nous allons voir dans la sous-section suivante.

III.4.4 HMM *versus* DBN : étude comparative

Un nouveau corpus, entièrement acquis *via* le module *GEST* sur nos robots, a été construite dans le but de disposer d'un corpus à la fois plus volumineuse, mais également plus représentative (*via* sa diversité). Ce corpus se compose de 12 gestes effectués de 5 à 10 fois par 11 personnes, ce qui représente 772 séquences d'observations. Dans nos expérimentations, nous nous sommes particulièrement intéressés à la comparaison du taux de reconnaissance des HMMs par rapport à celui des DBNs, mais aussi à la consommation en temps CPU de ces deux modélisations, le but étant de démontrer la supériorité d'une représentation par DBN.

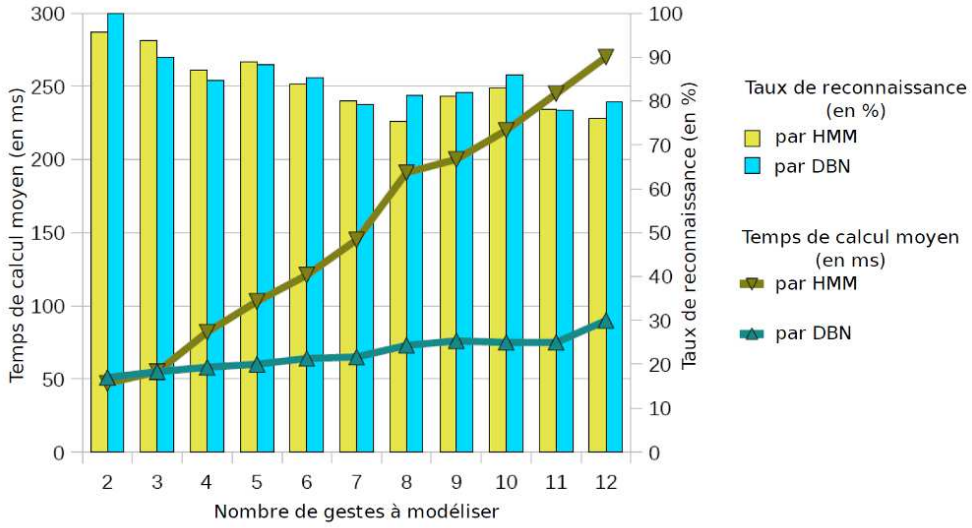


FIG. III.8: Comparaison en terme de taux de classification et de temps de calcul selon le type de modélisation (HMM ou DBN).

La figure III.8 montre les résultats obtenus sur ce corpus suivant le nombre de gestes pris en considération (c'est-à-dire le nombre de gestes modélisés et à reconnaître) et la modélisation utilisée (HMMs en jaune ou DBN en bleu). Ces évaluations ont été effectuées sur un PC équipé d'un Pentium 3.2 GHz. Les barres verticales représentent le taux de reconnaissance obtenu pour un nombre de gestes à reconnaitre variant de deux à douze (axe horizontal), tandis que les triangles représentent le temps de calcul moyen consommé pour le traitement d'une séquence d'observation. On constate logiquement une diminution du taux de reconnaissance avec

l'augmentation du nombre de gestes considérés, puisque le nombre de gestes se ressemblants augmente. Mais celui-ci est maintenu à un niveau satisfaisant, à savoir 79,7% du corpus complet a été correctement identifié en utilisant un unique DBN, tandis que la banque de HMMs arrive à un taux de 76%. Étant donné les aléas de la reconnaissance et l'allure générale de notre graphe, on peut considérer que les résultats en terme de taux de reconnaissance sont du même ordre pour les deux type de modélisation. Par contre, et ce qui est tout particulièrement intéressant dans notre contexte robotique, le DBN utilise une charge CPU bien moindre (inférieur d'un facteur 3 pour 12 gestes) que les HMMs. Cela est dû au fait que notre l'algorithme de reconnaissance doit tester l'ensemble des HMMs à chaque séquence, ce qui constitue un coup incompressible qui augmente linéairement avec le nombre de gestes à modéliser. Au contraire, l'unique DBN ne doit son augmentation de temps de calcul qu'à l'augmentation de l'arité du nœud G, ce qui représente un coup bien moindre.

De plus, il est intéressant de noter qu'il est possible de faire encore décroître de manière importante le temps de calcul des DBNs sans dégrader trop fortement le taux de reconnaissance. Par exemple, nous avons obtenu un temps de calcul moyen de 24 ms par séquence pour un taux de reconnaissance de 72,7%. Il est également à noter que lors de ces évaluations, la possibilité de terminer une reconnaissance par DBN avant la fin de la séquence en cours d'analyse n'était pas encore utilisée, mais sera abordée dans la sous-section suivante. Enfin, notons que ces résultats ont été obtenus après optimisation des différents paramètres libres de notre système (taille du réseau de Kohonen, taux d'adaptation, nombre de particule du filtre particulaire, etc) *via* le tracé de courbes ROC (voir annexe C). Le lecteur remarquera que la sélectivité des gestes bi-manuels est supérieure à celle des gestes mono-manuels.

Geste à reconnaître		**Geste reconnu**												
		1	2	3	4	5	6	7	8	9	10	11	12	sensibilité
« stop »	(1)	66,1	19,3	1,6	1,6	0	0	0	0	0	9,7	1,6	0	66,1
« viens vers moi (une main) »	(2)	12,8	62,8	0	1,3	0	1,3	4,8	1,3	8,0	1,3	6,4	0	62,8
« viens vers moi (deux mains) »	(3)	2,8	0	80,6	2,8	1,4	4,2	0	2,8	1,4	0	0	4,2	80,6
« pointage bas droite »	(4)	0	0	0	90,0	1,4	0	0	0	0	4,1	1,4	2,8	90,0
« pointage haut droite »	(5)	0	0	1,7	1,7	78,9	0	0	0	1,7	0	12,3	3,5	78,9
« pointage bas gauche »	(6)	0	0	0	0	0	86,2	7,0	3,5	1,7	1,7	0	0	86,2
« pointage haut gauche »	(7)	0	0	0	0	0	0	100	0	0	0	0	0	100
« pointage devant »	(8)	1,7	7,0	1,7	5,3	0	10,5	3,5	63,2	0	3,5	0	3,5	63,2
« je me présente »	(9)	0	6,8	2,7	1,4	0	4,1	0	1,4	83,6	0	0	0	83,6
« va-t-en »	(10)	4,2	0	0	6,9	1,4	0	0	0	2,8	83,3	1,4	0	83,3
« ohé (une main) »	(11)	1,6	3,1	0	0	3,1	0	1,6	0	3,1	0	87,5	0	87,5
« ohé (deux mains) »	(12)	0	0	3,6	5,3	8,9	0	1,8	0	0	0	1,8	78,6	78,6
sélectivité		70,7	68,1	89,2	78,8	81,8	78,7	83,1	83,7	80,3	82,2	78,9	83,0	

TAB. III.3: Matrice de confusion obtenue par notre DBN (en %).

Le tableau III.3 détaille les résultats de notre système de reconnaissance de gestes par DBN sur la totalité du corpus et des gestes (nous nommerons ce corpus $CORP_{12}$). Sur la diagonale nous pouvons observer le taux de gestes correctement reconnus pour chacun d'eux. La plus grande partie des erreurs surviennent de manière cohérente entre les gestes ayant une forte similarité. Ce type de mauvaise classification augmente avec la variabilité des gestes dans le corpus qui découle du nombre de personnes ayant effectués le geste en question.

Au vu de cette étude comparative, force est de constater que le temps de calcul nécessaire à la reconnaissance par HMM est non seulement assez élevé, mais grandit avec le nombre de

gestes modélisés, ce qui est incompatible avec une utilisation de ce type de reconnaissance qui se veut intensive et la plus large possible. Prenant cette constatation en compte, en plus des nombreux avantages de la modélisation par DBN dans notre contexte (voir sous-section III.2.3), nous nous sommes focalisés sur l'utilisation des DBNs. C'est pourquoi dans la suite de cette section nous ne parlerons plus de modélisation par HMM et les résultats exposés ne concerneront que des reconnaissances par DBN.

III.4.5 Vers une segmentation automatique des gestes

a) Les non-gestes

Les résultats décrits précédemment ont été calculés sur la base d'un corpus « parfait », c'est à dire que ces corpus ne contiennent que des gestes parfaitement exécutés, suivis et segmentés. Afin d'évaluer notre système de reconnaissance de gestes de manière plus réaliste, nous définissons ici la notion de « non-gestes ». Cette nouvelle catégorie peut contenir tout mouvement qui n'est pas un geste, typiquement une séquence où l'utilisateur marche devant le robot, ou mouvements de l'utilisateur lors d'une discussion avec une tierce personne. À des fins de tests, nous étiquetons donc 29 séquences supplémentaires comme non-gestes. En nous inspirant des modèles de silence utilisés en reconnaissance de parole, nous apprenons un modèle de non-geste comme s'il s'agissait d'un geste supplémentaire. Ce dernier est donc chargé de reconnaître ces fausses détections simulées.

Geste à reconnaître		**Geste reconnu**													
		1	2	3	4	5	6	7	8	9	10	11	12	rien	sensibilité
« stop »	(1)	61,3	27,4	3,2	0,0	0,0	0,0	0,0	0,0	0,0	4,8	3,2	0,0	0,0	61,3
« viens vers moi (une main) »	(2)	17,9	61,5	2,6	1,3	1,3	0,0	5,1	1,3	2,6	1,3	5,1	0,0	0,1	61,5
« viens vers moi (deux mains) »	(3)	0,0	2,8	84,7	2,8	2,8	1,4	0,0	1,4	0,0	0,0	1,4	2,8	0,0	84,7
« pointage bas droite »	(4)	0,0	1,4	2,9	72,9	7,1	1,4	0,0	0,0	0,0	7,1	0,0	5,7	1,4	72,9
« pointage haut droite »	(5)	0,0	0,0	1,8	7,0	49,1	1,8	0,0	0,0	0,0	5,3	28,1	1,8	5,3	49,1
« pointage bas gauche »	(6)	1,8	0,0	3,5	0,0	1,8	87,7	0,0	0,0	1,8	0,0	0,0	0,0	3,5	87,7
« pointage haut gauche »	(7)	0,0	0,0	0,0	0,0	0,0	0,0	94,4	0,0	3,7	0,0	0,0	0,0	1,9	94,4
« pointage devant »	(8)	1,8	3,5	0,0	3,5	0,0	14,0	0,0	57,9	3,5	3,5	0,0	5,3	7,0	57,9
« je me présente »	(9)	0,0	6,8	0,0	0,0	2,7	2,7	1,4	0,0	80,8	5,5	0,0	0,0	0,0	80,8
« va-t-en »	(10)	1,4	4,2	0,0	2,8	9,7	0,0	0,0	0,0	5,6	72,2	2,8	0,0	1,4	72,2
« ohé (une main) »	(11)	3,1	3,1	0,0	1,6	15,6	0,0	1,6	0,0	0,0	3,1	70,3	0,0	1,6	70,3
« ohé (deux mains) »	(12)	0,0	0,0	0,0	3,6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	96,4	0,0	96,4
« non-geste »	rien	6,9	3,4	3,4	3,4	3,4	10,3	0,0	3,4	0,0	0,0	0,0	3,4	62,1	62,1
sélectivité		64,4	59,3	85,9	77,3	49,1	75,8	89,5	91,7	84,3	72,2	64,3	83,1	58,1	

TAB. III.4: Matrice de confusion obtenue par notre DBN en prenant en compte les non-gestes (en %).

Le tableau III.4 montre les résultats obtenus par le même système de reconnaissance que celui utilisé dans la section précédente incrémenté d'un modèle de non-gestes. Ces résultats ont été calculés par validation croisée sur le corpus $CORP_{12\_NG}$, c'est à dire sur le corpus $CORP_{12}$ augmenté des 29 séquences de non-gestes. Le taux de reconnaissance est logiquement diminué,

passant à 73,4% de reconnaissance pour une sélectivité de 73,9%, puisque la diversité du corpus a augmenté fortement. La reconnaissance des non-gestes est plus laborieuse avec 62,1% de reconnaissance. Ceci s'explique en grande partie par la très grande diversité que doit modéliser ce modèle de non-gestes.

b) Segmentation automatique de la fin des séquences

Les résultats exposés précédemment sont en réalité biaisés, l'algorithme permettant de déduire automatiquement la fin d'une reconnaissance n'étant pas utilisé. Le tableau III.5 montre les résultats obtenus sur les mêmes données et dans les mêmes conditions que précédemment en utilisant la déduction automatique de la fin d'une séquence. Afin de rendre cette évaluation réaliste pour notre cadre robotique, si un processus de reconnaissance dépasse la fin d'un geste tel qu'étiqueté sans avoir convergé vers une solution, on considère que rien n'a été reconnu (ce cas correspond à la colonne « rien » du tableau).

Geste à reconnaître		**Geste reconnu**													
		1	2	3	4	5	6	7	8	9	10	11	12	rien	sensibilité
« stop »	(1)	57,1	15,9	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,6	1,6	23,9	57,1
« viens vers moi (une main) »	(2)	7,7	61,5	0,0	0,0	0,0	0,0	1,3	1,3	1,3	0,0	2,6	0,0	24,4	61,5
« viens vers moi (deux mains) »	(3)	0,0	0,0	80,6	0,0	0,0	2,8	0,0	0,0	0,0	0,0	0,0	4,2	12,5	80,6
« pointage bas droite »	(4)	0,0	0,0	0,0	80,0	0,0	0,0	0,0	0,0	0,0	5,7	0,0	2,9	11,4	80
« pointage haut droite »	(5)	0,0	0,0	0,0	1,8	52,6	0,0	0,0	0,0	0,0	5,3	22,8	3,5	14,1	52,6
« pointage bas gauche »	(6)	0,0	0,0	7,0	0,0	0,0	82,5	3,5	1,8	0,0	0,0	0,0	0,0	5,2	82,5
« pointage haut gauche »	(7)	0,0	0,0	1,9	0,0	0,0	0,0	94,4	0,0	0,0	0,0	0,0	0,0	3,7	94,4
« pointage devant »	(8)	1,8	12,3	1,8	1,8	0,0	14,0	1,8	42,1	0,0	0,0	0,0	5,3	19,3	42,1
« je me présente »	(9)	0,0	6,8	0,0	1,4	0,0	4,1	2,7	0,0	67,1	1,4	0,0	0,0	16,5	67,1
« va-t-en »	(10)	0,0	2,7	0,0	2,7	5,5	2,7	5,5	4,1	5,5	54,8	1,4	0,0	15,1	54,8
« ohé (une main) »	(11)	10,9	3,1	0,0	1,6	7,8	0,0	1,6	1,6	0,0	6,2	65,6	1,6	0,0	65,6
« ohé (deux mains) »	(12)	0,0	1,8	7,1	8,9	1,8	0,0	0,0	0,0	0,0	0,0	0,0	80,4	0,0	80,4
« non-geste »	rien	0,0	0,0	0,0	6,9	0,0	0,0	0,0	3,4	0,0	0,0	0,0	0,0	89,7	89,7
sélectivité		72,0	64,0	85,3	81,2	75,0	75,8	82,3	77,4	90,7	76,9	71,2	78,9		

TAB. III.5: Matrice de confusion obtenue avec segmentation automatique de la fin des séquences (en %).

Comme nous pouvons l'observer sur ce tableau, le taux de reconnaissance est encore diminué, passant à 68,8% de reconnaissance. Cette diminution est logique étant donné la possibilité pour le système de ne déduire aucun geste d'une séquence. À l'inverse, la sélectivité augmente pour atteindre 77,6%, de même que le taux de gestes correctement rejetés (c'est-à-dire des non-gestes reconnus comme tels ou non reconnus) qui atteint près de 90%. Cette amélioration s'explique par la plus grande possibilité de sélectivité qu'offre le fait de déduire automatiquement la fin d'un geste. En effet, les gestes pour lesquels la reconnaissance est hésitante sont ici classés comme inconnus diminuant donc le risque de mal les classer. Il est à noter que l'un des inconvénients de la méthode utilisée pour classifier les gestes avant la fin de la séquence est que certains gestes sont quasi identiques sur une grande partie de leur exécution. Ainsi, étant donné que notre modèle ne tient pas compte de la forme de la main, les gestes « stop » et « viens vers moi (à une seule main) » sont quasiment identiques à la seule différence près que le second donne souvent lieu à des répétitions, entrainant une confusion non négligeable pour le système.

c) Segmentation automatique complète

La stratégie utilisée ici a été décrite dans la section III.3.2. Dans la pratique, et afin de permettre l'obtention de statistiques hors-ligne aussi proches que possible de la réalité, notre corpus est parcouru en totalité, c'est-à-dire que les fichiers composants ce dernier sont considérés comme une suite d'observations continue. En effet, dans les évaluations précédentes, seules étaient prises en compte les observations faisant partie d'une séquence étiquetée. Par conséquent, il convient de quantifier également notre corpus en taille : celui-ci contient 42680 observations correspondant à environ 83 minutes d'enregistrement et contenant 774 occurrences de gestes.

Geste à reconnaître		**Geste reconnu**												
		1	2	3	4	5	6	7	8	9	10	11	12	rien	sensibilité
« stop »	(1)	47,6	3,2	0,0	0,0	0,0	0,0	11,1	4,8	0,0	6,3	1,6	1,6	23,8	47,6
« viens vers moi (une main) »	(2)	9,0	41,0	0,0	0,0	0,0	0,0	11,5	0,0	0,0	3,8	2,6	0,0	32,1	41
« viens vers moi (deux mains) »	(3)	0,0	0,0	83,3	0,0	0,0	1,4	0,0	0,0	0,0	0,0	0,0	5,6	9,8	83,3
« pointage bas droite »	(4)	1,4	1,4	0,0	78,6	0,0	0,0	0,0	0,0	0,0	1,4	0,0	2,9	14,3	78,6
« pointage haut droite »	(5)	0,0	0,0	0,0	0,0	49,1	0,0	0,0	0,0	0,0	3,5	31,6	1,8	14,1	49,1
« pointage bas gauche »	(6)	0,0	0,0	5,3	1,8	0,0	57,9	1,8	0,0	1,8	0,0	0,0	0,0	31,6	57,9
« pointage haut gauche »	(7)	0,0	0,0	1,9	0,0	0,0	3,7	61,1	1,9	3,7	3,7	1,9	0,0	22,2	61,1
« pointage devant »	(8)	0,0	0,0	1,8	1,8	0,0	21,0	0,0	64,9	0,0	0,0	0,0	3,5	7,0	64,9
« je me présente »	(9)	1,4	4,1	0,0	0,0	0,0	4,1	4,1	0,0	57,5	12,3	0,0	0,0	16,5	57,5
« va-t-en »	(10)	5,5	1,4	0,0	1,4	2,7	1,4	2,7	2,7	2,7	64,4	0,0	0,0	15,0	64,4
« ohé (une main) »	(11)	10,5	0,0	0,0	0,0	2,5	0,0	2,1	0,0	0,0	0,0	81,2	3,1	1,7	81,2
« ohé (deux mains) »	(12)	0,0	0,0	1,8	7,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	87,3	3,6	87,3
« non-geste »	rien	6,6	3,5	6,6	7,5	4,8	4,2	4,1	3,3	1,3	4,0	4,4	5,3	44,1	44,1
sélectivité		58,8	82,1	89,6	82,1	84,8	63,5	57,9	82,2	87,5	67,1	70,3	82,4		

TAB. III.6: Matrice de confusion obtenue avec segmentation automatique complète (en %).

Le tableau III.6 montre les résultats obtenus par notre système de reconnaissance de gestes avec segmentation automatique. La ligne « non-geste » ne correspond plus ici au taux de rejet des séquences étiquetées comme telle, mais à l'ensemble des fausses alarmes. Les pourcentages représentent alors le nombre de séquences classifiées sur le nombre total de ces fausses détections. Une partie d'entre elles sont éliminées par l'algorithme lui-même (*via* les seuils d'acceptation), mais aussi *via* le modèle de non-gestes décrit précédemment.

Si le taux de reconnaissance a logiquement diminué, étant donné la plus grande complexité, il reste à un niveau satisfaisant (65,2%) de même que la sélectivité (76,6%). Mais le point noir du système, qui n'apparait malheureusement pas clairement dans ce tableau concerne les faux positifs. En effet, le nombre de ces fausses détections est important avec près de 500 cas, dont seul 44,1% sont détectées, ce qui correspond en réalité à un taux de faux positifs de près de 38%. Ceci est réellement problématique, mais doit être nuancé.

En effet, ce tableau n'est qu'un exemple des résultats qu'il est possible d'obtenir avec ce système et nous le qualifierons de résultat médian. Les résultats dépendent en réalité fortement des paramètres utilisés : plus le lissage est fort, plus le nombre de faux positifs diminue, mais le taux de reconnaissance chute d'autant, et inversement. Ainsi, parmi nos tests, il est possible de faire grimper le taux de reconnaissance à plus de 72%, mais c'est au prix d'une augmentation de 50% du nombre de fausses détections ainsi que d'une diminution de la sélectivité (65%). De même, d'autres tests nous ont permis de diminuer de plus de moitié le nombre de fausse

alarmes, mais là encore, c'est au prix d'un taux de reconnaissance décevant (38,2%).

D'autre part, et à titre de comparaison, [Stiefelhagen et al., 2004], grâce à sa reconnaissance basée sur 3 HMMs, arrive à un taux de reconnaissance de l'ordre de 80%, pour 26% de faux positifs. Sachant que les auteurs ne reconnaissent dans ces travaux que des gestes de pointage et que leur cadre d'expérimentation est relativement restreint, nos résultats peuvent en réalité être considérés comme équivalents. Les auteurs précisent par ailleurs que l'ajout de l'orientation de la tête à leurs vecteurs d'observations leur a permis d'abaisser ce taux de faux positif à 13%.

Enfin, il est à noter qu'en terme de temps de calcul, cet algorithme semble parfaitement crédible dans notre cadre robotique. En effet, malgré la superposition des processus de reconnaissance, le temps moyen de calcul n'est que d'une dizaine de ms par observation.

III.4.6 Vers une reconnaissance incluant l'orientation du regard

Toutes les évaluations décrites précédemment ont été effectuées sur des données issues du module *GEST* (suivi de gestes) et modélisées par le modèle $MG1$ décrit par la figure III.6. Dans cette sous-section, nous nous intéressons à l'utilisation de données issues du module *GAZE* (suivi de l'orientation du visage) conjointement aux précédentes à travers le modèle $MG2$ permettant de fusionner ces deux types d'entrées (voir figure III.6). Le but est ici de prouver l'apport de l'orientation du visage pour la reconnaissance de gestes, en particulier déictiques. En effet, et comme nous l'avons vu dans le chapitre II, lors d'un geste de pointage les humains ont une forte tendance à regarder l'endroit désigné. Utiliser l'orientation du visage dans notre modélisation de gestes doit par conséquent permettre de discriminer plus facilement des gestes de pointage vers des directions différentes, mais également de diminuer le risque de confondre un geste de pointage avec un geste symbolique, ce dernier n'entrainant aucun mouvement particulier de la tête.

C'est dans ce but que nous avons construit un nouveau corpus $CORP_{GG}$ composé de cinq gestes (dont quatre sont déictiques). Ce corpus a été acquis sur notre robot HRP-2 (qui sera décrit dans le chapitre suivant) dans le cadre d'une interaction proximale homme-robot. Chaque geste a été répété en moyenne 15 fois.

Geste à reconnaître		**Geste reconnu**						
		1	2	3	4	5	rien	sensibilité
« pointage bas droite »	(1)	58	7	0	21	0	14	58
« pointage devant »	(2)	0	79	7	7	7	0	79
« pointage bas gauche »	(3)	7	7	86	0	0	0	86
« pointage bas très à droite »	(4)	0	6	0	88	0	6	88
« stop »	(5)	7	14	0	7	65	7	65
sélectivité		82	71	92	74	90		

TAB. III.7: Matrice de confusion obtenue sur le corpus $CORP_{GG}$ en utilisant le modèle $MG1$ décrit par la figure III.6 (en %).

Le tableau III.7 montre la matrice de confusion obtenue sur ce corpus grâce à une modélisation par $MG1$, c'est-à-dire sans utiliser l'orientation du visage. Ces résultats, avec une moyenne de 75% de reconnaissance pour 80% de sélectivité, sont conformes à ceux décrits précédemment dans ce chapitre compte tenu de la grande similarité des gestes de ce corpus :

– tous les gestes déictiques pointent le sol (ou une table),
– le geste « pointage bas très à droite » est très proche du geste « pointage bas droite »,
– les gestes « stop » et « pointage devant » sont également assez similaires.

Ce tableau sert de base de comparaison avec le traitement des mêmes données *via* une modélisation par $MG2$. Le tableau III.8 montre les résultats obtenus par ce dernier modèle. L'apport de l'orientation du visage est assez net, avec un taux de reconnaissance atteignant 84% pour une sélectivité de plus de 86%. La matrice de confusion permet également de voir que l'orientation du visage apporte les améliorations attendues.

Geste à reconnaître		**Geste reconnu**						
		1	2	3	4	5	rien	sensibilité
« pointage bas droite »	(1)	86	0	0	14	0	0	86
« pointage devant »	(2)	0	86	0	0	14	0	86
« pointage bas gauche »	(3)	7	0	93	0	0	0	93
« pointage bas très à droite »	(4)	0	0	12	88	0	0	88
« stop »	(5)	0	21	0	0	65	14	65
sélectivité		92	81	87	87	82		

TAB. III.8: Matrice de confusion obtenue sur le corpus $CORP_{GG}$ en utilisant le modèle $MG2$ décrit par la figure III.6 (en %).

Enfin, il est à noter que le coût supplémentaire, en terme de temps de calcul, engendré par l'utilisation du modèle $MG2$, légèrement plus complexe que $MG1$, est négligeable (de l'ordre de la milli-seconde par geste à reconnaître).

III.5 Conclusion et perspectives

Nous avons présenté ici nos travaux sur la reconnaissance de gestes. Dans notre cadre robotique, nous avons développé un module nommé $DREC$ dédié à la modélisation par DBN (ou HMM). La figure III.9 rappelle notre architecture définie en introduction et complétée ici par le module $DREC$ décrit dans ce chapitre.

Les contributions apportées par ce module concernent la reconnaissance de gestes dynamiques dans un formalisme DBN qui reste marginale dans la littérature. Ce type de modélisation a été testé et évalué sur des séquences réelles en provenance de notre module de suivi de gestes. Nous avons également comparé les performances relatives des HMMs et des DBNs, cette comparaison a par ailleurs donné lieu à la publication suivante : [Burger et al., 2009b]. Cette étude a prouvé non seulement la faisabilité d'un tel système basé sur une modélisation

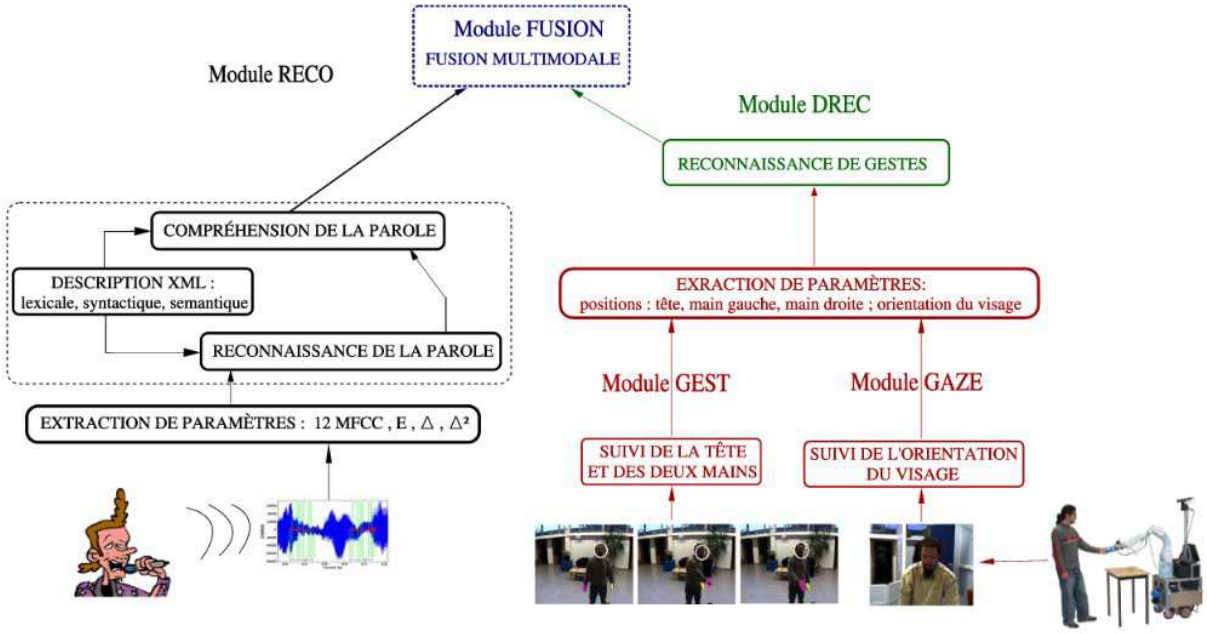


FIG. III.9: Architecture globale de notre interface homme-robot.

par DBN, mais aussi que les nombreux avantages des DBNs peuvent être exploités afin d'économiser les ressources CPU qui sont de fait limitées sur nos plateformes robotiques. Nous avons également pu démontrer la faisabilité d'une segmentation automatique des gestes, bien que ces investigations récentes mériteraient quelques développements supplémentaires afin de limiter davantage les fausses détections. Nous avons montré que ces fausses alarmes peuvent être réduites en considérant l'orientation du regard dans le processus de reconnaissance. Enfin, des évaluations qualitatives et quantitatives sur notre plateforme robotique, plutôt marginales dans la littérature, ont validé ces travaux.

Bien que notre implémentation de la reconnaissance de gestes par DBN ait montré ici son utilité et l'éventail de ses avantages, certaines évolutions récentes mériteraient quelques investigations complémentaires. En particulier, notre modélisation n'utilise pas, pour l'instant, d'autres données que la position des mains par rapport à la tête, alors que nous disposons également de la forme et de l'orientation de ces dernières. Une autre voix d'amélioration directe serait de tester de nouvelles structures de DBN et de trouver une manière plus efficace et rapide pour caractériser leurs performances, ainsi que pour optimiser les très nombreux paramètres libres du système (ou d'en diminuer significativement le nombre). En effet, l'ensemble de cette procédure reste extrêmement lourde et le nombre et les valeurs des paramètres libres influencent grandement les performances.

Chapitre IV

Fusion de données audio-visuelles et démonstrations robotiques

La finalité de nos travaux est de voir un utilisateur interagir le plus naturellement possible avec un robot grâce aux différentes modalités présentées. Nous nous intéressons en particulier à l'utilisation d'expressions gestuelles en confirmation ou en complément d'une expression verbale. Dans ce cadre, les précédents chapitres ont présenté les systèmes permettant de traiter les entrées des deux canaux, audio et vidéo, considérés ici.

Le présent chapitre traite de la fusion des données en provenance de ces deux canaux dans le cadre d'une interface dédiée à l'interaction multimodale homme-robot. Le but de ce chapitre est par conséquent de démontrer l'utilité d'une telle interface multimodale dans le cadre de démonstrations complètes, c'est-à-dire de scénarios exploitant l'ensemble des capacités du robot comme celles de notre interface. Les situations homme-robot, mais aussi les tâches robotiques utilisées dans nos scénarios, impliquent des stratégies de fusion spécifiques. C'est la raison pour laquelle nous avons choisi de les présenter ensemble dans cet unique chapitre.

Ce chapitre débute par un rapide état de l'art de la fusion de données audio-visuelle appliquée à la robotique mobile afin de positionner nos travaux par rapport à la littérature (section IV.1). La section IV.2 présente ensuite nos plateformes expérimentales, ainsi que les scénarios imaginés afin d'évaluer notre interface. La section IV.3 qui présente, successivement et pour chaque scénario, la stratégie de fusion adoptée et les résultats obtenus sur nos plateformes. Enfin, la section IV.4 conclut ce chapitre en rappelant nos contributions et en énonçant quelques perspectives.

IV.1 État de l'art et positionnement de nos travaux

Construire un système multimodal implique de prendre en compte les inter-corrélations des modalités concernées afin de construire une représentation du message global véhiculé par ces dernières. Pour ce faire, la fusion de ces données peut s'effectuer à différents niveaux, du niveau signal au niveau sémantique/symbolique. On parle alors respectivement de fusion précoce et tardive, la seconde impliquant, contrairement à la première de procéder au départ à l'analyse de chaque modalité.

IV.1.1 Fusion audio-visuelle en IHM

Une fusion de données au niveau du signal consiste à extraire des vecteurs de caractéristiques de chaque modalité (audio et vidéo), puis à les concaténer. On parle de fusion de descripteurs. Si la dimension des vecteurs résultants est grande, une réduction dimensionnelle est souvent effectuée. Les observations construites de cette manière peuvent alors être modélisées *via* une méthode de classification classique, par exemple par HMM. Une autre possibilité est de considérer directement les deux modalités sans passer par une phase préalable de concaténation grâce à un modèle conjoint des flux. Les deux vecteurs de données sont alors utilisés tels quels à travers une solution plus spécifique, comme un HMM multi-canal (par exemple le FHMM, pour "factorial HMM", de [Kulic et al., 2007] ou le CHMM, pour "Coupled HMM", de [Oliver et al., 2000]) qui permet de modéliser chaque flux en forçant des points de synchronisation entre des observations indépendantes. Dans les deux cas, une fusion au niveau signal implique de traiter des flots de données synchrones ou très fortement corrélées. C'est par exemple le cas pour des applications de reconnaissance de parole audio-visuelle [Heracleous et al., 2009, Beautemps et al., 2007, Potamianos et al., 2003, Meyer, 2002]. Ce type d'applications a pour but de fusionner des données acoustiques (paroles) avec des données visuelles, typiquement les mouvements des lèvres prononçant les paroles à traiter. On trouvera d'autres exemples liés à l'indexation multimédia dans [Joly, 2007].

Considérant au contraire deux modalités asynchrones, une autre approche consiste à effectuer une fusion au niveau symbolique. Il s'agit alors d'effectuer, pour chaque modalité indépendamment de l'autre, un processus de reconnaissance spécifique, puis de combiner les résultats obtenus, ceux-ci pouvant par exemple prendre la forme de listes des N meilleures reconnaissance accompagnées de leurs vraisemblances respectives. On parle alors de fusion au niveau sémantique [Delgado and Araki, 2005]. Il est à noter que ce type d'approche, si elle considère des données asynchrones, doit tout de même prendre en compte la cohérence temporelle des évènements en entrée.

IV.1.2 Application IHR

Pour les applications robotiques comme la nôtre, le premier type de fusion décrit précédemment n'est pas envisageable. En effet, une communication naturelle peut être verbale ou non, et avec ou sans geste. Ainsi, la parole peut être accompagnée par des gestes complémentaires dans un cas déictique (par exemple, « Pose la bouteille ici. »), mais peut également se suffire à elle-même (par exemple, « Prends la bouteille rouge »). De la même manière, certains gestes symboliques, compte tenu du contexte dans lequel ils sont effectués, sont assez significatifs en eux-même. Ainsi, saluer de la main est parfaitement compréhensible lors de l'initiation ou de la clôture d'une interaction, et dans ces cas, une parole complémentaire sert uniquement à renforcer l'interprétation du geste.

Citons quelques exemple de fusion multimodale en IHR. [Yoshizaki et al., 2002] n'effectuent pas de fusion de données audio-visuelles à proprement parler, mais utilisent la vision uniquement après que leur système de reconnaissance de parole en ait détecté le besoin. Dans la même veine, [Hanafiah et al., 2004] considèrent que parole et gestes sont parfaitement corrélés et, ne disposant pas de reconnaissance de gestes, extraient de l'image les informations supplémentaires nécessaires pour compléter l'occurrence verbale. [Rogalla et al., 2004] vont un peu plus loin en parlant de gestion d'événements, les actions étant alors déterminées par la fusion des événements associés. Enfin, [Stiefelhagen et al., 2004] utilisent la méthode la plus avancée en fusionnant la liste des N meilleures reconnaissances de chaque modalité dans une stratégie hiérarchique. La parole est alors utilisée comme mode principal de communication et permet de définir une fenêtre temporelle dans laquelle doit se trouver le geste complémentaire, ce qui permet de filtrer d'éventuels gestes parasites. Nos investigations s'inspirent de ces derniers travaux.

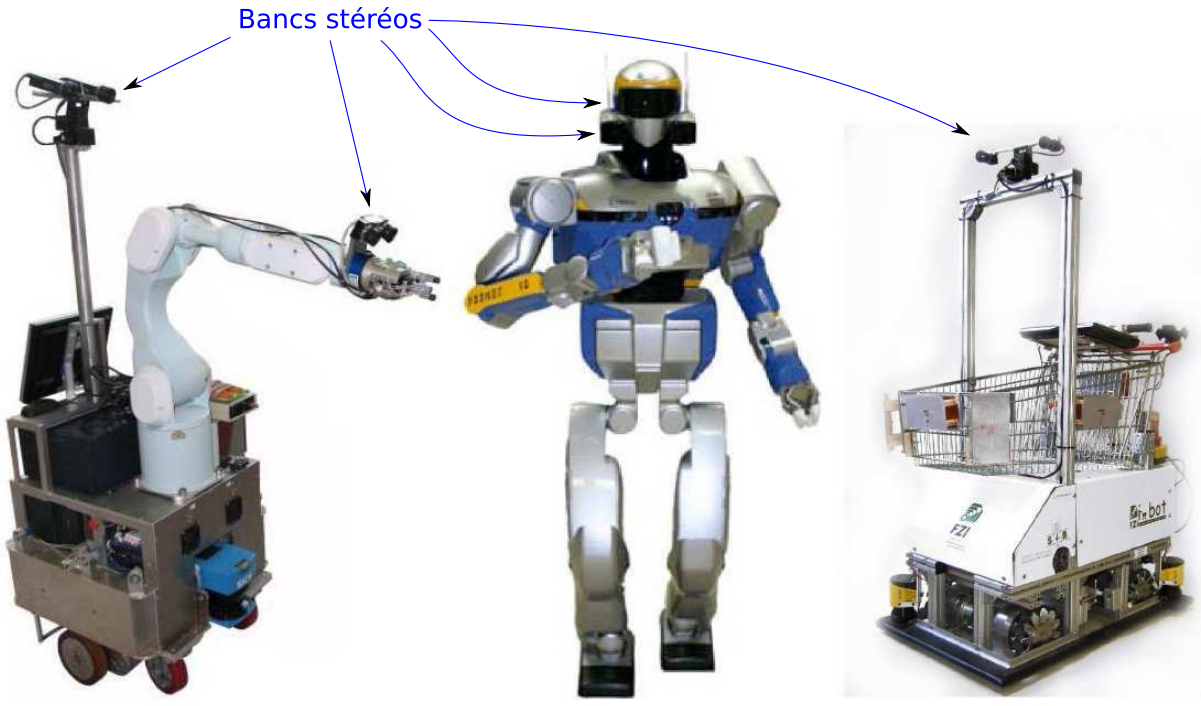
Si un certain nombre de travaux sur la fusion existent en IHR, un gros problème est leur intégration sur une plateforme robotique qui reste marginale dans la littérature IHR. Pour notre part et bien que cela représente un investissement conséquent en terme de temps de travail, nous attachons une grande importance à cette phase d'intégration. nous visons à rendre nos modules assez génériques pour leur permettre d'être utilisés sur différentes plateformes et cherchons à prouver l'intérêt et la validité de nos méthodes à travers ces démonstrations robotiques.

IV.2 Plateformes robotiques et scénarios associés

Étant donné notre cadre applicatif, la validation de nos travaux passe logiquement par leur intégration sur des plateformes robotiques et leur évaluation à travers différents scénarios. L'objet de cette section est la description des robots et des scénarios associées.

IV.2.1 Les robots

Cette sous-section a pour but de présenter rapidement les plateformes robotiques, sur lesquelles nous avons eu l'occasion d'implémenter et d'évaluer les travaux décrits dans les chapitres précédents. Précisons que l'intégration de nos travaux sur plusieurs plateformes se justifie par notre volonté de généricité qui seule prouve que notre approche et nos modules sont robuste dans un large panel de situations. La figure IV.1 montre ainsi des photos des trois robots concernés, le reste de cette sous-section donnant des détails concernant ces plateformes et leurs capacités. Il est à noter que dans les trois cas, l'utilisateur du robot est équipé d'un micro-casque sans fil afin de donner ses ordres vocaux au robot. Bien qu'une interaction naturelle voudrait que le robot embarque tous les capteurs, nous ne disposons pas, à l'heure actuelle, de microphone assez performant pour permettre la reconnaissance vocale à une distance de quelques mètres lors d'une interaction gestuelle. Des travaux cherchant à combler cette lacune ont été initiés par [Argentieri, 2006] et se poursuivent actuellement au laboratoire.



(a) Jido (b) HRP-2 (c) Inbot

FIG. IV.1: Plateformes robotiques concernés par nos travaux.

a) Le robot compagnon Jido

Jido est une plateforme MP-L655 construite par la société Neobotix et équipée d'un bras à six degrés de libertés PA-10 de Mitsubishi. Ce bras est prolongé par une pince permettant la saisie d'objets. Ses deux lasers SICK permettent une localisation précise dans son environnement,

mais aussi la détection des jambes d'utilisateurs potentiels. Il est également équipé de deux bancs stéréos : le premier au bout de son bras lui permet de détecter et localiser précisément des objets à saisir, tandis que le second est juché sur une plateforme pan-tilt au sommet d'un mat dans le but de lui donner une vue d'ensemble de son environnement. Enfin, des capteurs de contact sur ses pinces lui permettent de saisir et relâcher des objets de manière convenable et réactive.

Du point de vue informatique, il est, à l'heure actuelle, équipé de trois ordinateurs :

– un P4-3GHz dédié au contrôle des mouvements du robot,
– un P4-3GHz dédié à la planification et au contrôle des mouvements du bras, ainsi qu'à la vision et à la parole,
– un Core2-2.33GHz dédié à la planification des mouvements du robot prenant en compte son environnement et l'être humain.

Du fait de son physique disgracieux et de son encombrement, Jido n'est pas fait pour une interaction proximale (1,5m et moins), il n'en reste pas moins une excellente plateforme de développement, sans aucun doute la plus pratique et fonctionnelle. C'est pourquoi, comme nous le verrons plus loin, nous l'utilisons dans le cadre d'une interaction à moyenne distance (1,5m à 3m) dans un but de manipulation conjointe d'objets et de déplacements dans notre environnement d'intérieur.

b) Le robot humanoïde HRP-2

HRP-2 est un robot humanoïde de taille humaine (154 cm) développé par le laboratoire japonais AIST (National Institute of Advanced Industrial Science and Technology) et construit par le fabricant japonais Kawada. Il a été acquis par le CNRS dans le cadre de sa collaboration avec l'AIST à travers le laboratoire franco-japonais JRL (Joint Robotics Laboratory). Ce robot possède 30 degrés de libertés et divers capteurs lui permettant notamment de conserver dynamiquement son équilibre lors de la marche bipède, ainsi que la manipulation d'objets *via* les pinces qui lui servent de mains. Il est également équipé de deux jeux de caméras installées dans sa tête (et par conséquent orientable suivant ses trois axes) dont nous nous servons dans ces travaux afin d'acquérir des données visuelles sur son utilisateur. Les caméras centrales ont un champ de vision restreints, et permettent donc de filmer des détails proches du robot, tandis que les caméras extérieures ont au contraire un champ large, donnant une vision globale de son environnement au robot. Il est également à noter que la supervision n'est pas encore mise en œuvre sur ce robot. Il est équipé de deux ordinateurs Core2-2.33GHz dont l'un est dédié à la gestion temps-réel de la marche et de l'équilibre du robot.

Par sa physionomie, proche de celle de l'être humain, les applications d'interaction entre ce robot et l'homme s'orientent naturellement vers une interaction proximale et directe (échange d'objets, éventuels contacts physiques avec l'homme). Il est cependant à noter que la fragilité (et le prix) de ce robot ne permettent pas une utilisation aussi aisée que Jido. En effet, les contraintes d'équilibre alliées à des programmes qui n'assurent pas forcément la non auto-collision et encore moins la non collision avec l'environnement rendent son utilisation dangereuse pour le robot. C'est pourquoi, contrairement à Jido, nous privilégions une utilisation « statique » de ce robot, c'est-à-dire que nous préférons ne pas utiliser la marche.

c) Le caddie intelligent Inbot

Inbot est un caddie dit « intelligent » construit dans le cadre du projet européen CommRob. Ce robot est développé par le FZI (Forschungszentrum Informatik) de Karlsruhe et est utilisé dans diverses démonstrations dans des environnements de type supermarché. Ce cadre a été choisi pour les situations qu'il génère qui sont à la fois courantes et habituelles pour l'homme, mais qui se déroulent dans des environnements encombrés et fortement dynamiques. Le rôle du robot est de pouvoir remplir les mêmes missions qu'un caddie classique, à la différence près qu'il est motorisé (et commandable à travers un poignée haptique) et doit pouvoir être commandé par la voix et le geste.

Le robot est équipé d'un système de caméras stéréo, de lasers et de deux ordinateurs industriels dont l'un est dédié au mouvements du robot.

IV.2.2 Scénarios associées à ces plateformes

Nous avons choisi de valider notre interface multimodale à travers divers scénarios décrits dans cette sous-section. Les différents modules décrits dans les chapitres successifs de ce manuscrit étant trop nombreux et différents pour être validés tous en même temps, nous avons choisi d'effectuer cette validation suivant une approche graduelle. Cette dernière nous a permis, à travers des scénarios à difficulté croissante, d'évaluer notre interface par partie, mais également au fur et à mesure des développements des différents modules constituant celle-ci.

Rappelons que ces scénarios ne sont que des exemples de fonctionnement de notre interface dans un cadre limité, nos modules étant utilisés à pleine capacité (possibilités de reconnaître tous les gestes modélisés et toutes les phrases faisant partie de la grammaire) quels que soit leur utilisation.

a) Scénario n°1 : le robot assistant

Ce premier scénario envisagé consiste à se placer dans le cadre de l'assistance à une personne handicapée, cette dernière ayant par exemple une jambe dans le plâtre. Après s'être présentée au robot, cette personne lui demande de prendre l'une des bouteilles (il y en a deux) présentes sur la table en désignant celle-ci de la main. Le robot doit alors se déplacer afin de pouvoir saisir cette bouteille. Une fois cette dernière dans la pince du robot, l'utilisateur lui demande à nouveau de se déplacer en désignant ce nouveau but de la main. Enfin, l'utilisateur demande au robot de lui donner la bouteille en tendant la main.

Le but de ce scénario est de montrer un exemple dans lequel l'utilisation conjointe de la parole et de la vision est nécessaire, et ce à travers :

- la désignation d'une bouteille sur une table, l'endroit devant être assez précisément désigné pour permettre la localisation et la saisie de celle-ci,
- la désignation au sol d'un endroit où le robot devra se placer,
- l'échange d'objets, le robot devant coordonner ses mouvements avec ceux de l'homme.

Il est à noter que ce scénario n'utilise que les modalités de base de notre interface, à savoir le traitement de la parole et le suivi de gestes (c'est-à-dire les modules $RECO$ et $GEST$).

b) Scénario n°2 : le robot de service en environnement intérieur

Le tableau IV.1 résume le second scénario mis en place afin d'évaluer notre approche. Son but est de montrer non seulement la complémentarité du geste et de la parole pour des ordres déictiques, comme l'a déjà fait la démonstration précédemment décrite, mais également la possibilité de renforcement des ordres multimodaux impliquant notamment des gestes symboliques. Cela implique quelques difficultés supplémentaires, telles que la perception des mouvements globaux de l'homme, ainsi que la création d'une réelle stratégie de fusion de données.

#	Ordres de l'utilisateur humain	**Actions associées du robot**	**Commentaires**
1.	« Bonjour, je suis ici. » accompagné d'un geste symbolique de salutation	Déplacement en direction de l'utilisateur	Le robot interrompt sa tâche courante, avance et s'arrête en face de l'utilisateur.
2.	« Salut RobotX, c'est X. »		L'utilisateur est identifié afin qu'il ai le droit d'utiliser le robot.
3.	« Viens vers moi. » accompagné du geste symbolique associé	Déplacement en direction de l'utilisateur	L'exécution de la commande nécessite la connaissance de la position 3D de l'utilisateur.
4.	« Stop. » accompagné du geste symbolique associé	Le robot s'arrête	Cette commande est utilisée pendant que le robot bouge.
5.	« Prends cet objet. » accompagné d'un geste de pointage	Le robot se déplace vers l'objet, puis le saisit	Le robot cherche l'objet pointé et le prend s'il est présent
6.	« Viens à ma gauche. »	Déplacement en direction de l'utilisateur	L'exécution de la commande nécessite la connaissance de la position 3D de l'utilisateur.
7.	« Donne moi l'objet. »	Manipulation de l'objet	L'exécution de la commande nécessite la position 3D de la main de l'utilisateur.
8.	« Merci, tu peux t'en aller. » accompagné du geste symbolique associé	Déplacement s'éloignant de l'utilisateur	

TAB. IV.1: Résumé du scénario n°2 d'interaction entre utilisateur et robot incluant la reconnaissance de gestes.

Dans cette démonstration, nous voulons gérer les possibles incompréhensions du robot en nous basant sur la communication homme-homme. Ainsi, en cas d'échec de la reconnaissance de parole ou de gestes, et bien que nous n'ayons pas développé de module de gestion de dialogue destiné à gérer les interactions complexes, nous avons donné la possibilité au robot de

demander à l'utilisateur de reformuler sa dernière requête à chaque fois qu'une étape du scénario échoue sans conséquence irréversible (c'est-à-dire que le robot n'est ni perdu, ni dans une position l'empêchant de poursuivre le scénario). Le robot, pour sa part, relance alors cette étape. Le nombre de répétition est toutefois limité à trois, puisque nous considérons qu'au delà, un utilisateur lambda considérera ces incompréhensions comme excessives. Précisons, que mis à part les modules développés durant nos travaux ($GEST$, $RECO$, $DREC$, $FUSION$), d'autres modules prennent ici une part importante dans le scénario. Ces derniers seront décrits succinctement dans la section suivante.

c) Scénario n°3 : le partenaire de jeu

Cette nouvelle démonstration poursuit deux buts. Le premier est notamment de démontrer l'apport de l'intégration du suivi de regard dans la reconnaissance de geste. Le second est d'introduire une plus grande liberté dans l'utilisation du robot par l'homme et par conséquent de créer des démonstrations plus dynamiques et naturelles que les précédentes.

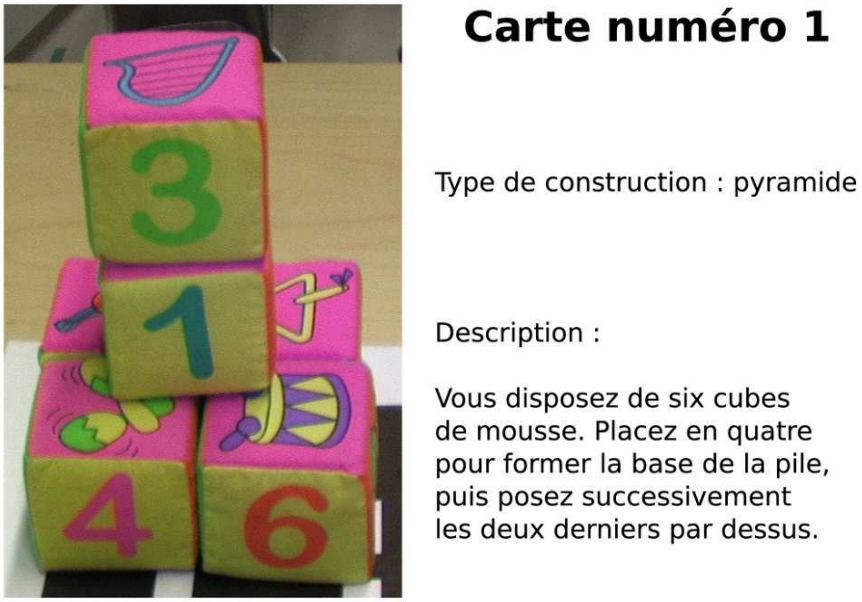


FIG. IV.2: Exemple de carte donnant la forme géométrique à faire construire au robot.

Le contexte général de cette démonstration est un jeu interactif en face à face, mené par le robot, mais guidé par l'homme. Il se base sur des cubes en mousse numérotés que le robot doit, suivant les instructions de l'homme, placer de manière à construire une forme géométrique prédéterminée. Le scénario se déroule de la manière suivante :

1 . L'utilisateur tire une carte sur laquelle se trouve la forme géométrique à faire réaliser au robot (un exemple d'une telle carte est donné par la figure IV.2).

2 . L'utilisateur salue le robot, signalant ainsi son intention de débuter le jeu.

3 . L'utilisateur, à travers des commandes verbales et/ou des gestes symboliques ou surtout déictiques, fait placer un à un les cubes de mousse par le robot sur la table.

4 . L'utilisateur clôture l'interaction en remerciant et saluant le robot.

5 . Le but est atteint si la forme géométrique voulue au départ a bien été construite par le robot.

Dans le cadre du placement des cubes de mousse, les différentes commandes, multimodales ou non, disponibles sont décrites dans le tableau IV.2. Les paroles sont ici données à titre indicatif, plusieurs dizaines voire centaines de possibilités de phrases et de prononciations étant en réalité disponibles pour chaque commande.

#	Type de commandes de l'utilisateur humain	**Actions associées du robot**	**Commentaires**
1.	« Prends ce cube. » accompagné d'un geste de pointage	Le robot prend le cube pointé.	commande multimodale
2.	« Prends-le cube numéro deux. »	Le robot prend le cube en question.	commande avec référent
3.	« Pose-le ici. » accompagné d'un geste de pointage	Le robot pose le cube qu'il a en main à la position pointée.	commande multimodale
4.	« Pose-le derrière le cube numéro 4. »	Le robot pose le cube qu'il a en main à l'endroit en question.	commande avec référent
5.	« Donne-le moi. », l'utilisateur tend la main vers le robot	Le robot pose le cube dans la main tendue.	l'utilisateur obtient l'objet à poser
6.	« Monte un peu ta pince. »	Le robot monte légèrement sa pince.	commande directe des mouvements du bras du robot
7.	« Stop. » accompagné du geste symbolique associé	Le robot stoppe son mouvement	arrêt d'une commande en cours

TAB. IV.2: Résumé des différentes commandes disponibles pour le scénario de jeu (n˚3).

IV.3 Intégration et évaluations

Cette section présente les architectures informatiques dans lesquelles s'intègre notre interface multimodale, la manière dont nos modules s'y intègrent, ainsi que les résultats obtenus par ces derniers à travers des expérimentations qui suivent les scénarios décrits précédemment.

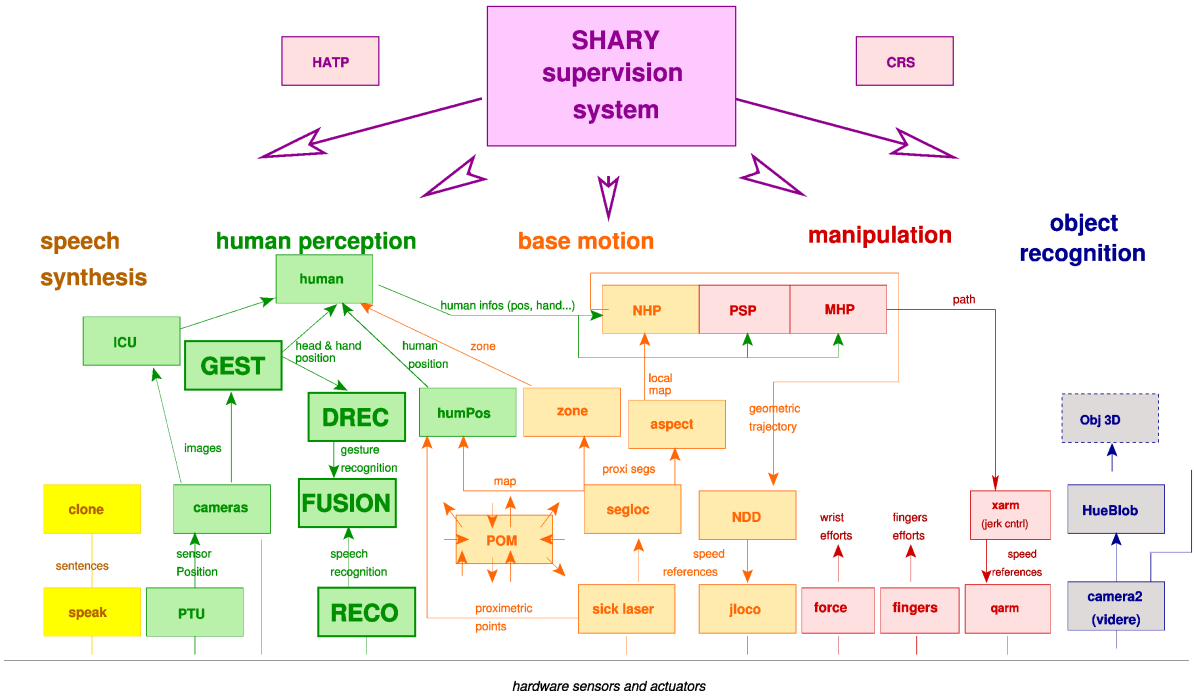


FIG. IV.3: Architecture logicielle Genom du robot Jido.

IV.3.1 Notre architecture logicielle : Genom et ses modules

a) Description générale d'une architecture Genom

Dans le domaine de la conception d'architectures de contrôle pour l'autonomie des robots, le pôle Robotique et Intelligence Artificielle du LAAS-CNRS utilise et développe l'outil Genom (Génération de modules logiciels ou en anglais "GENerator Of Modules")[Alami et al., 1998]. Ce dernier permet de générer automatiquement des composants logiciels temps-réel en s'appuyant sur un langage de description de composants (interfaces, propriétés temporelles, incompatibilités entre services, etc). Grâce à cette description, la partie algorithmique du composant est, entre autres, interfacée automatiquement aux couches de communication.

Le superviseur [Clodic et al., 2005] est une couche de haut niveau destinée à se servir des informations renvoyées par les différents modules fonctionnant en parallèle sur le robot afin de prendre des décisions concernant le comportement de ce dernier. Ses prises de décisions se traduisent par l'envoi de requêtes à exécuter aux différents modules concernés par la tâche en cours.

b) Architecture Genom de nos plateformes

Cette architecture étant générique, elle est utilisée sur nos différents robots. Les modules utilisés ne sont pourtant pas forcément les mêmes, ceux-ci dépendants souvent des capacités

propres des robots, qui dérivent de leurs attributs physiques et de leur conditions d'évolution.

➢ **Jido**

Ainsi, la figure IV.3, représente l'ensemble des modules Genom nécessaires au fonctionnement de la plateforme Jido. Ils sont classés ici suivant leur fonction :

- les modules jaunes permettent la synthèse de la voix, utile au robot pour communiquer à l'homme des informations sur son état ou ses intentions, ainsi que pour lui poser des questions afin de l'inviter à interagir,
- les modules verts sont destinés à la perception de l'homme, parmi lesquels les modules de notre interface, excepté $GAZE$,
- les modules oranges permettent la modélisation de l'environnement et la navigation dans celui-ci,
- les modules rouges servent à la manipulation d'objets (*via* le bras du robot),
- les modules bleus permettent la modélisation et/ou la reconnaissance d'objets
- enfin, en violet, SHARY et ses dépendances sont destinés à la supervision et ne font, à ce titre, pas directement partie de l'architecture Genom.

➢ **HRP-2**

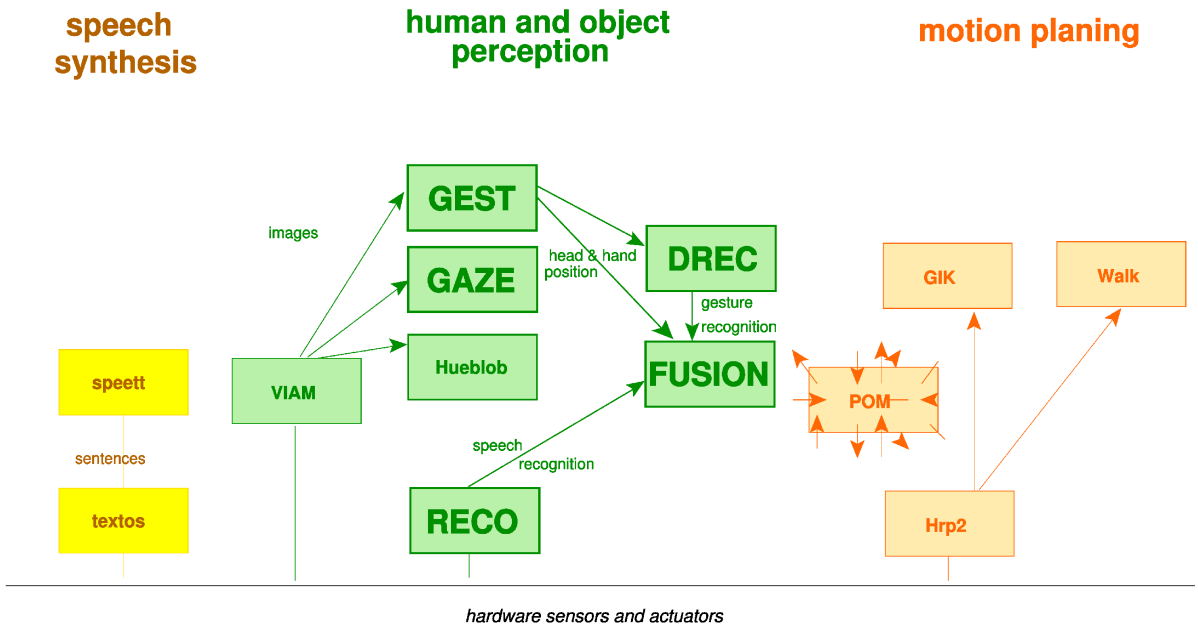


FIG. IV.4: Schéma de l'ensemble des modules Genom nécessaires au fonctionnement de la plateforme HRP-2.

De même, la figure IV.4, représente l'ensemble des modules Genom nécessaires au fonctionnement de la plateforme HRP-2, qui sont classés de la même manière que ceux de Jido. On remarquera que si les fonctions visuelles et auditives sont proches pour les deux plateformes, ce n'est pas le cas des autres. En effet, ce robot étant anthropomorphe et devant par conséquent

gérer son propre équilibre, les fonctions de manipulation utilisent les mêmes méthodes de planification de mouvement que les fonctions de déplacement par la marche. Il est à noter que dans le cadre de notre interface multimodale, le module $GAZE$ utilise les caméras centrales du robot afin d'obtenir l'image la plus fidèle du visage de l'utilisateur, tandis que le module $GEST$, qui fournit à $GAZE$ la position de la tête de l'utilisateur, utilise les caméras extérieures afin que les mains de l'utilisateur ne sortent pas du champs de vision du robot lorsqu'il en est proche (interaction à moins de deux mètres). Il est également à noter que, contrairement à Jido, HRP-2 n'est pas équipé du récepteur nécessaire à l'utilisation d'un micro-casque, c'est pourquoi la fonction d'acquisition et de traitement de la parole est faite sur un ordinateur portable séparé, son résultat étant ensuite envoyé au robot par Wifi.

➢ **Inbot**

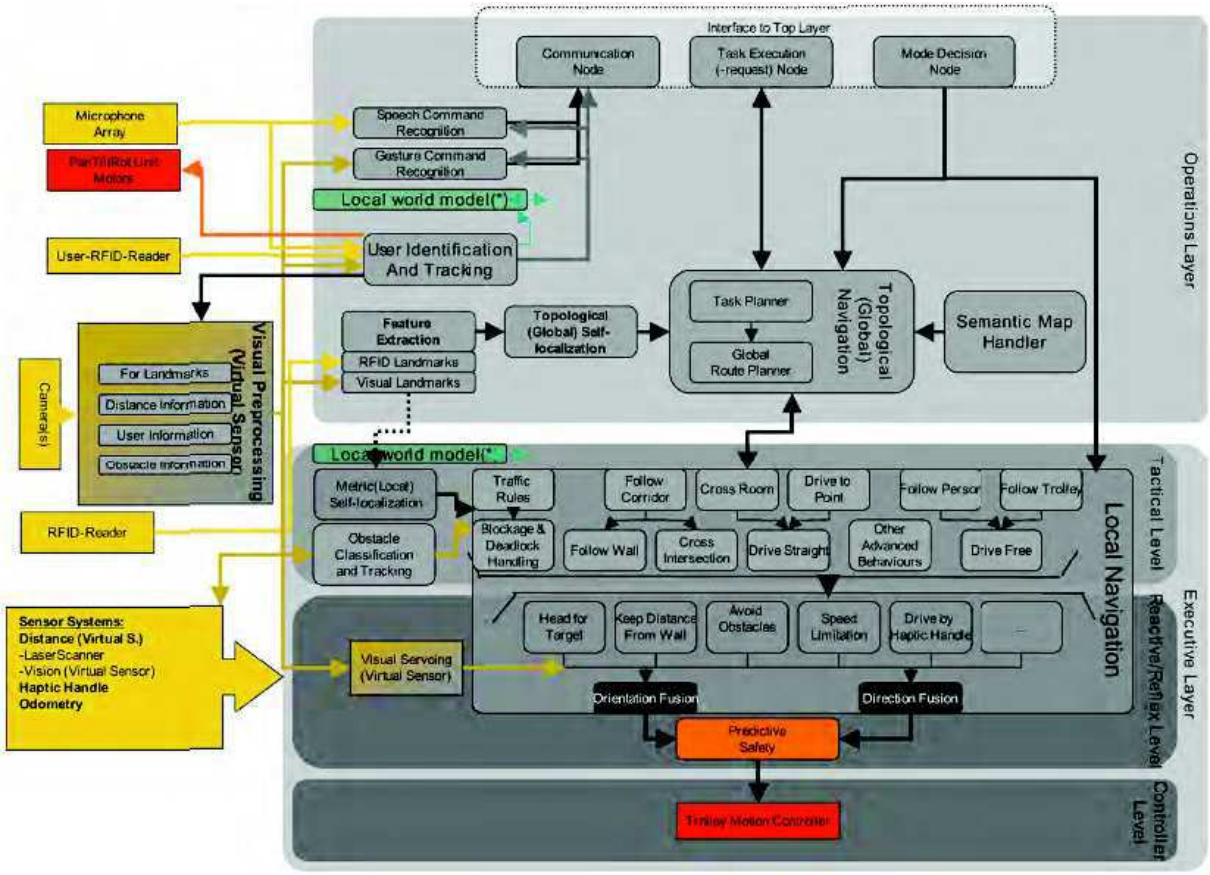


FIG. IV.5: Architecture fonctionnelle du robot-caddie Inbot.

Comme le montre la figure IV.5, ce robot fonctionne sous MCA2, développé par le FZI et qui est un équivalent de Genom. Notre apport dans ce projet concerne le suivi et la reconnaissance de gestes, la reconnaissance de parole ainsi que sa fusion avec le geste étant gérées par un autre partenaire du projet CommRob.

c) Intégration de notre interface pour l'interaction homme-robot

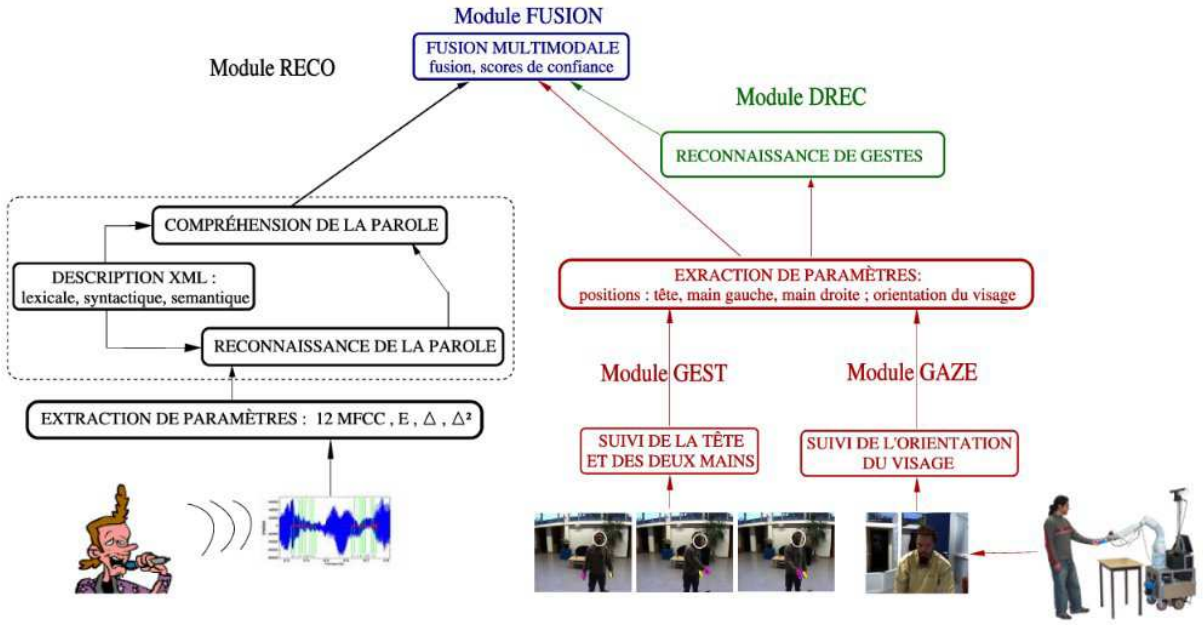


FIG. IV.6: Architecture globale de notre interface homme-robot.

Comme le montre la figure IV.6, notre interface destinée à la communication homme-robot regroupe les différents modules développés tout au long de cette thèse et est maintenant complète. Le module $FUSION$ s'intègre dans ce schéma en captant les sorties standards des modules de base ($GEST$, $RECO$, $DREC$) afin de les traiter de manière coordonnée, puis, dans le cadre d'une architecture globale (telle que celle décrite par la figure IV.3), d'envoyer le résultat de ce processus de fusion au superviseur.

IV.3.2 Stratégies de fusions et évaluations robotiques

Cette sous-section décrit les stratégies de fusion adoptées afin de permettre de réaliser les différents scénarios décrits précédemment. Elle donne également les résultats qualitatifs, mais aussi quantitatifs, obtenus lors de leur exécution.

a) Scénario n°1

Ce scénario a pour but de démontrer la faisabilité et l'utilité de notre démarche en utilisant uniquement les modules de base de notre interface, à savoir $GEST$ et $RECO$. Compte tenu des distances d'interaction et de la nécessité pour la plateforme de pouvoir se mouvoir facilement dans une pièce, Jido est le robot le plus à même de participer à ce scénario.

FIG. IV.7: De gauche à droite et de haut en bas : une réalisation du scénario n˚1 montrant la complémentarité de la parole et de la vision (vue d'ensemble -image principale- , résultats du module $GEST$ -images incrustées-, la reconnaissance et l'interprétation de la parole sont assurées par le module $RECO$).

➢ **Description de la stratégie de fusion**

Dans le cadre de ce scénario, nous avons cherché à réaliser une stratégie de fusion simple permettant de compléter des ordres vocaux avec des informations visuelles ou d'utiliser directement l'ordre vocal si ce dernier ne nécessite aucun complément. Ainsi, une phrase du type « Pose la bouteille sur la table. » est assez précise sur la désignation et la localisation de l'objet sur lequel agir, ainsi que du but de l'action à fournir par le robot. Le module $RECO$ est alors capable d'en extraire une interprétation complète et satisfaisante pour le superviseur ou un script chargé de mener une démonstration à son terme : *POSER(OBJET=bouteille, LOCALISATION=sur la table)*. Par contre, dans le cas de phrases incluant des déictiques, comme « Pose la bouteille là-bas. », les informations fournies par la parole ne sont pas suffisantes. Notre interpréteur marque alors certains champs de l'interprétation comme « à remplir ». C'est ici qu'intervient le module $FUSION$ dont la mission est de compléter, dans une stratégie de fusion tardive, les ordres incomplets fournis par le module $RECO$ avec des informations visuelles, notamment celles issues du module $GEST$.

En pratique, un ordre nécessitant un geste de désignation sera complété par les coordonnées, dans l'espace de navigation du robot, de l'objet ou du lieu pointé en extrayant une droite tête-main et en calculant son intersection avec un plan de l'environnement (typiquement, une

table ou le sol). De la même manière, un ordre nécessitant un référant, comme « Viens sur ma gauche. », pourra être complété à partir de la position de l'homme extraite par le module $GEST$. Enfin, n'utilisant pas encore, dans ce scénario, de reconnaissance de gestes, l'instant image choisi pour extraire les informations visuelles est déterminé par la parole.

➢ **Résultats qualitatifs**

La figure IV.7 illustre le scénario décrit précédemment. À chaque moment important du scénario est associé une image montrant une vue d'ensemble de la situation homme-robot et, incrustée dans celle-ci en bas à droite, la sous-figure montre le résultat du suivi de gestes par le module $GEST$. La vidéo complète est disponible à l'adresse suivante : `www.laas.fr/~bburger/`.

Cette expérimentation réussie montre la faisabilité et l'intérêt de notre approche. Les principales erreurs empêchant le déroulement complet du scénario, c'est-à-dire l'obtention de la bouteille par l'humain, proviennent de la précision du geste de pointage qui décroit avec l'angle de la ligne tête-main avec la table. Ce scénario et les développements liés ont donné lieu à la publication suivante : [Burger et al., 2008b].

b) Scénario n°2

Le développement d'un système de reconnaissance de gestes et d'une stratégie de fusion plus avancée autorisée par cette dernière, nous a permis de réaliser une seconde démonstration basée sur le scénario numéro 2. Là encore, étant données les distances moyennes à grandes (2m à 4m) mises en jeu et les nombreux déplacements du robot dans ce scénario, il sera exécuté par Jido.

➢ **Description de la stratégie de fusion**

Dans ce scénario, une stratégie de fusion tardive et hiérarchique à un niveau sémantique a été mise en place. Si, lors d'une interaction, l'utilisateur utilise la parole et le geste dans une même fenêtre temporelle, notre module $FUSION$ sera chargé de la fusion des liste des N meilleures hypothèses de reconnaissance de chaque modalité. Le but de cette stratégie est de permettre une amélioration du taux de réussite de nos démonstrations à travers la reconnaissance combinée d'un ordre multimodal, plutôt qu'à l'utilisation indépendante de chaque modalité. Pour ce faire, nous fusionnons les scores des deux modalités de la manière suivante :

– si un geste g_j se trouve dans la même fenêtre temporelle qu'une hypothèse de reconnaissance de parole h_i qu'il peut complémenter, un score L_f est associé à h_i :

$$L_f(h_i) = L(g_j)^{\alpha} \cdot S(h_i)^{(1-\alpha)};$$

– sinon, un score ne prenant pas en compte la reconnaissance de geste est calculé :

$$L_f(h_i) = {L_M}^{\alpha} \cdot S(h_i)^{(1-\alpha)};$$

où les paramètres sont décrits ci-après. $\alpha = \frac{T_g}{2 \cdot T_s}$ est un paramètre permettant de contre-balancer la différence des taux de reconnaissance des deux modalités (T_g pour les gestes, T_s pour la

parole). $L(g_j)$ est un score calculé à partir de la vraisemblance normalisée de chaque hypothèse g_j de la manière suivante :

$$L(g_i) = \exp^{-\frac{\left(\frac{LL(g_j)}{moy(LL)}\right)^2}{\sigma_2}},$$

avec $moy(LL) = \sum \frac{LL(g_j)}{N_G}$ la moyenne des scores de la reconnaissance de gestes, N_G la taille de la liste des meilleurs gestes et σ_2 une covariance déterminée empiriquement. $L_M = \exp^{-\frac{1}{\sigma_2}}$ permet, quand aucun geste n'est nécessaire, de procéder comme si nous disposions d'un geste dont le score est égal à la moyenne des scores de la reconnaissance de gestes $moy(LL)$, le but étant de ne privilégier ni discriminer les ordres vocaux accompagnés de gestes par rapport à ceux qui en sont dépourvus. Enfin, $S(h_i)$ est le score calculé pour une hypothèse vocale h_i selon la méthode détaillée en section I.4.3 du chapitre I.

Le lecteur trouvera les valeurs utilisées pour ces paramètres dans le tableau IV.3. Il est à noter qu'en pratique, la reconnaissance de parole est utilisée comme la modalité principale lors d'une interaction et la fenêtre temporelle dans laquelle doit se trouver un geste est en réalité déterminée par la détection d'une parole pour son début et par une temporisation pour sa fin.

Symbole	signification	valeur
σ_1	facteur de normalisation de la vraisemblance parole	0.2
σ_2	facteur de normalisation de la vraisemblance geste	0.5
N_S	taille de la liste de N meilleures hypothèses de parole	10
N_G	taille de la liste de N meilleures hypothèses de gestes	12

TAB. IV.3: Valeur des paramètres utilisés dans le module $FUSION$.

➢ **Performances hors-ligne de cette stratégie de fusion**

Avant d'étudier la mise en œuvre du scénario, nous avons cherché à connaître les performances de notre stratégie de fusion sur des ordres multimodaux du même type que ceux utilisés dans le scénario. Ainsi, le tableau IV.4 représente les résultats de ce processus de fusion pour les sept catégories d'ordres multimodaux utilisés dans le scénario. Chacune de ces catégories a été répétée dix fois , ce qui nous donne un petit corpus de 70 requêtes multimodales.

Ce tableau est à comparer aux résultats de chaque modalité isolée. Ainsi, alors que le tableau I.7 du chapitre I montre un taux d'interprétation de phrases de 87,8% et que le tableau III.3 du chapitre III montre un taux de reconnaissance de gestes de 79,7%, le taux de commandes multimodales correctement interprétée s'élève à 92%. Ceci valide logiquement que ce processus de fusion permet une amélioration du taux de reconnaissance combiné de chaque modalité indépendante.

➢ **Résultats qualitatifs et quantitatifs sur le scénario**

Le scénario numéro 2 a été joué plusieurs fois afin de déterminer sa répétabilité et de permettre le calcul de statistiques de réussite. Le but global de la démonstration est, suivant le résumé du tableau IV.1, de faire venir le robot à une table, de lui faire prendre une bouteille

#	fusion de : geste + type de phrase prononcée	1	2	3	4	5	6	7	autre
1 :	« Présentation » + type présentation	91	0	0	0	0	0	0	9
2 :	« Salutation » (à une ou deux mains) + type salutation	0	82	0	0	0	0	0	18
3 :	« Stop » + type stop	0	0	64	0	0	0	18	18
4 :	Geste de pointage + type « Prends cet objet. »	0	0	0	91	0	0	0	9
5 :	« Viens vers moi » (à une ou deux mains) + type « Viens vers moi. »	0	0	0	0	100	0	0	0
6 :	Geste de pointage + type « Viens ici. »	0	0	0	0	0	100	0	0
7 :	« Va t'en » +type « Va t'en. »	0	0	9	0	0	0	91	0

TAB. IV.4: Matrice de confusion résultat du processus de fusion (en %).

et de la donner à l'utilisateur. La figure IV.8 illustre les moments clés lors d'une réalisation de ce scénario. À chaque pas, la figure principale représente la situation homme-robot courante, tandis que la sous-figure montre les résultats du suivi de gestes par le module $GEST$, ou d'un autre module tel que ICU (reconnaissance faciale) ou HueBlob (localisation d'objets, ici des bouteilles). Dans ces évaluations, les commandes multimodales ont été interprétées avec succès et le robot a réussi à donner l'objet pointé par l'utilisateur à celui-ci. La vidéo complète est disponible à l'adresse suivante : `www.laas.fr/~bburger/`.

#	RECO	GEST/DREC	FUSION	Autres	Commentaires
1.	0	1	0	0	
2.	0	0	0	1 ICU	reconnaissance faciale
3.	1	3	1	0	la distance du robot rend les gestes difficiles à suivre
4.	3	2	2	0	le temps de calcul est parfois trop long quand le robot est en mouvement
5.	0	0	0	2 HueBlob	la bouteille n'est pas toujours localisée
6.	0	0	0	0	la gauche n'est pas toujours exactement à gauche...
7.	0	0	0	2 MHP (1 fatale)	main trop éloignée, erreurs de localisation du robot
8.	2	4	1	0	

TAB. IV.5: Erreurs ayant eu lieu durant l'exécution de nos différents tests pour chaque module.

À partir de ce scénario, nous avons également mené des évaluations quantitatives lors de l'exécution du scénario. Moins il y a d'erreur provenant de notre interface multimodale, plus l'interaction est naturelle et agréable pour l'utilisateur. Le tableau IV.5 expose les statistiques récoltées durant les 14 exécutions accomplies du scénario.

Commentons ces résultats. Sur les 14 exécutions du scénario complet, nous n'avons observé qu'une seule erreur fatale (notée comme telle dans le tableau). Celle-ci est due à une erreur de localisation rendant impossible par la suite tout mouvement du robot, et n'est donc pas attribuable à notre interface multimodale. Les principales erreurs de notre interface sont apparues pour des situations touchants aux limites du système, comme par exemple, rappelons-le, la précision des gestes de pointage qui diminue avec l'angle entre la droite tête-main et la table. De la même manière, les phrases courtes restent les plus difficiles à reconnaître, en particulier dans un environnement pollué par des bruits secs. Mis à part ces limitations, notre interface multimodale

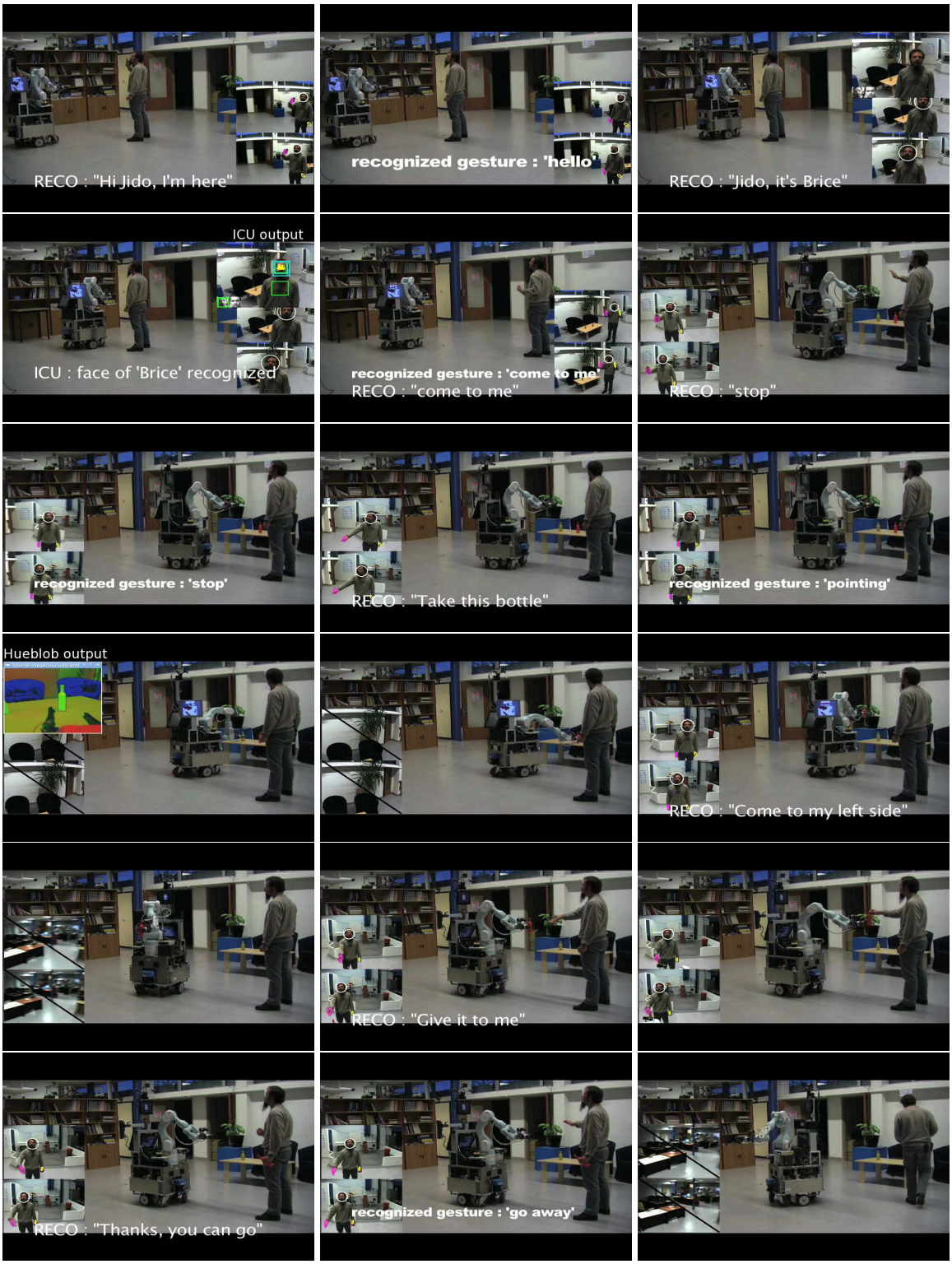


FIG. IV.8: Moments choisis d'une exécution du scénario avec fusion des données vocales et gestuelles : vue d'ensemble -image principale- , résultats du module $GEST$ et/ou d'autres modules (HueBlob, ICU) -images incrustées-, la reconnaissance et l'interprétation de la parole sont assurées par le module $RECO$.

a montré une robustesse suffisante pour permettre le bon déroulement du scénario.

Ces développements ont contribué aux publications suivantes : [Burger et al., 2009c, Burger et al., 2009a, Burger et al., 2010].

➢ **Application à Inbot**

Le contexte applicatif du robot-caddie Inbot relève de la même stratégie et du même type de scénario que celui-ci. L'évaluation sur cette plateforme ne sera pourtant pas développée ici car cette dernière relève d'un partenaire du projet CommRob et qu'elle aura lieu au cours du mois de décembre. Le lecteur intéressé pourra toutefois se rendre sur le site du projet `www.commrob.eu` pour en savoir plus. Les travaux menés sur ce robot ont conduit à la publication suivante en collaboration avec l'université de Vienne [Vallée et al., 2009].

c) Scénario n°3

Ce scénario implique le robot HRP-2. Ce robot a été choisi afin de mener des expérimentations dans un contexte d'interaction proximale, permettant notamment l'utilisation du module de suivi du visage $GAZE$, mais également pour sa forme humanoïde ouvrant la voix à des expérimentations plus naturelles pour des utilisateurs non roboticiens. HRP-2 est une plateforme bien plus adapté à une démonstration voulant le transformer en partenaire de jeu. De plus, il est intéressant d'évaluer la faisabilité d'une démonstration de ce type sur un robot humanoïde, ces derniers étant actuellement essentiellement dédiés aux recherches sur la planification de mouvements.

➢ **Description de la stratégie de fusion**

La stratégie de fusion adoptée ici est globalement la même que précédemment, mis à part quelques aménagements visant à permettre l'utilisation de la reconnaissance de gestes avec suivi de l'orientation du visage ainsi que l'utilisation d'un ordinateur supplémentaire pour palier l'absence de récepteur audio sur le robot.

➢ **Résultats qualitatifs**

La figure IV.9 illustre une exécution du scénario de jeu décrit précédemment. La figure géométrique à construire grâce au robot est celle de la carte IV.2. À chaque pas, la figure principale représente la situation homme-robot courante, tandis que la sous-figure montre les résultats du suivi de gestes par le module $GEST$ et du suivi de visage par le module $GAZE$. Comme expliqué plus haut, ce scénario n'est pas linéaire : aucun ordre dans lequel les commandes doivent être exécutées n'est imposé à l'utilisateur. Lors de cette exécution, les commandes multimodales ont été pour la plupart interprétées correctement, permettant au robot de construire la figure voulue suivant les ordres de l'utilisateur. Les quelques erreurs d'interprétations ont pu être rattrapés facilement grâce à la plus grande liberté avec laquelle l'utilisateur peut interagir avec le robot. En effet, si le robot pose un cube à une position erronée, rien n'empêche l'utilisateur de lui demander de le reprendre, puis de lui indiquer à nouveau l'endroit où le poser. La vidéo complète est disponible à l'adresse suivante : `www.laas.fr/~bburger/`. Il est à noter que, pour sa sécurité, le robot est relié par les épaules à un pont roulant : en cas d'appui

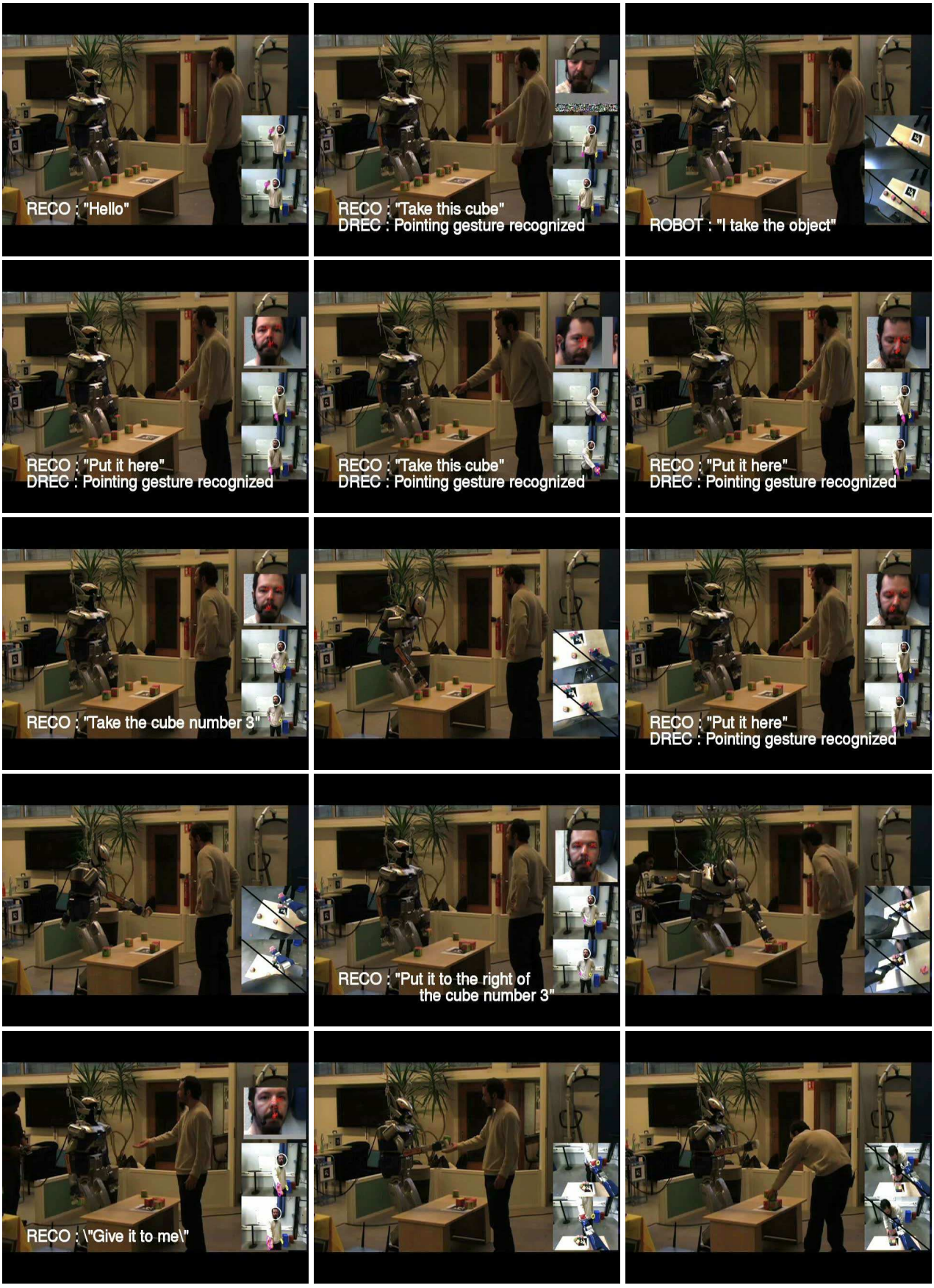

RECO : "Hello"
RECO : "Take this cube"
DREC : Pointing gesture recognized
ROBOT : "I take the object"
RECO : "Put it here"
DREC : Pointing gesture recognized
RECO : "Take this cube"
DREC : Pointing gesture recognized
RECO : "Put it here"
DREC : Pointing gesture recognized
RECO : "Take the cube number 3"
RECO : "Put it here"
DREC : Pointing gesture recognized
RECO : "Put it to the right of
the cube number 3"
RECO : \"Give it to me\"

FIG. IV.9: Une réalisation du scénario de jeu (n°3) avec fusion des données vocales et gestuelles : vue d'ensemble -image principale- , résultats des modules $GEST$ et $GAZE$ -images incrustées-, la reconnaissance et l'interprétation de la parole sont assurées par le module $RECO$.

sur l'arrêt d'urgence, le robot est alors retenu et ne s'effondrera pas au sol. Mais cette sécurité n'a aucune incidence sur le déroulement d'un scénario nominal.

Le fait que ce scénario laisse à l'utilisateur bien plus de libertés que les précédents, accompagné du fait que nous utilisons ici HRP-2 et non Jido, nous permet d'envisager une étude à plusieurs utilisateurs, et donc des résultats quantitatifs en terme de scénarios réalisés avec succès, à l'instar de Jido. Les essais menés jusqu'à présent ne concernant qu'un seul utilisateur et étant trop peu nombreux, ils ne nous permettent pas de publier des statistiques. Elles donnent néanmoins une idée des difficultés qui restent à surmonter avant de réaliser une étude de grande envergure, notamment avec des non-roboticiens dans le rôle d'utilisateur. En effet, sur ces 6 essais, seul deux ont été couronnés de succès, mais un seul échec est directement imputable à nos modules, les trois autres étant dûs à l'arrêt d'urgence du robot lors de manœuvres dangereuses.

Les principales difficultés, et sources d'échec, de l'utilisation d'un robot tel que HRP2 pour une démonstration de ce type sont donc en réalité liées à sa fragilité ainsi qu'au fait que ses mouvements ne soient pas sûr du point de vue de l'auto-collision et des collisions avec l'environnement. Sa fragilité alliée à ces manquements imposent des mesures de précautions telles l'arrêt d'urgence à la moindre manœuvre dangereuse, des déplacements ralentis qui rendent les démonstrations très longues et finalement assez peu naturelles, et la mobilisation de plusieurs personnes durant chaque démonstration. Mais ces difficultés sont essentiellement liées à la jeu-

nesse du robot et des recherches qui lui sont associées et nous sommes persuader qu'elle se dissiperont assez rapidement.

IV.4 Conclusion

Nous avons présenté dans ce chapitre nos travaux sur la fusion de données audio-visuelle dans un contexte IHR, ainsi que des démonstrations liées à cette fusion. Nous avons ainsi développé un module nommé $FUSION$ et dédié à la fusion des données vocales issues du module $RECO$ avec les données visuelles issues des modules $GEST$ (pour la position 3D de l'homme et de ses mains), $GAZE$ (pour l'orientation du visage) et $DREC$ (pour la reconnaissance de gestes) selon une approche hiérarchique pouvant être schématisée par la figure IV.6.

Nous avons validé notre approche, mais également tous les modules de notre interface un par un, à travers une démarche incrémentale les mettant en jeu dans divers scénarios. Ces études ont permis de mettre à l'épreuve nos diverses fonctions et de nous fournir des résultats qualitatifs, mais également quantitatifs, permettant de prouver l'intérêt et la faisabilité de notre approche globale. Il est à noter que ce travail conséquent d'intégration et d'évaluation robotique est peu présent dans la littérature.

Actuellement des développements importants sont encore en cours afin, d'une part, d'améliorer la fusion des données sensorielles et, d'autre part, de créer des scénarios d'interaction encore plus naturelles. Le plus important de ces développements concerne la gestion des sources de données que nous voulons rendre indépendantes et ainsi permettre l'utilisation de gestes sans parole associé, notamment grâce à l'intégration complète de la segmentation automatique des gestes. Ceci implique de prendre en compte une fenêtre temporelle réelle et coulissante, mais surtout qui n'est plus basée sur la parole, contrairement aux approches existantes en IHR. De cette manière, nous utilisons la corrélation temporelle des données et permettons une interaction plus naturelle, en donnant par exemple la possibilité de fusionner deux gestes avec un énoncé comme dans la commande « Prends ça et pose le ici. ». Enfin, nous espérons également tirer avantage des développements du module $RECO$ allant vers plus de généricité et une quasi indépendance des plateformes.

Conclusion et perspectives

Dans ce manuscrit nous avons présenté nos travaux visant à fournir une interface multimodale pour une interaction naturelle de homme avec le robot. Pour ce faire, nous nous inspirons de la communication homme-homme et partons du constat que les deux modalités principales pour cette dernière sont la parole et le geste. Ces deux modalités peuvent en effet être combinées afin de se compléter et/ou se renforcer. De plus, ces travaux ont abouti à de nombreuses intégrations sur des plateformes robotiques autonomes. En effet, si une évaluation composante par composante est nécessaire, il est aussi primordial de pouvoir évaluer globalement l'ensemble du système. C'est ce que nous avons souhaité faire à travers l'ensemble d'expérimentations qui ont été menées au cours de ces travaux de thèse. Nos travaux se doivent également d'être assez générique pour permettre leur utilisation sur toute plateforme robotique dûment équipée sans besoin d'adaptation. Enfin, évoluant dans un contexte de robotique mobile autonome, nous respectons les contraintes propres à tout robot mobile autonome.

Plus concrètement, nos travaux conduisent à doter un robot de la capacité de reconnaître des commandes vocales simples, telles que « Avance de trois mètres s'il-te-plait. », des commandes impliquant des références à l'homme (« Viens à ma droite. ») ou à l'environnement « Pose le à coté du cube vert. », ou encore des commandes multimodales combinant parole et gestes symboliques (« Salut Jido, c'est Brice. » + geste de salutation) ou déictiques (« Prends cette bouteille. » + geste de pointage). L'ensemble est validé par un ensemble de démonstrations validant l'interprétation des commandes par une action appropriée du robot.

Ce document débute par une introduction présentant le contexte et les objectifs de nos travaux. Nous y présentons également un état de l'art général des interfaces multimodales pour l'IHR tout en nous positionnant par rapport à cette littérature. Enfin, nous décrivons notre approche ainsi que les principales spécificités qu'elle comporte. Rappelons que nos travaux, souvent novateurs dans le pôle RIA du LAAS, couvrent un spectre scientifique très large :

1 . le traitement de la parole et la compréhension du langage naturel,

2 . le suivi visuel multi-cible robuste par filtrage particulaire,

3 . la reconnaissance de gestes par réseau bayésien dynamique,

4 . la fusion de données audio-visuelles dans un contexte IHR.

Le premier chapitre décrit notre module de reconnaissance et d'interprétation de la parole continue, adapté à notre contexte applicatif, $RECO$. Une interprétation générique de la parole et totalement embarquée, peu décrite dans la littérature IHR, en est un point clef.

Le second chapitre présente la partie visuelle de notre interface multimodale à travers deux modules chargés du suivi de l'utilisateur du robot : $GEST$ et $GAZE$. Le premier encapsule un traqueur multi-cible chargé du suivi visuel 3D conjoint de la tête et des deux mains de l'utilisateur du robot. Il permet la gestion des interactions entre cibles, donc une meilleure gestion des occultations mutuelles lors de l'exécution de gestes. La principale spécificité de ce module par rapport à la littérature concerne notre stratégie de filtrage particulaire distribué et interactif. Il est aujourd'hui utilisé couramment sur nos robots. De même, $GAZE$ encapsule un système de suivi de l'orientation du regard basé sur l'utilisation des SURFs. Ces deux modules coopèrent en exploitant les flots vidéos issus de deux systèmes de vision dédiés afin de profiter de leurs diverses caractéristiques intrinsèques.

Le module $DREC$ est l'objet du troisième chapitre de ce mémoire. Il est chargé de la reconnaissance de gestes par DBNs, formalisme peu utilisé dans la communauté IHR. Pour cela il peut s'appuyer sur les données fournies par le module $GEST$, mais également par $GAZE$, la fusion de ces deux sources de données étant un point important. Il aborde également la problématique de la segmentation automatique de gestes.

Enfin, le quatrième et dernier chapitre de ce document décrit le module $FUSION$ de fusion tardive et hiérarchique des deux modalités gérées ici : la parole et le geste.

Chacun de ces cinq modules a été évalué séparément sur des données réelles acquises à partir de nos plateformes robotiques. Chaque chapitre en a présenté les résultats de manière qualitatives et quantitatives. Nous avons ainsi prouvé que notre module de traitement de la parole permet un niveau tout à fait convenable de reconnaissance pour des personnes aux accents pourtant très divers. De même, nous montrons que notre module de suivi de gestes est robuste à la plupart des problèmes liés à la vision et que le suivi du visage est exploitable et intéressant dans notre contexte. Nous prouvons également la pertinence de la modélisation par DBN pour la reconnaissance de gestes. Nous démontrons enfin que la fusion audio-visuelle permet une amélioration des taux de reconnaissance conjoints tout en permettant d'exploiter la complémentarité des deux canaux exploités.

L'intégration sur nos plateformes robotiques étant, comme nous l'avons dit, un point clef de nos travaux, nous avons mené durant cette thèse des expérimentations dont le but a été de valider de manière incrémentale notre interface, les différents modules qui la composent et leur symbiose dans cet ensemble. Ces démonstrations, concluantes sur plusieurs plateformes, prouvent l'intérêt, la généricité et la validité de notre approche, tout en mettant à l'épreuve nos diverses fonctions pour montrer leur robustesse. Ces évaluations, ainsi que la description de nos stratégies de fusion ont fait l'objet du dernier chapitre de cette thèse.

Cette thèse a été à la fois un travail de synthèse (combinaison de différentes techniques et domaines existants) et un travail exploratoire (qui ouvre de nombreuses voix car cette combinaison de domaines et cette application à la robotique d'assistance crée de nouveaux besoins). Elle ouvre donc un certain nombre de perspectives à plus ou moins long terme.

Commençons par évoquer les principales perspectives spécifiques à chacun des modules décrits dans ce manuscrit.

Concernant le module $RECO$, il semble de plus en plus nécessaire de lui permettre de prendre en compte l'historique des discussions entre le robot et l'utilisateur ainsi que les don-

nées contextuelles acquises par les différents modules du robot, afin d'influencer positivement la reconnaissance de parole.

Pour les modules de suivi, $GAZE$ et $GEST$, il serait alors intéressant de développer un filtre particulaire auto-adaptatif permettant une adaptation des différentes mesures utilisées à un moment image donné en fonction du contexte image courant. De plus, si $GEST$ est aujourd'hui robuste et utilisé couramment par un grand nombre de personnes sur différents robot, le module $GAZE$ reste embryonnaire, et devra notamment intégrer un bien plus grand nombre de points d'intérêt et gérer leur apparition/disparition.

Pour le module de reconnaissance de gestes $DREC$, il serait intéressante de tester une manière originale d'effectuer une reconnaissance de geste en utilisant une grammaire de gestes. L'idée serait, en se basant sur des méthodes utilisées en reconnaissance de parole, d'imaginer une division des gestes en sous-gestes élémentaires, ce qui permettrait, à l'aide d'une grammaire (ou plutôt d'une sorte de lexique), de reconstituer n'importe quel geste sans devoir nécessairement procéder à un apprentissage supplémentaire.

Enfin, concernant le module $FUSION$, il serait intéressant d'aller vers un système de plus haut niveau incluant gestion de dialogue et gestion de l'environnement. Ce système serait réellement à mi-chemin entre les couches dite « basses » (extraction et traitement des données capteur, exécution de commandes physiques) et la supervision. De plus, si nos travaux ont prouvé leur validité à travers différentes expérimentations, pour faire sortir ce travail du monde de la recherche et permettre son intégration dans un contexte plus grand public, il faudra à l'avenir monter des expérimentations bien plus larges et incluant notamment des personnes réellement naïves (c'est-à-dire ne sachant absolument pas comment fonctionne un robot et quelles en sont les limites). Ce type d'expérimentations est primordial afin d'évaluer l'acceptabilité de notre interface par l'homme.

De manière plus générale, nous pensons que le champs des investigations de travaux comme les nôtres doit encore être élargi. Ainsi, le robot doit pouvoir disposer d'un panel plus large d'informations sur son environnement : les objets qui le composent, les personnes qui l'entourent, même si elles ne sont pas en interaction directe avec lui, etc. Pour se faire, l'utilisation de la vision doit se généraliser afin de permettre la perception simultanées de plusieurs humains [Zuriarrain et al., 2008], mais également d'objets [Trujillo-Romero and Devy, 2009], et ainsi permettre la reconnaissance de tâches homme-homme ou homme-objet. De même, l'équipement des robots qui doit être étoffé, notamment par des capteurs sonores qui permettront à un ou plusieurs utilisateurs de ne plus devoir porter de micro-casque [Argentieri, 2006]. L'utilisation de tels capteurs permettra également d'effectuer un suivi des locuteurs entourant le robot. Enfin, un soin particulier devra également être apporté à la généricité et à la centralisation de toutes les données collectées afin notamment de permettre un modèle de fusion plus évolué intégrant plus finement les relations temporelles entre geste et parole.

Liste des publications

[Brochard et al., 2009] Brochard, R., Burger, B., Herbulot, A., and Lerasle, F. (2009). Measuring gaze orientation for human-robot interaction. In *Int. Workshop during IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN'09)*, Toyama, Japan.

[Burger, 2008] Burger, B. (2008). Fusion de données audiovisuelles pour l'interaction homme/robot. In *Congrès de l'École Doctorale SYStèmes (EDSYS'08)*, Toulouse, France.

[Burger et al., 2008a] Burger, B., Ferrané, I., and Lerasle, F. (2008a). Multimodal interaction abilities for a robot companion. In *Int. Conf. on Computer Vision Systems (ICVS'08)*, pages 549–558, Santorini, Greece.

[Burger et al., 2010] Burger, B., Ferrané, I., and Lerasle, F. (2010). Two-handed gesture recognition and fusion with speech to command a robot. *Submitted to journal of Autonomous Robots*.

[Burger et al., 2009a] Burger, B., Ferrané, I., and Lerasle, F. (2009a). Towards multimodal interface for interactive robots : challenges and robotic systems description. In *International Journal of Advanced Robotic Systems*. IN-TECH, http ://intechweb.org/.

[Burger and Germa, 2007] Burger, B. and Germa, T. (2007). Fonctions visuelles pour la robotique personnelle. In *Vision par ordinateur en Midi-Pyrénées (VisioMiP'07)*, Toulouse, France.

[Burger et al., 2009b] Burger, B., Infantes, G., Ferrané, I., and Lerasle, F. (2009b). Dbn versus hmm for gesture recognition in human-robot interaction. In *Int. workshop on Electronics, Control, Modelling, Measurement and Signals (ECMS'09)*, pages 59–65, Mondragon, Spain.

[Burger et al., 2009c] Burger, B., Lerasle, F., and Ferrané, I. (2009c). Evaluations of embedded modules dedicated to multimodal human-robot interaction. In *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN'09)*, Toyama, Japan.

[Burger et al., 2008b] Burger, B., Lerasle, F., Ferrané, I., and Clodic, A. (2008b). Mutual assistance between speech and vision for human-robot interaction. In *Int. Conf. on Intelligent Robots and Systems (IROS'08)*, Nice, France.

[Fontmarty et al., 2007] Fontmarty, M., Germa, T., Burger, B., Marin, L., and Knoop, S. (2007). Implementation of human perception algorithms on a mobile robot. In *IFAC Symp. on Intelligent Autonomous Vehicles (IAV'07)*, Toulouse, France.

[Vallée et al., 2009] Vallée, M., Burger, B., Ertl, D., Lerasle, F., and Falb, J. (2009). Improving user interfaces of interactive robots with multimodality. In *Int. Conf. on Advanced Robotics (ICAR'09)*, Munich, Allemagne.

Lexique

LAAS : Laboratoire d'Analyse et d'Architecture des Systèmes
IRIT : Institut de Recherche en Informatique de Toulouse
IHR : interaction homme-robot
IHM : interaction homme-machine
MFCC : Mel Frequency Cepstrum Coefficient
HMM : Hidden Markov Model (en français : modèle de Markov caché)
LVCSR : Large Vocabulary Continuous Speech Recognition (en français : reconnaissance de parole continue grand vocabulaire)
WER : Word Error Rate (en français : taux d'erreur sur les mots)
SER : Sentence Error Rate (en français : taux d'erreur sur les phrases)
SVM : Support Vector Machine (en français : machines à vecteurs de support)
MOT : Mutiple Object Tracking (en français : suivi multi-cibles)
IDMOT : Interactively Distributed MOT (en français : suivi multi-cibles interactivement distribué)
SIR : Sampling Importance Resampling
SURF : Speeded Up Robust Features
DBN : Dynamic Bayesian Network (en français : réseaux bayésiens dynamiques)
Genom : GENerator Of Modules (en français : génération de modules logiciels

Annexes

Annexe A : L'alphabet phonétique français

représentation	exemples d'utilisation
/ a /	lac, cave, agate, il plongea
/ ɑ/	tas, vase, bâton, âme
/ e /	année, pays, désobéir
/ ɛ/	bec, poète, blême, Noël, il peigne, il aime
/ i /	île, ville, épître
/ ɔ/	note, robe, Paul
/ o /	drôle, aube, agneau, sot, pôle
/ u /	outil, mou, pour, goût, août
/ y /	usage, luth, mur, il eut
/ œ/	peuple, bouvreuil, boeuf
/ ø/	émeute, jeûne, aveu, noeud
/ ə /	me, grelotter, je serai
/ ɛ̃/	limbe, instinct, main, saint, dessein, lymphe, syncope
/ ɑ̃/	champ, ange, emballer, ennui, vengeance
/ ɔ̃/	plomb, ongle, mon
/ œ̃/	parfum, aucun, brun, à jeun
/ j /	yeux, lieu, fermier, liane, piller
/ ɥ/	lui, nuit, suivre, buée, sua
/ w /	oui, ouest, moi, squale
/ p /	prendre, apporter, stop
/ b /	bateau, combler, aborder, abbé, snob
/ d /	dalle, addition, cadenas
/ t /	train, théâtre, vendetta

représentation	exemples d'utilisation
/ k /	coq, quatre, carte, kilo, squelette, accabler, bacchante, chrome, chlore
/ g /	guêpe, diagnostic, garder, gondole
/ f /	fable, physique, Fez, chef
/ v /	voir, wagon, aviver, révolte
/ s /	savant, science, cela, façon, patience
/ z /	zèle, azur, réseau, rasade
/ ʒ/	jabot, déjouer, jongleur, âgé, gigot
/ ʃ/	charrue, échec, schéma, shah
/ l /	lier, pal, intelligence, illettré, calcul
/ r /	rare, arracher, âpre, sabre
/ m /	amas, mât, drame, grammaire
/ n /	nager, naine, neuf, dictionnaire
/ ɲ/	agneau, peigner, baigner, besogne

Annexe B : Principe général des SVMs

Les machines à vecteurs de support, aussi appelés séparateurs à vaste marge (ou SVM, pour *Support Vector Machine*) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination (que nous confondrons avec le terme de classification par abus de langage) et de régression. Les SVMs sont en fait une généralisation des classificateurs linéaires introduite par [Vapnik, 1979] et mise en place dans [Vapnik, 1995, Vapnik, 1998].

Les séparateurs à vastes marges sont des classificateurs qui reposent sur deux idées clefs qui permettent de traiter des problèmes de discrimination non-linéaire. La première idée clef est la notion de marge maximale. La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Ces derniers sont appelés vecteurs supports. Dans les SVMs, la frontière de séparation est choisie comme celle qui maximise la marge. Le problème est de trouver cette frontière séparatrice optimale, à partir d'un ensemble d'apprentissage. Ceci est fait en formulant le problème comme un problème d'optimisation quadratique, pour lequel il existe des algorithmes connus. Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables, la deuxième idée clef des SVMs est de transformer l'espace de représentation des données d'entrée en un espace de plus grande dimension (possiblement de dimension infinie), dans lequel il est probable qu'il existe une séparatrice linéaire. Ceci est réalisé grâce à une fonction noyau, qui doit respecter certaines conditions, et qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace. Les fonctions noyau permettent de transformer un produit scalaire dans un espace de grande dimension, ce qui est coûteux, en une simple évaluation ponctuelle d'une fonction, d'où un gain en temps de calcul considérable.

Dans une formulation plus mathématique et en restant dans le cadre d'une classification binaire, le but d'un SVM, comme tout classificateur linéaire, est de trouver une fonction $h(x) = x.w + b$ qui permette de séparer deux jeux de vecteurs de telle manière que ceux pour qui $h(x) \geq 0$ soient rangés dans la classe 1 et les autres dans la classe -1. x est alors un vecteur donné en entrée, w et b les constantes formant l'hyperplan défini par $h(x)$. La frontière de décision $h(x) = 0$ est alors appelé hyperplan séparateur, ou séparatrice. Soit

$$\{(x_1, c_1), (x_2, c_2), \ldots, (x_m, c_m)\} \in \mathbb{R}^N \times \{-1, 1\}$$

la base de données que nous voulons séparer. Dans le cas où ce problème serait linéairement séparable, c'est-à-dire dans le cas où nos données sont séparables par un tel hyperplan (par exemple une droite dans un espace à 2 dimensions), le problème se résume à trouver une fonction $h(x)$ (et donc en réalité w et b) qui respecte $l_k.h(x_k) \geq 0 \quad \forall k \in \{1, \ldots, m\}$ et qui maximise la marge :

$$\arg\max_{w,b} \min_{k} \{\| x - x_k \| / x \in \mathbb{R}^N, h(x) = 0\}$$

Dans le cas où le problème ne serait pas linéairement séparable, c'est-à-dire dans quasiment tous les cas, la parade est de reconsidérer le problème dans un espace de dimension supérieure

dans lequel le problème le deviendra. Il s'agit donc d'appliquer une transformation non-linéaire φ aux vecteurs d'entrée x afin d'arriver dans un espace dit de redescription tel que $l_k.h(x_k) \geq 0$ avec $h(x) = w.\varphi(x) + b$. On retombe alors dans le même problème d'optimisation. Le lecteur intéressé peut se référé à [Burges, 1998] qui lui donnera tous les détails voulus sur la théorie des SVMs et la manière de s'en servir.

En pratique, on ne connait pas la transformation φ. On construit alors directement une fonction noyau K qui doit respecter les condition suivantes : correspondre à un produit scalaire dans un espace de grande dimension, être symétrique et être semi-définie positive. Les fonctions les plus usuelles sont :

- le noyau gaussien : $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$,
- le noyau polynomial : $K(x_i, x_j) = (sx_i.x_j + r)^d$,
- le noyau sigmoïd : $K(x_i, x_j) = \tanh(sx_i.x_j + r)$.

Mais le fait d'utiliser une fonction prédéfinie implique qu'il n'est pas non plus toujours possible de trouver une séparatrice linéaire dans l'espace de redescription. Il se peut aussi que des échantillons soient mal étiquetés, et que l'hyperplan séparateur ne soit pas la meilleure solution au problème de classement. C'est pourquoi [Cortes and Vapnik, 1995] ont introduit la notion de marge souple qui tolère des mauvais classements. Le but devient alors de rechercher un hyperplan séparateur qui minimise le nombre d'erreurs grâce à l'introduction d'une variable ressort ξ_k, qui permette de relâcher les contraintes sur les vecteurs d'apprentissage :

$$l_k(x_k.w + b) \geq 1 - \xi_k \quad \xi_k \geq 0, \quad \forall k \in \{1, \ldots, m\}$$

Avec les contraintes précédentes, le problème d'optimisation est modifié pour devenir la minimisation de :

$$\frac{1}{2} \parallel w \parallel^2 + C \sum_{k=1}^{p} \xi_k \quad , \quad C > 0$$

où C est une constante qui permet de contrôler le compromis entre nombre d'erreurs de classement, et la largeur de la marge.

Il est possible d'étendre cette méthode à des cas de classification non-binaire, c'est-à-dire multi-classes. Imaginons que nous avons comme objectif de classer des échantillons dans M catégories. Alors les deux méthodes les plus connues sont les suivantes. La méthode *one-versus-all* consiste à construire M classificateurs binaires en attribuant le label 1 aux échantillons de l'une des classes et le label -1 à toutes les autres. En phase de test, le classificateur donnant la marge la plus élevée remporte le vote. La méthode *one-versus-one* consiste pour sa part à construire $M(M-1)/2$ classificateurs binaires en confrontant chacune des M classes. En phase de test, l'échantillon à classer est analysé par chaque classificateur et un vote majoritaire permet de déterminer sa classe. Même si c'est un abus de langage, on désigne souvent ces ensembles de classificateurs SVM par le terme de SVM multi-classe.

Annexe C : Détermination des paramètres libres d'un système par tracé de courbes ROC

Souvent, les paramètres libres d'un module sont déterminés arbitrairement, généralement après quelques tests rarement représentatifs. Mais dans le but d'évaluer un système plus proprement et en en profitant pour l'optimiser, nous pouvons déterminer ces paramètres en utilisant des courbes ROC (pour "receiver operating curve"). Ces courbes sont construites à partir de points ROC, c'est-à-dire de points dont l'abscisse est le taux de faux négatifs (bonne reconnaissance déclarée négative, noté FRR) et l'ordonnée le taux de faux positifs (mauvaise reconnaissance déclarée correcte par le système, noté FAR) obtenus par le module sur un jeu de paramètres $\mathbf{q}$ donné. Ainsi, pour un classificateur donné, un jeu $\mathcal{Q}$ de l'ensemble des vecteurs de paramètres admissibles $\mathbf{q}$ génère un jeu de points ROC. Notre but est alors de trouver le point dit « dominant » qui est déterminé en traçant le front de Pareto optimal par rapport au nuage convexe de points ROC. Plus formellement, nous recherchons le sous-ensemble $\mathcal{Q}^*_{1:n} \subset \mathcal{Q}_{1:n}$ de vecteurs de paramètres libres $\mathbf{q}_{1:n}$ pour lesquels il n'y a aucun autre vecteur de paramètre respectant le mieux les objectifs suivants dans $\mathcal{O} = \{FRR, FAR\}$:

$$\mathcal{Q}^* = \{\mathbf{q} \in \mathcal{Q} | \forall\, \mathbf{q}^{'} \in \mathcal{Q},$$
$$\forall f_1 \in \mathcal{O}, f_1(\mathbf{q}) \geq f_1(\mathbf{q}^{'}) \wedge\ \exists f_2 \in \mathcal{O}, f_2(\mathbf{q}) > f_2(\mathbf{q}^{'})\}$$

$\mathcal{Q}^*$ désigne alors le sous-ensemble de vecteurs de paramètres potentiellement optimaux pour ce classificateur. Le lecteur intéressé pourra lire [Provost and Fawcett, 2001] pour obtenir de plus amples détails sur cette méthode et son utilisation.

Annexe D : Projection d'ellipsoïde 3D sur une image

Afin de déterminer les poids de chaque particule $\mathbf{x}_t^i$ du filtre particulaire du module $GEST$, nous devons tout d'abord les projeter sur les plans images (les mesures étant définies dans ces plans). Pour ce faire, et pour chaque particule, l'ellipsoïde définie par $\mathbf{x}_t^i = (X, Y, Z, a_x, a_y, a_z, \theta_x, \theta_y, \theta_z)$ est mise sous sa forme matricielle $Q' = H^{-1T} Q H^{-1} = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix}$, c'est à dire sous la forme d'une matrice symétrique 4×4 calculée à partir de la représentation Q d'une ellipsoïde 3D sans rotation ni translation et ses matrices de transformation H^{-1} contenant ces dernières. H est en réalité la matrice de passage du repère image $R_c = (M, i, j, n)$ au repère $R_o = (0, X, Y, Z)$ de l'ellipsoïde représentée par Q, avec $\{i, j\}$ les axes des images, M leur origine et n la normale au plan image.

Dans ces conditions, projeter l'ellipsoïde sur le plan image revient à résoudre l'équation suivante :

$$\chi^T (bb^T - Ac)\chi = 0 \text{ , où } \chi = (x, y, 1).$$

Si l'on pose $bb^T - Ac = \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix}$, l'ellipse résultante de cette équation aura les caractéristiques suivantes : des demi-axes de taille $a_i = \sqrt{\frac{f''}{a'}}$ et $a_j = \sqrt{\frac{f''}{b'}}$ et une orientation $\theta = \frac{1}{2} arctan(\frac{2b}{c-a})$, avec $f'' = -f + \frac{d'^2}{a'} + \frac{e'^2}{c'}$ et :

$a' = a\cos^2(\theta) - 2b\cos(\theta)\sin(\theta) + c\sin^2(\theta)$
$b' = b\cos(2\theta) - \frac{c-a}{2}\sin(2\theta)$
$c' = a\sin^2(\theta) + 2b\cos(\theta)\sin(\theta) + c\cos^2(\theta)$
$d' = d\cos(\theta) - e\sin(\theta)$
$e' = d\sin(\theta) + e\cos(\theta)$

Annexe E : Optimisation de notre traqueur de gestes

a) Optimisation de nos mesures

Un certain nombre d'optimisations ont été apportées aux mesures et détections de notre traqueur de gestes afin de les rendre plus robustes, mais également et surtout plus rapides. En effet, il est possible de profiter de la nature des cibles, mais aussi de la nature dynamique du suivi, pour réduire fortement le temps de calcul de certaines mesures.

La nature des cibles, pour commencer, nous permet de conditionner certains calculs par les résultats d'une première détection. Par exemple, aucun visage ne sera détecté dans une zone de l'image ne correspondant à aucun *blob* peau, on peut donc réduire la recherche du premier à des zones définies autour du second ce qui permet un gain considérable en temps de calcul par rapport à une recherche sur les images complètes (gauche et droite) à chaque pas. De plus, il est possible d'utiliser certaines connaissances *a priori* pour diminuer la taille des vecteurs d'état des cibles. C'est par exemple ce qui nous a poussé à modéliser dès le départ la tête par une sphère de taille prédéterminée. Les mains, pour leur part, ne peuvent pas se définir de manière aussi restrictives, car l'utilisateur doit être aussi libre possible de ses mouvements et par conséquent libre par exemple de fermer le poing. Mais une chose reste relativement constante en elles indépendamment de l'utilisateur et de la forme qu'il donne à sa main : il s'agit de leur volume. Connaissant ce dernier, il est par conséquent possible de réduire le vecteur d'état des mains à 8 paramètres : $(X, Y, Z\ ,\ \theta_x, \theta_y, \theta_z\ ,\ \sigma_x, \sigma_y)$. σ_x et σ_y représentent alors les tailles des demis-axes « visibles », c'est à dire dans un plan parallèle aux caméras. On a alors $\sigma_z = \dfrac{volume}{(8.\sigma_x.\sigma_y)}$ représentant la demi-profondeur approximative de l'ellipsoïde. En plus de gagner une dimension sur deux cibles, cette méthode nous permet d'effectuer un contrôle de taille sur les *blobs* détectés, connaissant les limites de taille d'une main, ce qui permet un gain en qualité des détections. Une dernière connaissances *a priori* que nous pouvons exploiter est le fait que nos cibles sont liées : une main ne se trouvera par exemple jamais à plus de 1,5 mètres de la tête. Cela nous permet encore une fois de filtrer certaines fausses détections.

La dynamique des cibles, ensuite, peut être exploitée sous forme de zones de recherches définies par le résultat précédent. En effet, la dynamique des cibles est déjà définie dans notre filtre particulaire à travers les écart-types μ déterminés au départ pour la génération des particules selon la dynamique $p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$. Il est alors facile de l'utiliser pour déterminer la taille d'une zone 3D, centrée sur la position de la cible au pas précédent $\mathbf{x}_{t-1}^i$, dans laquelle seront effectuées nos mesures. Le gain en temps de calcul de cette méthode est d'autant plus élevé que la dynamique est faible et par conséquent que le module est rapide.

Enfin, nous définissons un intervalle de fonctionnement de notre module de suivi : entre 1 mètre et 3,5 mètres dans la configuration couramment utilisée. Cet intervalle correspond à la visibilité des cibles du point de vue du robot : un homme trop proche des caméras n'entrera plus dans leur champs de vue. De même, un utilisateur trop éloigné du robot ne permettra plus au suivi un fonctionnement normal, comment en effet suivre des mains qui ne représentent plus que

quelques pixels. Cette seconde limite est évidemment dépendante de la résolution des images acquises.

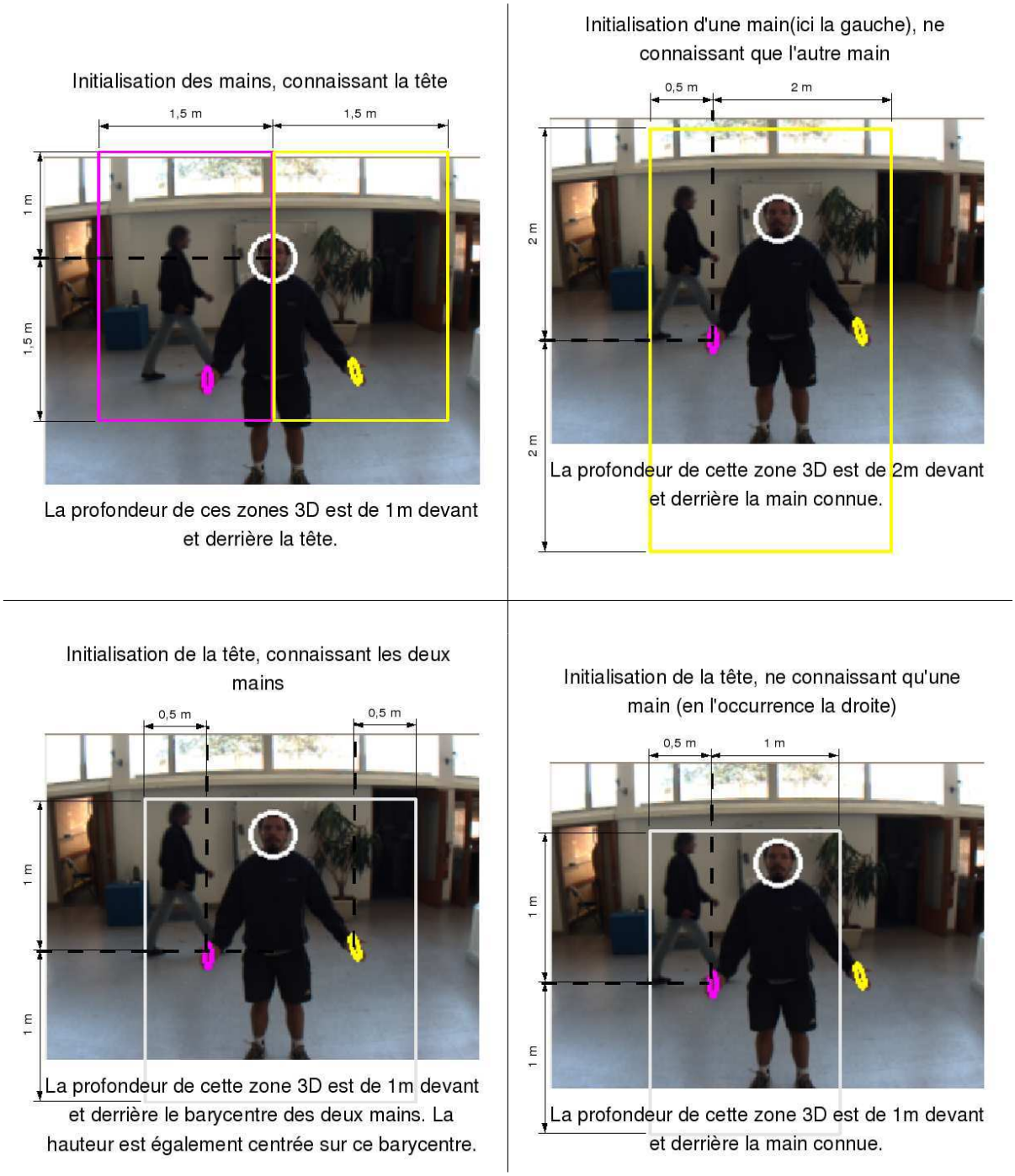


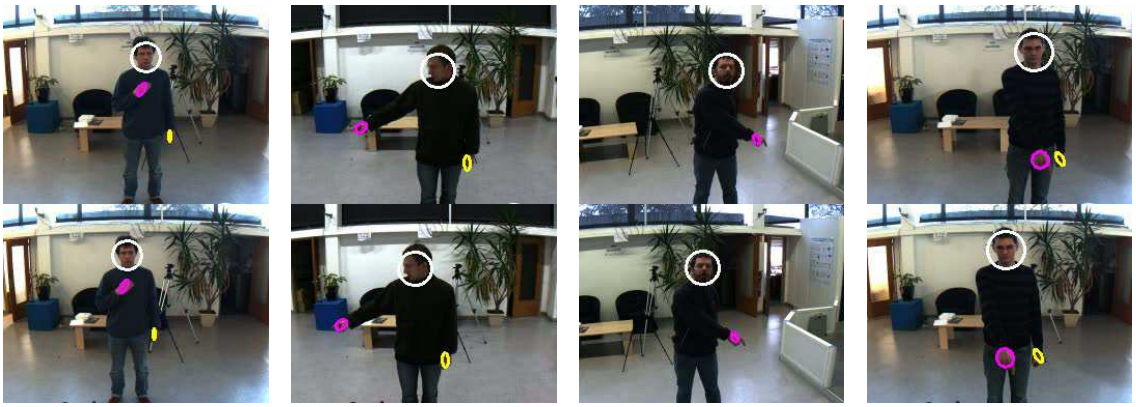
FIG. IV.10: Définition des zones 3D de réinitialisation des cibles en fonction de celle encore suivies.

b) Optimisation de l'algorithme général : phase d'initialisation et gestion des pertes de cible

La phase d'initialisation d'un algorithme se révèle souvent délicate quand il n'existe pas, comme c'est notre cas, de données *a priori* sur la position des cibles au démarrage. Une possibilité est d'initialiser le filtre de manière en considérant les particules comme des variables indépendantes identiquement distribuées, puis à laisser le filtre se charger, grâce à une évaluation suivie d'une nouvelle génération et aux particules générées selon les détections, de se stabiliser sur les cibles. C'est la méthode prévue théoriquement par l'algorithme SIR, mais elle a l'inconvénient de risquer d'initialiser le filtre sur une autre cible que celle voulue : initialisation d'une main sur un morceau de porte, ou plus simplement initialisation de la main gauche de l'utilisateur sur la main droite et inversement, etc. Ce dernier exemple est d'ailleurs plus problématique qu'il n'y parait puisque, bien que cela ne poserait aucun problème direct au suivi, une reconnaissance de geste basée sur ces données serait complètement faussée. Pour notre part, nous avons choisi d'utiliser une détection de visage (conditionnée par la présence d'un *blob* peau) afin d'initialiser le suivi à partir de la tête. L'hypothèse restrictive qui en découle est qu'à l'initialisation, l'utilisateur doit apparaître seul et de face sur les images. On tente ensuite d'initialiser les mains dans une zones 3D définie par la tête (voir figure IV.10) en partant d'une dernière hypothèse qui consiste à dire que les mains de l'utilisateur ne sont pas croisées à ce stade.

Une fois l'algorithme lancé, il est important de savoir détecter une dérive de ce dernier ou tout simplement une sortie du champs de vue de la cible. C'est pourquoi à l'étape 8 de notre algorithme (décrit dans la table II.2) a été inséré une étape d'évaluation de l'estimée. Si ce poids se trouve être inférieur à un seuil ω_{se} la cible est déclarée morte (elle n'est plus suivie) et sort par conséquent du cadre de l'algorithme. Mais détecter la perte de suivie d'une cible impose de pouvoir la réinitialiser dès que possible. Cette réinitialisation partielle (puisqu'il s'agit de réinitialiser un seul filtre et non l'ensemble de l'algorithme) est effectuée en respectant la dynamique de la cible à travers plusieurs niveaux de réinitialisation. Le premier de ces niveaux est utilisé pour réinitialiser une cible perdue un pas auparavant et consiste à doubler la zone 3D de recherche normalement définie pour le calcul des mesures en la centrant sur la position de la cible au dernier pas effectué avec succès $\mathbf{x}_{t-2}^{i}$. De la même manière, le second niveau de réinitialisation utilise une zone encore élargie suivant la dynamique. Le dernier niveau de réinitialisation, utilisé pour une cible perdue depuis un trop grand laps de temps mais aussi pour l'initialisation globale, considère qu'il n'y a plus de connaissance sur le passé de la cible et qu'il faut par conséquent la réinitialiser à partir des données actuelles du traqueur. La figure IV.10 montre les zones 3D de réinitialisation telles qu'elles sont définies en fonction des cibles encore suivies.

Annexe F :
Ensemble des gestes appris et reconnus par notre système.

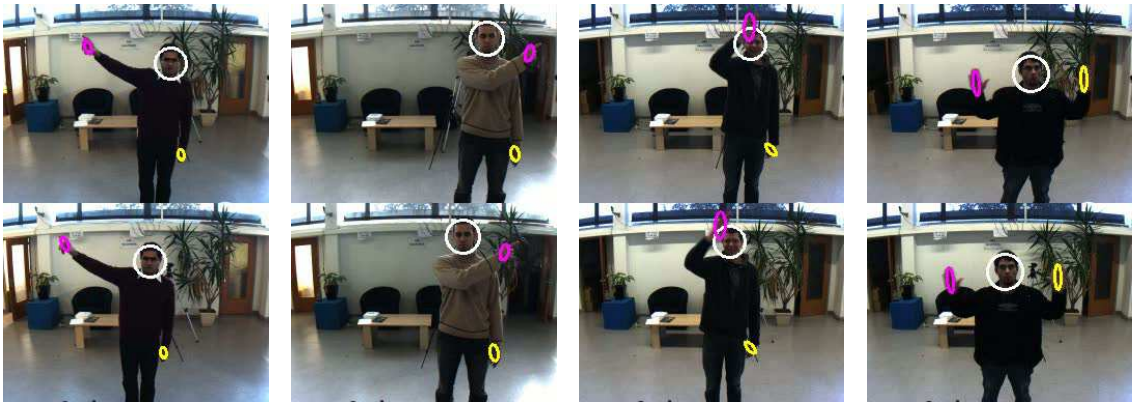


(a) Geste « je me présente »

(b) Geste « pointage bas droite »

(c) Geste « pointage bas gauche »

(d) Geste « pointage devant »

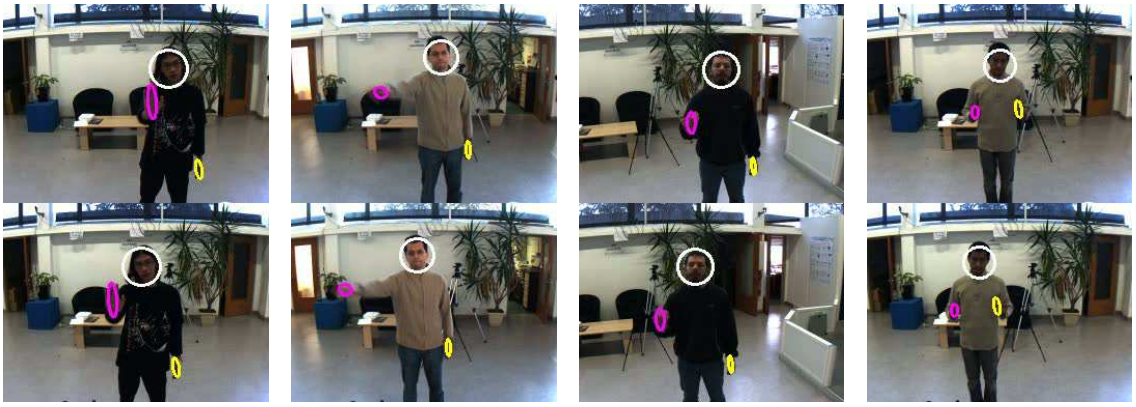


(e) Geste « pointage haut droite »

(f) Geste « pointage haut gauche »

(g) Geste « ohé (une main) »

(h) Geste « ohé (deux mains) »



(i) Geste « stop »

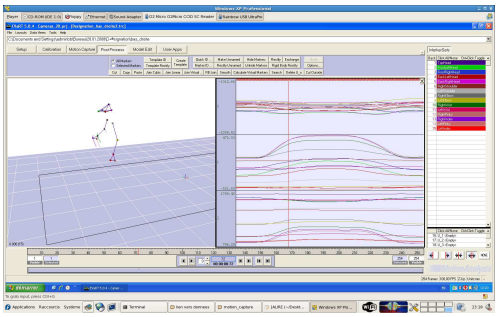
(j) Geste « va-t-en »

(k) Geste « viens vers moi (une main) »

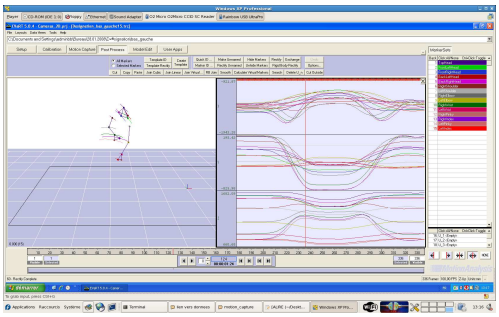
(l) Geste « viens vers moi (deux mains) »

FIG. IV.11: Extrais de la base de donnée de reconnaissance gestes illustrant l'ensemble des gestes appris et reconnus par notre système.

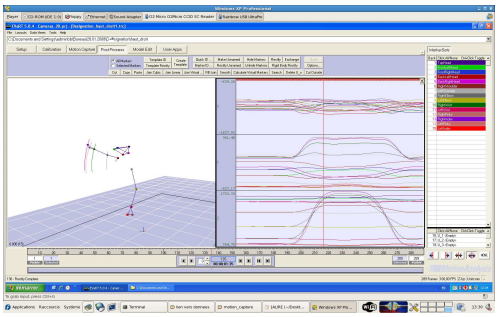
Annexe G : Exemples de gestes acquis avec un système commercial de capture de mouvements.



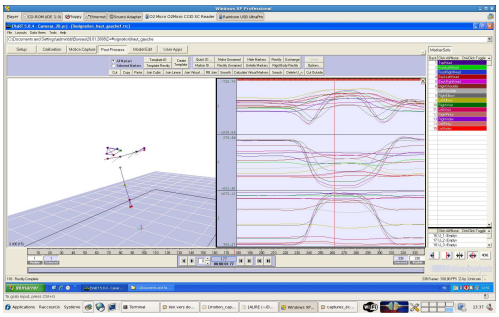
(a) Geste « pointage bas droite »



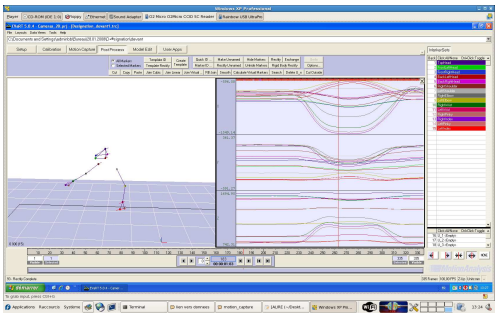
(b) Geste « pointage bas gauche »



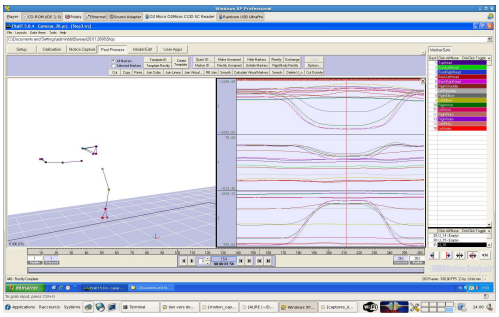
(c) Geste « pointage haut droite »



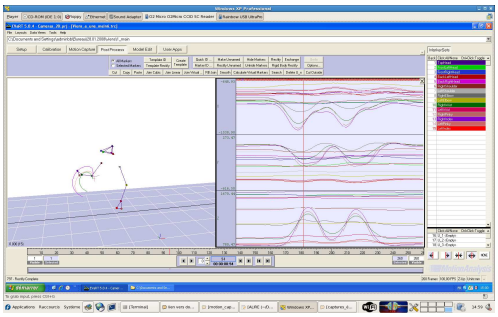
(d) Geste « pointage haut gauche »



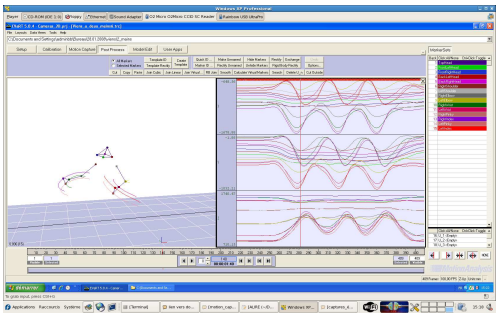
(e) Geste « pointage devant »



(f) Geste « stop »



(g) Geste « viens vers moi (une seule main) »



(h) Geste « viens vers moi (deux mains) »

FIG. IV.12: Exemples de gestes acquis avec un système commercial de capture de mouvements.

Table des figures

Liste des tableaux

Bibliographie

[Aggarwal and Cai, 1999] Aggarwal, J. and Cai, Q. (1999). Human motion analysis : A review. *Computer Vision and Image Understanding (CVIU'99)*, 73(3) :428–440.

[Aherne et al., 1997] Aherne, F., Thacker, N., and Rockett, P. (1997). The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 32(4) :1–7.

[Alami et al., 1998] Alami, R., Chatila, R., Fleury, S., and Ingrand, F. (1998). An architecture for autonomy. *International Journal of Robotic Research (IJRR'98)*, 17(4) :315–337.

[Antoniol et al., 1993] Antoniol, G., Cattoni, R., Cettolo, M., and Federico, M. (1993). Robust speech understanding for robot telecontrol. In *In Proceedings of the 6th International Conference on Advanced Robotics*, pages 205–209.

[Argentieri, 2006] Argentieri, S. (2006). *Conception d'un capteur sonore pour la localisation de source en robotique mobile*. PhD thesis, Université Paul Sabatier de Toulouse.

[Arriaga et al., 2003] Arriaga, H., Sucar, L., Mendoza, C., and Vargas, B. (2003). Visual recognition of dynamic gestures applied to command mobile robots. In *Int. Symp. on Robot and Human Interactive Communication (ROMAN'03)*, San Francisco, USA.

[Arulampalam et al., 2002] Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *Trans. on Signal Processing*, 2(50) :174–188.

[Asteriadis et al., 2008] Asteriadis, S., Tzouveli, P., Karpouzis, K., and Kollias, S., editors (2008). *Estimation of behavioral user state based on eye gaze and head pose - application in an e-learning environment*. Multimedia Tools and Applications. Springer.

[Azad et al., 2007] Azad, P., Ude, A., Asfour, T., and Dillman, R. (2007). Stereo-based markerless human motion capture for humanoid robot systems. In *Int. Conf. on Robotics and Automation (ICRA'07)*, Roma, Italy.

[Baggia et al., 1992] Baggia, P., Gerbino, E., Giachin, E., and Rullent, C. (1992). Real-time linguistic analysis for continuous speech understanding. In *ANLP*, pages 33–39.

[Balh et al., 1983] Balh, L., Jelinek, F., and Mercer, L. (1983). A maximum likelihood approach to continuous speech recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-5, pages 179–190.

[Bar-Shalom and Jaffer, 1998] Bar-Shalom, Y. and Jaffer, A. (1998). *Tracking and data association*. Academic Press, San Diego, USA.

[Bardet et al., 2009] Bardet, F., Chateau, T., and Ramasasan, D. (2009). Illumination aware MCMC particle filter for long-term outdoor multi-object simultaneous tracking and classification. In *Int. Conf. on Computer Vision (ICCV'09)*, Kyoto, Japan.

[Baum, 1972] Baum, L. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. In *Inequalities*, number 3, pages 1–8.

[Baum and Petrie, 1966] Baum, L. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. In *Annals of Mathematical Statistics (37)*, pages 1554–1563.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF : Speeded-Up Robust Features. In *European Conf. on Computer Vision (ECCV'06)*, pages 404–417, Graz, Austria.

[Beautemps et al., 2007] Beautemps, D., Girin, L., Aboutabit, N., Bailly, G., Besacier, L., Breton, G., Burger, T., Caplier, A., Cathiard, M.-A., Clarke, J., Elisei, F., Govokhina, O., Jutten, C., Le, V. B., Marthouret, M., Mancini, S., Mathieu, Y., Perret, P., Rivet, B., Sacher, P., Savariaux, C., Schmerber, S., Serignat, J.-F., Tribout, M., and Vidal, S. (2007). TELMA : Telephony for the Hearing-Impaired People. From Models to User Tests. In *ASSISTH'2007*, Toulouse France.

[Benewitz et al., 2008] Benewitz, M., Axenbeck, T., Behnke, S., and Burgard, W. (2008). Robust recognition of complex gestures for natural human-robot interaction. In *Workshop on Interactive Robot Learning*, Zurich, Switzerland.

[Benitez et al., 2000] Benitez, M., Rubio, A., and Torre, A. (2000). Different confidence measures for word verification in speech recognition. (32) :79–94.

[Bernier et al., 2009] Bernier, O., Cheung-Mon-Chan, P., and Bouguet, A. (2009). Fast nonparametric belief propagation for real-time stereo articulated body tracking. *Computer Vision and Image Understanding (CVIU'09)*, 113 :29–47.

[Bischoff and Graefe, 2004] Bischoff, R. and Graefe, V. (2004). HERMES - a versatile personal robotic assistant. *IEEE*, 92 :1759–1779.

[Boite, 2000] Boite, R. (2000). *Traitement de la parole*. Presses polytechniques et universitaires romandes.

[Boyen and Koller, 1998] Boyen, X. and Koller, D. (1998). Tractable inference for complex stochastic processes. *fourteenth Annual Conference on Uncertainty and Artificial Intelligence (UAI)*.

[Boyen and Koller, 1999] Boyen, X. and Koller, D. (1999). Exploiting the architecture of dynamic systems. *sixteenth national conference on artificial intelligence (AAAI)*.

[Brèthes, 2005] Brèthes, L. (2005). *Suivi visuel par filtrage particulaire. Application à l'interaction Homme-Robot*. PhD thesis, Université Paul Sabatier de Toulouse.

[Bretzner et al., 2002] Bretzner, L., Laptev, I., and Lindeberg, T. (2002). Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Int. Conf. on Automatic Face and Gesture Recognition (FGR'02)*, pages 405–410, Washington, USA.

[Brochard et al., 2009] Brochard, R., Burger, B., Herbulot, A., and Lerasle, F. (2009). Measuring gaze orientation for human-robot interaction. In *Int. Workshop during IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN'09)*, Toyama, Japan.

[Brown and Tian, 2002] Brown, L. and Tian, Y.-L. (2002). Comparative study of coarse head pose estimation. In *Workshop on Motion and Video Computing*, pages 125–130.

[Burger et al., 2008a] Burger, B., Ferrané, I., and Lerasle, F. (2008a). Multimodal interaction abilities for a robot companion. In *Int. Conf. on Computer Vision Systems (ICVS'08)*, pages 549–558, Santorini, Greece.

[Burger et al., 2010] Burger, B., Ferrané, I., and Lerasle, F. (2010). Two-handed gesture recognition and fusion with speech to command a robot. *Submitted to journal of Autonomous Robots*.

[Burger et al., 2009a] Burger, B., Ferrané, I., and Lerasle, F. (2009a). Towards multimodal interface for interactive robots : challenges and robotic systems description. In *International Journal of Advanced Robotic Systems*. IN-TECH, http ://intechweb.org/.

[Burger et al., 2009b] Burger, B., Infantes, G., Ferrané, I., and Lerasle, F. (2009b). Dbn versus hmm for gesture recognition in human-robot interaction. In *Int. workshop on Electronics, Control, Modelling, Measurement and Signals (ECMS'09)*, pages 59–65, Mondragon, Spain.

[Burger et al., 2009c] Burger, B., Lerasle, F., and Ferrané, I. (2009c). Evaluations of embedded modules dedicated to multimodal human-robot interaction. In *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN'09)*, Toyama, Japan.

[Burger et al., 2008b] Burger, B., Lerasle, F., Ferrané, I., and Clodic, A. (2008b). Mutual assistance between speech and vision for human-robot interaction. In *Int. Conf. on Intelligent Robots and Systems (IROS'08)*, Nice, France.

[Burges, 1998] Burges, C. (1998). A tutorial on support vector machines for pattern recognition. volume 2, pages 121–167.

[Cadoz, 1994] Cadoz, C. (1994). Le geste canal de communication homme/machine. la communication « instrumentale ». *Technique et science informatique*, 13(1) :31–61.

[Chase, 1997] Chase, L. (1997). Word and acoustic confidence annotation for large vocabulary speech recognition. In *European Conference on Speech Communication Technology*, pages 815–818.

[Chen et al., 2003] Chen, F., Fu, C., and C.L, H. (2003). Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing (IVC'03)*, 21(8) :745–758.

[Chong et al., 2000] Chong, S., Kuno, Y., Shimada, N., and Shirai, Y. (2000). Human-robot interface based on speech understanding assisted by vision. In *ICMI*, pages 16–23.

[Clodic et al., 2005] Clodic, A., Montreuil, V., Alami, R., and Chatila, R. (2005). A decisional framework for autonomous robots interacting with humans. In *IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*.

[Cootes et al., 2000] Cootes, T., Walker, K., and Taylor, C. (2000). View-based active appearance models. In *Int. Conf. on Automatic Face and Gesture Recognition (FG'00)*, Grenoble, France.

[Corradini and Gross, 2000] Corradini, A. and Gross, H. (2000). Camera-based gesture recognition for robot control. In *Int. Joint Conf. on Neural Networks (IJCNN'00)*, Roma, Italy.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support vector networks. In *Machine Learning*, pages 273–297.

[Davis, 1971] Davis, F. (1971). *Inside Intuition-What we know about non-verbal communication*. McGraw-Hill Book Co.

[Dean and Kanazawa, 1990] Dean, T. and Kanazawa, K. (1990). A model for reasoning about persistence and causation. In *Computational Intelligence*, volume 5(3), pages 142–150.

[Delgado and Araki, 2005] Delgado, R. L.-C. and Araki, M. (2005). *Spoken, multilingual and multimodal dialogues systems - Development ans assessment*. Wiley Editions.

[Derpanis, 2004] Derpanis, K. (2004). A review of vision-based hand gestures. Technical report, Center for Vision Research, York University.

[Deutscher and Reid, 2005] Deutscher, J. and Reid, I. (2005). Articulated body motion capture by stochastic search. *Int. Journal of Computer Vision (IJCV'05)*, 21(3) :185–205.

[Doucet et al., 2001] Doucet, A., De Freitas, N., and Gordon, N. J. (2001). *Sequential Monte Carlo Methods in Practice*. Series Statistics For Engineering and Information Science. Springer-Verlag, New York.

[Doucet et al., 2000] Doucet, A., Godsill, S. J., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3) :197–208.

[Du et al., 2006] Du, Y., Chen, F., Xu, W., and Li, Y. (2006). Recognizing interaction activities using dynamic bayesian network. In *Int. Conf. on Pattern Recognition (ICPR'06)*, pages 618–621, Hong-Kong.

[Duong et al., 2005] Duong, T., Bui, H., Phung, D., and Venkatesh, S. (2005). Activity recognition and abnormality detection with the switching hidden semi-markov model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 838–845, Washington, DC, USA. IEEE Computer Society.

[Eklund, 2000] Eklund, R. (2000). Crosslinguistic disfluency modeling : a comparative analysis of swedish and tok pisin human-human ATIS dialogues. In *ICSLP*, volume 2, pages 991–994, Beijing, China.

[Eklund, 2004] Eklund, R. (2004). *Disfluency in Swedish human-human and human-machine travel booking dialogues*. PhD thesis, Linköping University, Sweden.

[Erol et al., 2007] Erol, A., Bebis, G., Nicolescu, M., Boyle, R., and Twombly, X. (2007). Vision-based hand pose estimation : a review. *Computer Vision and Image Understanding (CVIU'07)*, 108 :52–73.

[Fels and Hinton, 1997] Fels, S. and Hinton, G. (1997). Glove-talk II : A neural network interface which maps gestures to parallel format speech synthesizer controls. *Trans. on Neural Networks*, 9(1) :205–212.

[Fidaleo and Medioni, 2007] Fidaleo, D. and Medioni, G. (2007). Model-assisted 3d face reconstruction from video. In *AMFG07*, pages 124–138.

[Fong et al., 2003] Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems (RAS'03)*, 42 :143–166.

[Fontmarty et al., 2007] Fontmarty, M., Lerasle, F., and Danès, P. (2007). Data fusion within a modified annealed particle filter dedicated to human motion capture. In *Int. Conf. on Intelligent Robots and Systems (IROS'07)*, pages 3391–3396, San Diego, USA.

[Forney, 1973] Forney, G. (1973). The viterbi algorithm. *Proceedings of the IEEE (61)*, pages 268–278.

[Gabsdil and Lemon, 2004] Gabsdil, M. and Lemon, O. (2004). Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems. In *Meeting of the Association for Computational Linguistics (ACL'04)*, pages 343–350, Barcelona, Spain.

[Galliano et al., 2005] Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K., Bonastre, J., and Gravier, G. (2005). The ESTER phase II evaluation campaign for the rich transcription of french broadcast news. In *Interspeech/Eurospeech*, Lisbon, Portugal.

[Gee and Cipolla, 1994] Gee, A. H. and Cipolla, R. (1994). Determining the gaze of faces in images. *Image and Vision Computing (IVC'94)*, 12 :639–647.

[Georghiades et al., 2001] Georghiades, A., Belhumeur, P., and Kriegman, D. (2001). From few to many : Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6) :643–660.

[Ghidary et al., 2002] Ghidary, S., Nakata, Y., Saito, H., Hattori, M., and Takamori, T. (2002). Multi-modal interaction of human and home robot in the context of room map generation. *Autonomous Robots (AR'02)*, 13(2) :169–184.

[Gorostiza et al., 2006] Gorostiza, J., Barber, R., Khamis, A., and Malfaz, M. (2006). Multimodal human-robot interaction framework for a personal robot. In *Int. Symp. on Robot and Human Interactive Communication (RO-MAN'06)*, pages 39–44, Hatfield, UK.

[Hanafiah et al., 2004] Hanafiah, Z., Yamazaki, C., Nakamura, A., and Kuno, Y. (2004). Human-robot speech interface understanding inexplicit utterances using vision. In *CHI 2004*, pages 1321–1324, Vienna, Austria.

[Hasanuzzaman et al., 2004] Hasanuzzaman, M., Ampornaramveth, V., Zhang, T., Bhuiyan, M., Shirai, Y., and Ueno, H. (2004). Real-time vision-based gesture recognition for human robot interaction. In *Int. Conf. on Robotics and Biomimetics*, Shenyang, China.

[Hasanuzzaman et al., 2007] Hasanuzzaman, M., Zhang, T., Ampornaramveth, V., and Ueno, H. (2007). Adaptive visual gesture recognition using a knowledge-based software platform. *Robotics and Autonomous Systems (RAS'07)*, 55(8) :643–657.

[Heinzmann and Zelinsky, 1999] Heinzmann, J. and Zelinsky, A. (1999). Robust real-time face tracking and gesture recognition. In *Int. Joint Conf. on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden.

[Heracleous et al., 2009] Heracleous, P., Aboutabit, N., and Beautemps, D. (2009). Lip shape and hand position fusion for automatic vowel recognition in Cued Speech for French. *IEEE Signal Processing Letters*, pages 1–4.

[Huang et al., 2002] Huang, Y., Huang, T., and Niemann, H. (2002). Two-handed gesture tracking incorporating template warping with static segmentation. In *Int. Conf. on Automatic Face and Gesture Recognition (FGR'02)*, pages 275–280, Washington, USA.

[Hüwel and Wrede, 2006] Hüwel, S. and Wrede, B. (2006). Spontaneous speech understanding for robust multi-modal human-robot communication. In *ACL*.

[Infantes, 2006] Infantes, G. (2006). *Apprentissage de modèles de comportements pour le contrôle d'exécution et la planification robotique*. PhD thesis, Université Paul Sabatier de Toulouse.

[Infantes et al., 2006] Infantes, G., Ingrand, F., and Ghallab, M. (2006). Learning behaviors models for robot execution control. *ICAPS*, pages 394–397.

[Isard and Blake, 1998a] Isard, M. and Blake, A. (1998a). CONDENSATION – conditional density propagation for visual tracking. *Int. Journal on Computer Vision (IJCV'98)*, 29(1) :5–28.

[Isard and Blake, 1998b] Isard, M. and Blake, A. (1998b). I-CONDENSATION : Unifying low-level and high-level tracking in a stochastic framework. In *European Conf. on Computer Vision (ECCV'98)*, pages 893–908, Freibourg, Germany.

[Isard and Blake, 2001] Isard, M. and Blake, A. (2001). BraMBLe : a bayesian multiple blob tracker. In *Int. Conf. on Computer Vision (ICCV'01)*, pages 34–41, Vancouver, Canada.

[J. Goulian, 2003] J. Goulian, J.Y. Antoine, F. P. (2003). How nlp techniques can improve speech understanding : Romus – a robust chunk based message understanding system using link grammars. In *European Conference on Speech Communication and Technology (EUROSPEECH)*, Geneva, Switzerland.

[Jeong et al., 2002] Jeong, M., Kuno, Y., Shimada, N., and Shirai, Y. (2002). Two-hand gesture recognition using coupled switching linear model. In *Int. Conf. on Pattern Recognition (ICPR'02)*, Quebec, Canada.

[Jiang, 2005] Jiang, H. (2005). Confidence measures for speech recognition : A survey. *Speech Communication*, 45 :455–470.

[Joly, 2007] Joly, P. (2007). Descriptions des séquences d'images. In Gros, P., editor, *L'indexation multimédia - Description et recherche automatiques*, Traité IC2, série Traitement du signal et de l'image, pages 119–136. Hermès.

[Jurafsky and Martin, 2000] Jurafsky, D. and Martin, J. (2000). *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall.

[Just et al., 2004] Just, A., Marcel, S., and Bernier, O. (2004). Hmm and iohmm for the recognition of mono and bi-manual 3d hand gestures. In *British Machine Vision Conference (BMVC'04)*, London, UK.

[Kahol and Kahol, 2003] Kahol, K. and Kahol, K. (2003). Gesture segmentation in complex motion sequences. In *Proceedings IEEE International Conference on Image Processing*, pages 105–108.

[Kamppari and Hazen, 2000] Kamppari, S. and Hazen, T. (2000). Word and phone level acoustic confidence scoring. In *International Conference on Acoustics, Speech and Signal Processing*, pages 1799–1802.

[Kanungo et al., 2002] Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., and Wu, A. (2002). An efficient k-means clustering algorithm : analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :881–892.

[Kendon, 1980] Kendon, A. (1980). *The relation between verbal and non-verbal communication*, chapter Gesticulation and speech : Two aspects of the process of the utterance, pages 207–227. Ed. Hague : Mouton.

[Khan et al., 2005] Khan, Z., Balch, T., and Dellaert, F. (2005). MCMC-based particle filtering for tracking a variable number of interacting targets. *Trans. on Pattern Analysis Machine Intelligence (PAMI'05)*, 27(11) :1805–1818.

[Kobayashi et al., 2007] Kobayashi, A., Onoe, K., Homma, S., Sato, S., and Imai, T. (2007). Word error rate minimization using an integrated confidence measure. In *IEICE TRANSACTIONS on Information and Systems*, pages 835–843.

[Kohonen, 1984] Kohonen, T. (1984). Self-organisation and associative memory. *Springer-Verlag*.

[Koller and Fratkina, 1998] Koller, D. and Fratkina, R. (1998). Using learning for approximation in stochastic processes.

[Kulic et al., 2007] Kulic, D., Takano, W., and Nakamura, Y. (2007). Representability of human motions by factorial hidden markov models. In *Int. Conf. on Intelligent Robots and Systems (IROS'07)*, pages 2388–2393, San Diego, USA.

[Lee et al., 2001] Lee, A., Kawahara, T., and Shikano, K. (2001). Julius — an open source real-time large vocabulary recognition engine. In *European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1691–1694, Aalborg, Denmark.

[Linarès et al., 2007] Linarès, G., Nocéra, P., Massonié, D., and Matrouf, D. (2007). The lia speech recognition system : from 10xrt to 1xrt. In *Lecture Notes in Computer Science, 4629 LNAI*, pages 302–308.

[Lindeberg, 1998] Lindeberg, T. (1998). Feature detection with automatic scale selection. *Int. Journal of Computer Vision (IJCV'98)*, 30(2) :77–116.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision (IJCV'04)*, 60(2) :91–110.

[Maas et al., 2006] Maas, J., Spexard, T., Fritsch, J., Wrede, B., and Sagerer, G. (2006). BIRON, what's the topic ? a multi-modal topic tracker for improved human-robot interaction. In *Int. Symp. on Robot and Human Interactive Communication (RO-MAN'06)*, Hatfield, UK.

[Meyer, 2002] Meyer, G. Mulligan, J. (2002). Continuous audio-visual digit recognition using decision fusion. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'02)*, volume 1, pages 305–308, Orlando, USA.

[Moeslund et al., 2006] Moeslund, T., Hilton, A., and Kruger, V. (2006). A survey of advanced vision-based human motion capture and analysis. *Computer Vision and Image Understanding (CVIU'06)*, 104 :174–192.

[Morimoto and Mimica, 2005] Morimoto, C. and Mimica, M. (2005). Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding (CVIU'05)*, 98 :4–24.

[Murphy-Chutorian and Trivedi, 2008a] Murphy-Chutorian, E. and Trivedi, M. (2008a). Head pose estimation in computer vision : A survey. *Trans. on Pattern Analysis Machine Intelligence (PAMI'08)*.

[Murphy-Chutorian and Trivedi, 2008b] Murphy-Chutorian, E. and Trivedi, M. (2008b). Head pose estimation in computer vision : a survey. *Trans. on Pattern Analysis and Machine Intelligence (PAMI'08)*, 31(4) :607–626.

[Nickel and Stiefelhagen, 2006] Nickel, K. and Stiefelhagen, R. (2006). Visual recognition of pointing gestures for human-robot interaction. *Image and Vision Computing (IVC'06)*, 3(12) :1875–1884.

[Nummiaro et al., 2003] Nummiaro, K., Koller-Meier, E., and Gool, L. V. (2003). An adaptative color-based particle filter. *Image and Vision Computing (IVC'03)*, 21(90) :90–110.

[Oliver et al., 2000] Oliver, N., Rosario, B., and Pentland, A. (2000). A bayesian computer vision system for modeling human interactions. *Trans. on Pattern Analysis Machine Intelligence (PAMI'00)*, 22(8) :831–843.

[Ong and Ranganath, 2005] Ong, S. and Ranganath, S. (2005). Automatic sign language analysis : A survey and the future beyond lexical meaning. *Trans. on Pattern Analysis Machine Intelligence (PAMI'05)*, 27(6) :873–891.

[OpenCv, 2008] OpenCv (2008). Opencv (open source computer vision) library, url=http ://opencv.willowgarage.com/wiki/.

[O'Shaughnessy, 1987] O'Shaughnessy, D. (1987). *Speech Communication : Human and Machine*. Addison-Wesley.

[Park et al., 2005] Park, H., Kim, E., Jang, S., and Park, S. (2005). HMM-based gesture recognition for robot control. In *Iberian Conf. on Pattern Recognition and Image Analysis (IbPRIA'05)*, Estoril, Portugal.

[Pavlovic et al., 1997] Pavlovic, V., Sharma, R., and Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction : A review. *Trans. On Pattern Analysis and Machine Intelligence (PAMI'97)*, 19(7) :677–695.

[Pérennou and de Calmès, 2000] Pérennou, G. and de Calmès, M. (2000). MHATLex : Lexical resources for modelling the french pronunciation. In *Int. Conf. on Language Resources and Evaluations*, pages 257–264, Athens, Greece.

[Pérez et al., 2004] Pérez, P., Vermaak, J., and Blake, A. (2004). Data fusion for visual tracking with particles. *Proc. IEEE*, 92(3) :495–513.

[Pérez et al., 2002] Pérez, P., Vermaak, J., and Gangnet, M. (2002). Color-based probabilistic tracking. In *European Conf. on Computer Vision (ECCV'02)*, pages 661–675, Berlin, Germany.

[Potamianos et al., 2009] Potamianos, G., Lamel, L., Wölfel, M., Huang, J., Marcheret, E., Barras, C., Zhu, X., McDonough, J., Hernando, J., Macho, D., and Nadeu, C. (2009). *Computers*

in the Human Interaction Loop, chapter Automatic speech recognition, pages 44–60. Springer.

[Potamianos et al., 2003] Potamianos, G., Neti, C., Gravier, G., Garg, A., Member, S., Senior, A. W., and Member, S. (2003). Recent advances in the automatic recognition of audiovisual speech. In *Proc. IEEE*, pages 1306–1326.

[Potamianos et al., 2004] Potamianos, G., Neti, C., Luettin, J., and Matthews, I. (2004). Audio-visual automatic speech recognition : An overview. In *Issues in Visual and Audio-visual Speech Processing*. MIT Press.

[Prodanov and Drygajlo, 2003a] Prodanov, P. and Drygajlo, A. (2003a). Bayesian networks for spoken dialogue managements in multimodal systems of tour-guide robots. In *European Conf. on Speech Communication and Technology (EUROSPEECH'03)*, pages 1057–1060, Geneva, Switzerland.

[Prodanov and Drygajlo, 2003b] Prodanov, P. and Drygajlo, A. (2003b). Multimodal interaction management for tour-guide robots using bayesian networks. In *Int. Conf. on Intelligent Robots and Systems (IROS'03)*, pages 3447–3452, Las Vegas, Canada.

[Provost and Fawcett, 2001] Provost, F. and Fawcett, T. (2001). Robust classification for imprecise environments. In *Machine Learning*, pages 203–231.

[Qu et al., 2007] Qu, W., Schonfeld, D., and Mohamed, M. (2007). Distributed bayesian multiple-target tracking in crowded environments using multiple collaborative cameras. *EURASIP Journal on Advances in Signal Processing*.

[Quek, 1994] Quek, F. K. H. (1994). « Toward a Vision-Based Hand Gesture Interface ». *Virtual Reality Software and Technology*.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *IEEE*, 77(2) :257–286.

[Rahim et al., 1997] Rahim, M., Lee, C.-H., and Juang, B.-H. (1997). Discriminative utterance verification for connected digits recognition. 5(3) :266–277.

[Richarz et al., 2006] Richarz, J., Martin, C., Scheidig, A., and Gross, H. (2006). There you go ! - estimating pointing gestures in monocular images for mobile robot instruction. In *Int. Symp. on Robot and Human Interactive Communication (RO-MAN'06)*, pages 546–551, Hartfield, UK.

[Rogalla et al., 2004] Rogalla, O., Ehrenmann, M., Zollner, R., Becher, R., and Dillman, R. (2004). *Advances in human-robot interaction*, volume 14, chapter Using gesture and speech control for commanding a robot. Springer-Verlag.

[Rose et al., 1995] Rose, R., Juang, B., and Lee, C. (1995). A training procedure for verifying string hypothesis in continuous speech recognition. In *International Conference on Acoustics, Speech and Signal Processing*, pages 281–284.

[San-Segundo et al., 2001] San-Segundo, R., Pellom, B., Hacioglu, K., and Ward, W. (2001). Confidence measures for spoken dialogue systems. In *International Conference on Acoustics, Speech and Signal Processing*.

[Schaaf and Kemp, 1997] Schaaf, T. and Kemp, T. (1997). Confidence measures for spontaneous speech recognition. In *International Conference on Acoustics, Speech and Signal Processing*, pages 875–878.

[Schwerdt and Crowley, 2000] Schwerdt, K. and Crowley, J. L. (2000). Robust face tracking using color. In *Int. Conf. on Face and Gesture Recognition (FGR'00)*, pages 90–95, Grenoble, France.

[Shimizu et al., 2006] Shimizu, M., Yoshizuka, T., and Miyamoto, H. (2006). A gesture recognition system using stereo vision and arm model fitting. In *Int. Conf. on Brain-Inspired Information Technology (BrainIT'06)*, Hibikino, Japan.

[Siegel, 2002] Siegel, M. E. A. (2002). Like : The Discourse Particle and Semantics. *Journal of Semantics*, 19(1) :35–71.

[Siegwart et al., 2003] Siegwart, R., Arras, O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., Jensen, B., Lorotte, A., Mayor, L., Meisser, M., Philippsen, R., Piguet, R., Ramel, G., Terrien, G., and Tomatis, N. (2003). Robox at expo 0.2 : a large scale installation of personal robots. *Robotics and Autonomous Systems (RAS'03)*, 42 :203–222.

[Skubic et al., 2004] Skubic, M., Perzanowski, D., Blisard, S., Schultz, A., and Adams, W. (2004). Spatial language for human-robot dialogs. *Journal of Systems, Man, and Cybernetics*, 2(34) :154–167.

[Sminchisescu and Triggs, 2003] Sminchisescu, C. and Triggs, B. (2003). Estimating articulated human motion with covariance scaled sampling. *Int. Journal of Robotics Research (IJRR'03)*, 6(22) :371–393.

[Smyth et al., 1997] Smyth, P., Heckerman, D., and Jordan, M. (1997). Probabilistic independence networks for hidden markov probability models. In *Neural Computation*, volume 9, pages 227–269.

[Stiefelhagen et al., 2004] Stiefelhagen, R., Fgen, C., Gieselmann, P., Holzapfel, H., Nickel, K., and Waibel, A. (2004). Natural human-robot interaction using speech, head pose and gestures. In *Int. Conf. on Intelligent Robots and Systems (IROS'04)*, pages 2422–2427, Sendai, Japan.

[Stolcke et al., 1997] Stolcke, A., König, Y., and Weintraub, M. (1997). Explicit word error minimization in n-best list rescoring. In *Eurospeech '97*, pages 163–166.

[Suk et al., 2008] Suk, H., Sin, B., and Lee, S. (2008). Robust modelling and recognition of hand gestures with dynamic bayesian network. In *Int. Conf. on Pattern Recognition (ICPR'08)*, Tampa, USA.

[Sukkar and Lee, 1996] Sukkar, R. and Lee, C.-H. (1996). Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition. 4(6) :420–429.

[Thayananthan et al., 2003] Thayananthan, A., Stenger, B., Torr, P., and Cipolla, R. (2003). Learning a kinematic prior for tree-based filtering. In *British Machine Vision Conf. (BMVC'03)*, volume 2, pages 589–598, Norwick, UK.

[Theobalt et al., 2002] Theobalt, C., Bos, J., Chapman, T., and Espinosa, A. (2002). Talking to godot : Dialogue with a mobile robot. In *Int. Conf. on Intelligent Robots and Systems (IROS'02)*, Lausanne, Switzerland.

[Triesch and Von der Malsburg, 2001] Triesch, J. and Von der Malsburg, C. (2001). A system for person-independent hand posture recognition against complex backgrounds. *Trans. on Pattern Analysis Machine Intelligence (PAMI'01)*, 23(12) :1449–1453.

[Trujillo-Romero and Devy, 2009] Trujillo-Romero, F. and Devy, M. (2009). Appearance-based and active 3d object recognition using vision. In *VISSAPP (1)*, pages 417–424.

[Valenti et al., 2008] Valenti, R., Sebe, N., and Gevers, T. (2008). Simple and efficient visual gaze estimation. In *Int. Conf. on Multimodal Interfaces (ICMI'08)*, Chania, Greece.

[Vallée et al., 2009] Vallée, M., Burger, B., Ertl, D., Lerasle, F., and Falb, J. (2009). Improving user interfaces of interactive robots with multimodality. In *Int. Conf. on Advanced Robotics (ICAR'09)*, Munich, Allemagne.

[Vapnik, 1979] Vapnik, V. (1979). Estimation of dependences based on empirical data (in russian). Nauka, Russia.

[Vapnik, 1995] Vapnik, V. (1995). The nature of statistical learning theory. In *Springer-verlag*, New York, USA.

[Vapnik, 1998] Vapnik, V. (1998). Statistical learning theory. In *John Wiley and Sons*, New York, USA.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'01)*, Hawaii.

[Waldherr et al., 2000] Waldherr, S., Thrun, S., and Romero, R. (2000). A gesture-based interface for human-robot interaction. *Autonomous Robots (AR'00)*, 9(2) :151–173.

[Wang et al., 2001] Wang, T.-S., yeung Shum, H., Xu, Y.-Q., and ning Zheng, N. (2001). Unsupervised analysis of human gestures.

[Wessel et al., 2001] Wessel, F., Schluter, R., Macherey, K., and Ney, H. (2001). Confidence measures for large vocabulary continuous speech recognition. 3(9) :288–298.

[Wu et al., 1999] Wu, L., Oviatt, S. L., and Cohen, P. R. (1999). Multimodal integration - a statistical view. *IEEE Transactions on Multimedia*, 1 :334–341.

[Yang et al., 2007] Yang, J., Park, A., and Lee, S. (2007). Gesture spotting and recognition for human-robot interaction. *Trans. on Robotics*, 23(2) :256–270.

[Yoshizaki et al., 2002] Yoshizaki, M., Kuno, Y., and Nakamura, A. (2002). Mutual assistance between speech and vision for human-robot interface. In *Int. Conf. on Intelligent Robots and Systems (IROS'02)*, pages 1308–1313, Lausanne, Switzerland.

[Young, 1994] Young, S. (1994). Detecting misrecognitions and out-ofvocabulary words. In *International Conference on Acoustics, Speech and Signal Processing*, pages 21–24.

[Young et al., 2006] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2006). The htk book v3.4. In *User Manual*, University of Cambridge's Engineering Department.

[Yu and Wu, 2004] Yu, T. and Wu, Y. (2004). Collaborative tracking of multiple targets. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, Washington, USA.

[Zhao, 2001] Zhao, L. (2001). *Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures*. PhD thesis, CIS, University of Pennsylvania.

[Zhou et al., 2009] Zhou, H., Yuan, Y., and Shi, C. (2009). Object tracking using SIFT features and mean shift. *Computer Vision and Image Understanding (CVIU'09)*, 113(3) :345–352.

[Ziegler et al., 2006] Ziegler, J., Nickel, K., and Stiefelhagen, R. (2006). Tracking of the articulated upper body on multi-view stereo image sequences. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, pages 774–781, New York, USA.

[Zieren et al., 2002] Zieren, J., Unger, N., and Akyol, S. (2002). Hands tracking from frontal view for vision-based gesture recognition. In *DAGM Symp.*, pages 531–539, Zurich, Switzerland.

[Zuriarrain et al., 2008] Zuriarrain, I., Lerasle, F., Arana-Arejolaleiba, N., and Devy, M. (2008). An mcmc-based particle filter for multiple person tracking. In *ICPR*, pages 1–4.

Table des matières