ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

THÈSE

pour obtenir le titre de

Docteur de l'École Normale Supérieure de Cachan
Spécialité : Informatique

Présentée par

Christophe COLLET

Capture et suivi du regard par un système de vision
dans le contexte de la communication homme-machine

Soutenue le 15 janvier 1999 *devant le jury composé de*

MM.	Joëlle	COUTAZ	Présidente
	Catherine	GARBAY	Rapporteurs
M.	Claude	LAURGEAU	
	Kevin	O'REGAN	Examinateurs
	Alain	FINKEL	(Directeur de thèse)
	Rachid	GHERBI	

Thèse préparée au sein du
Laboratoire Spécification et Vérification – LSV - CNRS / ENS Cachan
et du
Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur – LIMSI - CNRS

Table des matières

Remerciements

A l'origine de cette thèse, il y a une rencontre entre Alain Finkel et Rachid Gherbi, qui ont trouvé un intérêt commun pour la capture du regard. Cette coopération entre Alain en tant que directeur de thèse au LSV et Rachid co-encadrant au LIMSI, m'a permis de disposer d'un encadrement scientifique et pédagogique d'excellente qualité. Cette rencontre a été une vrai chance pour moi, et je ne les remercierais jamais assez pour tout ce qu'ils m'ont apporté au cours de cette thèse.

Merci à Catherine Garbay et Claude Laurgeau pour l'intérêt qu'ils ont porté à mes travaux et pour avoir bien voulu rapporter sur un manuscrit à peine terminé et rempli de fautes d'orthographe.

Merci à Joëlle Coutaz d'avoir accepté la présidence du jury et de s'être privée d'un voyage dans les îles pour participer à la soutenance.

Merci à Kevin O'Regan d'avoir apporté son regard de spécialiste en oculométrie dans le jury de cette thèse.

Quant aux divers personnes que j'ai côtoyées durant ces trois années et quelques mois au LSV et au LIMSI, je tiens à leur exprimer ma gratitude pour tous les instants de reflexion scientifique et pédagogique, et les moments de fête partagés ensembles. Avec une mention spéciale pour Annelies (*Môman*), David, Mike et Thierry pour leur amitié, ainsi que les mécaniciens du LIMSI qui m'ont appris à avoir le reflexe "pôt" pour marquer les événements de la vie, ne serait-ce que la fin de semaine.

Merci à Boris Doval pour ses connaissances en traitement du signal et sa disponibilité.

Merci à Sylvie Gibet qui m'a recommandé auprès d'Alain Finkel et m'a fait confiance.

Merci à Sophie Pageau pour avoir relu et corrigé ma thèse.

Un grand merci également à mes parents qui m'ont toujours encouragé et ont préparé un buffet mémorable pour la soutenance.

Enfin merci aux concepteurs des logiciels suivants : LaTeX, nedit, showcase, Scilab, Netscape, qui m'ont été fort utile pendant toutes ces années.

Introduction

Communiquer avec une machine devient une activité courante qui dépasse le cadre de la bureautique. Du simple agenda électronique aux bornes interactives permettant d'acheter des billets de trains par un dialogue oral, en passant par le téléphone portable et les systèmes d'aide à la navigation automobile, notre quotidien est progressivement envahi par des machines communicantes. Ces machines sont destinées à être utilisées par un grand nombre de personnes, souvent non-spécialistes. Elles offrent un nombre croissant de fonctionnalités, qui doivent être facilement accessible à l'utilisateur par le biais de l'interface de communication. La facilité d'interaction permet à l'utilisateur de se focaliser sur la tâche qu'il veut accomplir. Ce qui est particulièrement important lorsque l'attention de l'utilisateur est déjà focalisée sur une autre activité. Lorsque le conducteur d'un véhicule veut utiliser un téléphone ou un système d'aide à la navigation, il doit pouvoir le faire tout en conduisant en toute sécurité. Dans d'autres exemples où l'utilisateur gère un nombre important d'informations, il est nécessaire de développer des outils communicants qui l'aident dans sa tâche sans risquer de le perturber. C'est le cas des pilotes d'avions, des contrôleurs aériens ou encore des chirurgiens.

La recherche en communication homme-machine propose des solutions rendant l'interaction simple pour les utilisateurs. Parmi ces solutions, on trouve les interfaces dont l'aspect et le comportement s'inspirent de situations réelles et connues de la plupart des utilisateurs. Le bureau virtuel représente les dossiers, les documents, la corbeille et la calculatrice que chacun a en général dans son bureau. Une autre solution consiste à prendre modèle sur la communication entre humains. De là sont nés les dispositifs de reconnaissance de la parole et les systèmes de gestion du dialogue qui permettent un dialogue vocal entre un homme et une machine. Les recherches sont désormais assez avancées pour que ces systèmes soient accessibles au grand public, notamment sous forme de billetteries automatiques vocales dans les gares ou de serveurs vocaux téléphoniques. Le dialogue entre humains ne se limite pas à la parole, il exploite d'autres canaux de communication, comme par exemple le geste et les expressions du visage. Les recherches en linguistique [Calbris85] et en communication humaine [Kendon94], ont montré l'importance de la communication "non-verbale" dans le dialogue. Notamment, on sait que les gestes, qu'ils soient "co-verbaux" ou utilisés sans parole, ont un potentiel communicatif important qui les rend parfois plus efficaces et plus concis que la parole. Des recherches ont été entreprises sur la reconnaissance automatique des gestes humains, principalement les gestes de la main [Cadoz93] (pour les utiliser en communication homme-machine).

Le geste humain peut assurer trois fonctions principales [Cadoz93]. La fonction ergotique consiste à agir physiquement sur les objets en utilisant par exemple la souris ou le gant numérique. La fonction épistémique permet de capter des informations sur l'environnement (terminal Braille, dispositifs à retour d'effort ...). La fonction sémiotique consiste à transmettre des informations à l'environnement, par exemple en désignant avec le doigt l'objet dont on est en train de parler. Elle est plus difficile à exploiter, car elle nécessite non

seulement de capter le geste mais aussi de l'interpréter. Des études récentes s'intéressent à cette fonction en communication homme-machine (geste co-verbal [Cassell98], langue des signes des sourds [Braffort96]). Cependant une des composantes du geste est encore peu exploitée : le regard. Or, le regard fait partie des moyens utilisés dans la communication entre humains pour transmettre de l'information.

On sait qu'il est possible d'utiliser le regard pour interagir avec une machine (sélection ou le déplacement d'objets sur un bureau virtuel [Jacob95]). Il est aussi possible de connaître l'endroit où se porte l'attention visuelle de l'utilisateur. Une machine peut ainsi afficher à l'écran des informations sur le dernier objet regardé, ce qui rend la consultation ou la recherche d'informations extrêmement rapide [Jacob95]. Connaître la direction du regard permet d'observer les modalités d'attention de l'utilisateur (vigilance, fatigue) dans des situations où la sécurité entre en jeu (le conducteur d'un véhicule ou le contrôleur d'une centrale nucléaire). Pour prendre en compte le regard, il faut nécessairement le capter. Les dispositifs de capture du regard utilisés actuellement, que ce soit pour des recherches en optométrie, en psychologie de la perception ou en interaction homme-machine, nécessitent un matériel spécifique, difficile à mettre en œuvre, souvent contraignant pour l'utilisateur (immobilisé ou nécessité de porter du matériel sur la tête . . .) et dont le coût est très élevé (rendant difficile leur accès par un grand nombre d'utilisateurs).

L'objet de ce travail de thèse est d'étudier et de développer un système de capture du regard qui pallie aux inconvénients listés précédemment. Le dispositif de capture ne doit pas être intrusif, ni gêner l'utilisateur dans ces mouvements, ni perturber sa concentration sur la tâche qu'il accomplit. En deuxième lieu, le système doit être facile à mettre en œuvre, comme l'est par exemple la souris, qui après un éventuel réglage lors de son installation, est toujours prête à servir. Le dispositif de capture du regard doit fonctionner en permanence et doit détecter la présence de l'utilisateur automatiquement pour commencer à mesurer la direction du regard. Par ailleurs, d'autres contraintes sont liées à l'interaction et aux spécificités des mouvements des yeux. En particulier, le système doit être suffisamment rapide pour mesurer et suivre les mouvements oculaires en temps réel. Cette contrainte dépend du type de mouvement que l'on veut exploiter dans l'interaction. Par exemple, si l'on souhaite exploiter le temps de fixation du regard, il faut que le système puisse détecter la plus rapide des fixations, c'est-à-dire 200 millisecondes. Ensuite, le système doit produire des mesures relativement précises selon l'exploitation escomptée. Par exemple, si l'on veut savoir quelle fenêtre affichée à l'écran est observée par l'utilisateur, une précision de l'ordre de quelques degrés est suffisante. Par contre, si l'objet est une icône, la précision doit être alors inférieure à un degré. Enfin, le système doit être robuste pour fonctionner avec des personnes différentes et dans des conditions variables, notamment de luminosité. Sachant que la contrainte de robustesse est difficile à satisfaire complètement, on introduit une contrainte de fiabilité du système. Le système doit être capable d'évaluer la validité des résultats produits.

Sur la base de ces contraintes, nous avons mis en place une plate forme matérielle et

logicielle de capture du regard. Pour la partie matérielle, nous optons pour une capture du regard avec une caméra vidéo monochrome. Cette caméra n'a aucun dispositif bruyant ou mobile risquant de gêner l'utilisateur. Elle est suffisamment petite pour être placée entre l'écran et le clavier d'un ordinateur. Cette position permet de capter les images de l'utilisateur qui interagit avec la machine, dans lesquelles les yeux sont visibles quelque soit la direction du regard sur l'écran. Le choix de la fréquence d'échantillonnage et de la taille des images a une incidence sur les capacités globales de précision, de robustesse et de fonctionnement en temps réel du système. Un compromis temps réel/précision permet de mettre au point le système malgré les caractéristiques limitées du matériel dont nous disposons. Pour la partie la plus importante qui concerne l'aspect logiciel du système, celui-ci exploite les séquences d'images captées par la caméra, en procèdant par une série de traitements pour calculer la direction du regard de l'utilisateur. Les contraintes décrites ci-dessus conduisent à une spécification algorithmique du système. Nous développons deux aspects du fonctionnement du système : un aspect statique concernant les traitements effectués sur chaque image de la séquence ; et un aspect dynamique, qui tient compte des caractéristiques temporelles de la séquence vidéo. L'aspect statique consiste à exécuter une série de processus visant à détecter les yeux dans l'image. L'œil est une composante complexe qui est difficile à détecter tout en respectant la spécification du système. La stratégie adoptée consiste à détecter d'autres composantes moins complexes en premier. Connaître la localisation de ces composantes dans l'image permet de réduire l'espace de recherche des yeux. Le système détecte la boîte englobante du visage, puis recherche le nez à l'intérieur de cette boîte, et enfin recherche chaque œil dans une zone délimitée par le haut du visage et le nez. Cette stratégie permet au système de fonctionner avec des calculs rapides et garantie une certaine robustesse quant à la localisation des yeux. Cependant, ces traitements statiques doivent être guidés par l'aspect dynamique pour augmenter la fiabilité de détection.
Les processus qui détectent chaque composante (visage, nez, yeux) fonctionnent selon deux états différents : un état d'initialisation et un état d'adaptation. Dans un premier temps, tout processus se trouve dans l'état d'initialisation. Il utilise des paramètres généraux pour détecter la composante recherchée. Grâce à l'auto-évaluation de la fiabilité des mesures qu'il réalise, le processus "sait" s'il a réussi à détecter la composante ou pas. Si la composante est correctement détectée pendant plusieurs images successives, le processus passe en état d'adaptation. Dans cet état, le processus suit la composante en utilisant des paramètres spécifiques mesurés dans les images précédentes. Le processus s'adapte donc au visage de l'utilisateur et les calculs sont plus robustes et plus fiables. Au cours du suivi, il est possible que le processus ne détecte plus la composante. Par exemple si l'utilisateur passe sa main devant son visage ou s'il s'en va. Dans ce cas, le processus décide qu'il a perdu la composante et il repasse dans l'état d'initialisation.

Une fois que le système a détecté les yeux dans la séquence d'images, il s'agit ensuite de calculer la direction du regard à partir des images des yeux. Ce problème est complexe car plusieurs paramètres entrent en jeu, notamment l'orientation des yeux par rapport au visage et la position du visage dans l'espace par rapport à l'écran de l'ordinateur.

Nous proposons d'évaluer une solution pour la partie du problème qui consiste à calculer l'orientation des yeux par rapport au visage.

Il est difficile de démontrer que les différentes solutions proposées dans le système, respectent précisément les contraintes définies. Cette difficulté est principalement due à la variabilité des données en entrée des processus et à la complexité des traitements. Nous avons spécifié et réalisé un corpus de séquences d'images, et nous avons effectué une évaluation des performances du système appliqué à ce corpus. Cette évaluation s'inspire des méthodes utilisées dans les domaines de l'interaction homme-machine et de la vision par machine. La conception du corpus est réalisée dans des conditions précises, tenant compte du fait que le système de capture doit être exploité pour l'interaction. Un scénario d'interaction a été défini et différentes personnes l'ont exécuté. Pour chaque utilisateur, le système enregistre le film de la séquence et les interactions par la souris. Les films sont ensuite étiquetés pour ce qui est de la localisation du nez et des yeux. Il est ainsi possible de comparer les valeurs de l'étiquetage et celles renvoyées par le système de capture pour calculer les erreurs de mesure et évaluer les performances. Nous présentons les résultats de cette évaluation de manière détaillée, notamment en terme de fiabilité et de robustesse. Cette présentation met en évidence les qualités et les défauts du système, permettant ainsi d'envisager des applications et des évolutions possibles de celui-ci.

Ce travail de thèse pose le problème de la conception d'un système de capture du regard en communication homme-machine. Il s'appuie sur des connaissances dans les domaines de la communication homme-machine et de la vision par ordinateur, mais aussi des connaissances spécifiques en optométrie, en psychologie de la perception et en interaction par le regard. Nous proposons des solutions adaptées à l'interaction homme-machine et une méthode de validation pour un outil de capture du regard par un système de vision.

Chapitre 1

Communiquer par le regard

1.1 La communication homme-machine

Évolution

La communication homme-machine (CHM) est un problème qui s'est posé dès les débuts de l'informatique et de la création de machines automatiques programmables. À cette époque, les besoins en communication se réduisent à l'entrée des données et des programmes, et à la lecture des résultats. Communiquer n'est alors que convertir une représentation en une autre, le plus souvent des caractères en bits [Caelen92]. Dans ce contexte, l'utilisateur n'interagit pas avec la machine, ou du moins pas au sens de la réalisation d'une action qui génère une réponse immédiate. Les premiers pas significatifs en matière de communication viennent avec le développement des langages de programmation et de langages évolués plus proche du langage naturel que le langage machine. Les organes matériels de cette communication étaient depuis les années 50, le clavier alphanumérique, la console de visualisation, le crayon optique, et l'imprimante. La plupart des outils ou des concepts modernes de la communication homme-machine : manipulation directe d'objets graphiques (métaphoriques) , souris, fenêtre, éditeur de texte, hypertexte et reconnaissance de gestes, ont été inventés dans les années 60, mais n'ont été disponibles pour le grand public que vingt ans plus tard [Myers98].

Au début des années 70, on s'intéresse à la parole comme mode de communication, et on développe un nouveau domaine de recherche, la communication parlée. Ce domaine sert d'exemple pour le développement d'une communication homme-machine plus anthropomorphique. Il pose des problèmes tels que l'analyse, la reconnaissance et l'interprétation de la parole dans les sens homme vers machine, et la construction de phrases et la synthèse dans l'autre sens. On s'est aperçu après une dizaine d'années, que la difficulté de ces problèmes nécessite de faire intervenir plusieurs disciplines, comme les sciences physiques de l'ingénieur, la phonétique, la psychoacoustique et la linguistique, pour réussir à progresser [Cadoz93]. Aujourd'hui encore, le développement de nouveaux moyens de communication, comme le geste ou le regard, nécessite les compétences de diverses disciplines.

Malgré ces progrès, la CHM reste peu conviviale et peu naturelle, et ceci jusque dans les années 80 sans que cela semble gêner les concepteurs de systèmes informatiques. Cependant, certains chercheurs comme J. Foley s'interrogent [Foley87] : « *pourquoi des ordinateurs très perfectionnés devraient-ils être difficiles à utiliser?* ». Les utilisateurs d'ordinateurs passent la plupart de leur temps à saisir des données. Or les déficiences en matière de CHM créent un goulet d'étranglement, que l'augmentation des performances des ordinateurs ne peut résorber. C'est l'amélioration des systèmes de saisie et donc de la CHM, qui permettra l'amélioration de la productivité des utilisateurs d'ordinateur en général [Rubine91].

De nombreux travaux en CHM ont eu lieu dans des domaines et pour des applications très divers : télé manipulation et commande de robots ; imagerie médicale ; ergonomie des postes de pilotage, des stations de contrôle de systèmes et chaînes de processus complexes ; bureautique . . . Ils se sont développés de la même manière, sans liens systématiques les uns avec les autres, et sans se réclamer explicitement de la CHM. C'est l'apparition du premier ordinateur personnel Macintosh®, qui fit une révolution dans le domaine de la CHM, avec l'apport de la souris, des icones et des fenêtres, et tout un mode de manipulation de représentations métaphoriques évoquant des manipulations d'objets naturels [Cadoz93]. Cette machine dotée de son interface, le Finder®, donna un sens aux termes de communication et d'interaction homme-machine.

État actuel

La CHM est une discipline à part entière et une profession reconnue depuis le début des années 90 [Nielsen90]. La finalité de la CHM est clairement établie : elle a pour objet l'étude des phénomènes sensori-moteurs, des processus cognitifs et des procédés techniques mis en jeu dans l'accomplissement d'une tâche au moyen d'une machine. Elle a pour objectif la conception et la réalisation de machines adaptées aux besoins, servant si possible d'extension aux facultés sensori-motrices et intellectuelles de l'homme. En général, la machine considérée en CHM est en fait un système informatique. Pour atteindre son but, la CHM fait appel à des ressources disponibles dans les disciplines les plus diverses, telles que [Coutaz et al.91] :

- la physiologie, la psychologie et l'ergonomie pour l'étude des processus sensori-moteurs et cognitifs. La linguistique pour le traitement de la langue naturelle (écrite ou parlée) et la phonétique en reconnaissance de la parole ;
- le traitement du signal et les mathématiques (analyse, topologie, statistiques) en vision, en synthèse d'images, en reconnaissance et en synthèse de la parole et du geste ;
- les techniques informatiques et le génie logiciel pour l'organisation logicielle des systèmes. L'intelligence artificielle apparaît un peu partout en filigrane dès qu'il

s'agit de mener un raisonnement en utilisant une représentation des connaissances du domaine.

Les derniers outils issus de la recherche en CHM et disponibles pour le grand public sont les logiciels pour l'accès à Internet et les systèmes de reconnaissance et de synthèse de la parole continue. La réalité virtuelle ou augmentée, les systèmes de communication médiatisée, les systèmes de reconnaissance de gestes et les interfaces multimodales ne sont pas encore sorti des laboratoires de recherche et des départements de conception des industriels, ou bien sont financièrement inaccessibles.

Aujourd'hui les recherches sur le dialogue homme-machine tendent à rendre celui-ci le plus proche possible du dialogue inter humains. Elles s'inspirent pour cela de situations spécifiques de dialogue : demande de renseignements, réservation [Caelen et al.97], travail coopératif [Flanagan et al.97][Ishii et al.94] . . . Un autre aspect important dans l'évolution de la CHM est la priorité qu'elle donne à l'usager. En effet, jusqu'au début des années 90, la CHM était centrée sur la technologie et de ce fait générait des moyens de communication pas toujours très ergonomiques. On peut citer par exemple le stylo optique, qui nécessite de lever le bras pour interagir sur l'écran, ce qui rend son utilisation inconfortable et fatigante. La souris, par exemple, le surpasse par son ergonomie et sa précision ([Shneiderman87], p.245). Cette démarche est indépendante de la technologie selon le principe de Buxton [Ishii et al.94] : « *Faisons des choses intelligentes avec la technologie stupide d'aujourd'hui, au lieu d'attendre pour faire des choses stupides avec la technologie intelligente de demain* ». Ceci conduit à l'étude d'interfaces à partir d'une technologie simple, comme le travail coopératif où l'espace de travail commun est une feuille de papier partagée par vidéo interposée [Ishii et al.94], ou à se passer de technologie comme dans les expérimentations de magicien d'Oz où les réponses de l'ordinateur sont simulées par un compère [AC et al.96]. L'étude préalable de la situation pour laquelle on veut améliorer l'interaction homme-machine ou l'interaction entre l'homme et son environnement via une machine, permet de réaliser une spécification précise et adéquate du système [Caelen et al.97]. C'est d'autant plus important si des conditions de sécurité sont critiques, comme par exemple pour les contrôleurs aériens [Mackay et al.97]. Ces études permettent de mettre au point des moyens ou des systèmes d'interaction adaptés à l'utilisateur et à ses pratiques.

Dans cette évolution de la CHM, nous soulignons la récente utilisation du geste comme moyen de communication. En effet, l'interaction gestuelle semblait être oubliée : autant les canaux communicationnels de la voix, de l'ouïe et de la vision ont été abordés, analysés et modélisés, autant le canal gestuel a été sous-développé. Il a fallu attendre l'invasion des souris et l'apparition du gant numérique, pour commencer à parler "officiellement" du geste. Et pourtant, le canal gestuel est peut-être le plus singulier et le plus riche des canaux de communication [Cadoz93]. Récemment de nombreux travaux sur l'utilisation du canal gestuel dans la CHM ont déjà donné des résultats, ce qui permet de disposer d'une base théorique et technique importante pour étudier de nouveaux moyens d'interaction par le

geste. Ainsi, Nielsen [Nielsen93] prédit que les prochaines générations d'interfaces utiliseront le principe de l'interface utilisateur "sans commande". Il décrit comment grâce, entre autre, à la capture des gestes et du regard de l'utilisateur, l'ordinateur "devine" quelle commande doit être exécutée.

Dans le cadre de ce travail, nous avons tenu compte des différents aspects de la CHM que nous venons de rappeler. L'axe principal est cependant l'utilisation du geste et plus précisément du regard dans l'interaction homme-machine. Il existe de nombreux travaux sur l'interaction gestuelle, mais ils n'exploitent généralement que les gestes de la main. On peut étendre les concepts issus de ces travaux à l'interaction par le regard, car il y a des similitudes et des complémentarités entre ces deux modalités de communication. Nous présentons donc les résultats de travaux effectués sur la communication gestuelle, en élargissant ou en adaptant ces concepts de manière à décrire la communication par le regard. Aussi, de manière générale, lorsque nous parlons de communication gestuelle, nous y incluons le regard.

1.2 De la communication gestuelle à la communication par le regard

La communication entre humains consiste à transmettre des informations par le biais des divers canaux que nos sens peuvent appréhender : visuel, auditif, olfactif, gustatif et tactile. On peut considérer que la vue et l'ouïe sont les plus utilisés et souvent d'une manière conjointe. Le toucher l'est aussi dans une moindre mesure, cela dépend essentiellement du contexte socioculturel dans lequel se situe cette communication. Les gestes sont exploités dans cette communication, ils permettent de porter des informations sur ces trois canaux. Selon le cas, on peut "voir" le geste lorsqu'il est exécuté, entendre le bruit qu'il génère ou le sentir sur soi.
Par ailleurs, l'action d'un humain sur le monde qui l'entoure est principalement gestuelle. L'homme peut capter de l'information en utilisant tous ses sens, mais il ne peut agir physiquement que par le geste.
Ces deux aspects du geste peuvent être exploités dans la communication homme-machine. Nous verrons comment ils peuvent l'être aussi grâce au regard. Nous allons définir d'abord, le geste de manière plus précise pour faire apparaître son potentiel communicatif.

1.2.1 Les trois fonctions du geste

Pour commencer, nous donnons des définitions concernant les fonctions du geste. Cadoz donne trois fonctions différentes pour le geste de la main [Cadoz93] :

- une fonction **ergotique**, qui est une action matérielle de modification de l'environnement. Ce qui caractérise l'action motrice, c'est la prise directe avec la matière, qu'elle

peut modeler, transformer, usiner, briser... La fonction ergotique ne lui communique pas d'information, mais de l'énergie, en appliquant des forces, des déformations et des déplacements sur les objets. Le regard et les mouvements des yeux n'ont pas cette fonction. Cependant, il est possible de la créer grâce à un système asservi aux mouvements oculaires, mais cela n'est pas naturel pour l'homme ;

- une fonction **épistémique**, prise de connaissance sur l'environnement. On pense immédiatement au sens du toucher, mais derrière cette fonction se cache une capacité de perception extrêmement performante qui se compose de trois parties :

  - la perception *tactile cutanée simple* : la surface entière du corps est pourvue de terminaisons sensibles au contact mécanique, mais c'est sur la surface interne des doigts qu'elles sont les plus nombreuses. Elles nous renseignent tout au plus sur la température ou l'état de surface des objets ;
  - la perception *tactilo-kinesthésique* ou *haptique* : par la combinaison de stratégies de palpations, de mouvements exploratoires et de l'évolution des informations envoyées par les différentes cellules sensibles de la peau, des muscles et des jointures, on acquiert des informations de formes (d'une précision voisine de la perception visuelle), orientation, distance et grandeur sur les objets ;
  - la perception *proprioceptive* : celle des muscles et des articulations. La sensibilité articulaire apprécie l'ajustage correct des segments osseux et rend compte des sensations de fermeté perçues au niveau du doigt. Ainsi est-on informé sur la position des différentes parties du corps les unes par rapport aux autres et par rapport à l'espace extérieur.

  Grâce à ces perceptions par le canal gestuel et sous certaines conditions de proximité et d'action motrice, il est possible d'être informé, sur la température, l'état de surface, la dureté ou la mollesse, la forme, l'orientation, la distance ou sur la grandeur des objets. On perçoit également leur poids, leur structure articulatoire ou leurs propriétés de plasticité ou de déformabilité et pour finir leurs mouvements. La fonction **épistémique** est la fonction principale du regard. Certaines de ces informations peuvent aussi être perçues par le regard et en général ces deux moyens de perception sont complémentaires ;

- une fonction **sémiotique**, il s'agit cette fois de comportements gestuels qui ont pour fonction de produire un message informationnel à destination de l'environnement. Il existe une grande variété de cas dont le geste co-verbal, le geste langage utilitaire, le geste langage esthétique, le geste graphique, le geste de commande, le geste instrumental... Nous montrons qu'il existe plusieurs liens entre les gestes de la main et le mouvement des yeux lors de l'utilisation de cette fonction (cf. pages 20 et 21).

1.2.2 Quelle fonction gestuelle exploiter dans le cadre de la CHM?

la fonction épistémique

Elle est exploitée de manière évidente pour le regard, via l'écran qui est le périphérique dominant dans le sens de la machine vers l'homme. Par quel autre moyen un ordinateur peut-il nous transmettre des informations destinées à être captées par le sens tactile? Le problème s'est posé pour les personnes ne disposant justement pas de perception visuelle. Ainsi, les personnes aveugles utilisent un terminal Braille, qui leur permet de communiquer en entrée et en sortie avec un ordinateur. Le terminal Braille dispose de touches pour la saisie et d'une ligne de caractères représentés chacun par une matrice de 8×2 picots. Ces picots commandés par l'ordinateur permettent d'afficher du texte en Braille. De ce fait, la personne peut utiliser ses doigts pour lire et communiquer avec la machine.

On trouve aussi la fonction épistémique du geste dans les interfaces à réalité virtuelle ou augmentée. En effet, ces interfaces permettent de manipuler des objets dont la représentation visuelle se veut réaliste au point qu'ils peuvent être dotés de caractéristiques physiques. Cependant, l'effet de ces caractéristiques physiques n'est souvent perceptible que de manière visuelle (si on lâche l'objet, il tombe), mais pas de manière proprio-tactilo-kinesthésique. Or il est important si l'on veut interagir de manière réaliste, de pouvoir faire le lien perceptif entre la vision et le "toucher". Des dispositifs à retour d'effort, issus de la recherche en robotique, permettent de simuler toutes ou partie de ces caractéristiques physiques. Il est donc possible de sentir le poids, l'état de surface, l'élasticité ou même la forme d'un objet virtuel. Ces dispositifs peuvent se présenter sous la forme d'un stylet raccordé au bras d'un robot [Pappu et al.98], d'un manche à balai à 3 degrés de liberté [Bouzouita et al.96], de petits vérins attachés entre la paume de la main et les doigts [Fabiani et al.96] ou d'un mécanisme dont l'extrémité est manipulable comme une paire de ciseaux notamment pour simuler des opérations chirurgicales [Cotin et al.96]. La précision et la vitesse de réaction de ces dispositifs rendent la perception plus réaliste que ce que peut produire la partie visuelle de l'interface. Cela permet un gain important en temps d'apprentissage [Bouzouita et al.96] et en précision [Fabiani et al.96] dans la manipulation des objets virtuels.

On voit dans les exemples cités ci-dessus, que l'exploitation de la fonction épistémique du geste est essentielle pour aboutir à l'objectif de l'interaction visé. Le lien entre le regard et le geste est nécessaire dans les interfaces à réalité virtuelle ou augmentée.

la fonction ergotique

De même que la fonction épistémique, la fonction ergotique se trouve dans les interfaces à réalité virtuelle ou augmentée. Le retour d'effort est là aussi nécessaire pour communiquer de l'énergie aux objets virtuels. En l'absence d'un tel dispositif, le geste

n'est pas ergotique au sens de la définition qu'en fait Cadoz, il est sémiotique. En effet, si on désire par exemple déplacer ou déformer un objet virtuel sans retour d'effort, on ne produit pas physiquement le déplacement ou la déformation, on lui transmet l'information correspondante. De fait, il n'y a pas de lien direct entre l'énergie que l'on dépense pour ce geste et l'énergie nécessaire pour réaliser cette tâche. Les mouvements réalisés ne sont contrôlés que par le retour visuel de leur action (boucle visuo-motrice [Jeannerod88]).

L'action que l'on réalise sur les touches d'un clavier ou sur la souris consiste bien à transmettre de l'énergie à ces objets. Si l'on prend le cas de la souris, la fonction ergotique du geste se prolonge dans l'interaction par le mouvement du curseur à l'écran. Mais cette relation souris-curseur s'arrête là. En effet, si le curseur atteint le bord de l'écran, il s'y immobilise alors que l'on continue à déplacer la souris. Encore une fois, l'absence de retour d'effort limite l'interaction à la boucle de contrôle visuo-motrice. Cependant, ce système est efficace et largement suffisant pour réaliser des interactions avec les objets en deux dimensions que l'on trouve en général dans le bureau virtuel d'une interface.
Si l'on prend le cas de la touche du clavier, la fonction ergotique du geste ne concerne que la touche parce qu'on appuie dessus, et pas l'ordinateur. La touche sert à transmettre de l'information à la machine et c'est donc la fonction sémiotique du geste qui est utilisée vis-à-vis de l'ordinateur au travers de la touche.

On trouve des dispositifs permettant d'agir sur un clavier ou le curseur d'une souris par le regard. Dans ce cas, on ajoute une fonction ergotique au regard, qui n'existe pas naturellement. Ce type de dispositif sert essentiellement à pallier à l'absence de la fonction ergotique du geste, notamment pour les personnes handicapées (cf. Section 1.3.3).

la fonction sémiotique

Cette fonction est utilisée de manière prépondérante dans la communication, notamment par le biais du clavier et dans une moindre mesure par la souris. On se rend compte que le geste de frappe sur un clavier, ne transmet qu'une information assez pauvre par rapport à son potentiel expressif. Si l'on observe le geste sémiotique utilisé dans la communication entre humains, on constate qu'il prend divers aspects et qu'il est capable de véhiculer dans certains cas plus d'informations, de manière plus précise et plus concise que la parole. Pierre Rabischong résume bien cette lacune de la CHM dans son commentaire d'un article de Foley en 1987 [Foley87] :

> « *... il est choquant de constater que l'accès conversationnel aux ordinateurs se fait par un clavier, frappé le plus souvent avec un ou deux doigts, et dans un langage qui est loin du langage naturel de la conversation inter humaine. D'où le souci de faire entrer le dialogue homme/ordinateur dans une configuration poly sensorielle, utilisant le sens tactile et la parole, afin de faire entrer l'homme, avec tous ses systèmes d'acquisitions biologiques, en vraie grandeur, dans l'interface.* »

Aujourd'hui la situation s'est améliorée avec l'usage de la souris et des interfaces à manipulation directe d'objets métaphoriques, et même si nous sommes loin de l'interface de configuration poly sensorielle de Rabischong, les études sur l'interaction gestuelle, comprenant la direction du regard, les expressions faciales et la multimodalité tendent à aller dans cette direction.

Avant d'énumérer des exemples d'utilisation du geste dans l'interaction homme-machine, il nous semble important de définir deux formes que prend le geste dans sa fonction sémiotique : le geste co-verbal et le geste langage utilitaire. D'une part, parce que ce sont de bons exemples de la capacité d'expression du geste sémiotique. D'autre part, parce que le regard y est exploité comme faisant tout ou partie du geste.

1.2.3 Le geste co-verbal

Naturel et spontané, plus ou moins développé selon les cultures mais particulièrement irrépressible dans certaines, c'est le geste qui accompagne la parole [Cadoz93]. Il est prépondérant dans la communication entre humains, puisque 90 % des gestes sémiotiques que nous utilisons sont des gestes co-verbaux [Cassell98]. Ils sont si naturellement associés au discours, qu'ils sont parfois réalisés même lorsqu'ils ne sont pas visibles, notamment lors de conversations téléphoniques [Cassell98]. Ils intéressent les linguistes, qui les ont étudiés et ont défini les différents types de gestes co-verbaux que l'on rencontre. Il est important d'exploiter le résultat de ces études pour évaluer la pertinence de l'utilisation des gestes co-verbaux dans le cadre de la CHM et pour spécifier les systèmes permettant d'intégrer ce canal dans l'interaction. Annelies Braffort fait une présentation détaillée des gestes dans sa thèse [Braffort96]. Nous nous en inspirons pour faire une présentation synthétique des gestes co-verbaux. Elle signale qu'il existe différentes classifications et que les chercheurs ne semblent pas unanimes sur celles-ci. Les classifications les plus répandues sont les suivantes :

- les gestes **symboliques** ou **emblématiques** sont des gestes ayant un sens spécifique qui les rend indépendants du canal verbal. Ils peuvent accompagner ou remplacer un mot ou un groupe de mots et ils sont propres à des communautés sociolinguistiques. Par exemple, le geste de salut est un emblème. S'il est exécuté en même temps que les mots "Au revoir", les deux messages sont sémantiquement redondants. Lever les yeux au ciel est aussi un geste emblématique, en général accompagné d'une expression faciale. Le geste symbolique ne représente que 10 % des gestes réalisés lors de conversations [Cassell98] ;

- les gestes **illustrateurs** sont dépendants du canal verbal. Le sens complet du message est obtenu en combinant les contenus du message oral et du message gestuel. Par exemple, la phrase "Je veux celle-là !" n'est interprétable que si elle est accompagnée d'un geste désignant l'objet désiré par le locuteur parmi les autres objets présents.

Les illustrateurs sont décomposés en quatre sous-classes :

- les gestes **déictiques** sont des mouvements de désignation généralement exécutés avec l'index tendu et les autres doigts pliés, mais parfois aussi avec d'autres parties du corps (tête, yeux, nez, menton...) ou par l'intermédiaire d'artefacts (règles, stylo...). Ils désignent un objet qui est simultanément référencé dans le discours ;

- les gestes **iconiques** sont les gestes qui représentent un objet, une action ou un événement concret, eux-mêmes simultanément référencés dans le discours oral. Ils indiquent la forme, la taille ou d'autres caractéristiques propres et parfois complexes comme la manière dont se déroule une action ou le point de vue physique du narrateur par rapport à l'action. Par exemple, un narrateur qui décrit le dialogue qu'il mène avec une personne en regardant vers le bas, indique la taille de son interlocuteur ;

- les gestes **métaphoriques** sont aussi des représentations, mais les concepts qu'ils représentent n'ont pas de formes physiques. Ces concepts abstraits sont donc représentés par leurs formes métaphoriques. Par exemple, la phrase "la réunion s'est poursuivie" et accompagnée d'un mouvement de roulement de la main ;

- les gestes de **battements** (ou bâtons) sont des mouvements qui rythment le discours, dont la forme est indépendante du contenu du discours. Ils ont une fonction pragmatique, utilisée pour commenter le discours, comme accentuer ou donner de l'importance à un mot, ou pour signifier une erreur. Par exemple, la phrase "Elle a parlé en premier, je veux dire en deuxième" est dite accompagnée d'un geste sec vers le bas puis vers le haut.

Ces classifications sont inspirées de celles données par Ekman et Friesen [Ekman et al.72]. Ceux-ci précisent qu'elles ne sont pas exclusives et qu'un même geste peut appartenir à l'une ou l'autre des classes selon le contexte dans lequel il est réalisé. Les gestes emblématiques et métaphoriques sont liés aux langues et aux cultures, cependant globalement les gestes co-verbaux sont plus universels que particuliers. Cela en fait un moyen de communication généralement indépendant de la langue utilisée. On constate que les mains, le corps, les yeux et le visage peuvent être exploités dans ces différentes classes pour l'expression gestuelle.

La caractéristique du geste qui lui permet d'être un moyen d'expression parfois plus efficace que la parole, est la simultanéité des informations véhiculées. Dans un même geste, on peut exprimer un symbole et plusieurs paramètres liés à ce symbole. Par exemple, dans un geste de désignation, la configuration de la main permet d'interpréter qu'il s'agit d'une désignation, et la direction du doigt ou de la main indique l'objet désigné. Une autre caractéristique intéressante des gestes co-verbaux est qu'ils sont rarement erronés. De ce fait, les auditeurs peuvent rectifier l'erreur orale par le geste. Par exemple, quand l'orateur dit "droite" alors qu'il pense "gauche", son geste indiquera probablement la

gauche [Cassell98].

Nous ajoutons à cette description des gestes co-verbaux, les utilisations spécifiques du regard. Celui-ci est considéré comme étant un des principaux signaux non-verbaux, avec l'expression faciale, la posture, la proximité et la tonalité vocale [Cook84]. Daly-Jones et al. donnent quatre fonctions utilisant le regard dans une conversation [DJ et al.98] :

- créer le contact : le croisement des regards fait partie des signes permettant d'amorcer une discussion. Et de fait, éviter de croiser un regard signifie ne pas vouloir communiquer ;
- distribuer la parole : le regard sert à redonner la parole. Lorsque le locuteur a fini de parler, il peut le signifier en abaissant le ton de sa voix, en laissant "traîner" sa voix à la fin de la phrase, en finissant un geste, en utilisant une phrase de clôture ou en changeant la direction de son regard [Torres et al.97]. Le locuteur a tendance à détourner son regard au début de sa prise de parole mais le reporte sur son auditoire à la fin de celle-ci [Cook84] ;
- vérifier l'attention et la compréhension : le regard est un moyen pour l'auditeur de signifier l'attention qu'il porte au discours du locuteur et un moyen pour le locuteur de s'informer sur la compréhension de l'auditeur ;
- désigner : le regard permet de diriger l'attention de l'auditoire vers un autre endroit que le visage du locuteur. Dans ce cadre, cela correspond à un geste déictique. Mais il peut aussi modifier l'intensité d'expression d'un geste. Notamment lorsqu'il est utilisé pour désigner les mains elles-mêmes en train de réaliser un geste sémiotique, cela a pour effet de renforcer l'expression de ce geste.

Les gestes co-verbaux sont d'un soutien important à la communication orale puisque l'on a constaté que la compréhension de phrases est deux fois plus précise lorsque celles-ci sont accompagnées de gestes que lorsqu'elles ne le sont pas [Kendon94]. On sait de plus que les gestes co-verbaux aident non seulement la compréhension de l'auditeur mais aussi l'expression du locuteur. Il semble que les gestes soient si profondément associés à l'expression orale, qu'ils permettent un accès à la mémoire plus rapide tant pour exprimer un énoncé que pour le comprendre [MS90]. Compte tenu de toutes ces qualités, il est naturel que le geste co-verbal soit utilisé dans le cadre de la CHM. Ainsi, nous verrons que plusieurs recherches visant à exploiter le geste conjointement à la parole dans le dialogue homme-machine ont été entreprises.
Avant cela, nous présentons un type de geste qui n'est pas lié à la parole, le geste langage utilitaire.

1.2.4 Le geste langage utilitaire

Cette catégorie de gestes est constituée des langages utilisant exclusivement le geste comme canal de communication. Ce type de langage est en général lié à une déficience du

canal auditif (interne ou externe aux personnes) : la langue des signes des sourds, le langage des grutiers, celui des plongeurs sous-marins ou la gestique du chef d'orchestre [Cadoz93]. En général, le type de gestes utilisé dans ces langages correspond au geste emblématique.

Le geste langage utilitaire le plus développé est la langue des signes des sourds. En effet, c'est une langue à part entière, avec une grammaire et un vocabulaire constitué de signes spécifiques. Cependant les recherches en linguistique en sont à leurs balbutiements et beaucoup de divergences existent à ce sujet [Cuxac93]. La grammaire associée n'est pas basée sur l'enchaînement chronologique des mots mais sur la spatialisation des signes. L'ordre d'exécution des signes importe moins que leur localisation par rapport au signeur. L'intérêt de cette langue vient du fait qu'elle s'est développée de manière naturelle et spontanée, dès que plusieurs sourds se trouvaient ensemble pour communiquer[Moody83]. Cette langue est suffisamment riche pour permettre aux sourds de converser aussi rapidement et efficacement que le font des entendants. Elle est conçue de telle sorte que la réalisation des signes économise les charges physique et cognitive du signeur et de ses interlocuteurs.

L'iconicité est la caractéristique principale de la langue des signes. Les signes et la manière de les réaliser dans l'espace, permettent de représenter visuellement l'objet, l'action ou l'abstraction qu'ils signifient. Cela permet notamment à des sourds de communautés linguistiques différentes d'arriver à converser facilement après une courte période d'adaptation [Cuxac93]. Cette langue utilise aussi la simultanéité des informations véhiculées par le geste pour augmenter sa capacité d'expression tout en restant très concise. A. Braffort décrit comment avec les signes standards "petit récipient", "tomber" et "surface plane", on peut exprimer en un seul geste "le verre tombe de la table" ([Braffort96], p.58). Ainsi ce ne sont pas seulement les signes utilisés qui permettent de s'exprimer mais aussi les paramètres associés comme la localisation dans l'espace ou la dynamique.

Il existe des similitudes entre certains gestes co-verbaux et des signes de la langue des signes, notamment pour représenter le temps [Calbris85] et les mouvements du regard. Bahan et Supalla [Bahan et al.95] décrivent les mêmes fonctions utilisant le regard pour distribuer la parole et vérifier l'attention et la compréhension de l'auditoire que celles données pour les gestes co-verbaux. Ils remarquent que le regard reste plus longtemps en contact avec l'auditoire dans une conversation en langue des signes que dans un dialogue oral. La raison est liée à une utilisation linguistique spécifique du regard dans la narration. On trouve deux fonctions linguistiques liées au regard, et cela dans plusieurs langues des signes comme le remarquent Bahan et Supalla [Bahan et al.95] et Cuxac [Cuxac93] :

- Le transfert personnel : lors d'une conversation en langue des signes, on peut s'exprimer de deux manières : soit avec des signes et la grammaire standard, soit en utilisant la pantomime, notamment en incarnant le ou les personnages que l'on met en scène. Mais le passage de l'un à l'autre doit être clair pour les auditeurs. C'est le mouvement des yeux, puis du corps qui prend place où se trouve le personnage,

qui signifient le transfert. Le regard devient alors celui du protagoniste de l'énoncé. Le signeur quitte le personnage en reprenant la place qu'il occupait en tant que narrateur ;

- La construction d'une référence : la direction du regard, sur les mains ou devant le signeur, transforme la référence de l'objet ou de l'action signé. Par exemple, le signe standard "bateau" lorsqu'il est réalisé sans l'apport du regard signifie "un bateau". Si le signeur regarde ses mains pendant la réalisation du signe, celui-ci devient "ce bateau". Le signeur peut aussi donner une référence spatiale simultanément à la réalisation d'un signe. Pour cela, il réalise le signe à l'endroit où il veut placer l'objet représenté. Mais il ne peut le faire que si ce signe n'est réalisé qu'avec une ou deux mains, sans intervention du corps. Dans le cas contraire comme pour le signe "école" réalisé avec les deux mains sur le ventre, c'est le regard qui désigne l'emplacement de l'objet dans l'espace, simultanément à la réalisation du signe.

Les caractéristiques de la langue des signes, tant sur le plan de la communication que de l'ergonomie, font de celle-ci un exemple intéressant à exploiter dans la réalisation d'interfaces de communication gestuelle appropriées. En effet, il serait dommage d'inventer des gestes "artificiels", au risque qu'ils souffrent d'un manque de naturel et de spontanéité, alors que nous disposons d'une panoplie de gestes éprouvés, qui sont associés à une grammaire établie. Même si pour l'instant, celle-ci n'est pas encore complètement définie par les linguistes et qu'elle est donc difficile à spécifier, des exemples d'applications existent [Braffort96].

Il est donc important d'exploiter le regard dans la communication homme-machine si l'on veut que celle-ci soit plus proche de la communication inter humains et plus adaptée à l'homme. Il nous faut répondre aux deux questions suivantes :

1. Comment interpréter le regard dans l'interaction ?
2. Comment doter la machine d'un système de perception lui permettant de capter le regard ?

La première question est discutée dans le chapitre suivant. La seconde constitue le sujet des chapitres suivants du mémoire.

1.3 Interagir par le regard

Nous avons vu que l'utilisation du geste langage utilitaire est généralement corrélée à une déficience du canal auditif. De la même manière, l'utilisation du regard dans la CHM est due principalement au défaut des autres moyens d'interactions : soit parce que

le regard est le seul mouvement possible, par exemple pour des personnes handicapées ; soit parce que la parole et les mouvements des mains sont déjà occupés, par exemple lors d'une opération chirurgicale. Les expériences que nous décrirons ne concernent que l'exploitation du regard seul, mais nous avons vu qu'il serait aussi intéressant de l'exploiter conjointement à d'autres moyens de communication comme le geste des mains et/ou la parole. Jacob [Jacob95] souligne que ce domaine de recherche est peu développé et que l'on a peu de pratique dans l'utilisation du regard pour la CHM, contrairement aux travaux qui analysent les mouvements des yeux des points de vue psychologique et physiologique. Cependant les connaissances dont nous disposons dans ces différents domaines nous permettent de réaliser des spécifications précises pour l'exploitation du regard dans la CHM.

Nous présentons dans un premier temps les potentialités du regard pour l'interaction. Puis, nous décrirons les expériences qui ont été réalisées dans le but de tester les différents moyens d'interagir avec les yeux. Enfin, nous présentons les applications qui ont été développées à base d'interactions par le regard.

1.3.1 Potentialités du regard

Diverses expériences et réflexions ont été menées, qui visent à déterminer le potentiel du regard pour une exploitation dans le cadre de l'interaction. L'idée la plus courante consiste à remplacer la souris par le regard. Cette idée est discutée notamment par Jacob comme nous le verrons. Nous présentons d'abord les performances de l'utilisation du regard en ce qui concerne une tâche de sélection.

Ware et Mikaelian [Ware et al.87] ont mesuré la vitesse et la précision de la sélection par le regard. Ils montrent qu'utiliser le regard pour désigner une cible sur un écran est plus rapide que les autres moyens de sélections (souris, manche à balai ...). Ils expliquent cela par le fait que les yeux se déplacent directement sur la cible, alors que les autres moyens de sélection nécessitent de déplacer la main pour attraper le périphérique, puis pour déplacer le curseur et sélectionner la cible. De toute façon, les yeux arrivent toujours sur la cible avant le curseur puisque ce sont eux qui contrôlent le mouvement de la main (boucle visuo-motrice [Jeannerod88]). D'après Card et al. [Card et al.80], l'opération de désignation d'une cible à l'aide d'un périphérique, suit la loi de Fitt [1] :

$$T = C + I \log_2 \left(\frac{D}{S} + 0,5 \right) \tag{1.1}$$

Cette équation permet de calculer le temps T (en secondes), utile pour réaliser un mouvement impliquant la coordination yeux-main, en tenant compte de la distance à parcourir D et de la taille S de la cible. C et I sont des constantes dont la valeur est déterminée empiriquement. En utilisant la souris, Card et al. [Card et al.80] ont déterminé : $C = 1,1$

1. *Fitt's law* : l'équation originale est $T = a + b \log_2 2\frac{D}{S}$ (cf. chapitre 3.1.3 du livre de Jeannerod [Jeannerod88]), les auteurs n'expliquent pas pourquoi le coefficient à l'intérieur du log passe de $\times 2$ à $+0,5$!

et $I = 0,1$. Ware et Mikaelian [Ware et al.87] constatent que seule la constante C est différente pour le regard, $C = 0,6$. Ils remarquent aussi que la sélection est plus rapide et plus précise si la cible a une taille supérieure à un degré d'angle de vision.

Istance et Howarth [Istance et al.94] ont mené le même type d'expérience, en comparant dans la même session les performances en temps de la sélection d'une cible par le regard et avec une souris. Ils ont mesuré des temps un peu plus longs que ceux de Ware et Mikaelian, et ils constatent surtout qu'il y a peu de différence entre les deux modes de sélection. Le regard est notablement plus rapide que la souris, lorsque la cible a une taille de $2,5^o$ d'angle visuel et plus. Ils expliquent les différences des résultats par le fait qu'ils utilisent des cibles plus larges et des sujets novices en manipulation de souris. Ils ne disent rien sur la loi de Fitt.

Mais les yeux sont plus qu'un simple outil permettant de déplacer un curseur rapidement à l'écran. Ils permettent souvent de savoir où se porte l'attention visuelle de l'utilisateur, sans que celui-ci ait à le spécifier. Cette propriété présente un inconvénient si l'on utilise le regard comme périphérique d'entrée. En effet, les yeux bougent constamment (cf. Chapitre 2.1), il est difficile de contrôler consciemment et précisément ces mouvements de manière continue. De plus, contrairement à la souris, ils sont toujours activés (*on*) [Jacob95]. Jacob souligne que la plupart des mouvements oculaires sont non-intentionnels et non-conscients, et qu'il convient de les interpréter prudemment pour éviter d'ennuyer l'utilisateur avec des réponses non voulues à ses actions. Il nomme ce problème *Midas Touch*. Il n'y a pas de manière naturelle d'enclencher l'interaction visuelle comme on le fait avec une souris simplement en la déplaçant. Et enfin, toujours par rapport à la souris, nos yeux n'ont pas de "bouton clic". Selon Jacob, il n'est pas question de fermer les yeux ou de faire des clins d'œil pour résoudre ces problèmes. Charbonnier [Charbonnier95] a testé cette méthode et a constaté qu'elle n'était pas praticable parce que trop fatigante.

Jacob se demande aussi, s'il doit y avoir un curseur qui suit la direction du regard. Mis à part le fait que celui-ci risque de disparaître (cf. Chapitre 2.1, page 35), il procure l'information la moins intéressante que l'on puisse trouver sur un écran, puisqu'il indique à l'utilisateur où il est en train de regarder, ce qu'il sait déjà. De plus, s'il y a un léger décalage entre le curseur et le regard, l'œil a tendance à compenser ce défaut, créant une boucle de rétroaction positive anormale [Jacob95]. Cela semble donc à proscrire, même si ceci s'est révélé efficace dans un cadre thérapeutique [2].

Jacob conseille donc d'utiliser des mouvements naturels des yeux comme une entrée implicite pour l'interface, plutôt que de demander à l'utilisateur de s'entraîner à bouger ses yeux d'une certaine manière pour interagir avec la machine. Il explique aussi que les réactions de l'interface peuvent être naturelles ou non. C'est-à-dire similaires aux réactions

2. Lusted et Knapp [Lusted et al.96] décrivent comment avec l'aide d'un oculomètre et d'un ordinateur, le docteur Warner a aidé une fillette de 18 mois qui avait subi une grave lésion de la moelle épinière étant bébé, à développer ses capacités visuo-motrices.

du monde qui nous entoure, ce qui n'est pas facile puisque que nous n'interagissons pas avec le monde par le regard, sauf dans les échanges entre individus (cf. pages 20 et 21). Cependant, nous verrons qu'il existe deux exemples probants de réponses naturelles au regard dans l'interaction. Jacob, quant à lui, a expérimenté des réactions non-naturelles que nous décrivons ci-dessous.

1.3.2 Expérimentations sur l'interaction par le regard

Jacob [Jacob95], Istance et Howarth [Istance et al.94], Charbonnier et Massé [Charbonnier et al.94] ont réalisé une série d'expérimentations, que nous présentons brièvement. Dans tous les cas, la machine connaît la direction du regard de manière continue et elle peut évaluer la durée d'une fixation. C'est cette propriété qui est principalement utilisée pour l'interaction :

- **Sélection d'objet** [Jacob95] : l'utilisateur doit sélectionner selon l'expérience soit une icone, soit un bateau, parmi plusieurs. Le problème est de remplacer le bouton de la souris. Il propose deux solutions : utiliser un bouton ou mesurer le temps que passe l'utilisateur à regarder l'icone, temps de fixation (*dwell time*). Cette deuxième solution semble être la plus efficace. En réglant le système de manière à ce qu'il attende un temps assez long, on est sûr de ne pas avoir d'erreur de sélection. Mais l'utilisateur préfère alors le bouton, pour gagner du temps. S'il est facile de corriger une erreur de sélection, par exemple en sélectionnant immédiatement après le bon objet, ce temps de fixation peut être réduit à une valeur entre 150 et 250 ms. La réponse du système est alors tellement rapide qu'il donne l'impression d'exécuter l'intention de l'utilisateur avant que celui-ci ne l'exprime. Dans le cas, où la correction de l'erreur n'est pas simple, il vaut mieux utiliser le bouton pour sélectionner. Jacob n'a pas trouvé de cas où un long temps de fixation (3/4 de seconde) est utilisable, probablement parce que ce n'est pas naturel.

  Istance et Howarth [Istance et al.94] font la même expérience et obtiennent les mêmes résultats. Ils testent en plus la sélection par le clin d'œil, et constatent que cela génère beaucoup d'erreurs (40 à 60 %).

  Charbonnier et Massé [Charbonnier et al.94] ont fait cette expérience pour la sélection de lettres dans un clavier virtuel affiché à l'écran. Le temps de fixation pour une sélection est d'une demi-seconde. Ils ont aussi testé un mode de sélection en deux temps : une première fixation (0,3 secondes) fait afficher une croix sur la lettre regardée, puis une seconde fixation sur la fenêtre texte, fait afficher la lettre. Leurs résultats sont en accord avec ceux des autres et le mode sélection en deux temps est le plus long de tous. Ils ont mesuré que le taux maximum d'écriture par le regard est de moins d'une lettre par seconde. C'est un bon résultat comparé au taux moyen de l'écriture manuscrite de 2 à 3 lettres par seconde et au taux moyen obtenu lorsque l'on tape sur un clavier avec un doigt de 2 lettres par seconde ;

- **Affichage continu d'attributs** [Jacob95] : l'écran est séparé en deux parties, d'un côté il y a une carte avec des bateaux, de l'autre une fenêtre de texte. Lorsque l'utilisateur regarde la carte, le système affiche dans la fenêtre de texte les informations concernant le dernier bateau fixé par le regard. Ainsi, à chaque fois que l'utilisateur regarde la fenêtre de texte, il dispose des informations dont il a probablement besoin. Les constantes mises à jour du texte sont suffisamment rapides pour ne pas être perçues par l'utilisateur quand il regarde la carte, et ne le gênent donc pas ;

- **Déplacer un objet** [Jacob95] : Cette opération nécessite de sélectionner l'objet, d'indiquer son nouvel emplacement et de désélectionner l'objet. La sélection est réalisée par le regard avec l'aide d'un bouton actionné avec la main, mais l'objet reste sélectionné tant que l'on appuie sur le bouton. La position de l'objet est mise à jour à chaque nouvelle fixation du regard durant plus de 100 ms. Ainsi, le mouvement de l'objet est relativement lisse, prévisible et semble instantané. Quand le bouton est relâché, l'objet reste sur sa position courante. Cela marche mieux si la destination est représentée graphiquement, car il est difficile de fixer par le regard sur une zone vide. Cette méthode fonctionne mieux que celle qui consiste à sélectionner par le regard et déplacer avec la souris. En effet, avant de déplacer l'objet avec une souris, c'est le regard qui cherche la nouvelle localisation et à ce moment-là le déplacement de la souris est vécu comme une perte de temps ;

- **Défilement d'un texte** [Jacob95] : En général, les textes sont présentés dans une fenêtre dont la taille permet de contenir le texte en largeur mais pas en hauteur. Jacob, ajoute en haut et en bas de la fenêtre, un bouton représentant une flèche respectivement vers le haut et vers le bas. Lorsque l'utilisateur lit le texte, son regard finit naturellement sur le bouton du bas après la dernière ligne affichée. Cela déclenche le défilement vers le bas. L'auteur ne dit pas quelle est la vitesse de défilement du texte, ni ce que pensent les utilisateurs de ce type d'interaction ;

- **Commande de menus déroulant** [Jacob95] : Si l'utilisateur regarde l'en-tête du menu pendant plus de 400 ms, le corps de celui-ci est affiché. Ensuite, l'utilisateur peut regarder les items du menu, un item étant sélectionné (mais pas activé) après 100 ms. Si la fixation sur un item dure plus d'une seconde, celui-ci est activé, la commande sous-jacente exécutée et le menu effacé. Si le regard quitte le menu plus de 600 ms, celui-ci est effacé. L'utilisateur peut aussi activer un item en tapant sur un bouton, et c'est cette solution qui est préférée car le temps de fixation pour activer par le regard est trop long et peu naturel. Cependant les actions qui précèdent l'activation semblent appropriées au regard ;

- **Sélection d'une partie d'un texte** : Istance et Howarth [Istance et al.94] comparent la sélection d'une partie de texte, matérialisée par une sur-brillance, par le regard et la souris. Le problème est comme pour le déplacement d'objet, de remplacer le bouton de la souris. Ils testent le clin d'œil avec deux modes, en continu où l'œil reste fermé le temps de sélectionner tous les caractères voulus, et en discret où

un premier clin d'œil enclenche la sélection et un second l'arrête. Évidemment, ils constatent que d'une part la souris est bien plus rapide que le regard et d'autre part les erreurs de sélection sont très faibles avec la souris contrairement aux yeux. Ils observent de plus que la taille des caractères utilisés, n'influe pas sur ces résultats de manière significative.

Enfin, Khan et al. [Khan et al.95] ont monté une expérience qui vise à vérifier s'il existe une corrélation entre la variation du diamètre de la pupille et le niveau d'intérêt d'une cible visuelle pour l'utilisateur, comme un bouton affiché à l'écran. L'objectif étant de résoudre le problème de la sélection sur le critère du temps de fixation, en le remplaçant par la détection d'une variation de la taille de la pupille. Ils démontrent que ce critère n'est pas assez sûr, puisque les variations lumineuses dues aux changements dans l'affichage sont les principales causes de variations de l'ouverture de la pupille. Cette méthode ne peut donc pas être utilisée pour rendre plus rapides et plus fiables les interactions par le regard.

1.3.3 Applications

Les applications de l'interaction par le regard sont majoritairement orientées vers les personnes handicapées. On trouve notamment, plusieurs applications qui permettent de taper sur un clavier virtuel affiché à l'écran.

Hutchinson et al. [Hutchinson et al.89] ont mis au point à la fin des années 80, un système appelé ERICA (Eye-gaze-Response Interface Computer Aid), composé d'un ordinateur personnel et d'un oculomètre. Ils ont intégré plusieurs logiciels spécifiques dont : un logiciel de contrôle de l'environnement et un système de communication non verbal pour des besoins personnels (appel d'une infirmière) ; un traitement de texte et un synthétiseur de parole ; des jeux vidéo et des programmes de musique numérique et éducatifs ; une petite bibliothèque de livres et quelques textes. Les applications sont lancées à partir de menus hiérarchiques. Les sélections sont réalisées par le temps de fixation du regard (2 à 3 secondes) en deux temps : après le premier laps de temps, une icone apparaît sur l'item fixé dans le menu, signifiant que la commande est sélectionnée ; si le regard n'a pas bougé après un second laps de temps, la commande est exécutée. La méthode permettant de saisir du texte est laborieuse, puisque la résolution du système de capture du regard ne permet pas d'utiliser un clavier complet affiché à l'écran. C'est donc avec des menus hiérarchiques que sont sélectionnés les caractères. Cela prend 85 mn pour taper une page de texte, ce qui ne semble pas plus performant que les systèmes mécaniques utilisés pour cette opération (tige fixée sur le front servant à actionner un clavier par des mouvements de tête) [Charbonnier95]. Pour accélérer la saisie, ils utilisent deux méthodes : les phrases les plus courantes sont intégrées dans les menus hiérarchiques ; un algorithme de prédiction change les caractères proposés dans le menu principal en fonction des deux derniers caractères entrés, ce qui permet un gain de 25 % sur le temps de saisie [Frey et al.90].

On arrive aujourd'hui à faire des systèmes de saisie de texte par le regard plus performants, comme nous l'avons vu précédemment avec l'expérience de Charbonnier et Massé [Charbonnier et al.94], ou dans le système de Gips et al. [Gips et al.93].

Istance et al. [Istance et al.96] proposent un environnement complet avec plusieurs claviers virtuels permettant d'interagir avec le système selon le contexte : clavier texte pour taper du texte ; clavier de dialogue pour contrôler les boîtes de dialogue ; clavier menu pour interagir avec les menus ; et clavier zoom permettant de déplacer une zone rectangulaire dans une fenêtre et affichant un agrandissement de l'intérieur de cette zone. Toutes les sélections sont réalisées par le temps de fixation du regard (entre 500 ms et 2 s). Celui-ci peut être modifié et avoir une valeur différente pour chaque touche. Les claviers virtuels agissent sur le système en générant des événements souris (déplacement ou clic). Ainsi, il est possible d'utiliser les applications standard du commerce sans modification spécifique de celles-ci. Les auteurs ne donnent pas d'indications sur le temps moyen pour taper un texte, mais ils précisent que cela doit être amélioré.

L'autre type privilégié d'application est celui qui vise les situations où les mains et la parole sont déjà occupées. Charlier et al. [Charlier et al.92] ont réalisé un système de commande de microscope par le regard en cours d'opération. En effet, les mains du chirurgien étant occupées par la manipulation des instruments chirurgicaux, il ne reste que trois possibilités pour déplacer le champ visuel du microscope : l'aide d'un assistant, qui peut faire des erreurs et a un temps de réponse assez long ; une commande par les pieds, qui augmente la charge cognitive du chirurgien pour dissocier les mouvements des mains et des pieds ; une commande vocale, ce qui n'est pas pratique pour réaliser des réglages fins et en continu. L'idée est d'exploiter l'information contenue dans le regard sur l'attention du chirurgien. Car pour qu'il puisse travailler, le champ de vision du microscope doit être centré sur ce qui intéresse le chirurgien. Un oculomètre est intégré au microscope de manière à ne pas gêner son utilisation. L'interaction se fait grâce à trois zones concentriques dans le champ de vision :

- dans la zone centrale, le regard ne déclenche aucun déplacement ;
- dans la zone intermédiaire, le regard déclenche après une seconde, un déplacement lent ($2\,^{\circ}/_{s}$) du champ de vision de manière à centrer la "cible" fixée par l'œil. À cette vitesse, l'œil a tendance à suivre naturellement la cible grâce aux mouvements de poursuite (cf. page 36) ;
- dans la zone périphérique ou en dehors du champ de vision du microscope, le regard déclenche un déplacement du champ de vision à vitesse maximum jusqu'à la zone intermédiaire.

Cette méthode est suffisamment naturelle pour qu'il ne soit pas nécessaire de faire un apprentissage ou une initiation préalable. Elle a été testée avec 80 chirurgiens et a fonctionnée sans réglage spécifique dans 90 % des cas. Les limites des zones d'interaction sont 30 % du champ de vision pour la zone centrale et 60 % pour la zone intermédiaire. Certains

sujets, ont préféré régler ces limites à 25 % et 50 % avec un temps de validation en zone périphérique de 200 ms. Les solutions retenues dans ce système d'interaction permettent de préserver la charge cognitive du chirurgien et ne perturbent pas sa perception visuelle.

On trouve des applications dans d'autres domaines, visant un public plus large. Selon Jacob [Jacob95], l'utilisation la plus pertinente du regard dans la CHM est celle de Starker et Bolt [Starker et al.90]. Ils ont mis au point un système qui donne des informations visuelles et orales selon l'intérêt que l'utilisateur porte sur ce qui est affiché. Le système affiche une scène en 3D, par exemple la planète du "Petit Prince" de S^t Exupéry, parsemée de volcans et de fleurs. En fonction de l'endroit où le regard de l'utilisateur se focalise, le système détermine la zone d'attention de celui-ci. Si l'utilisateur regarde toute la scène, le petit prince donne des informations générales. Si l'attention se resserre sur un groupe d'objets, les informations concernant ces objets sont données. Si le regard se focalise sur un objet précis, c'est celui-ci qui est décrit. L'un des problèmes rencontré est de discerner l'intérêt "fortuit" de l'intérêt "intense" de l'utilisateur dans son comportement oculaire, pour que le système réagisse plus efficacement. On peut remarquer que ce système peut être utilisé avec des interactions passives ou actives du regard, car même si l'on n'est pas au courant du fonctionnement du système, on s'en rend rapidement compte.

De manière générale, l'interface utilisateur "sans commande" telle que la décrit Nielsen [Nielsen93] est l'exemple de l'interface exploitant des mouvements des yeux naturels et y réagissant de manière naturelle. En effet, dans ces interfaces, il n'y a pas que des dialogues directs entre la machine et l'utilisateur. La machine n'attend pas des commandes explicites de l'utilisateur avant d'agir. Elle observe constamment le comportement de celui-ci pour exécuter les commandes adéquates, comme dans l'application présentée ci-dessus.

Un autre exemple d'application où l'on exploite la direction du regard de l'utilisateur sans que celui-ci ne s'en rende compte est l'expérience montée par Charbonnier [Charbonnier95]. Celle-ci consiste à asservir la définition de l'image affichée à la direction du regard : la partie de l'image regardée par l'utilisateur a une résolution maximum, et le reste est dégradé. L'expérience consiste à afficher plusieurs icones à l'écran. Lorsque la personne regarde une icone, celle-ci est nette (haute résolution). Dès qu'elle en regarde une autre, l'ancienne icone est dégradée (faible résolution) et la nouvelle devient nette. Charbonnier a mesuré que lorsque la distance entre l'icone nette et les autres icones dégradées est de 13^o d'angle visuel, la différence n'est pas perçue. Par contre si la distance est de 5^o d'angle visuel, elle est perçue. Une telle application permettrait d'optimiser les calculs d'affichage de scènes 3D complexes, par exemple en réalité virtuelle ou augmentée, en dégradant par des techniques de niveau de détail [Krus et al.97], les parties de l'image qui sont en dehors du point de fixation du regard.

Le domaine de la réalité virtuelle n'est pas en reste, cependant des expériences réalisées par Istance et al. [Istance et al.95] montrent que cela pose des difficultés supplémentaires.

Ils ont étudié un système permettant de mesurer la distance de vergence, c'est-à-dire la localisation du point de fixation du regard dans l'espace dans un environnement virtuel. Malheureusement, les mesures réalisées ne sont pas cohérentes et donc pas exploitables. Ils ont aussi testé le guidage d'un curseur 3D dans l'environnement virtuel, via le regard ou un *3D tracker*. Le guidage par le regard est bien moins précis que par le *3D tracker*. Ils pensent que ce type de mesures nécessite un outil de capture extrêmement précis et rapide (250 Hz). D'autre part, la perception de la troisième dimension dans un environnement virtuel semble changer d'une personne à l'autre ce qui complique la mise au point du système de calcul de vergence.

Enfin, les mouvements des yeux peuvent aussi être exploités en sortie. Torres et al. [Torres et al.97] ont mis au point un modèle d'interaction permettant à un personnage virtuel de réaliser des mouvements des yeux "naturels" en fonction de la structure de son discours. Ces mouvements correspondent à ceux définis dans la section sur les gestes co-verbaux (page 20). Dans ce cadre, l'exploitation du regard de l'utilisateur permet de mettre en place une interaction plus réaliste dans le dialogue entre l'homme et le personnage virtuel.

Les différentes expériences et systèmes d'interaction que nous avons présentés démontrent que l'utilisation du regard est possible comme entrée dans la CHM. Ils définissent aussi certaines limites de celle-ci. Jacob [Jacob95] écrit que le bénéfice réel de l'interaction par le regard, pour la majorité des utilisateurs, viendra de son caractère naturel, de sa fluidité, de la faible charge cognitive qu'elle engendre et du fait que les opérations sont réalisées de manière pratiquement inconsciente. Ces bénéfices sont atténués si des mouvements non-naturels et donc plus conscients sont nécessaires.

Il y a cependant, un problème à régler, celui de la capture du regard. En effet, peut-on espérer des mouvements naturels de l'utilisateur si le système de capture du regard ne le permet pas ? L'utilisateur peut-il réaliser des mouvements de manière inconsciente, si le système de capture n'a de cesse de lui rappeler qu'il est "suivi" ? Il faut donc prévoir dans la spécification du système de capture du regard, que celui-ci doit respecter les principes ergonomiques, mais aussi physiologiques et psychologiques garantissant une utilisation naturelle du regard. Nous allons donc explorer ces différents points dans le chapitre suivant.

Chapitre 2

Capter le regard

Capter le regard correspond à mesurer des valeurs relatives aux yeux. Ces valeurs relèvent des caractéristiques fonctionnelles ou morphologiques de ceux-ci. Il est important d'établir quelles valeurs on peut mesurer et de choisir celles qui nous intéressent. Dans un premier temps, nous présentons les connaissances [1] issues de la recherche en biologie, en neurologie et en psychologie, concernant la vision et les mouvements des yeux. Grâce à ces connaissances, il est possible de spécifier les caractéristiques d'un outil de capture du regard. Nous décrivons ensuite, les solutions techniques apportées au problème de la mesure de la direction du regard et des mouvements oculaires, à travers la présentation des différents outils de capture qui existent. Enfin, à partir des informations exposées dans ce chapitre et en tenant compte du contexte dans lequel est réalisée cette étude, c'est-à-dire la communication homme-machine, nous donnons les principes qui guident le développement de notre outil de capture du regard.

2.1 Les mouvements des yeux

Nous utilisons naturellement nos yeux sans avoir à nous soucier de leur fonctionnement. Pourtant, les mécanismes sous-jacents aux mouvements oculaires, sont extrêmement complexes et nécessitent plusieurs années d'apprentissage pour atteindre leur maturité de fonctionnement. Aussi, la plupart de ces mouvements sont réalisés de manière inconsciente, ce qui ne les empêche pas d'être efficaces et de nous servir à chaque instant. Les mouvements des yeux sont décrits à travers plusieurs domaines de recherche : la biologie, la neurologie et la psychologie. L'étude des yeux et de leurs mouvements constitue des sous-parties de ces domaines de recherche et elle trouve ses origines chez les philosophes grecs de l'antiquité. Nous nous inspirons donc des articles produits dans ces différents domaines pour décrire le fonctionnement des mouvements oculaires. Le premier aspect

1. Il est parfois difficile de trouver une information sûre à propos des yeux et de leurs mouvements. En effet, les scientifiques disposent d'instruments de mesures leur permettant de donner des valeurs précises, cependant ils ne donnent pas tous les mêmes. À titre d'exemple, on trouve dans le même livre [Chekaluk et al.92] deux valeurs différentes pour la vitesse maximale de rotation des yeux lors d'un mouvement de vergence : $10^o/_s$ page 220 et $20^o/_s$ page 260. Toutes ces valeurs sont donc citées à titre indicatif.

que nous décrivons ne correspond cependant pas aux mouvements. En effet, la fonction des yeux est avant tout de permettre la perception visuelle du monde et rare sont les mouvements qui ne sont pas associés à celle-ci. On observe tout de même des mouvements oculaires dont l'objectif n'est pas d'accéder à la vision, mais de servir à la communication comme nous l'avons vu dans le premier chapitre (cf. Section 1.2). Nous commençons donc notre description par le fonctionnement de la vision, pour ensuite parler des mouvements oculaires.

2.1.1 L'œil et la vision

David Hubel dans "L'œil, le cerveau et la vision" [Hubel94], donne la description suivante de l'œil en introduction :

> *« L'œil est souvent comparé à un appareil photographique, mais il serait plus juste de le comparer à une caméra de télévision fixée sur un tripode asservi à un système de poursuite automatique ; cette caméra posséderait une mise au point automatique, un contrôle de l'ouverture du diaphragme selon l'intensité lumineuse et une lentille auto-lavable ; en outre, elle serait reliée à un ordinateur très élaboré, capable de traiter en parallèle l'information. »*

Voyons dans un premier temps comment fonctionne la "caméra". Le globe oculaire contient le dispositif optique et le dispositif de perception de l'œil (Figure 2.1). Le dispositif de perception s'appelle la rétine. Celle-ci est placée dans le fond du globe oculaire. C'est en fait une partie du cerveau, elle en a été séparée au début du développement embryonnaire, mais elle conserve avec lui des connexions nerveuses constituant le nerf optique. La rétine a la forme d'une plaque d'un quart de millimètre d'épaisseur, elle est formée de trois couches de corps cellulaires, séparées par deux couches de fibres nerveuses reliant les cellules d'une couche vers la suivante. La couche à l'arrière de la rétine contient les cellules sensibles à la lumière, les photorécepteurs [2] ([Hubel94], p.44) :

- les **cônes** : servent à la perception des détails et des couleurs. Ils ne réagissent pas à une faible intensité lumineuse et sont courts et coniques ;
- les **bâtonnets** : assurent la vision crépusculaire, assez peu précise et en noir et blanc. Ils ne servent pas en plein jour et ont une forme longue et cylindrique. Ils sont beaucoup plus nombreux que les cônes.

Ces cellules renferment des pigments photosensibles. Cependant, alors que les bâtonnets n'en contiennent qu'une sorte, ceux des cônes sont de trois types différents. Ces pigments sont sensibles à des longueurs d'ondes lumineuses différentes ("cônes bleus", "cônes verts" et "cônes rouges" [3]), permettant la perception des couleurs.

2. Il y a 127 millions de cellules réceptrices dans la rétine de chaque œil, dont 120 millions de bâtonnets et 7 millions de cônes ([Bruce et al.84], p.25).

3. Les trois pigments des cônes absorbent les longueurs d'ondes de 430, 530 et 560 nanomètres, ce qui correspond plutôt aux couleurs violet, bleu-vert et jaune-vert ([Hubel94], p.173).

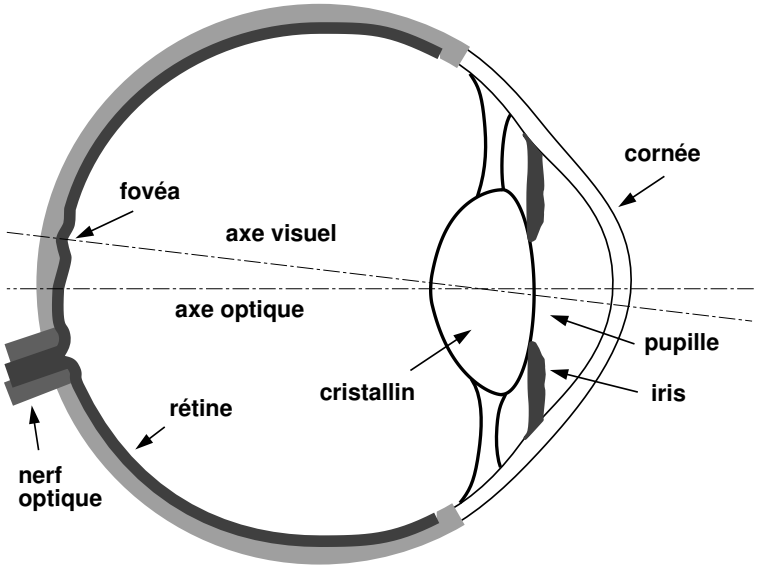


FIG. 2.1 – *Coupe de l'œil vu du dessus, d'après [Reuchlin86] et [Hubel94].*

Les photorécepteurs ne sont pas répartis de manière uniforme sur la rétine. Au centre de celle-ci se trouve la **fovéa** : la partie qui nous permet de voir avec le plus de précision [4]. Elle est constituée uniquement de cônes et mesure environ un demi-millimètre de diamètre (2^o d'angle visuel). Les cônes se trouvent dans toute la rétine mais ils sont particulièrement plus nombreux et plus dense dans la fovéa [5].

On peut distinguer deux systèmes visuels ([Reuchlin86], p.49) :

- la vision **photopique** : fonctionne surtout en lumière diurne et nous permet de distinguer les couleurs et les détails des objets. Cette fonction est essentiellement assurée par les cônes de la fovéa ;

- la vision **scotopique** : peut fonctionner en faible lumière et permet de percevoir des degrés d'intensité lumineuse, les directions de l'espace et les mouvements. La rétine périphérique (hors fovéa) riche en bâtonnets assure cette fonction.

Ces systèmes permettent de capter des informations lumineuses. Mais pour que le système de perception visuelle soit efficace, les yeux disposent d'un système permettant de générer les mouvements appropriés aux diverses situations de la perception. Par exemple, l'apparition d'un objet dans le champ du système de vision **scotopique** génère un réflexe

4. La résolution maximale de l'œil est 0,5 minutes d'angle de vision ([Bruce et al.84], p.23).

5. La zone qui contient la plupart des cônes est nommée *macula lutea* et son centre est la fovéa. Le reste de la rétine ne contient pratiquement que des bâtonnets ([Bruce et al.84], p.25). On trouve peu de références à la *macula lutea* dans la littérature. En général, les auteurs parlent de fovéa et de rétine périphérique.

amenant l'image de l'objet dans le champ du système de vision **photopique**, la fovéa.

2.1.2 Les différents types de mouvements oculaires

Dans l'œil, les éléments qui n'appartiennent pas à la rétine servent à produire sur la rétine une image focalisée et nette du monde extérieur. Chaque œil est maintenu en place par six petits muscles extra oculaires. Ces muscles sont groupés par paires agissant en opposition de telle sorte que l'œil bouge dans trois plans perpendiculaires. Afin de réaliser des mouvements de la précision de quelques minutes d'arc, nous mettons en jeu un ensemble de réflexes finement synchronisés, qui commandent notamment l'orientation de la tête ([Hubel94], p.41).

2.1.2.1 Les mouvements oculaires volontaires

Nous disposons de quatre systèmes oculomoteurs utilisés lorsque notre tête est immobile ([Godaux et al.89], p.230) : le système de maintien de position ou fixation ; le système saccadique ; le système de poursuite et celui des mouvements de vergence.

Le système de fixation et ses micromouvements

Ce système permet de maintenir une image stable d'un objet immobile sur la rétine ([Godaux et al.89], p.230). La fixation a lieu lorsque les yeux sont immobiles. Le temps d'une fixation comprend en général, un temps d'acquisition de l'information visuelle et un temps de programmation du mouvement oculaire suivant. Ces temps dépendent du type d'information regardée, notamment s'il s'agit de la lecture d'un texte ou de la scrutation d'une scène. Pour la lecture, le temps d'acquisition de l'information visuelle est de 50 ms. La plupart des fixations durent entre 225 et 450 ms ([McConkie83b], p.117). Lors de la scrutation d'une scène, le temps nécessaire à l'acquisition de l'information est de 100 ms ([Loftus83], p.373). Cependant, le temps de fixation est très variable ([Loftus83], p.367). En résumé, les yeux peuvent rester entre 100 ms et plusieurs secondes sur un point donné, mais la durée moyenne d'une fixation est entre 200 et 400 ms selon l'individu et l'activité sous-jacente.

Pendant une fixation, les yeux ne sont pas complètement immobiles. Ils effectuent trois types de micromouvement :

- les micronystagmus : fins tremblements de très petite amplitude et de fréquence assez élevée ;
- les dérives ou glissades : mouvements lents dont la vitesse ne dépasse pas $15'/_s$ et l'amplitude $5'$;

– les microsaccades : mouvements rapides dont l'amplitude varie, suivant les sujets et les circonstances, de 2 à 20′.

Lors de la fixation, l'œil balaie une plage qui ne dépasse pas 10 minutes d'angle autour de la position moyenne ([Charbonnier95], p.16). Les micromouvements sont involontaires. Il y a plusieurs hypothèses sur la fonction remplie par ces mouvements ([McConkie83a], p.77), ce qui est sûr c'est qu'ils sont indispensables à la vision. En effet, si on supprime artificiellement ces mouvements en maintenant une image stable sur la rétine, on perd la perception des couleurs et des contours au bout de quelques secondes ([Bruce et al.84], p.146).

Le système saccadique

Ce système est utilisé lorsque l'on veut centrer l'image d'un objet sur la fovéa, scruter une scène visuelle et localiser des cibles ou constater leur absence. Les yeux réalisent des mouvements extrêmement rapides, appelés saccades, pour parcourir la distance séparant le point de fixation actuel de celui convoité. Les yeux bougent en parallèle et dans la même direction (mouvements conjugués), horizontalement, verticalement ou en oblique ([Godaux et al.89], p.230).

Le mouvement des yeux en direction d'une cible, présente dans le champ de vision, est en général composé d'une saccade principale, dont l'amplitude correspond à 90 % de la distance à parcourir, et d'une saccade secondaire dite "corrective" amenant les yeux sur la cible ([Jeannerod88], p.102). Le reste de la distance peut aussi être parcouru par un lent mouvement correctif appelé "glissage" ou alors par une combinaison des deux mouvements([Shea92], p.276).

La saccade est un mouvement préprogrammé par notre cerveau, dit mouvement balistique ([Godaux et al.89], p.235). C'est-à-dire qu'une fois lancé, il ne peut être interrompu ou modifié avant d'avoir atteint sa cible ([Shea92], p.276). La saccade est déclenchée par une bouffée d'activité des motoneurones oculaires. La durée de cette activité détermine la durée de la saccade et donc son amplitude ([Godaux et al.89], p.235).

En général, la vitesse moyenne de la saccade est proportionnelle à la distance à parcourir : $400\,^o/_s$ pour $10\,^o$ et $700\,^o/_s$ pour $80\,^o$. La vitesse maximale est $700\,^o/_s$. La plupart des saccades sont inférieures à $15\,^o$ ([Shea92], p.276). Le temps de latence pour une saccade est en moyenne entre 175 et 200 ms ([McConkie83b], p.112). Une saccade corrective a quant à elle, un temps de latence de 150 à 190 ms. Malgré ce temps court, elle n'est pas préprogrammée en même temps que la saccade principale, mais semble être générée par rétroaction, après réalisation de la première saccade ([Jeannerod88], p.105).

Lors de la lecture, le temps de latence d'une saccade peut être beaucoup plus court, 125 à 175 ms, précédé d'un temps d'acquisition de l'information visuelle de 50 ms. Cependant, la plupart des fixations durent entre 225 et 450 ms ([McConkie83b], p.117).

Ces mouvements saccadiques sont trop rapides pour la perception visuelle, aussi sommes-nous "aveugles" pendant chaque saccade. Nous ne nous en apercevons pas, car notre cerveau maintient en mémoire l'image perçue avant la saccade, et ce tout le long de celle-ci ([Godaux et al.89], p.230).

Enfin les saccades obliques n'ont pas une trajectoire rectiligne, il semble qu'elles soient gérées par deux processus indépendants, l'un générant la composante horizontale et l'autre la composante verticale du mouvement, de manière non simultanée ([Godaux et al.89], p.236).

Le système de poursuite

Si l'on veut observer un objet en mouvement, il est nécessaire d'avoir une image nette de celui-ci en permanence. Pour cela, les yeux réalisent un mouvement conjugué, lent et continu appelé poursuite oculaire. Le système de poursuite a une structure en boucle fermée avec rétroaction négative, c'est-à-dire que la vitesse de rotation de l'œil est calculée en fonction de la vitesse de l'image de l'objet sur la rétine ([Godaux et al.89], p.231). La vitesse de rotation des yeux est en général de 30 à 40 $^{o}/_{s}$, mais peut aller jusqu'à 100 $^{o}/_{s}$. Le temps de réaction à un stimulus en mouvement est de 125 ms. Les yeux bougent à une vitesse proportionnelle à celle de la cible. Si celle-ci excède la vitesse maximum de poursuite des yeux, des saccades correctives viennent morceler la poursuite ([Shea92], p.265).

Le système de vergence

Les mouvements précédents sont adaptés à la vision d'objets éloignés. Lorsque l'on désire observer un objet proche des yeux, les axes oculaires ne peuvent pas rester parallèles, ils se croisent de manière à permettre de focaliser l'objet sur chaque fovéa. C'est ce que l'on appelle un mouvement de vergence ([Godaux et al.89], p.231). Les mouvements de vergence sont en général lents, avec une vitesse maximum de 20 $^{o}/_{s}$, une durée approchant une seconde et une latence de réponse d'approximativement 160 ms ([Shea92], p.259).

2.1.2.2 Les systèmes de stabilisation du regard

Lorsque nous réalisons des mouvements de la tête ou que nous nous déplaçons, nous utilisons deux systèmes permettant de stabiliser l'image sur les rétines ([Godaux et al.89], p.232) : le réflexe vestibulo-oculaire et la réponse optocinétique.

Le réflexe vestibulo-oculaire

Un mouvement de rotation de la tête génère un réflexe de rotation des yeux dans le sens opposé. Ce réflexe permet de garder une image stable sur la rétine lors d'une rotation de la tête. C'est un mouvement lent pendant lequel l'œil va réaliser, comme pour le

mouvement de poursuite, des saccades de correction permettant de diriger avec précision l'axe du regard sur la cible, et cela notamment si le mouvement de la tête se prolonge ([Shea92], p.272). Ce réflexe est engendré par le vestibule, dans l'oreille interne, qui transmet au cerveau la vitesse de rotation de la tête ([Godaux et al.89], p.117). Il est exploité par deux mécanismes : la coordination des mouvements yeux-tête et le *nystagmus vestibulaire*.

Ce réflexe permet de coordonner les mouvements de la tête et des yeux, pour le déplacement du regard. Lorsqu'un objet (une cible) apparaît sur la périphérie du champ visuel, nous réalisons en premier une saccade oculaire, puis un mouvement de la tête en direction de l'objet accompagné d'un mouvement des yeux dans le sens opposé (mouvement compensatoire des yeux [Godaux et al.89], p.172). Le mouvement de la tête n'est pas systématique, il est déclenché dès que la rotation de l'œil dépasse 30^o ([Charbonnier95], p.17). Le résultat de cette coordination est excellent : le regard est fixé très rapidement sur la cible et y reste accroché en dépit du mouvement de la tête ([Godaux et al.89], p.173).

Si la tête tourne de manière prolongée dans le même sens, les yeux eux ne peuvent pas continuer à tourner dans le sens inverse. Un système de repositionnement ramène les yeux du côté correspondant au sens de rotation, grâce à des mouvements rapides similaires à des saccades, appelés *phases rapides*. Les yeux réalisent une alternance de *phases lentes* (réflexe vestibulo-oculaire) et de *phases rapides*, appelée *nystagmus vestibulaire*. Si la rotation de la tête est sinusoïdale, on observe les mêmes mouvements alternés ([Godaux et al.89], p.232).

La réponse optocinétique

Lorsque l'on est immobile mais que le paysage bouge, par exemple si l'on regarde à travers la fenêtre dans un train en marche, on a tendance à suivre ce paysage automatiquement et de façon réflexe. C'est ce que l'on appelle la réponse optocinétique. Elle présente les mêmes phases lentes et rapides que celles que l'on trouve dans le réflexe vestibulo-oculaire, nommées alors *nystagmus optocinétique*. La réponse optocinétique combine les effets du réflexe de poursuite de l'ensemble du paysage et ceux de la poursuite volontaire d'un élément de ce paysage. On constate que le réflexe vestibulo-oculaire reste actif à des vitesses de rotations plus élevées que la réponse optocinétique ([Godaux et al.89], p.232).

2.1.2.3 Les différents types de saccades oculaires

Selon la situation, on distingue cinq types de saccades oculaires, mettant en jeu des circuits cérébraux différents ([Godaux et al.89], p.242) :

- **les saccades déclenchées par une cible visuelle**, consistent à centrer sur la fovéa, l'image d'une cible apparue dans le champ visuel. Elles naissent de la conjonction d'une commande volontaire et d'un stimulus visuel extérieur, la cible ;

- **les saccades volontaires** sont réalisées lorsque l'on décide de diriger notre regard dans une certaine direction sans qu'il y ait pour autant de cible visuelle précise ;
- **les saccades exploratoires** sont utilisées pour scruter une scène visuelle, notre regard saute sans arrêt d'un point à l'autre ;
- **les saccades spontanées dans l'obscurité** se font dans le noir en l'absence de tous stimuli visuels ou d'ordre volontaire net ;
- **les phases rapides du nystagmus**, vestibulaire ou optocinétique, ne sont pas à proprement parler des saccades, mais y ressemblent fortement et utilisent les mêmes structures.

2.1.2.4 Les clignements des yeux

Ce ne sont pas à proprement parler des mouvements oculaires mais ils interviennent de façon régulière dans le champ visuel. Le clignement correspond à l'abaissement de la paupière qui assure le nettoyage automatique de la face antérieure de la cornée. Lorsqu'une irritation survient sur la cornée, ses fibres nerveuses transmettent l'information qui provoque le réflexe de clignement de l'œil et la sécrétion de larmes ([Hubel94], p.43). Ces clignements, effectués de manière involontaire, sont réalisés en moyenne toutes les 3 secondes et leur durée n'excède pas 160 ms. Les clignements volontaires, qui portent en général une fonction sémiotique, ont une durée supérieure à 250 ms ([Charbonnier95], p.17).

Les différents aspects présentés ci-dessus, nous permettent de spécifier les caractéristiques techniques d'un outil de capture, en fonction des paramètres du regard que l'on désire mesurer. Mais les solutions techniques qui s'offrent en général pour la capture du regard, sont accompagnées de contraintes. Il convient de les connaître de manière à vérifier leur cohérence vis-à-vis de notre objectif, c'est-à-dire l'exploitation du regard pour la CHM. Nous allons donc passer en revue les différentes méthodes et outils utilisés pour la capture du regard.

2.2 Les outils de mesure

Des outils de mesure des mouvements oculaires, ou **oculomètres**, existent depuis plusieurs décennies. Ils ont été nécessaires pour réaliser les recherches portant sur le fonctionnement des mouvements oculaires. Plusieurs domaines scientifiques s'y intéressent à des degrés différents. Nous citons ici une liste proposée par Charbonnier ([Charbonnier95], p.23) :

- la physiologie s'intéresse à la caractérisation des mouvements oculaires et tente de mettre à jour les liens entre le système oculomoteur et la vision ;

- la médecine, dans le cadre de la pathologie cérébrale en particulier, étudie les troubles du comportement oculomoteur. La mesure des mouvements des yeux intervient également dans les expériences de neurophysiologie ;

- en psychologie, la stratégie visuelle d'un sujet renseigne sur les mécanismes psychophysiques de perception de l'espace visuel. Dans le domaine de la psychologie de l'enfant, les mouvements des yeux permettent d'étudier les stratégies perceptivo-motrices qui conditionnement l'apprentissage. En particulier, l'oculométrie est très utilisée pour l'étude de la stratégie visuelle et cognitive pendant la lecture ;

- en ergonomie, l'analyse d'un poste de travail passe par l'étude de la stratégie visuelle, en particulier dans les domaines de la conduite de véhicules ou du travail devant des écrans de contrôle ;

- enfin dans un autre domaine, les publicitaires mesurent l'impact d'un emballage ou d'une affiche sur les consommateurs, notamment grâce aux mouvements oculaires.

Ces différentes expérimentations n'ont pas les mêmes besoins en précision et en fréquence d'échantillonnage pour la mesure des mouvements oculaires. On trouve donc plusieurs outils de mesures, ayant chacun des caractéristiques adaptées à l'utilisation qui en est faite. Nous verrons que pour une utilisation dans le cadre de la CHM, il faut tenir compte d'autres caractéristiques que celles citées pour une utilisation "classique" des oculomètres. Parmi celles-ci, nous avons déjà évoqué le fait qu'il est difficile de réaliser des gestes naturels et spontanés si l'on doit porter un équipement pour les mesurer. La contrainte la plus discriminante est donc le fait que le système de mesure du regard ne gène pas l'utilisateur dans ses mouvements. Mais il est aussi important que l'utilisateur ne soit pas gêné dans son activité par du bruit ou des mouvements réalisés par l'outil de capture. Tout système qui par son fonctionnement oblige à porter de l'équipement ou génère des perturbations visuelles ou sonores est appelé système intrusif. Nous présentons donc les systèmes ou techniques existants pour la mesure de la direction du regard en les classant dans deux familles : les systèmes intrusifs et les systèmes non-intrusifs.

2.2.1 Les systèmes intrusifs

Lentilles à bobines magnétiques

Inventée par D. Robinson en 1963 ([Hubel94], p.86), cette technique permet de mesurer le mouvement directement sur l'œil. Une fine bobine de fil électrique est collée sur une lentille de contact, elle-même posée sur l'œil du sujet. De cette lentille, partent deux fils reliés à un instrument qui mesure le courant induit dans la bobine. Ce courant est généré par les champs magnétiques émanant de deux grandes bobines placées perpendiculairement l'une par rapport à l'autre, autour de la tête du sujet. Le courant dépend de la position de l'œil par rapport aux deux grandes bobines. Il est donc possible, après étalonnage du système, de mesurer les mouvements des yeux du sujet. Cette technique est pratique pour des tests cliniques, mais elle n'est pas utilisée plus d'une vingtaine de minutes, pour

éviter les risques de réaction allergique de l'œil. D'autre part, si les mouvements des yeux sont trop amples, la lentille risque de glisser, ce qui provoque un décalibrage du système ([Hallett86], p.10.26).

Électro-Oculographie

Placées autour des yeux, des électrodes mesurent le potentiel électro-oculographique, c'est-à-dire la variation du potentiel électrique entre la rétine et la cornée, en fonction de la position des yeux par rapport à la tête. La différence des potentiels mesurés par les électrodes, placées au-dessus et en dessous l'œil, indique la position verticale de celui-ci par rapport à la tête. Il en est de même pour la différence entre les électrodes placées de chaque côté de l'œil, qui indique sa position horizontale. Cette technique permet une précision des mesures de l'ordre de 0,5 à 1^o sur un large champ de vision ([Hallett86], p.10.28). Peu onéreuse, elle a été exploitée par Gips et al. [Gips et al.93] et par Kaczmarek [Kaczmarek93] dans des systèmes d'interaction par le regard. Cependant, elle ne permet pas de mesurer la position de la tête de l'utilisateur par rapport à l'écran, ce qui oblige celui-ci à garder sa tête fixe.

Localisation du limbe

Le limbe est la frontière entre le blanc de l'œil (sclère) et l'iris. La différence de luminosité de ces deux parties de l'œil doit permettre de détecter et de suivre facilement ce limbe. À l'aide d'un simple instrument de mesure de luminosité constitué de diodes LED et de capteurs infrarouge, le tout monté sur des lunettes, il est possible de réaliser un système de mesure peu coûteux. On trouve des systèmes commercialisés, dont la précision est de l'ordre de $0{,}1^o$ en mesure horizontale ou verticale seule [HVS Image], ou $0{,}5^o$ en horizontal et 1^o en verticale mesurés simultanément [Permobil inc.]. L'occultation occasionnelle des parties supérieure et inférieure du limbe par les paupières, rend ces mesures instables ([Glenstrup et al.95], p.6). Cette technique ne permet pas non plus de mesurer la position de l'utilisateur, qui doit donc rester immobile. Pour résoudre ce problème, Iida et al. [Iida et al.89] proposent d'utiliser un *3D tracker*, qui mesure la position de la tête du sujet et permet de calculer la direction du regard par rapport à un écran.

Image de l'œil

Plusieurs techniques consistent à analyser l'image de l'œil captée par une caméra vidéo, pour y localiser une de ses composantes comme la pupille ou l'iris. Le suivi de cette composante de l'œil permet de mesurer les mouvements réalisés relativement à la tête de l'utilisateur. Pour réaliser des mesures précises, il faut que l'image de l'œil couvre toute l'image captée par la caméra. D'autre part, la prise de vue doit être fixe par rapport à la tête du sujet. Il y a deux moyens pour immobiliser la tête par rapport à la caméra : soit on fixe la caméra sur la tête du sujet pour filmer l'œil directement ou par le biais d'un miroir semi-réfléchissant [NAC]. Dans ce cas le sujet est relativement libre de ses mouvements mais la mesure de la direction du regard est relative à la tête ; soit on fixe

la tête du sujet sur un support solidaire du bureau, qui intègre la ou les caméras. Ainsi on dispose, après étalonnage, de la position absolue des yeux, mais l'utilisateur n'est pas libre de ses mouvements.

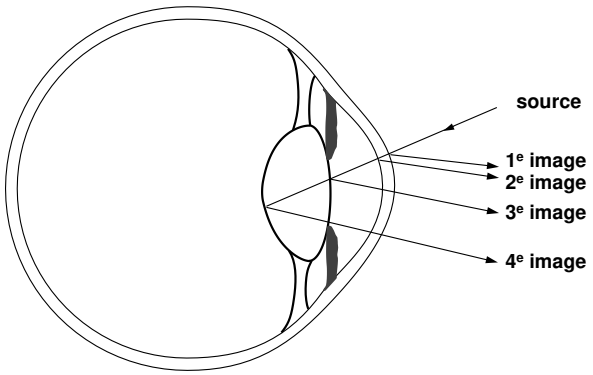


FIG. 2.2 – *Les quatre images de Purkinje, d'après [Glenstrup et al.95].*

D'autres techniques sont basées sur la mesure de reflets, en utilisant généralement une lumière infrarouge projetée sur l'œil. On trouve quatre reflets d'une même source lumineuse dans l'œil, les deux premiers sont dus à la cornée et les deux suivants au cristallin (Figure 2.2). On les nomme images de Purkinje ([Glenstrup et al.95], p.7). Ces reflets sont exploités de deux manières :

- ***Dual-Purkinje-Image*** : on peut mesurer la position de la première et de la quatrième image de Purkinje. Cela permet d'évaluer la rotation de l'œil indépendamment des translations horizontales ou verticales. L'utilisation d'un miroir mobile asservi à ces translations permet de garder une image de l'œil centrée dans la caméra. Cependant, la quatrième image de Purkinje est difficile à suivre, notamment quand la pupille est trop petite ([Hallett86], p.10.27), il est donc important de bien contrôler les conditions lumineuses lors des mesures [Cleveland et al.92]. Les dispositifs basés sur cette technique sont précis, $1'$ d'angle de vision, mais nécessitent un matériel lourd et complexe, n'autorisant pas ou peu de mouvements de la tête (25 mm) [HFT] ;

- **Reflet cornéen et pupille** : le premier correspond à la première image de Purkinje et le second est localisé aisément grâce à la réflexion de la lumière sur la rétine qui rend la pupille particulièrement brillante. Il est possible, à partir de la position relative du centre de la pupille et du reflet cornéen, de calculer la direction du regard par rapport à la caméra. Comme précédemment, on trouve des dispositifs portés par l'utilisateur, des dispositifs fixés sur le bureau bloquant la tête de l'utilisateur ([Charbonnier95], p.27), mais aussi des dispositifs posés sur le bureau autorisant des mouvements de quelques centimètres de débattement [Cleveland92]. Plus le sujet est contraint, plus le système est précis, cela va de quelques degrés à quelques minutes

d'arcs ([Charbonnier95], p.40).

L'analyse de l'image de l'œil, avec les techniques citées ci-dessus, permet aussi de mettre au point des systèmes de capture du regard qui laissent l'utilisateur libre de ses mouvements et ne nécessitent pas de porter sur la tête du matériel de capture. C'est ce que nous appelons les systèmes non-intrusifs.

2.2.2 Les systèmes non-intrusifs

Il se trouve que dans plusieurs situations, il est difficile d'utiliser un système de capture du regard intrusif : pour mesurer les mouvements oculaires d'enfants en bas âge, pour permettre l'accès à un ordinateur à des personnes handicapées ... De même, si l'on veut exploiter le regard comme modalité d'interaction avec un ordinateur, il est important, comme nous l'avons signalé, que le système de capture soit le moins intrusif possible. Les personnes intéressées par l'une ou l'autre de ces applications, ont donc cherché à améliorer les techniques utilisées pour la mesure des mouvements oculaires. Les systèmes mis au point, sont tous à base d'images vidéo, captées par une ou plusieurs caméras posées plus ou moins en face du sujet. On trouve deux sortes de systèmes : les systèmes à lumière et caméras infrarouges et les systèmes à lumière visible munis de caméras vidéo monochromes ou couleurs. Tous ces systèmes ont à résoudre deux problèmes principaux : suivre les mouvements de la tête et mesurer le regard.

Parmi les systèmes qui exploitent la lumière infrarouge, on peut citer le *Eyegaze System*® de LC Technologie [Cleveland92] qui n'autorise pas des déplacements de la tête de plus de quelques centimètres. Charlier a mis au point un système, le Visioboard® de Metrovision [Metrovision], qui suit automatiquement les mouvements de la tête et permet un débattement de la position de l'œil dans une zone de 20 cm de côté. La précision du système permet de discriminer des fixations du regard sur des cibles de 4^o d'angle de vision.

Les systèmes de capture du regard qui exploitent des caméras vidéo sensibles à la lumière visible, sont peu nombreux et existent seulement dans les laboratoires de recherche. L'objectif sous-jacent à leur développement est de ne pas utiliser de matériel spécifique, afin de pouvoir les intégrer à n'importe quel ordinateur pourvu d'une carte d'acquisition vidéo et d'une caméra. L'essentiel dans ces systèmes est donc le logiciel qui traite les images fournies par la caméra. La difficulté est d'arriver à mettre au point des traitements suffisamment rapides et performants pour obtenir un système qui mesure la direction du regard de manière précise tout en permettant à l'utilisateur de bouger librement. Ces systèmes sont les suivants :

- Baluja et Pomerleau [Baluja et al.94] utilisent le reflet spéculaire d'une source lumineuse qui apparaît sur les yeux, pour les détecter dans l'image. Ainsi, ils éliminent

le problème de localisation des yeux, comme dans les systèmes à base de lumière infrarouge. Ensuite, ils utilisent un réseau de neurones pour à la fois reconnaître l'œil et mesurer la direction du regard. Ce système ne tenant pas compte de la position et de l'orientation de la tête pour calculer la direction du regard, il ne semble pouvoir garder la même précision de mesure que si l'utilisateur ne bouge pas. D'autre part, l'utilisation d'un réseau de neurones ne leur permet d'exploiter qu'une image de 30×15 de l'œil à la fréquence de 15 Hz. La précision annoncée est de $1{,}5^o$ et le débattement de la tête de 30 cm ;

- Stiefelhagen, Yang et Waibel [Stiefelhagen et al.96], utilisent une série de traitements pour localiser le visage, puis les pupilles, la bouche et enfin les narines de la personne. Ensuite, ils évaluent l'orientation du visage grâce à un modèle 3D, POSIT [DeMenthon et al.92], développé par DeMenthon. Comme dans le système précédant, un réseau de neurones [Stiefelhagen et al.97] permet de calculer la direction du regard à partir de l'image des yeux. Leur système fonctionne à 15 Hz et la précision est de $1{,}9^o$, mais l'utilisateur doit rester dans la même position ;

- Varchmin, Rae et Ritter [Varchmin et al.98] utilisent principalement, quant à eux, des réseaux de cartes linéaires locales (*Local Linear Map-network*) pour localiser les yeux, les narines et les coins de la bouche, et évaluer la direction du regard. Ils ont recours à une lumière spéculaire, comme Baluja et Pomerleau, pour centrer l'œil lors de l'évaluation de la direction du regard. Celle-ci tient compte de la position du visage, elle est donc tolérante aux mouvements de la tête. Ces traitements très longs font que le système fonctionne à la fréquence d'une image par seconde. Ils obtiennent la précision de $1{,}5^o$ horizontalement et $2{,}5^o$ verticalement.

On constate que la capture du regard à l'aide d'une caméra est un problème complexe, mais qu'il est possible d'obtenir un système relativement précis, de l'ordre d'un degré d'angle de vision, et rapide, d'une fréquence supérieure à 10 Hz. Ces valeurs sont suffisantes pour exploiter la direction du regard dans le cadre de la CHM, si l'on se réfère aux applications citées dans la section 1.3.

Ces deux types de systèmes non-intrusifs, se différencient plus par leur technologie que par leurs performances potentielles. En effet, l'utilisation de la lumière infrarouge simplifie quelque peu les traitements mais les images obtenues sur une caméra infrarouge et sur une caméra à lumière visible, doivent permettre d'évaluer des mesures avec une résolution équivalente. Ces systèmes ont les mêmes inconvénients, notamment ils sont sensibles aux conditions d'éclairages et à la qualité de la prise de vue en particulier des yeux de l'utilisateur. Ils nécessitent aussi un calibrage pour pouvoir réaliser des mesures précises. On peut cependant noter quelques cas particuliers pouvant incommoder les systèmes exploitant les reflets d'une lumière (infrarouge ou visible) dans l'œil, par exemple si l'utilisateur porte des lunettes ou des lentilles de contact, ou si son œil n'est pas assez "humide". Le port de lunettes pose aussi des problèmes aux autres systèmes, mais le traitement d'image peut y

apporter des solutions. Enfin, on trouve dans le commerce des cartes d'acquisition vidéo et des caméras vidéo, sensibles à la lumière visible, à bas prix, contrairement aux systèmes exploitant la lumière infrarouge qui sont très onéreux. Il est donc plus pratique et moins coûteux d'utiliser ce matériel standard et commun pour mettre au point un système de capture du regard.

Conclusion

Les expériences diverses du développement de la CHM nous apprennent qu'il est nécessaire de s'appuyer sur divers domaines scientifiques si l'on veut exploiter les ressources communicatives de l'homme. La tendance anthropomorphique des recherches sur le dialogue homme-machine, incite à utiliser tous les canaux de communication pour faciliter et rendre plus naturelle cette communication. Dans ce cadre, l'exploitation du geste est de plus en plus étudiée et bénéficie de techniques de capture et de reconnaissance ainsi que de définitions en linguistique et en physiologie. Nous nous sommes intéressés aux gestes réalisés par les yeux aux travers des mouvements oculaires. Les recherches réalisées dans les divers domaines intéressant le regard et les mouvements oculaires, ainsi que le développement d'applications et de systèmes commercialisés exploitant le regard, ont mis en évidence le potentiel de ce canal de communication. Cependant, il reste un certain nombre de problèmes à régler si l'on veut utiliser efficacement le regard dans le CHM. Pour cela, il est important de tenir compte des contraintes émanant des domaines exposés précédemment. En guise de synthèse, nous proposons la liste suivante :

- **Système non-intrusif** : l'utilisateur doit pouvoir se concentrer sur la tâche qu'il veut accomplir, aussi le système lui permettant de communiquer avec la machine doit être le plus discret possible. En premier lieu, tout équipement porté par l'utilisateur est à proscrire, ainsi que les systèmes immobilisant ou restreignant les mouvements. Il n'y a pas que l'entrave physique qu'il convient d'éviter : un système déporté, par exemple sur le bureau, se déplaçant selon les mouvements de l'utilisateur est aussi perturbant, à cause des mouvements et du bruit qu'il génère (il peut s'agir de caméras ou de miroirs mobiles) ;

- **Fonctionnement en temps réel** : pour pouvoir exploiter les mouvements des yeux dans l'interaction il est nécessaire de disposer d'informations sur ceux-ci au moment où ils ont lieu. Ceci n'est évidement pas toujours possible, cependant on peut définir un temps de réponse minimum garantissant un fonctionnement confortable et naturel pour l'utilisateur. Si l'on veut exploiter le temps de fixation du regard pour réaliser des commandes, on peut utiliser comme temps minimum 200 ms (cf. Chapitre 2.1, page 34). Si l'on désire utiliser des mouvements comme les saccades oculaires, il est nécessaire d'être dix à vingt fois plus rapide ;

- **Précision des mesures** : celle-ci dépend de la taille des objets présents à l'écran et du fait qu'il y a besoin de retour visuel ou non. Une précision de 2^o d'angle visuel permet de discriminer différentes zones perçues par la fovéa et de sélectionner des

cibles avec précision. Mais une précision plus faible permet tout de même d'interagir comme nous l'avons vu dans les exemples d'applications ;

- **Fiabilité du système** : contrairement aux périphériques d'entrée classiques tels que le clavier ou la souris, un système de capture du regard doit réaliser des traitements pour interpréter des informations complexes. Ces traitements consistent par exemple à localiser des yeux dans une image et à calculer la direction du regard. Compte tenu de la complexité de la tâche, il est possible que ces traitements ne soient pas complètement fiables et qu'ils renvoient des résultats erronés. Dans ce cas, il ne faut pas transmettre ces résultats au système d'interaction, car il risque alors d'avoir un comportement incohérent, ce qui est peut-être gênant pour l'utilisateur. Il faut donc que le système de capture puisse évaluer la validité de ses résultats, pour ne renvoyer que des informations sûres et signaler d'éventuels problèmes au système d'interaction ;
- **Utilisation immédiate** : la mise en œuvre d'un système de capture du regard doit être aussi simple que celle d'une souris. Cette contrainte a deux conséquences :
  - le système doit fonctionner quel que soit l'utilisateur. On sait qu'il y a des différences significatives entre les visages des personnes, ce qui est une difficulté pour les traitements d'images ;
  - le réglage du système, appelé aussi calibrage, joue pour beaucoup dans la précision des mesures. Si l'on ne peut pas l'éviter, cela doit être une opération simple pour l'utilisateur.
- **coût faible** : l'autre inconvénient de l'usage des oculomètres commercialisés est leur prix. Si nous voulons pouvoir réaliser plusieurs études exploitant un système de capture du regard, l'aspect financier n'est pas à négliger.

Partant de ces contraintes et ayant connaissance des objectifs applicatifs que nous avons exposés dans les deux premiers chapitres, nous pouvons étudier et mettre en œuvre un système de capture du regard exploitable dans le cadre de la CHM. Ce travail est décrit dans le chapitre suivant.

Chapitre 3

CapRe : un outil de capture du regard

Étudier la conception d'un outil de capture du regard nécessite la prise en compte de plusieurs paramètres. Nous avons décrit le contexte dans lequel nous souhaitons utiliser cet outil. Ensuite, l'objet de la capture, c'est-à-dire le regard et les mouvements oculaires, a été décrit à travers les connaissances dont nous disposons en biologie, neurologie et psychologie. Nous avons montré que les systèmes de capture du regard qui existent et qui sont utilisés dans ces domaines scientifiques, ne sont pas compatibles avec les contraintes de la communication homme-machine. À partir de là, nous avons pu spécifier les contraintes propres à un système de capture du regard dans le contexte de la communication homme-machine.

Dans ce chapitre, nous proposons une solution répondant aux contraintes décrites précédemment et qui servent alors de spécifications. Cette solution se décompose en deux parties : une partie matérielle et une partie logicielle. Dans un premier temps, nous décrivons une solution matérielle appelée plate-forme expérimentale, qui tient compte des spécifications du système mais aussi du matériel disponible pour réaliser cette étude. Puis, à partir de cette solution matérielle, nous étudions la mise au point algorithmique du système de capture. Les spécifications sont traduites en termes de contraintes algorithmiques, qui guident les choix réalisés à chaque étape de l'étude. Ces choix sont ensuite évalués de façon quantitative dans le chapitre 4.

3.1 Description de la plate-forme expérimentale

La plate-forme expérimentale doit servir de support au développement et à la mise au point de notre système de capture. Mais, elle doit aussi permettre de réaliser des expériences sur l'interaction Homme-Machine. La mise en place d'une telle plate-forme doit donc suivre des contraintes liées à ces deux aspects, qui peuvent parfois être contra-

dictoires [1] et donc nécessiter de faire des compromis.

Quel est le cadre de l'interaction entre l'utilisateur et l'ordinateur? Le cadre le plus courant est encore aujourd'hui composé d'un écran, un clavier et une souris posés sur un bureau. L'utilisateur est quant à lui généralement assis face à l'écran. Nous avons donc décidé de monter la plate-forme à partir de cette configuration. Nous n'écartons pas l'idée que CapRe peut être exploité dans d'autres cadres, comme par exemple pour surveiller la vigilance des conducteurs de véhicules (cf. Chapitre 2.2).

Nous montrons dans le chapitre sur les capteurs de regard, qu'il est nécessaire d'utiliser un capteur non-intrusif si l'on veut que l'utilisateur puisse réaliser ses mouvements naturels. La solution que nous choisissons est d'utiliser une caméra vidéo comme capteur. Pour pouvoir être posée face à l'utilisateur tout en s'intégrant dans le cadre de l'ensemble écran/clavier/souris, la caméra doit être de petite taille. Elle ne doit pas contenir ni être associée à un mécanisme bruyant ou en mouvement, qui risquerait de gêner l'utilisateur. De même, nous n'utilisons pas de mécanisme de mise au point automatique, de zoom motorisé ou d'orientation motorisée de la caméra. Cette configuration respecte la contrainte selon laquelle le système doit être non-intrusif (cf. Contraintes page 44). Nous présentons les divers aspects techniques de la plate-forme expérimentale.

Poste de travail

Les caractéristiques techniques de la caméra doivent permettre de capter des images de l'utilisateur de "qualité" suffisante pour être exploitées par notre système de traitement d'image. Ces caractéristiques sont liées au système de numérisation qui permet d'échantillonner le signal vidéo provenant de la caméra. Pour cela nous disposons d'une station de travail SGI–INDIGO^2, équipée d'un processeur Mips R4400 cadencé à 250 MHz, de 64 Mo de mémoire centrale et d'une carte d'acquisition vidéo "Galileo". Cette machine n'est pas spécialisée pour le traitement d'images, mais le processeur est suffisamment puissant pour réaliser en temps réel les traitements nécessaires pour notre application (cf. Chapitre 3.2). La carte d'acquisition vidéo quant à elle, est capable de numériser un signal vidéo et permet d'afficher les échantillons réalisés en temps réel. Nous nous sommes rendu compte par la suite que cette configuration matérielle ne permet pas d'acquérir des images entières en mémoire centrale en temps réel. En effet, la carte d'acquisition sollicite le processeur pour réaliser le transfert des images numérisées vers la mémoire centrale. Ce mécanisme réduit la fréquence d'échantillonnage en fonction de la taille des images. Nous avons mesuré la bande passante du système qui est de l'ordre de 690 KHz. Cela a amené à faire les compromis que nous détaillons plus loin.

La carte d'acquisition numérise un signal vidéo PAL ou NTSC, et peut coder les images avec différents formats de pixel: niveaux de gris sur 8 bits, RGB sur 8 bits, RGB sur 24 bits, etc. Ces valeurs, par composante de gris ou de couleur, sont codées au mieux

1. par exemple, la nécessité d'éclairer le visage de l'utilisateur pour le filmer, ce qui risque de le gêner. Le compromis est de mettre des sources lumineuses de biais par rapport à l'utilisateur.

sur 8 bits. Donc la dynamique maximale de l'information numérisée est de 48 décibels ($20 \log 2^8$). Pour tirer le meilleur parti de ce matériel, la dynamique de la caméra vidéo que nous utilisons doit donc être au moins de 48 dB par composante captée (gris ou couleur).

Caméra : caractéristiques et emplacement

Le choix entre une caméra noir et blanc et une caméra couleur est essentiellement lié à la différence du rapport qualité/prix de ces deux types de capteurs. Il est évident qu'il est préférable d'utiliser une caméra couleur, afin de capter le plus d'informations visuelles possible, quitte à n'en traiter qu'une partie dans l'application. Cependant, il est aussi nécessaire d'avoir une caméra dont le capteur CCD et le circuit électronique générant le signal vidéo, soient de bonne qualité. A qualité égale, la caméra vidéo couleur est sensiblement plus chère que la noir et blanc. Nous nous sommes donc équipés d'une caméra vidéo Hitachi KP-M3 monochrome, muni d'un capteur CCD 1/3 de pouce (490 000 pixels). Son rapport signal/bruit est de 56 dB, ce qui correspond à notre attente. Elle a l'avantage d'être de taille suffisamment réduite pour être insérée discrètement dans la plate-forme.

L'image captée par la caméra doit nous permettre de mesurer la direction du regard de l'utilisateur. Cela signifie d'une part, que le point de vue de la caméra doit être optimal, notamment pour ce qui est de l'image des yeux, quel que soit la direction du regard. D'autre part, le visage de l'utilisateur doit se trouver dans le champ de la caméra, au moins lorsqu'il interagit avec la station de travail.

Nous avons procédé à plusieurs tests de prises de vue d'un utilisateur regardant vers des directions clés par rapport à l'écran de l'ordinateur : vers le haut, vers le bas, en face, vers la droite, vers la gauche, vers les coins droite-gauche/haut-bas et vers le clavier (Figures page 50). Au cours de ces tests, la caméra était placée soit au-dessus du moniteur, soit sur un côté du moniteur, soit entre le clavier et le moniteur. Nous avons tiré les conclusions suivantes de ces tests :

- Caméra au-dessus du moniteur : lorsque l'utilisateur regarde vers le bas, ses paupières se referment obstruant ainsi l'image des iris (Figure 3.1). Il est donc impossible de mesurer dans ces conditions la direction du regard ;
- Caméra sur le côté du moniteur : lorsque l'utilisateur regarde vers le côté opposé de celui où se trouve la caméra, le nez cache un œil (Figure 3.2). Or il est préférable d'avoir la direction des deux yeux (cf. Section 3.2.4) ;
- Caméra entre le clavier et le moniteur : la visée très basse et en contre-plongée tasse le visage de la personne. Cependant, l'image des yeux est toujours visible, quel que soit la direction du regard ou l'orientation du visage, vers l'écran ou vers le clavier (Figure 3.3).

Les images où les zones des yeux sont les plus exploitables sont celles recueillies par la caméra placée entre le clavier et l'écran, en prise de vue en contre-plongée. Cette orientation de la caméra présente aussi l'avantage de mettre dans le fond de l'image le

FIG. 3.1 – *Vues avec la caméra placée au-dessus du moniteur.*

FIG. 3.2 – *Vues avec la caméra placée sur le coté du moniteur.*

FIG. 3.3 – *Vues avec la caméra placée entre le moniteur et le clavier.*

plafond de la salle. Cela simplifiera une partie du traitement d'image réalisé par la suite, notamment en ce qui concerne la séparation entre le fond de l'image et le visage de l'utilisateur. La position idéale serait entre le centre et le bas de l'écran.

Caractéristiques de l'optique de la caméra

Si l'on ne veut pas contraindre la personne quant à sa position physique face à la machine, le champ de vue de la caméra doit être suffisamment large pour qu'il y ait peu de chance que le visage ne soit pas dedans quand la personne utilise l'ordinateur. Nous évaluons la largeur de ce champ de manière à pouvoir calculer la focale de l'objectif de la caméra. Les mesures faites sur un utilisateur placé devant la plate-forme, ont permis d'établir les paramètres suivants : la distance entre la caméra et le visage de l'utilisateur (d_u) est au moins de 450 mm, la largeur du champ (l_c) dans lequel se trouve le visage est d'environ 250 mm (cf. Figure 3.4). Sachant que la largeur de la matrice CCD (l_m) est 4.89 mm, nous pouvons déterminer la focale de l'objectif (d_f) qui convient, grâce à l'équation suivante (3.1):

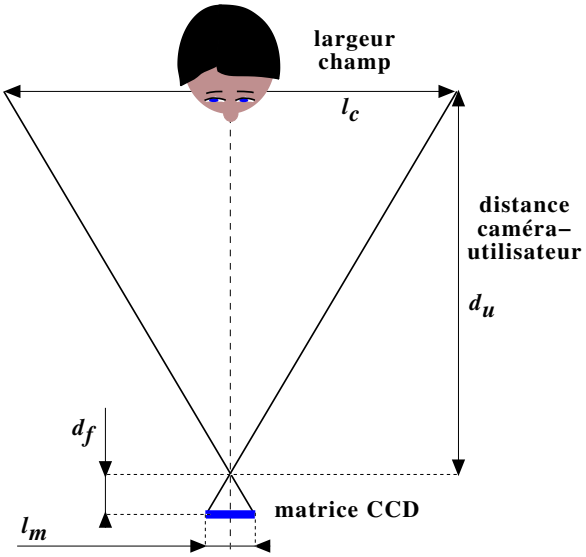


FIG. 3.4 – *Détermination de la focale de l'objectif (d_f), adaptée à la plate-forme.*

$$\frac{l_c}{d_u} = \frac{l_m}{d_f} \Leftrightarrow \begin{aligned} d_f &= l_m \times \frac{d_u}{l_c} \\ &= 4.89 \times \frac{450}{250} \\ &\approx 8,802\ mm \end{aligned} \tag{3.1}$$

Nous avons utilisé un objectif *Computar*, le plus proche de nos besoins avec une focale de 8,5 mm.

Il reste à régler un des problèmes majeurs en photographie, qu'est l'éclairage. Pour que la caméra capte des images exploitables du visage de l'utilisateur, il faut que celui-ci soit correctement éclairé. Le problème est moins la quantité de lumière que son homogénéité sur tout le visage. La dynamique du capteur CCD d'une caméra vidéo, même de bonne qualité, est bien trop faible pour permettre de capter des images très contrastées. Or, la limite de cette dynamique est souvent atteinte. Notamment, si le sujet est éclairé par la lumière du jour, les variations du temps sont suffisamment importantes pour saturer ou insensibiliser le capteur CCD. Il est donc nécessaire de maîtriser les conditions d'éclairage du sujet qui doit être filmé. Pour cela, nous avons installé un rideau opaque entre la plate-forme et les fenêtres de la salle. Cela permet d'éviter les variations lumineuses qui se produisaient sur un seul côté du visage, le rendant soit trop clair, soit trop sombre par rapport à l'autre côté. Nous avons aussi installé une source lumineuse de chaque côté du moniteur. Cela permet d'avoir toujours une lumière homogène sur le visage. Cependant, cette installation risque de gêner l'utilisateur, ce qui va à l'encontre des principes de non-intrusion définis précédemment (cf. Chapitre 2.2.2). Nous avons donc placé ces sources lumineuses sur le côté, de manière à éviter à l'utilisateur de voir la lumière directement lorsqu'il regarde vers le moniteur. La lumière émise a été adoucie en orientant les sources vers le bas et en y ajoutant des réflecteurs en polystyrène. La quantité de lumière est suffisante dans la journée, avec comme appoint à la lumière diffuse venant de l'extérieur. Mais, le soir, il est nécessaire d'y adjoindre une source placée au-dessus du moniteur et orientée vers le mur, pour ajouter de la lumière diffuse. Cette lumière générera par la suite des problèmes pour la reconnaissance des yeux, notamment pour les personnes portant des lunettes (cf. page 127).

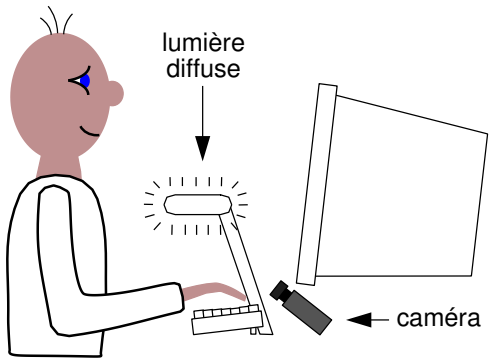


FIG. 3.5 – *Disposition de la Plate-forme.*

Grâce à cet éclairage, il est possible de filmer avec l'obturateur de la caméra réglé à une vitesse de 1/100^{e} de seconde. Le diaphragme de l'objectif est quant à lui ouvert pratiquement au maximum (f 1.3). Il n'est donc pas possible d'augmenter la vitesse d'obturation de la caméra sans ajouter plus de lumière. Cette vitesse est suffisante pour capter des images dans lesquelles il n'y a pas de mouvements rapides, ce qui est généralement le cas. Cependant, nous verrons que lors de certains déplacements du visage, les images sont floues et cela pose des problèmes lors du processus de reconnaissance des formes (cf.

Chapitre 4.1). Nous gardons néanmoins cette configuration pour la plate-forme car elle constitue un bon compromis entre les contraintes de qualité d'images et de confort pour l'utilisateur.

Moniteur

Enfin, la plate-forme est équipée d'un moniteur dont la diagonale mesure 53 cm (21 pouces), ce qui nous permet de réaliser des mesures sur le regard à l'intérieur d'un cadre d'interaction (les objets affichés à l'écran) assez grand.

FIG. 3.6 – *Une personne dans la Plate-forme.*

Conditions d'acquisitions des images

A partir des images captées dans la plate-forme, le système CapRe va réaliser une série de traitements, dont le but est de détecter les yeux et de calculer la direction du regard. Ces traitements nécessitent la prise en compte d'une donnée dynamique : le mouvement (cf. Section 3.2.3). Cela ne peut se faire qu'en utilisant les informations recueillies dans une séquence d'images et à condition de connaître le temps qui s'écoule entre chaque image. Pour mener cela à bien, il faut choisir une fréquence d'échantillonnage appropriée. Nous avons vu que le temps de fixation du regard est en moyenne, au minimum de 200 ms (cf. Contrainte de fonctionnement en temps réel, page 44). Partant de cette valeur, nous savons que la fréquence de changement d'état du regard, c'est-à-dire le passage d'un état de fixation à un autre état de fixation, se fait à la fréquence maximale de 5 Hz. Le théorème de Shannon [Bellanger81c] implique qu'il est nécessaire d'échantillonner la séquence vidéo à plus de 10 Hz si l'on veut être sûr d'avoir toutes les informations requises de la séquence. Or, nous avons expliqué précédemment que la taille des images numérisées

par la carte d'acquisition vidéo, influe sur la fréquence d'échantillonnage. La bande passante du système d'acquisition étant 690 KHz, si l'échantillonnage est réalisé à 10 Hz, il n'est théoriquement pas possible d'avoir des images plus grandes que 304×228 pixels. À cela, vient s'ajouter le fait que la carte d'acquisition n'autorise pas de choisir la taille de l'image au pixel près.

Ratio	1:1	1:2	1:4	1:8
Taille des images	768×576	384×288	192×144	96×72
Fréquence max	1,5 Hz	6 Hz	25 Hz	25 Hz

TAB. 3.1 – *Fréquence d'échantillonnage maximale en fonction de la taille de l'image.*

Le tableau 3.1 montre les possibilités offertes par le système dans le meilleur des cas. Nous voyons qu'il est difficile de choisir des valeurs proches de ce que l'on recherche. En effet, si l'image est trop petite (192×144), le traitement d'image sera plus difficile et les mesures effectuées moins précises. Nous avons donc utilisé une possibilité de la carte d'acquisition, permettant d'avoir un rapport taille/fréquence plus intéressant. Les images vidéo sont composées de 2 trames entrelacées. Il est possible de n'avoir qu'une trame (paire ou impaire) par image. Cela a pour effet de réduire de moitié la hauteur de l'image et de multiplier par 2 la fréquence d'échantillonnage (cf. Tableau 3.2).

Ratio	1:1	1:2	1:4	1:8
Taille des images	768×288	384×144	192×72	96×36
Fréquence max	3 Hz	12,5 Hz	25 Hz	25 Hz

TAB. 3.2 – *Fréquence d'échantillonnage maximale en fonction de la taille de l'image d'une trame (paire ou impaire).*

La fréquence 12,5 Hz pour des images 384×144 d'une trame est un bon compromis. Cela nous permet d'avoir une fréquence élevée tout en gardant des images d'une taille suffisante pour le traitement.

L'idéal serait d'une part de pouvoir échantillonner les images à la fréquence du signal vidéo : 25 Hz pour avoir une plus grande précision des mesures de dynamique du mouvement et permettre un suivi plus fin. D'autre part, s'il était possible d'échantillonner les images à leur taille d'origine (768×576), nous aurions une plus grande précision dans les calculs de mesures de direction du regard. Cependant, les traitements seraient de fait plus longs et cela nécessiterait une machine plus puissante.

Les conditions d'acquisitions des séquences d'images étant définies, nous pouvons décrire l'aspect logiciel du système CapRe au travers de son fonctionnement.

3.2 Fonctionnement

Nous avons vu dans le chapitre précédent que la plate-forme expérimentale permet l'acquisition d'un flot d'images provenant de la caméra. Il faut maintenant décrire comment à partir de ce flot d'images, on peut mesurer la direction du regard de l'utilisateur. La structure logicielle générale du système est présentée, puis chacune de ses sous parties est décrite de manière plus détaillée.

3.2.1 Spécification

Les contraintes définies précédemment engendrent les spécifications logicielles suivantes :

- **Système non-intrusif** : l'utilisateur est donc libre de ces mouvements. Il faut en tenir compte dans les processus de traitements d'image, de manière à ce que le système fonctionne dans les diverses positions et orientations que peut prendre le visage dans l'image. On peut envisager deux types de situations : lorsque l'utilisateur interagit avec la machine et lorsqu'il fait autre chose. Dans la première situation, l'image du visage est exploitable, c'est-à-dire qu'elle contient les informations nécessaires aux traitements, comme nous l'avons vu dans la section précédente. Il suffit donc que les traitements d'images tiennent compte des diverses positions et orientations que peut prendre le visage de l'utilisateur en train d'interagir. Cette spécification est appelée *robustesse* du système aux transformations affines [Varchmin et al.98]. Dans la seconde situation, le visage risque d'être de profil ou de sortir du cadre de l'image, ce qui peut mettre en difficulté les traitements. Le système doit donc tenir compte du fait que les changements de situation de l'utilisateur rendront par moments les mesures difficiles voir impossibles à réaliser. Nous nommons cette spécification *robustesse* du système aux changements de situation de l'utilisateur ;

- **Fonctionnement en temps réel** : dans le cadre de cette application, cela signifie que le système est capable de réaliser tous les calculs nécessaires sur une image avant l'acquisition de l'image suivante. Compte tenu de la fréquence d'échantillonnage de 12 Hz, le système dispose au mieux de 80 millisecondes pour traiter chaque image. Pour réaliser cela, il faut utiliser des algorithmes dont la complexité est faible, c'est-à-dire linéaire. Les traitements du système les plus gourmands en temps de calcul sont les algorithmes de traitement d'image. Or, ces algorithmes sont rarement de complexité linéaire. Il faut donc mettre au point une stratégie de traitement permettant de réduire au maximum l'utilisation d'algorithmes de complexité polynomiale et lorsque l'on doit les utiliser, ne le faire que sur un nombre de données faible ;

- **Précision des mesures** : deux paramètres influent sur cette contrainte, la précision des données en entrée et la précision des algorithmes utilisés pour traiter ces données. Les conditions de capture des images étant fixées, cette contrainte ne dépend plus que de la précision des algorithmes dans la réalisation des diverses mesures nécessaires

au calcul du résultat. La précision d'un algorithme dépend de sa complexité, il s'agit donc de trouver un compromis entre cette contrainte et la précédente. Cette spécification est appelée *précision* du système (*accuracy* [Stiefelhagen et al.96]) ;

- **Fiabilité du système** : il faut inclure dans les processus de traitements des outils permettant d'évaluer la validité des résultats calculés. Ces outils peuvent être des calculs d'erreurs ou de probabilités, cela dépend du contenu des traitements réalisés. Ces évaluations spécifiques aux traitements, doivent permettre ensuite de calculer des scores plus globaux et indépendant des traitements pour que le système prenne des décisions de haut niveau, comme par exemple si un visage a été détecté [2] sans erreur dans l'image ou pas. Cette spécification est appelée *auto-évaluation* (*selfdiagnosis* [Förstner94]) du système. On remarque que cette spécification peut être utile pour apporter une solution pour la prise en compte des changements de situation de l'utilisateur spécifiée ci-dessus ;

- **Utilisation immédiate** :

  - **Système indépendant de l'utilisateur** : des utilisateurs différents ont des visages différents qui donnent des images différentes en entrée du système. Les processus traitant ces images doivent tenir compte de ces différences pour que cela n'influe pas sur les mesures réalisées et les résultats renvoyés. Ces processus devront autant que possible s'appuyer sur les paramètres les plus universels des images des visages et réaliser des traitements spécifiques lorsque cela n'est pas possible. Il est donc nécessaire de spécifier ces paramètres visuels avant de mettre au point les traitements sous-jacents ;
  - **Mise en œuvre simple** : pour que l'utilisateur n'ait pas à se soucier du fonctionnement du système de capture, ce dernier doit décider seul de ce qu'il doit faire. En l'occurrence, il doit être en permanence en état de marche, pour pouvoir détecter la présence de l'utilisateur et lancer le cas échéant des processus permettant le calcul de la direction du regard. Si nécessaire, le système sollicite l'utilisateur pour initialiser certains de ces paramètres de fonctionnement, mais cela doit rester exceptionnel et l'initialisation doit être la plus automatisée possible. Nous nommons cette spécification *auto-initialisation* du système.

Quelles solutions permettent de suivre ces spécifications ? Il est difficile de les respecter toutes. À notre connaissance, il n'existe qu'un seul système qui tienne compte de l'en-

2. Afin d'établir une différence claire entre la détection et la reconnaissance, nous utilisons ces termes dans le sens suivant :

Détection : consiste à vérifier la présence ou l'absence d'un événement ;

Reconnaissance : consiste à associer une valeur symbolique à un événement ;

Événement : caractéristique mesurable de l'information traitée : valeur de luminosité, différence avec un patron, différence temporelle... ;

semble de ces spécifications, celui de Stiefelhagen, Yang et Waibel [Stiefelhagen et al.96]. On trouve des similarités entre ce système et le système CapRe. Les autres études dont nous avons connaissance, ne suivent pas toutes ces spécifications. Par exemple, le système de Varchmin, Rae et Ritter [Varchmin et al.98] ne tient pas compte de la contrainte temps réel, puisqu'il fonctionne à 1 Hz. D'autres études n'ayant pas le même objectif, ne réalisent qu'une sous partie du système de capture. Par exemple, le système de suivi des mouvements du visage de Saulnier, Viaud et Geldreich [Saulnier et al.95] nécessite d'initialiser les paramètres de reconnaissance pour chaque visage, il n'est donc ni indépendant de l'utilisateur, ni auto-initialisant. Nous ne citons ce type de travaux que lorsque nous pensons qu'une partie des solutions utilisées, peut servir d'alternative à celle que nous proposons, sans pour autant sacrifier une des spécifications que nous avons définies.

3.2.2 Structure générale

Quelle structure générale adopter pour un système de capture du regard par caméra ?

Pour répondre à cette question, il faut d'abord définir quels traitements doivent être réalisés par le système. Il s'agit principalement de localiser les yeux dans l'image. Pour arriver à cela, il peut être utile de chercher aussi d'autres éléments dans l'image comme le visage, le nez ou la bouche. Ensuite il faut calculer la direction du regard. De la même manière, il peut être nécessaire de réaliser d'autres calculs avant d'aboutir à la direction du regard. On trouve principalement deux approches dans la littérature : l'une consiste à réaliser une suite de processus qui vise à détecter les éléments les uns à la suite des autres, en utilisant dans chaque processus les informations relevées dans les processus précédents ; l'autre consiste à exécuter des processus permettant de détecter plusieurs candidats possibles pour chaque élément, puis à décider de manière globale quels sont les éléments recherchés parmi tous ces candidats.

Parmi les études exploitant la première approche, on trouve :

- Stiefelhagen, Yang et Waibel [Stiefelhagen et al.96] qui proposent de réaliser une suite de processus. Celle-ci consiste à rechercher dans l'ordre : le visage dans l'image ; les yeux à l'intérieur du visage ; la bouche dans le visage et sous les yeux ; et enfin le nez entre les yeux et la bouche. À partir de la localisation de ces composantes, ils calculent l'orientation du visage. Ils utilisent ce système comme pré-traitement [Stiefelhagen et al.97] pour calculer la direction des yeux, en s'inspirant du système de Baluja et Pomerleau [Baluja et al.94] (cf. page 43) ;
- Varchmin, Rae et Ritter [Varchmin et al.98] utilisent le même principe de suite de processus en cherchant dans l'ordre : le visage dans l'image ; le nez à l'intérieur du visage ; la bouche dans le visage et sous le nez ; et enfin les yeux au dessus du nez. Ils calculent ensuite la direction du regard à partir de l'image des yeux et des coordonnées des composantes du visage ;

– Machin [Machin96] propose de rechercher les yeux en premier dans l'image. Puis à partir de la localisation des yeux, il détermine la position de la boîte rectangulaire qui englobe le visage. Enfin, il cherche la bouche dans le visage et sous les yeux.

Cette approche est utilisée pour deux raisons : elle permet de réduire l'espace de recherche au fur et à mesure de la suite des processus, car elle exploite les informations dès qu'elles sont disponibles. Cela permet de réduire le temps de calcul et donc d'assurer un fonctionnement en temps réel du système ; elle permet aussi de commencer par chercher l'élément le plus "facile" à trouver dans l'image, ce qui rend la détection plus sûre et évite de propager des erreurs dans la suite des processus.

La seconde approche est plus coûteuse en temps de calcul. De ce fait, les études qui l'exploitent ne fonctionnent pas en temps réel ([Yow et al.97] [Yow et al.98] [DeCarlos et al.98]) ou bien ne réalisent qu'une sous partie d'un système de capture du regard ([Graf et al.95] [Darrell et al.96] [Reinders et al.96] [Birchfield98]). Elle a cependant l'avantage de réaliser une détection plus robuste des éléments et d'éviter toute propagation d'erreur de détection d'un élément vers les autres.

Nous choisissons la première approche, pour permettre le fonctionnement du système en temps réel. Il faut déterminer quelle suite de processus nous devons réaliser pour d'une part localiser les yeux dans l'image et d'autre part calculer la direction du regard.

Localiser les yeux dans l'image

Nous pouvons constater que selon les études présentées ci-dessus, les éléments ne sont pas détectés dans le même ordre.

Machin [Machin96] cherche les yeux en premier. Il explique que les yeux varient peu selon les sujets et l'orientation du visage, et qu'ils sont rarement cachés quand le sujet regarde l'écran. Les traitements utilisés pour détecter les yeux sont gourmands en temps de calcul parce qu'ils sont basés sur des algorithmes de complexité polynomiale appliqués sur toute l'image. De ce fait, son système ne fonctionne qu'à une fréquence de 5 Hz. Crowley et Bérard [Crowley et al.97] proposent de détecter les clignements des yeux. Cela peut être réalisé par des algorithmes de complexité linéaire mais cette approche présente quelques défauts : l'échantillonnage doit être fait à une fréquence supérieure à 12 Hz (cf. Section 2.1.2.4) ; il faut attendre un clignement (3 secondes en moyenne) ; la technique utilisée ne détecte pas des yeux mais un mouvement ce qui peut générer des erreurs. Nous pensons qu'il est préférable de détecter dans un premier temps le visage dans l'image, comme cela est fait dans les autres études. En effet, il est possible de détecter le visage en utilisant des traitements rapides basés sur des algorithmes de complexité linéaire.

Savoir où se trouve le visage dans l'image suffit-il pour rechercher les yeux de manière efficace, c'est-à-dire en temps réel ? Les yeux sont en fait, des éléments complexes et

donc difficiles à détecter ([Saulnier et al.95]), même si l'on n'applique la recherche qu'à l'intérieur du visage. Il peut être intéressant de rechercher une autre composante du visage, pour réduire encore l'espace de recherche des yeux. On sait que les yeux sont dans la partie supérieure du visage, il est donc intéressant de rechercher une composante générant une borne inférieure dans la visage. Connaître la localisation du nez permet de déterminer cette borne inférieure mais aussi de séparer un espace de recherche pour chaque œil, d'un côté et de l'autre par rapport au nez. De plus, c'est un élément simple à détecter ([Varchmin et al.98] [Petajan et al.96]) surtout si l'image est captée en contre-plongée, car cela permet de voir clairement les narines. Le second élément de l'image à rechercher dans CapRe est donc le nez. Enfin, à partir de la localisation du visage et du nez, on peut réaliser une recherche complexe des yeux dans une partie réduite de l'image.

Calculer la direction du regard

Le fait de chercher le nez dans l'image ne sert pas uniquement à préparer la recherche des yeux. Il faut aussi penser à la suite des traitements, c'est-à-dire le calcul de la direction du regard.

On sait qu'il est possible de calculer la direction du regard à partir de l'image des yeux ([Baluja et al.94]), mais c'est une mesure locale par rapport au visage. En effet, si la tête se déplace, la direction change selon ce déplacement en translation ([Charbonnier95], p.37) et/ou en rotation. Stiefelhagen et al. [Stiefelhagen et al.97] constatent comme nous l'avons fait [Collet et al.97a], qu'il est nécessaire de réaliser une mesure d'orientation des yeux par rapport au visage et une mesure de l'orientation et de la localisation du visage dans l'espace pour pouvoir calculer la direction du regard dans l'espace, c'est-à-dire par rapport à la plate forme expérimentale.

C'est en fait, un problème complexe car le calcul de la direction des yeux par rapport au visage est dépendant de l'orientation du visage. L'information de base pour ces calculs est l'image des yeux, or celle-ci varie en fonction de l'orientation du visage. Varchmin et al. [Varchmin et al.98] proposent une solution qui consiste à évaluer la direction des yeux par rapport au visage puis à intégrer les deux autres calculs, orientation du visage et direction du regard dans l'espace, dans un seul outil de calcul à apprentissage automatique appelé un réseau de projections linéaires locales (*Local Linear Map-network*). Ils n'ont pas réussi pour l'instant à intégrer cette partie dans leur système de manière à ce qu'elle soit indépendante de l'utilisateur. Stiefelhagen et al. [Stiefelhagen et al.96] proposent d'utiliser un modèle 3D du visage permettant de calculer son orientation dans l'espace à partir des localisations d'éléments dans l'image. Ils réalisent le calcul de la direction des yeux par rapport au visage mais n'ont pas intégré le calcul de la direction du regard dans l'espace [Stiefelhagen et al.97]. Il serait nécessaire de réaliser une étude spécifique pour déterminer une méthode de calcul appropriée. Le préambule à cette étude est de disposer d'un système de détection des yeux et des autres éléments utiles, renvoyant des résultats exploitables en

terme de précision, de fiabilité et de robustesse. Il est donc important de définir la structure générale du système en tenant compte de ces problèmes, même si nous n'étudions pas une solution complète pour le calcul de la direction du regard dans l'espace.

Structure générale de CapRe

Nous distinguons deux parties dans l'enchaînement des processus nécessaires au traitement. La première partie dite processus de détection et de suivi, doit extraire, à partir des images, les informations de type visuel : détection du visage, du nez et des deux yeux. À partir des résultats produit par celle-ci, la seconde partie dite processus de mesures, doit calculer des projections dans l'espace pour évaluer la direction du regard. La figure (3.7) décrit la succession des opérations effectuées à l'intérieur de ces deux parties. Chacune de ces opérations est nécessaire pour le traitement à effectuer selon les spécifications décrites dans la section précédente (cf. Section 3.2.1).

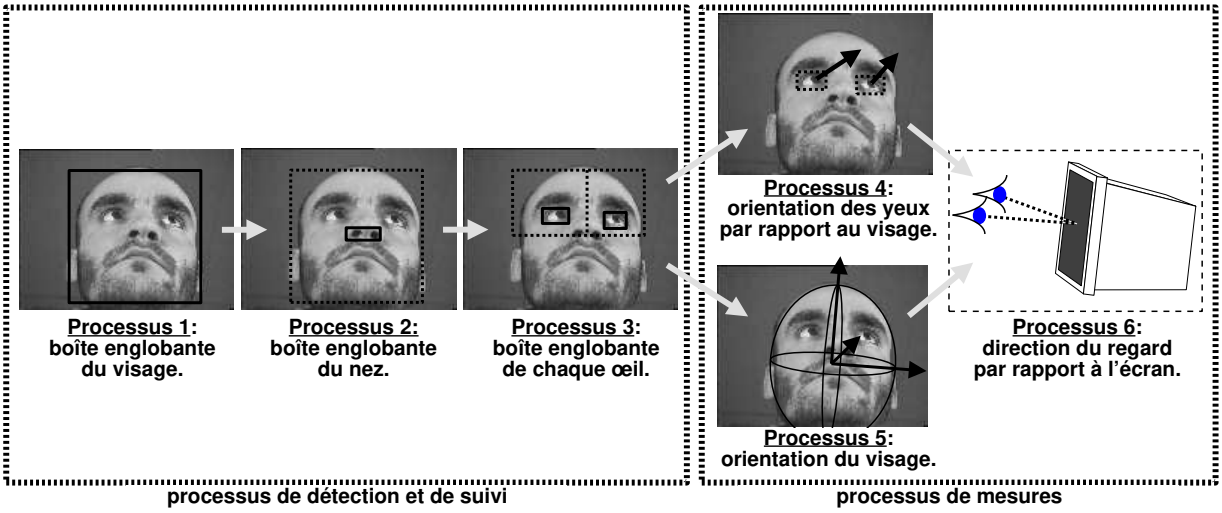


FIG. 3.7 – *Structure générale du fonctionnement de CapRe.*

Les processus de détection et de suivi

Processus 1 : *Calcul de la boîte englobante du Visage*

Entrée : ce processus prend en entrée une image monochrome fournie soit par le système de numérisation d'images vidéo, soit par un fichier contenant une séquence préalablement enregistrée ;

Traitement : il cherche à extraire de cette image la boîte englobante du visage de l'utilisateur, c'est-à-dire une zone rectangulaire contenant le visage ;

Sortie : il renvoie comme résultat soit une zone rectangulaire de l'image incluant le visage (boîte englobante), soit l'information de non-détection de visage.

Processus 2 : *Recherche du Nez*

Entrée : ce processus prend en entrée l'image utilisée par le premier processus et la boîte englobante détectée par celui-ci ;

Traitement : il recherche le nez à l'intérieur de la boîte englobante du visage et calcule sa boîte englobante ;

Sortie : il renvoie soit la localisation et la boîte englobante du nez dans l'image, soit l'information de non-détection du nez.

Processus 3 : *Recherche des Yeux*

Entrée : ce processus prend en entrée l'image utilisée par les processus précédents, les boîtes englobantes détectées par ceux-ci et la localisation du nez ;

Traitement : il détermine deux zones, une pour chaque œil, à partir de la boîte englobante du visage et de la localisation du nez. Ces zones sont au-dessus et de chaque côté du nez, dans la boîte englobante du visage. Puis il recherche un œil dans chacune de ces zones, calcule sa boîte englobante et la position du centre de l'iris (la pupille) ;

Sortie : il renvoie soit la localisation et la boîte englobante de chaque œil ainsi que la localisation de la pupille dans l'image, soit l'information de non-détection d'un œil ou des deux yeux.

Les processus de mesures

Processus 4 : *Orientation des yeux*

Entrée : ce processus prend en entrée l'image utilisée par les processus précédents, les boîtes englobantes des yeux et les localisations des pupilles ;

Traitement : il évalue l'orientation de chaque œil par rapport au visage en calculant un vecteur dans le plan de l'image ;

Sortie : il renvoie un vecteur d'orientation pour chaque œil.

Processus 5 : *Orientation du visage*

Entrée : ce processus prend en entrée l'image utilisée par les processus précédents et les boîtes englobantes des yeux et du nez ;

Traitement : il évalue l'orientation du visage par rapport à la plate-forme expérimentale en calculant un vecteur dans l'espace grâce à un modèle de projection 2D vers 3D ;

Sortie : il renvoie un vecteur d'orientation du visage.

Processus 6 : *Direction du regard*

Entrée : ce processus prend en entrée les vecteurs d'orientation des deux yeux (Processus 4) et du visage (Processus 5) ;

Traitement : il évalue la direction du regard par rapport à l'écran ;

Sortie : il renvoie les coordonnées du regard dans le plan défini par l'écran.

Ayant présenté la structure générale du système, nous allons pouvoir entrer plus en détail dans la description de chaque processus.

3.2.3 Processus de détection et de suivi

La description que nous avons fait des processus correspond à une vision statique du système. En effet, nous avons spécifié la structure du système pour traiter une image. En l'occurrence, chaque processus doit détecter un élément dans l'image : le visage, le nez ou les yeux. Mais, nous avons à traiter une séquence d'images et cela doit nous permettre de mesurer et d'utiliser des informations liées à la dynamique.

Quelle structure pour traiter des séquences d'images ?

L'approche utilisée en général lors du traitement d'une séquence d'images, consiste à "suivre" un ou plusieurs objets d'une image à l'autre. Cette approche permet de restreindre l'espace de recherche de l'objet dans la nouvelle image, autour de sa localisation dans l'image précédente. Elle est appliquée dans divers contextes comme le suivi d'objets rigides ([Courtney et al.97]), de personnes ([BG et al.94] [Wren et al.96] [Gavrila et al.96]), de visages ([Hunke et al.94] [Collobert et al.96] [DeCarlos et al.96] [Sobottka et al.96] [Crowley et al.97] [Birchfield98] [McKenna et al.96] [Yang et al.98]), de gestes ([Bérard et al.96]), d'expressions ([Essa et al.95]) ou de véhicules ([BG et al.94]). Cette approche est aussi appliquée pour suivre des composantes du visage comme les yeux, le nez ou la bouche ([Oliver et al.96] [Stiefelhagen et al.96]). Cette approche nécessite de résoudre deux problèmes :

- Le premier problème est que le suivi est réalisé sur un objet préalablement localisé dans la première image de la séquence. Cette localisation peut être manuelle ([Black et al.95] [DeCarlos et al.96] [Bérard et al.96] [Birchfield98]) ou automatique en réalisant la détection de l'objet dans l'image ([BG et al.94] [Essa et al.95] [Sobottka et al.96] [McKenna et al.96] [Collobert et al.96] [Stiefelhagen et al.96] [Courtney et al.97]) ;

- le second problème est qu'il est possible que le système "perde" l'objet qu'il suit. Cette perte peut être due à une occultation totale ou partielle de l'objet ou à tout autre événement perturbant le système de suivi [McKenna et al.96], par exemple lors des changements de situation de l'utilisateur que nous évoquons dans les spécifications du système (cf. Section 3.2.1). Il peut être nécessaire dans ce cas de refaire une localisation de l'objet dans l'image pour pouvoir reprendre le suivi.

Quelles solutions apporter à ces deux problèmes ?
Le premier problème est associé à une des spécifications du système qui est l'auto-initialisation. Le système doit donc être capable de localiser par des processus de détection, les composantes du visage avant de les suivre, comme cela se fait dans les études que nous référençons ci-dessus. Le second trouve sa solution dans la spécification appelée auto-évaluation du système. En effet, un système qui est capable d'évaluer s'il suit une composante du visage ou s'il l'a perdue, peut décider de poursuivre ou non les processus de suivi. S'il doit arrêter, il se ré-initialise en appliquant les processus de détection de la composante concernée. Il est possible d'exploiter ces deux solutions en créant une structure dynamique du fonctionnement du système. Nous proposons de modéliser cette structure dynamique en utilisant deux "états" de fonctionnement : un "état" qui utilise les processus de détection des composantes dans une image et un "état" qui utilise des processus de suivi de ces composantes d'une image à l'autre. Le premier se nomme **"état d'initialisation"** et le second **"état d'adaptation"**. La transition d'un "état" vers l'autre est décidée par le système grâce à ses capacités d'auto-évaluation. Dans un premier temps, le système se trouve dans l'"état d'initialisation" et cherche les composantes du visage dans chaque image. Si le système évalue qu'il a détecté les composantes du visage sans erreur (ce qui peut prendre plusieurs images), il transite vers l'"état d'adaptation". Dans cet "état d'adaptation", le système suit les composantes du visage d'une image à l'autre. S'il évalue qu'il a perdu une composante (ce qui peut aussi prendre plusieurs images), il transite vers l'"état d'initialisation", et ainsi de suite (Schéma 3.8). Crowley et Bérard [Crowley et al.97] utilisent le même principe mais avec deux états d'initialisations : l'un détectant uniquement le clignement des yeux qui est utilisé pour amorcer le système et l'autre détectant en plus le visage grâce à la couleur de la peau et qui est utilisé pour ré-initialiser le système.

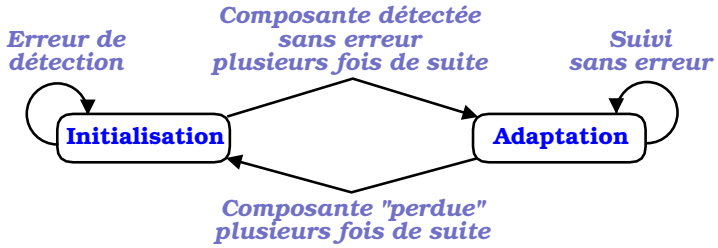


FIG. 3.8 – *Transitions entre les deux "états" du système.*

Cette solution permet en outre de mettre en œuvre une stratégie plus générale sur les traitements pour respecter les spécifications du système. Notamment, en ce qui concerne la spécification du fonctionnement indépendamment de l'utilisateur (cf. Section 3.2.1), dans laquelle nous proposons d'utiliser des paramètres universels des images des visages pour y détecter les composantes recherchées. Par ailleurs, il est clair que pour que le système soit fiable, il est préférable d'utiliser des paramètres spécifiques au visage de la personne filmée. Les paramètres spécifiques pour une composante du visage, peuvent être mesurés

dans une image à partir du moment où l'on a détecté cette composante. Cette méthode est utilisée dans plusieurs systèmes de suivi ([BG et al.94] [Oliver et al.96] [Stiefelhagen et al.96] [Crowley et al.97] [Birchfield98]). Nous proposons d'étendre la définition des deux "états" de fonctionnement du système pour y intégrer cette spécification. Cette structure étant dépendante de la "réussite" du processus appliqué à chaque composante, il est plus efficace de l'appliquer indépendamment (avec des "états" différents) pour chaque processus de recherche de composante (visage, nez et yeux) :

- dans l'"état d'initialisation", le système tente de détecter la composante du visage sans connaissance spécifique de celle-ci. Les seules connaissances qu'il peut utiliser sont des informations générales (ou universelles) et *a priori* sur cette composante ;

- dans l'"état d'adaptation", le système tente de suivre la composante du visage en tenant compte des mesures réalisées dans les images précédentes. Il utilise par conséquent des informations spécifiques au visage de la personne filmée.

Nous allons décrire le détail de fonctionnement de chacun de ces deux états.

3.2.3.1 État d'initialisation

Lorsqu'un processus se trouve dans l'état d'initialisation, il tente de détecter la composante qui le concerne (le visage, le nez ou les yeux) dans l'image. Nous avons défini cet état de manière à permettre au système de s'amorcer automatiquement lorsque que l'utilisateur vient prendre position devant la machine. A cet instant nous n'avons pas de connaissances spécifiques sur cet utilisateur. Nous ne pouvons utiliser que des connaissances générales applicables à la plupart des individus. Nous devons notamment tenir compte des différences inter individus, comme la morphologie du visage ou la couleur de la peau, et des différences intra individus comme la coiffure, la pilosité ou le port de lunettes. De ce fait, même si l'on utilise un système d'exploitation dans lequel l'utilisateur doit s'identifier (avec un nom et un mot de passe) pour pouvoir travailler, comme sur UNIX, ou si l'on demande à l'utilisateur de s'identifier ou de donner des informations sur son visage pour utiliser le système de capture, il restera au système à gérer les différences intra individus. Nous avons choisi d'étudier la faisabilité d'un système capable de gérer à la fois les différences intra individus et les différences inter individus.

Comment satisfaire le spécification de fiabilité du système ? Nous proposons de réaliser une succession de traitements visant à détecter la composante par reconnaissance, puis de valider cette détection en utilisant la cohérence temporelle, et enfin de décider si la composante à été détectée en fonction de la confiance que l'on accorde au résultat. Le schéma (3.9) présente la structure générale d'un processus dans l'état d'initialisation. Ce schéma est décrit par procédure ci-dessous.

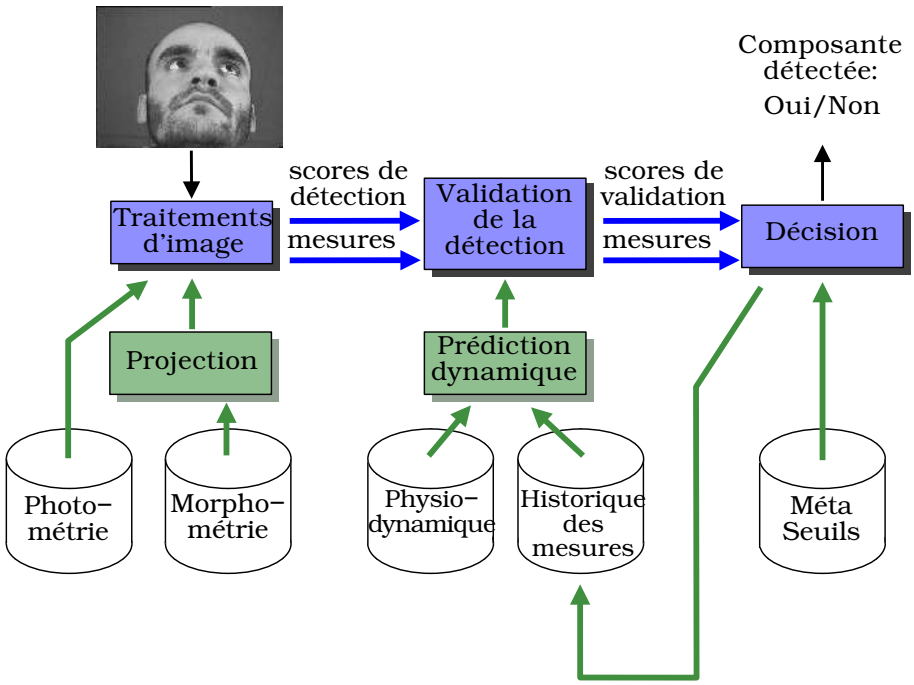


FIG. 3.9 – *Structure d'un processus dans l'état d'initialisation.*

Traitements d'image

Dans un premier temps, le processus (dans l'état d'initialisation) réalise des traitements sur toute ou partie de l'image. Ceux-ci utilisent deux types d'informations : des informations photométriques et des informations morphologiques. Les informations photométriques sont principalement des seuils ou des modèles, liés à la luminosité ou permettant aux processus de s'adapter aux conditions de luminosité de l'image. Elles ont été déterminées par mesure et réglées plus précisément par expérimentation. Les informations morphologiques (ou morphométriques) sont des seuils ou des intervalles définissant grossièrement la forme de la composante. Elles correspondent à des valeurs définies en anthropométrie mais que nous avons adaptées en fonction des mesures faites sur les images captées par la caméra. Une fonction de **projection** permet de transformer ces valeurs de mesures métriques en unités de pixels utilisées par les traitements d'image. Ceux-ci produisent à la fois des valeurs de mesure sur la composante (localisation, taille ...) et un ou plusieurs scores de détection. Ces scores sont le reflet de la fiabilité de la reconnaissance et des mesures effectuées. De plus, ces processus ne renvoient pas un seul résultat mais proposent, quand cela est possible, plusieurs candidats potentiellement considérés comme étant la composante recherchée. Chacun des candidats est accompagné des mesures et scores le concernant. C'est seulement, à la fin de la série de traitements que le meilleur candidat est sélectionné en fonction de son ou de ses scores de détection. C'est ce candidat qui est utilisé pour la suite.

Validation de la détection

L'étape suivante prend en entrée les valeurs mesurées et les scores du candidat. Elle utilise des informations sur la physiologie du mouvement et les mesures réalisées dans les images précédentes. Ainsi, elle peut vérifier la cohérence spatio-temporelle entre la composante candidate et les composantes précédentes. Cette opération consiste principalement à calculer l'accélération du déplacement de la composante pour extrapoler sa nouvelle localisation dans l'image (**prédiction dynamique**). Cette valeur est utilisée avec les autres scores de détection comme score de validation.

Décision

Enfin, il est possible à partir des scores de validation et de détection de décider si la composante a été reconnue. La méthode la plus simple consiste à utiliser des seuils, qui comme ils ne sont pas liés directement aux données traitées, sont appelés méta seuils. Le processus calcule un score "global" dit de confiance de détection à partir des scores de validation et de détection.

D'une image à l'autre, le processus va accumuler des scores de confiance de détection. Si ces scores sont assez élevés sur plusieurs images successives, on peut considérer que non seulement la composante est bien reconnue mais de plus cette reconnaissance est stable dans le temps. Étant sûr d'avoir détecté la composante, le processus peut transiter vers l'état d'adaptation.

3.2.3.2 État d'adaptation

Lorsqu'un processus se trouve dans cet état, il n'est plus en train de faire de la détection de composante. En effet, on considère que la détection a déjà été effectuée dans l'état d'initialisation et qu'elle est correcte (sans erreur). Cet état consiste donc à faire du suivi de composante. Il dispose d'informations spécifiques sur la composante pour réaliser la reconnaissance, mais surtout il sait où la chercher dans l'image. Cet état permet à la fois de satisfaire les contraintes de robustesse et de fonctionnement en temps réel.

Comment réaliser le suivi de la composante ? Nous écartons les techniques de suivi employant des algorithmes de complexité non linéaire comme la corrélation ([Courtney et al.97] [Crowley et al.97] [Bérard et al.96]), le flot optique ([DeCarlos et al.96]), de convolution temporelle de pixels ([McKenna et al.96]). On peut aussi utiliser la localisation de l'élément dans l'image précédente et réaliser une recherche dans la nouvelle image, en utilisant des techniques comme les contours actifs [Sobottka et al.96], l'appariement de formes géométriques simples comme une ellipse [Birchfield98] ou la segmentation par la couleur de la peau [Collobert et al.96]. Ces techniques utilisent une localisation de départ pour la recherche mais n'ont pas de limites *a priori* pour terminer le traitement. Hunke et Waibel [Hunke et al.94] utilisent en plus de la localisation, la boîte englobante de la composante détectée pour définir une zone de recherche limitée dans l'image suivante. Pour

permettre une certaine tolérance aux mouvements entre les deux images, ils multiplient la taille de la zone de recherche par un coefficient fixe : deux fois la taille de la boîte englobante. Les mouvements étant variables en direction et en vitesse, il semble intéressant que ce coefficient soit lui aussi variable pour éviter de trop agrandir ou trop réduire la zone de recherche. On peut utiliser un estimateur récursif de Kalman pour calculer la localisation et la taille de la boîte englobante de la composante, en tenant compte de sa vitesse et de la direction de déplacement mesurées dans les images précédentes ([Oliver et al.96] [Courtney et al.97] [DeCarlos et al.97] [Crowley et al.97]).

Nous proposons une solution plus simple mais proche de celle utilisant l'estimateur de Kalman. Elle consiste à prédire la localisation et la taille la boîte englobante de la composante en les extrapolant à partir de la boîte englobante, de la vitesse et de l'accélération mesurées dans l'image ou les images précédentes. Cette procédure appelée **prédiction dynamique** est aussi utilisée dans l'état d'initialisation, mais pas au même moment.

Ensuite, on peut réaliser une succession de traitements visant à valider cette prédiction en détectant la composante dans la zone extrapolée de l'image (**Traitements d'image**). Enfin le système peut décider si la composante a bien été suivie en fonction de la confiance qu'il accorde au résultat (**Décision**). Le schéma (3.10) présente la structure générale d'un processus dans l'état d'adaptation.

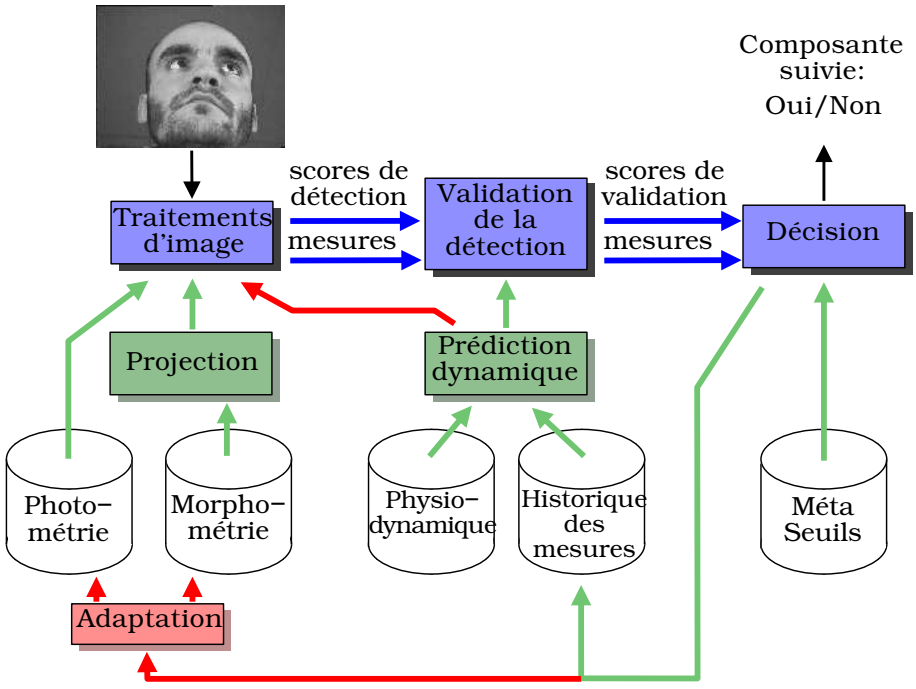


FIG. 3.10 – *Structure d'un processus dans l'état d'adaptation.*

Si l'on compare ce schéma avec celui décrivant la structure de l'état d'initialisation, on constate qu'un flot de donnée a été ajouté entre les procédures de **Prédiction dynamique** et de **Traitements d'image**, et qu'il y a un traitement en plus correspondant à l'adaptation des paramètres de reconnaissance (**Adaptation**). Les deux procédures de **Traitements d'image** et de **Prédiction dynamique**, changent de rôle mais réalisent les mêmes fonctions que dans l'état d'initialisation. La **Prédiction dynamique** permet de réduire la zone de recherche de la composante dans l'image et par la même le nombre de candidats potentiels à traiter par les **Traitements d'image**, et donc le nombre d'erreurs ainsi que le temps nécessaire. À la sortie de cette procédure, on dispose comme dans l'état d'initialisation d'un candidat pour la composante à suivre et de son score de détection.

Adaptation

D'une image à l'autre, le processus va continuer à accumuler des scores de confiance de détection. Si ces scores sont assez élevés sur plusieurs images successives, le processus est d'autant plus sûr d'avoir détecté et de suivre la bonne composante. Il va donc en profiter pour adapter ses paramètres de reconnaissance pour qu'ils deviennent plus spécifiques à l'utilisateur et plus proches des conditions de capture des images. Cette adaptation consiste principalement à reprendre des mesures de la composante sur l'image afin de mettre à jour les valeurs photométriques et morphométriques utilisées pour reconnaître cette composante. Ces valeurs mises à jour seront utilisées tant que le processus reste dans l'état d'adaptation.

Si les scores de confiance de détection sont faibles sur plusieurs images successives, le processus considérant qu'il a "perdu" la composante ou qu'il suit une mauvaise composante, va transiter vers l'état d'initialisation. Les valeurs spécifiques, dans l'état d'adaptation, sont alors remplacées par les valeurs universelles utilisées initialement.

La structure et le fonctionnement des deux états dans lesquels peuvent se trouver les trois processus étant décrits, nous allons détailler chaque processus en commençant par le calcul de la boîte englobante du visage.

3.2.3.3 1er processus : Calcul de la boîte englobante du Visage

Ce processus prend en entrée l'image numérisée saisie par la caméra et doit déterminer les coordonnées d'une zone rectangulaire dans laquelle se trouve le visage de l'utilisateur : la boîte englobante. Ce processus a pour but, d'une part de permettre de réduire l'espace de recherche pour les processus suivants 2 et 3 (recherche du nez et des yeux, cf. Section 3.2.3.4) ; et d'autre part de collecter des informations, notamment sur la dynamique, utiles pour les processus suivants et aussi pour traiter les images suivantes dans

ce processus. Cette démarche apporte les avantages suivants :

- Les processus de détection du nez et des yeux (Processus 2 et 3) utilisent des traitements gourmands en temps de calcul. Les appliquer dans une partie réduite de l'image, permet par conséquent un gain sur ce temps de calcul ;

- Ces mêmes traitements ne donnent pas un résultat sûr à 100 %. Ils peuvent confondre l'élément qu'ils recherchent dans l'image avec d'autres éléments notamment dans le fond de l'image, à l'extérieur du visage. S'ils ne sont appliqués que dans la boîte englobante du visage, cela réduit le nombre d'erreurs potentielles ;

- Les algorithmes et les paramètres utilisés dans ces traitements peuvent être optimisés compte tenu du fait qu'ils ne sont appliqués la plupart du temps que sur une image ne contenant que le visage. Ceci étant une conséquence directe de l'avantage précédent.

Comment calculer la boîte englobante du Visage ?

Il existe plusieurs méthodes permettant de déterminer quels sont les pixels d'une image qui appartiennent à un visage. On peut en distinguer deux principales : l'une utilisant la couleur de la peau ([Wu et al.95] [Sobottka et al.96] [Stiefelhagen et al.96] [Crowley et al.97]), l'autre utilisant les éléments qui composent le visage (yeux, nez, bouche, sourcils, menton, cheveux . . .) ([Samaria93] [Moghaddam et al.95] [Yow et al.98]). En général, et lorsque cela est possible, c'est une combinaison de ces deux approches qui est utilisée ([Collobert et al.96] [Birchfield98]). Nous décrivons deux méthodes qui sont proches des spécifications de notre système.

Stiefelhagen, Yang et Waibel [Stiefelhagen et al.96], utilisent un modèle statistique de la couleur de la peau. Ce modèle consiste en une distribution gaussienne en deux dimensions des couleurs de la peau normalisées [Hunke et al.94]. Cela permet de faire correspondre à chaque pixel de couleur une valeur dans la surface gaussienne. Ils obtiennent ainsi une image de probabilités des pixels d'appartenir à la couleur de la peau. Ils utilisent ensuite un seuil et calculent la boîte englobante de la plus grande surface de composantes connexes, considérée comme étant le visage. Leur système fonctionne à 15 Hz, mais ils expliquent qu'il n'est pas nécessaire de localiser le visage dans chaque image, car celui-ci ne se déplace pas vite. De ce fait, ils ne réalisent la détection du visage qu'une image sur cinq ou sur dix. Nos observations montrent que cela n'est pas suffisant, car le visage peut se déplacer rapidement et il est nécessaire de le localiser dans toutes les images s'il l'on ne veut pas risquer de le perdre momentanément.

Stan Birchfield [Birchfield97] a mis au point une technique de suivi (sans reconnaissance) de la tête d'une personne, utilisant un simple modèle 2D : une ellipse. Il ajuste la position et la taille de l'ellipse en maximisant la somme des gradients de l'image autour

de l'ellipse. Le traitement est effectué localement dans le voisinage de la position détectée dans l'image précédente. Le système fonctionne en temps réel (30 Hz) et peut piloter le système d'orientation d'une caméra pour maintenir le sujet au centre de l'image. Le suivi est possible quelle que soit l'orientation de la tête. Il évalue un score de confiance des mesures d'appariement entre l'ellipse et la tête qui semble efficace, l'auteur ne donnant pas d'évaluation quantitative des résultats. Il y a cependant des cas où le système a du mal à suivre une tête, notamment lorsque le fond de l'image est complexe. Pour résoudre ce problème, Stan Birchfield ajoute à son système un module s'intéressant à l'histogramme des couleurs à l'intérieur de l'ellipse (et donc de la tête) [Birchfield98]. Ce module effectue un appariement entre histogrammes dans l'ellipse d'une image sur l'autre. Les deux modules se complètent et forment un système plus performant, tout en fonctionnant en temps réel. Il peut fonctionner avec des images où se trouvent plusieurs personnes (il n'en suit qu'une) et tolère les mouvements de caméra et les occultations momentanées. Cette évaluation reste qualitative.

Aucune de ces méthodes ne peut être employée pour notre système, soit parce qu'elles ne sont pas assez rapides, soit parce qu'elles utilisent la couleur, ce dont nous ne disposons pas. Nous avons donc choisi de développer une technique correspondant aux contraintes du système et qui exploite les conditions d'utilisations de la plate forme expérimentale. Nous considérons que la plupart du temps il n'y a qu'une seule personne face à la station de travail. Donc, le processus de détection et de suivi du visage n'a pas à discriminer l'utilisateur parmi d'autres visages. D'autre part, la position et l'orientation de la caméra en contre-plongé (cf. page 49) permet d'acquérir des images où le fond est constitué principalement du plafond de la salle. Lorsque l'on observe les images prises par la caméra, on remarque que l'utilisateur est en général en mouvement, alors que le fond est lui toujours immobile. Un simple algorithme de détection de mouvement, doit donc permettre de discriminer le visage du reste de l'image.

3.2.3.3.1 Détection du mouvement

Nous cherchons à distinguer les pixels de l'image appartenant au visage de l'utilisateur. La première idée qui vient à l'esprit est d'utiliser une image dans laquelle il n'y a que le fond et qui sert d'image de référence. Si l'on fait la différence entre celle-ci et une autre image, on fait ressortir les pixels dans lesquels se trouve probablement l'utilisateur. Un seuillage de l'image résultante de cette différence permet de décider quels sont les pixels qui ont changé. Si aucun pixel ne ressort de cette différence, on en déduit qu'il n'y a pas d'utilisateur. Cette méthode, simple à mettre en œuvre, a deux défauts : d'une part il est nécessaire d'enregistrer une image de référence dès que l'on change la position de la caméra ; d'autre part les variations lumineuses font apparaître des différences entre les images sur des objets identiques et immobiles. Pour résoudre ce dernier problème, on peut mettre en œuvre des techniques d'adaptation de seuil. Mais celles-ci trouvent leurs limites dès que les variations sont trop grandes. Nous avons donc préféré agir différemment. La fréquence d'échantillonnage des images permet de disposer d'une image tous les 8/100$^{\text{es}}$ de

seconde (cf. page 54). Dans cet intervalle, l'utilisateur n'a le temps de bouger qu'au plus de quelques centimètres et les changements d'intensité de la lumière sont faibles. La différence entre deux images successives donne donc comme résultat une **image de mouvement** de l'utilisateur. Cette opération correspond au calcul d'un gradient temporel sur chaque pixel de l'image (cf. Schéma 3.11). Un seuillage de cette image permet de déterminer quels sont les pixels qui appartiennent à l'utilisateur dans les deux images. Ainsi, le résultat ne correspond pas strictement à ce que nous cherchons dans l'image, mais à l'union des deux images de l'utilisateur. Le mouvement entre les deux images étant de faible ampleur, l'erreur générée par cette méthode est aussi faible. Le seuil qui permet de décider quel est le niveau de gradient temporel minimum pour qu'un pixel soit considéré comme étant en mouvement, est fixe. Sa valeur est déterminée empiriquement.

Compte tenu de la taille de l'objet à détecter proportionnellement à l'image (le visage occupe entre la moitié et le tiers de l'image), nous pouvons réaliser cette opération sur une image sous-échantillonnée sans pour autant que la perte de précision soit significative. Cela nous permet un gain sur le temps de traitement. La taille de l'image est donc divisée par 32 : la longueur est divisée par 8 et la hauteur qui a déjà un ratio de 1/2 par rapport à la longueur (cf. page 54), est divisée par 4.

L'image de gradient temporel de l'utilisateur, une fois seuillée, contient des pixels qui lorsqu'ils ont pour valeur "en mouvement" appartiennent *a priori* à l'utilisateur. La position et l'orientation de la caméra permettent de réaliser un cadrage de l'utilisateur où le visage est centré. Nous pouvons donc considérer que le contour externe des "pixels en mouvements" correspond au contour de la tête de l'utilisateur. Le modèle de détection du contour du visage est un rectangle. Le résultat est appelé boîte englobante du visage. Le traitement consiste donc à décider où sont les bords de la boîte englobante dans l'image de manière à être sûr d'y trouver le visage. Dans un premier temps le système détecte les contours de part et d'autre du visage, en scrutant l'image horizontalement. Le traitement est réalisé du haut vers la bas de l'image. Dès qu'un mouvement est considéré, ses coordonnées sont enregistrées et la scrutation reprend en partant de l'autre côté de l'image. S'il n'y a pas deux mouvements sur cette ligne, le système n'en tient pas compte. Lorsque le traitement de la ligne est terminé, il passe à la ligne suivante (cf. Figure 3.12). Le système dispose, une fois ce traitement réalisé, des coordonnées des premiers "pixels en mouvement" rencontrés des deux côtés, sur chaque ligne où il y a du mouvement. En calculant la moyenne des abscisses des "pixels en mouvement" de chaque côté, on obtient les bords à gauche X_g et à droite X_d de la boîte englobante (cf. équations 3.2 et 3.4). Cette moyenne ne correspond pas réellement au bord du visage. La forme elliptique du visage fait qu'il faut ajouter une marge à la moyenne pour être sûr que le visage est à l'intérieur de la boîte englobante. Nous avons choisi de prendre l'écart type comme marge (cf. équations 3.3 et 3.4). Les équations suivantes ont été utilisées pour ces calculs, où : E_mvt est l'ensemble des lignes contenant au moins deux "pixels en mouvement" ; $x_d(l)$ est l'abscisse du premier "pixel en mouvement" de la ligne l dans E_mvt (contour droit du visage) ; $x_g(l)$ est l'abscisse du dernier "pixel en mouvement" de la ligne l dans E_mvt

(contour gauche du visage) ; et $card(E\_mvt)$ est le nombre de lignes dans E_mvt.

FIG. 3.11 – *Schéma du calcul du gradient temporel.*

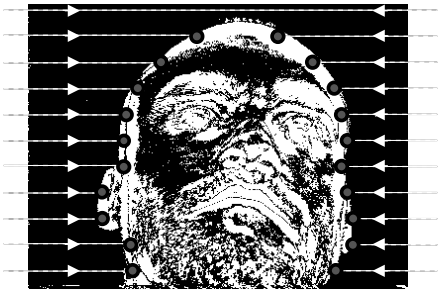


FIG. 3.12 – *Schéma de la détection des bords du visage.*

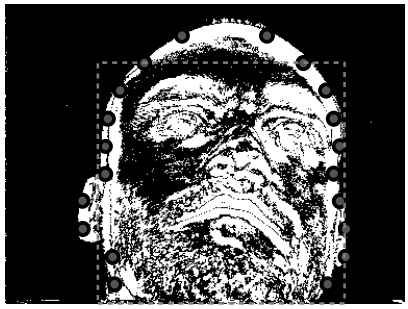


FIG. 3.13 – *Schéma de la boîte englobante du visage.*

$$
\begin{array}{rcl}
Soit \quad N & = & card(E\_mvt) \\
\overline{X_d} & = & \frac{\sum_{l=1}^{N} x_d(l)}{N} \qquad\qquad \overline{X_g} = \frac{\sum_{l=1}^{N} x_g(l)}{N} \qquad (3.2) \\
\sigma_d & = & \sqrt{\frac{\sum_{l=1}^{N} x_d^2(l)}{N} - \overline{X}_d^2} \qquad \sigma_g = \sqrt{\frac{\sum_{l=1}^{N} x_g^2(l)}{N} - \overline{X}_g^2} \qquad (3.3) \\
X_d & = & \overline{X}_d - \sigma_d \qquad\qquad X_g = \overline{X}_g + \sigma_g \qquad (3.4)
\end{array}
$$

La limite supérieure de la boîte englobante du visage est calculée avec l'algorithme suivant : pour chaque ligne l dans E_mvt en partant du haut de l'image, on calcule la distance $D_{mvt}(l)$ (cf. équation 3.5). La première ligne dont $D_{mvt}(l) \in [Min\_D,\ Max\_D]$ est considérée comme étant le haut du visage. Cela permet de ne pas tenir compte des premières lignes de mouvements correspondant au haut de la tête tout en incluant le haut du visage : les sourcils et les yeux.

$$
\begin{array}{rcl}
D_{mvt}(l) & = & x_g(l) - x_d(l) \qquad (3.5) \\
Max\_D & = & X_g - X_d \\
Min\_D & = & Max\_D - 2(\sigma_d + \sigma_g)
\end{array}
$$

Le bas du visage est plus complexe à détecter. Cependant, il se trouve en général près du bas de l'image captée par la caméra (cf. page 71). Il a été donc choisi de donner la valeur de l'ordonnée de la dernière ligne de l'image à la limite inférieure de la boîte englobante du visage (cf. Figure 3.13).

Afin d'avoir une référence permettant d'évaluer le résultat de ce traitement, nous considérons que la boîte englobante doit contenir au moins les deux yeux et le nez. Car ce sont ces composantes que nous allons chercher par la suite à l'intérieur de cette boîte englobante. La figure (3.14) montre l'évolution de l'abscisse du bord droit de la boîte englobante du visage sur plusieurs images successives. La position de l'œil droit dans ces mêmes images nous permet d'évaluer la validité de la mesure. Nous considérons la valeur x_d satisfaisante lorsqu'elle est inférieure à celle de l'abscisse "x *œil droit*". Nous pouvons décomposer la séquence décrite dans la figure (3.14) en 3 temps : au début le visage est fixe, puis l'utilisateur tourne la tête vers la droite (images 145 à 160) et se stabilise à nouveau. On constate que le seul moment où la mesure du bord du visage est cohérente par rapport à la position de l'œil, correspond à un mouvement de la tête. Le reste du temps, la mesure est en général bruitée et donne un résultat inexploitable quand elle dépasse la position de l'œil. Nous expliquons ce phénomène par le fait que le résultat est satisfaisant si le contour du visage est net dans l'image de mouvement. Le problème est que ce contour

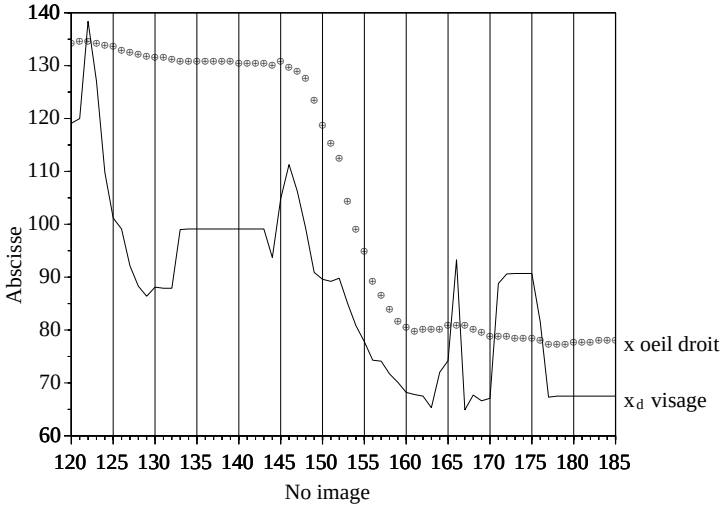


FIG. 3.14 – *Séquences des abscisses du bord droit de la boîte englobante du visage et de l'œil droit, de l'image 120 à l'image 185.*

n'apparaît distinctement que lorsqu'il y a eu un mouvement, si faible soit-il, entre les deux images. Or, il arrive que la tête de l'utilisateur ne bouge pas, ou du moins pas de manière perceptible par la caméra. Dans ce cas, le contour du visage n'apparaît pas de manière continue, voire n'apparaît pas du tout. Ce faisant, les composantes du visage (la bouche, le nez, les yeux, les sourcils...) peuvent toujours bouger et être détectées, formant des îlots de pixels dans l'image de mouvement. Si l'on applique notre traitement de détection de boîte englobante du visage à une telle image de mouvement (images 122, 129, 144, 166, 171...dans la figure 3.14), nous obtenons des informations incohérentes.

Cette situation est *a priori* simple à traiter, s'il n'y a pas de mouvement, il suffit d'utiliser la boîte englobante détectée dans l'image précédente. Cela pose deux problèmes : qu'est-ce qui permet de décider qu'il n'y a pas de mouvement dans une image ? Que se passe-t-il lors des transitions : immobile–en mouvement ?

Pour répondre à la première question, il faut analyser l'image de mouvement et en extraire un paramètre caractérisant le mouvement détecté. Cela doit être fait sans oublier que le traitement doit être le plus rapide possible. La solution que nous avons adoptée consiste à compter le nombre de lignes sur lesquelles on a détecté du mouvement à l'intérieur de la boîte englobante du visage. Ce nombre divisé par le nombre total de lignes dans la boîte englobante du visage, nous permet d'avoir une valeur proportionnelle au mouvement détecté dans l'image. Cette valeur est appelée **taux de mouvement**. Ce taux de mouvement est donc proportionnel au mouvement détecté pour mesurer la boîte englobante du visage (cf. Figure 3.15). On peut utiliser un seuil de taux de mouvement en dessous duquel on décide qu'il n'y a pas de mouvement et donc qu'il faut utiliser la boîte

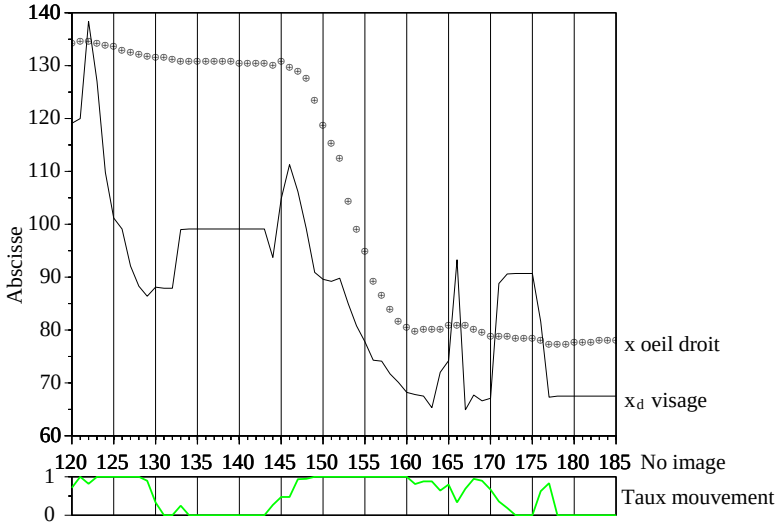


FIG. 3.15 – *Séquences des abscisses du bord droit de la boîte englobante du visage et de l'œil droit, et séquence des taux de mouvement de l'image 120 à l'image 185.*

englobante du visage de l'image précédente. Mais d'une part, il est difficile de choisir un seuil pertinent, et d'autre part cela ne répond pas à la deuxième question. En effet, les îlots de mouvement dus aux composantes du visage qui bougent, même quand le visage n'est pas en mouvement, génèrent un taux de mouvement très variable et qui peut être aussi élevé que celui généré par le mouvement de la tête. Cela rend la mesure de la boîte englobante instable dans le temps. Pour résoudre ce problème, nous avons utilisé un filtre que nous présentons dans la section suivante.

3.2.3.3.2 Filtrage des bornes de la boîte englobante

Filtrer une séquence de valeurs est un problème classique de traitement du signal. Il existe une littérature abondante y compris pour les problèmes de traitement d'image. Nous devons cependant définir quelle partie du signal (les séquences de coordonnées mesurées) représente le bruit qui doit être filtré, pour ensuite choisir le filtre adapté. Or, lorsque l'on observe ces séquences il est difficile de caractériser le bruit. De manière qualitative, nous pouvons dire que lorsqu'il y a du mouvement, il n'y a pas ou pratiquement pas de bruit. Mais lorsque le mouvement diminue le bruit est élevé. Cela dit, il est difficile de modéliser de manière mathématique ce bruit, d'une part parce que nous n'avons pas de signal de référence (séquence de valeurs correctes) qui nous permettrait d'évaluer une erreur de mesure. D'autre part, il n'y a pas dans le signal que nous mesurons de "silence" qui nous permettrait d'évaluer le bruit sans le signal utile.

La figure (3.16) montre le spectre moyen de la séquence correspondant à l'abscisse du bord droit des boîtes englobantes des visages enregistrés dans le corpus de films (cf. Chapitre 4.1). La figure (3.17) montre le spectre correspondant aux séquences d'abscisses

de l'œil droit. Si l'on ne tient pas compte des saccades oculaires, qui ne sont pas mesurées dans ce signal, nous pouvons considérer que l'œil bouge en général au moins à la même fréquence que le visage. Mis à part le fait que cela semble naturel, c'est ce que nous constatons à l'observation de ces deux figures. En effet, soit le visage effectue des translations et les yeux font les mêmes mouvements à la même fréquence. Soit le visage fait des rotations et la boîte englobante du visage est pratiquement immobile, alors que les yeux bougent. Ainsi, pour évaluer la bande passante utile du signal, le spectre correspondant à l'œil droit, peut être comparé au spectre correspondant aux boîtes englobantes du visage. Par ailleurs, nous pouvons observer que la fréquence dont la puissance est la plus élevée est 0,1 Hz, ce qui correspond à une période de 10 secondes. Cette fréquence est normale compte tenu de l'activité que l'on exerce face à un ordinateur, notamment celle demandée lors de la constitution du corpus de films (cf. Chapitre 4.1). Ensuite, le spectre s'atténue rapidement, -26 dB pour l'œil et -9,5 dB pour le visage, jusqu'à 0,35 Hz. Nous pouvons donc baser le filtrage sur un filtre passe–bas, qui atténue le signal à partir de 0,35 Hz.

Il est nécessaire d'utiliser un filtre permettant d'obtenir un résultat en temps réel. Nous devons appliquer le filtre sur une fenêtre temporelle décalée, ce qui a pour inconvénient de générer un léger retard sur le résultat du filtrage. Les filtres à réponse impulsionnelle finie (RIF [Bellanger81b]), de la forme (3.6), sont simples à mettre en œuvre et ont l'avantage d'être toujours stables. Cependant nous choisissons de nous inspirer des filtres à réponse impulsionnelle infinie (ou filtres récursifs [Bellanger81a], [Horaud et al.95b]) dont les coefficients sont moins nombreux (3.7).

$$y(n) = \sum_{i=0}^{N-1} a_i x(n-i) \tag{3.6}$$

$$y(n) = ax(n) + by(n-1) \tag{3.7}$$

Les essais que nous avons réalisés, montrent que l'application seule d'un filtre passe–bas, n'est pas suffisant pour corriger le signal. La figure (3.18) illustre l'utilisation du filtre (3.7) avec $a = 0,444$ et $b = 0,556$. La figure (3.20) montre l'atténuation réalisée par ce filtre. Cette atténuation est très faible (0,7 dB) à 0,35 Hz et augmente jusqu'à dépasser 10 dB avant 2 Hz, ce qui correspond à ce que nous souhaitions. Si nous observons la figure (3.18), nous notons que lorsqu'il y a du mouvement (images 145 à 160), le filtre atténue très peu le signal, de sorte que le résultat est proche de ce qui a été mesuré. Le filtre atténue suffisamment les impulsions comme dans l'image 122 ou l'image 166. Par contre, il ne peut pas supprimer les variations qui perdurent dans le temps, comme dans les images 171 à 176. Si l'on modifie les paramètres du filtre de manière à augmenter l'effet d'atténuation, cela ne corrige pas les longues erreurs de mesures et on perd la précision des mesures correctes. Dans la figure (3.19), nous montrons le résultat de l'application du même filtre avec pour $h_2 : a = 0,21$ et $b = 0,78$ et pour $h_3 : a = 0,08$ et $b = 0,91$. La figure (3.19) montre que les atténuations sont beaucoup plus importantes qu'avec les paramètres précédents. Cependant, le résultat ne s'améliore pas.

On voit bien que le filtrage doit s'adapter au signal. Le type de filtrage nécessaire doit être basé sur un filtre adaptatif. Cependant, les méthodes classique de filtrage adapta-

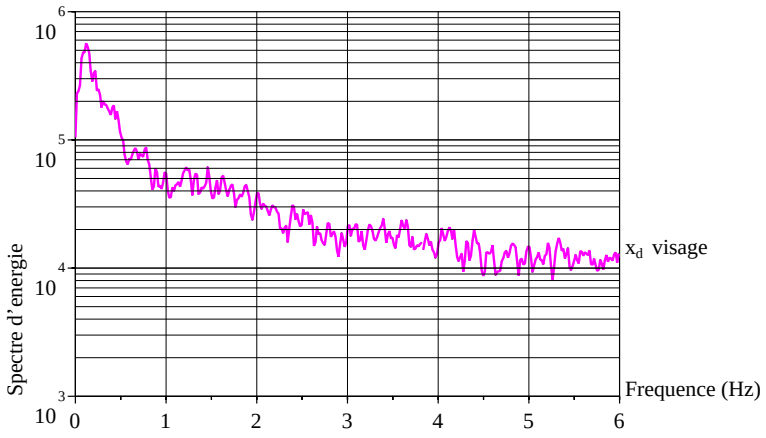


FIG. 3.16 – *Spectre des séquences d'abscisses du bord droit de la boîte englobante du visage.*

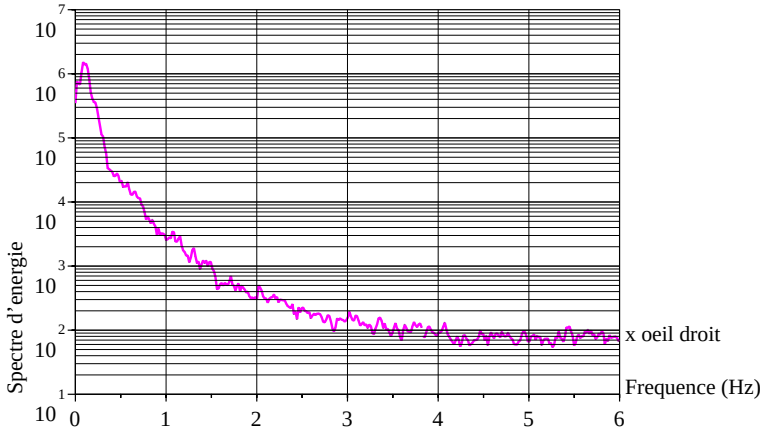


FIG. 3.17 – *Spectre des séquences d'abscisses de l'œil droit.*

tif nécessitent la connaissance d'un signal de référence, ce dont nous ne disposons pas. Comme nous l'avons montré dans la section précédente (cf. page 74), nous disposons d'un autre signal lié au signal mesuré : les séquences de taux de mouvement. Nous nous en sommes donc servi pour pondérer les paramètres du filtre. Lorsqu'il y a du mouvement, le taux de mouvement tend vers 1^- et dans ce cas, le filtre le plus adapté est h_1. Lorsque le mouvement diminue, le taux de mouvement tend vers 0^+ et le filtre atténue d'autant plus le signal, comme h_3. Nous utilisons dans un premier temps le taux de mouvement pour le comparer à un seuil. Si le taux de mouvement est inférieur ou égal au seuil de mouvement,

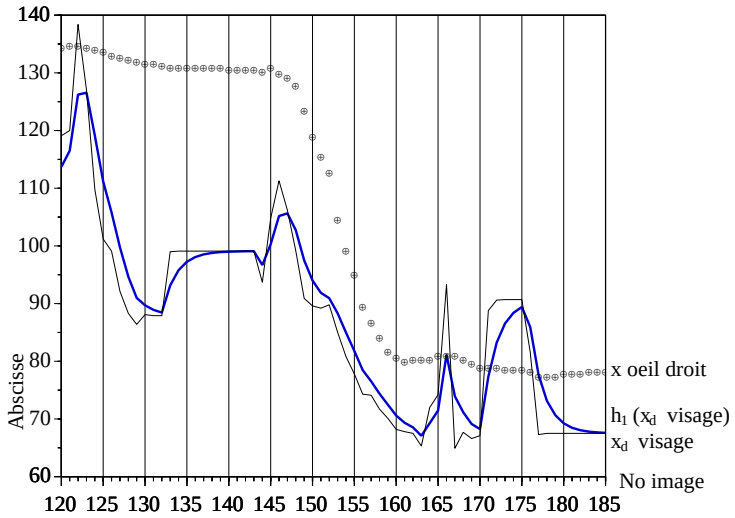


FIG. 3.18 – *Exemple de filtrage d'une séquence d'abscisses du bord droit de la boîte englobante du visage, de l'image 120 à l'image 185.*

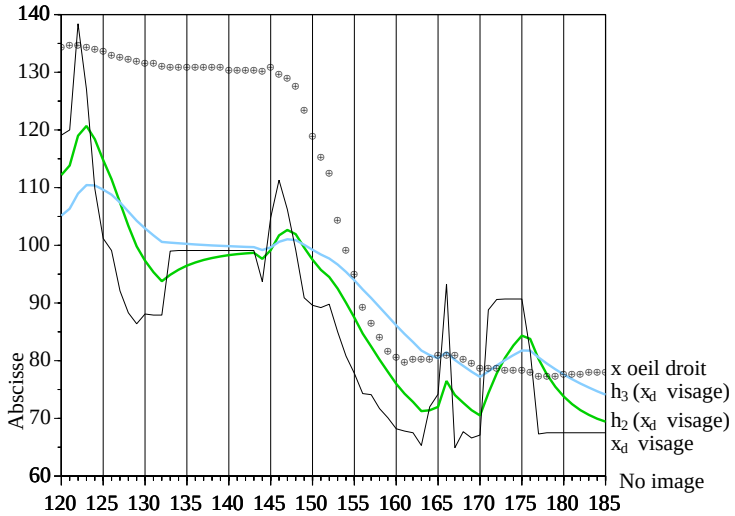


FIG. 3.19 – *Exemples de filtrage d'une séquence d'abscisses du bord droit de la boîte englobante du visage, de l'image 120 à l'image 185.*

on considère que le mouvement ne correspond pas au déplacement du visage. Dans ce cas, on ne tient pas compte de la boîte englobante mesurée, mais on utilise la précédente. Dans le cas contraire, le taux de mouvement est normalisé de la manière suivante :

$$\tau\_normal(n) = \frac{\tau\_mvt(n) - Seuil\_mvt}{1 - Seuil\_mvt}$$

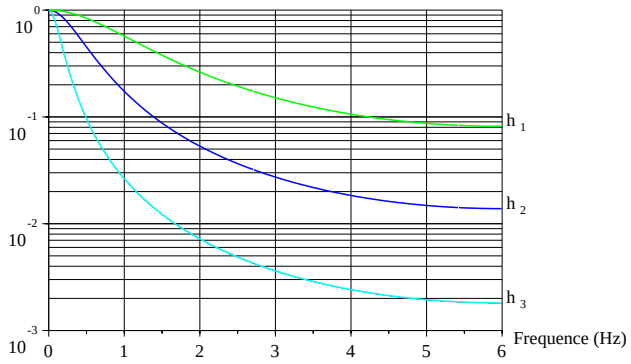


FIG. 3.20 – *Spectre des réponses impulsionnelles des filtres h_1, h_2 et h_3.*

Puis un coefficient α est calculé de manière à mettre au point le filtre suivant :

$$y(n) = \frac{\tau\_normal(n)\, x(n) + \alpha y(n-1)}{\tau\_normal(n) + \alpha} \tag{3.8}$$

Grâce à cela, nous sommes sûrs que quelles que soient les valeurs de $\tau\_normal(n)$ et de α, le calcul est normalisé et le filtre ne diverge pas. Enfin, α est calculé de manière à ce que le filtre tende vers h_1 quand $\tau\_normal(n)$ tend vers 1^- :

$$\alpha(n) = 1,25 + 1,25 \times (1 - \tau\_normal(n)) \tag{3.9}$$

L'équation (3.9) a été mise au point de manière expérimentale et donne un résultat satisfaisant (cf. Section 4.3.1) compte tenu de la simplicité du filtre utilisé. Il nous reste cependant à régler un problème lié aux filtres récursifs. En effet, la caractéristique de ces filtres est qu'ils peuvent être instables. D'autre part, ce n'est pas un filtre classique puisqu'il n'est pas invariant dans le temps, du fait de l'utilisation du taux de mouvement dans le calcul de ses coefficients. Nous devons donc démontrer que notre filtre est stable, pour pouvoir en garantir le bon fonctionnement. Boris Doval en a proposé la démonstration suivante [Doval98].

Nous pouvons donner comme équation générale du filtre :

$$y_n = a_n\, x_n + b_n y_{n-1} \tag{3.10}$$

1 linéarité :

soient x et z les séquences en entrée, y et w les réponses en sortie de filtre.

$$\left(\begin{array}{rcl} y_n &=& a_n\,x_n + b_n\,y_{n-1} \\ w_n &=& a_n\,z_n + b_n\,w_{n-1} \end{array}\right) \iff \left(\begin{array}{rcl} y_n - b_n\,y_{n-1} &=& a_n\,x_n \\ w_n - b_n\,w_{n-1} &=& a_n\,z_n \end{array}\right)$$
$$\iff (y_n + w_n) - b_n\,(y_{n-1} + w_{n-1}) = a_n\,(x_n + z_n)$$

La séquence $x_n + z_n$ donne en sortie $y_n + w_n$ et λx_n donne en sortie λy_n, donc le filtre est linéaire.

2 invariance : par construction, le filtre n'est pas invariant dans le temps.

3 stabilité, la démonstration est faite suivant les étapes suivante :

a : on peut définir des réponses impulsionnelles $h_{n,i}$

b : on peut trouver des conditions simples pour assurer que $h_{n,i}$ est absolument sommable $\forall i$

c : si $h_{n,i}$ est absolument sommable, alors le filtre est stable

a : On peut décomposer la séquence x_n de la manière suivante :

$$x_n = \sum_{i=-\infty}^{+\infty} x_i\,\delta_{n-i}$$

avec x_i : échantillon du signal x_n à l'instant i et δ_{n-i} : signal impulsion unité placée à l'instant i.

$$\begin{array}{rl} y_n &= H\{x_n\} \\ &= H\{\sum x_i\,\delta_{n-i}\} \\ &= \sum x_i\,H\{\delta_{n-i}\} \quad (linéarité) \end{array}$$

Si le système était invariant dans le temps, on pourrait écrire : $H\{\delta_{n-i}\} = h_{n-i}$ avec $H\{\delta_n\} = h_n$ (réponse impulsionnelle du filtre), mais comme nous utilisons un coefficient différent (τ_mvmt) pour chaque n, nous avons en réalité une réponse impulsionnelle différente à chaque n.
On note $H\{\delta_{n-i}\} = h_{n,i}$

b : Dans le cas particulier des filtres d'ordre 1 d'équation récursive : $y_n - b_n y_{n-1} = a_n x_n$, on peut exprimer $h_{n,i}$, avec i fixé :

$$h_{n,i} - b_n h_{n-1,i} = a_n \delta_{n-i} \begin{cases} pour\, n < i, & h_{n,i} = 0 \quad (filtre\ causal) \\ pour\, n = i, & h_{i,i} - b_i\,h_{i-1,i} = a_i\,.\,1 \iff h_{i,i} = a_i \\ pour\, n > i, & h_{n,i} - b_n\,h_{n-1,i} = 0 \\ & \iff h_{n,i} = b_n\,h_{n-1,i} \\ & \implies h_{n,i} = a_i \times \prod_{k=i+1}^{n} b_k \end{cases}$$

On note : $$\begin{aligned}\sum_{n=-\infty}^{+\infty} \mid h_{n,i} \mid &= \sum_{n=i}^{+\infty} \mid h_{n,i} \mid \qquad (filtre\ causal)\\ &= \mid a_i \mid \sum_{n=i}^{+\infty} \prod_{k=i+1}^{n} \mid b_k \mid\end{aligned}$$

Si $\exists b_{MAX}/\forall k, \mid b_k \mid \leq b_{MAX} < 1$ *alors*

$$\begin{aligned}\sum_{n=-\infty}^{+\infty} \mid h_{n,i} \mid &\leq a_i \sum_{n=i}^{+\infty} \prod_{k=i+1}^{n} b_{MAX}\\ &\leq \mid a_i \mid \sum_{n=i}^{+\infty} b_{MAX}^{n-i} \qquad = \mid a_i \mid \sum_{n=0}^{+\infty} b_{MAX}^{n}\\ &\qquad\qquad = \mid a_i \mid \frac{1}{1-b_{MAX}} < +\infty\end{aligned}$$

Or $\frac{1}{1-b_{MAX}}$ est indépendant de i, et converge ssi $\mid b_{MAX} \mid < 1$

Donc *si* $\exists b_{MAX}/\forall n, \mid b_n \mid \leq b_{MAX} < 1$ *alors* $\forall i$, $h_{n,i}$ *est absolument sommable* :

$$H_{Som} = \sum_{n=-\infty}^{+\infty} \mid h_{n,i} \mid < +\infty$$

c : si x_n est borné, $\exists X_{MAX}/\forall i, \mid x_i \mid < X_{MAX}$

$$\mid y_n \mid = \mid \sum_n x_i\, h_{n,i} \mid \leq \sum_n \mid x_i \mid\mid h_{n,i} \mid \leq X_{MAX} \sum_n \mid h_{n,i} \mid$$

Donc *si* $\exists b_{MAX}/\forall n, \mid b_n \mid \leq b_{MAX} < 1$ et si x_n est borné

alors y_n est borné par $\boxed{\mid y_n \mid \leq X_{MAX}\, H_{Som}}$

Par construction tous les b_i sont strictement inférieurs à 1 et sont en nombre fini. Il est donc toujours possible de déterminer un b_{MAX} et le filtre est stable.

On remarque que dans le cas particulier où le taux de mouvement est inférieur ou égal au seuil de mouvement, en utilisant la boîte englobante précédente on applique le même filtre mais avec comme coefficients $a = 0$ et $b = 1$. D'après la valeur de b, le filtre est instable mais comme a est nul, x_n n'entre pas en ligne de compte pour le calcul de y_n. C'est par ce choix que le filtre reste stable dans ce cas.

Les coordonnées de la boîte englobante, ainsi filtrées sont utilisables pour la suite du traitement. Nous pouvons donc rechercher les narines et les yeux de l'utilisateur à l'intérieur du visage.

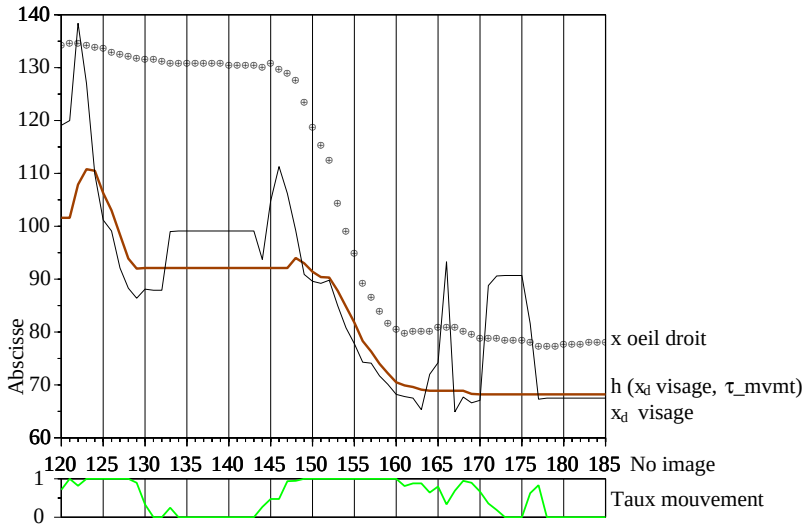


FIG. 3.21 – *Résultat du filtrage sur la même séquence que celle de la figure 3.18.*

3.2.3.4 2e et 3e processus : Nez et Yeux

Nous disposons pour chaque image d'une zone rectangulaire (la boîte englobante du visage) dans laquelle nous pouvons espérer trouver les deux yeux et le nez de l'utilisateur. Dans le pire des cas, notamment si dans les premières images traitées de la séquence, il n'y a pas de mouvement, cette zone rectangulaire correspond à l'image toute entière. Dans la pratique, cela ne doit pas arriver, l'utilisateur venant s'installer devant la machine est nécessairement en mouvement.

Baluja et Pomerleau [Baluja et al.94] utilisent le reflet spéculaire d'une source lumineuse fixe, qui apparaît sur les yeux pour détecter ceux-ci dans l'image. Ainsi, ils éliminent le problème de localisation. Mais cela sous-entend qu'ils utilisent une lumière spécifique placée en face du visage, non loin de la caméra. En effet, une source lumineuse mise de côté par rapport à l'utilisateur et donc loin de l'axe de vision de la caméra, peut ne pas générer de reflet dans les yeux selon l'orientation de la tête. Ensuite, ils utilisent un réseau de neurones pour à la fois reconnaître l'œil et mesurer la direction du regard. Ce système ne tenant pas compte de la position et de l'orientation de la tête pour calculer la direction du regard, il ne semble pouvoir garder la même précision de mesure si l'utilisateur bouge. D'autre part, l'utilisation d'un réseau de neurones ne permet d'exploiter qu'une image de 30×15 de l'œil à la fréquence de 15 Hz. C'est sans doute pour cette raison, que l'équipe qui a succédé à celle-ci sur ce projet à Carnegie-Mellon University, a utilisé une autre stratégie. Ainsi, Stiefelhagen, Yang et Waibel dans [Stiefelhagen et al.96], proposent une démarche proche de la notre. Après avoir détecté la boîte englobante du visage (cf. page 69), ils cherchent les points les plus sombres dans le visage. Ils considèrent que ces points sont les centres des iris. Pour cela, ils font varier un seuil de binarisation, jusqu'à obtenir quelques petits groupes de pixels. Puis ils utilisent des contraintes morphologiques

et l'hypothèse que le visage est bien de face et droit dans l'image, pour sélectionner les deux groupes constituant les yeux. La détection des narines est effectuée de la même manière, après celle de la bouche. La zone de recherche pour les narines est définie par un rectangle entre les yeux et la bouche. Cette méthode ne nous semble pas assez robuste, car elle ne reconnaît pas les yeux, ni les narines. Elle privilégie la cohérence globale des composantes du visage entre-elles et les contraintes morphologiques. Nous pensons que ces deux aspects sont importants pour la reconnaissance mais pas suffisants pour la robustesse du système. D'autre part, on ne sait pas quelle est la tolérance du système pour l'orientation du visage lors de l'amorçage. En effet, ils précisent que pour que la localisation fonctionne au début du traitement, le visage doit être de face et droit. Mais ils ne donnent pas d'évaluation de l'amorçage de la détection selon l'orientation du visage.

Nous avons donc à trouver des solutions différentes. Pour cela, il nous faut analyser les images et en extraire des primitives caractéristiques des composantes à reconnaître. Nous avons à répondre à la question : qu'est-ce qui caractérise une narine (respectivement un œil) par rapport au reste du visage? Les narines peuvent être de forme très différentes d'un visage à l'autre. Dans le livre de Farkas, "Anthropometry of the Head and Face" [Farkas94], on peut observer des narines plutôt rondes, ou fines, ou larges, ou allongées avec des orientations diverses. Il semble donc difficile de les lier à un modèle rigide. Cela d'autant plus que l'image de ces narines changent selon l'orientation du visage. Nous pouvons cependant donner comme caractéristiques communes à toutes les narines les descriptions suivantes :

1. ce sont des trous qui dans l'image sont représentés par un ensemble compact de pixels sombres [Stiefelhagen et al.96] ;

2. la couleur du visage n'est pas aussi sombre que celle des narines [Stiefelhagen et al.96] ;

3. le bord des narines est constitué de pixels clairs qui s'assombrissent graduellement en se rapprochant de la narine (narines sombres entourées de régions plus claires [Varchmin et al.98]);

4. on peut définir une fourchette de valeurs en longueur et en hauteur dans lesquelles on peut englober la narine (quelle que soit sa forme) [Petajan et al.96].

5. La plupart du temps (cela dépend de l'orientation de la tête), les deux narines sont visibles et très proches l'une de l'autre. On peut aussi définir une fourchette de distance minimum et maximum entre les deux narines [Petajan et al.96].

Par ailleurs, nous pouvons considérer que tous les yeux sont formés d'une sphère de couleur claire (blanche), avec sur leur surface, un iris qui est rond et de couleur plus foncée. Cet iris peut être plus ou moins foncé selon les individus, mais la pupille au centre est extrêmement sombre. En effet, sauf en cas de reflet, la pupille est un trou qui ne renvoie pas de lumière. Pour que la lumière ressorte de la pupille, il faut une source

orientée en face de celle-ci et d'assez forte puissance, comme un flash d'appareil photo par exemple. Cet ensemble est derrière des paupières qui peuvent être plus ou moins fermées. Le problème principal pour la détection vient des paupières qui cachent en général la plus grande partie de l'œil. D'un point de vue visuel, nous utilisons les descriptions suivantes pour caractériser les yeux :

1. ils sont représentés par un cercle de pixels sombres, bordés de part et d'autre de pixels clairs [Herpers et al.96] ;
2. la pupille contient les pixels les plus sombres de l'iris, et parmi les plus sombres du visage [Stiefelhagen et al.96] ;
3. on peut définir une fourchette de valeurs en longueur et en hauteur dans lesquelles on peut englober l'iris.

Les techniques classiques de traitement d'image ou de reconnaissance des formes pour répondre à un problème aussi complexe (réseaux de neurones [Baluja et al.94] [Reinders et al.96], détection de contour [Yow et al.95] [Herpers et al.96], contours actifs [Sobottka et al.96] [Leroy et al.96]...), utilisent des algorithmes de complexité non linéaire. Si on les applique sur l'image du visage, ces traitements permettent difficilement un fonctionnement en temps réel. Nous pensons qu'il est possible de découper le problème de manière à n'avoir à réaliser que des processus qui seront exécutés rapidement par la machine.

Notre approche consiste dans un premier temps à décrire, d'un point de vue de la reconnaissance des formes, une des caractéristiques commune aux narines et aux yeux : les transitions de pixels sombres vers des pixels clairs. En effet, si l'on observe le niveau de luminosité des pixels sur une ligne de l'image, on remarque qu'il y a de fortes transitions (i.e. des gradients élevés) sur les bords des narines et sur les bords des iris. On ne remarque pas de gradient aussi élevé à l'intérieur du visage sauf autour des sourcils ou de la barbe. On en trouve par contre sur les bords du visage, notamment entre les cheveux et la peau. Dès lors, nous pouvions mesurer la forme du gradient correspondant au bord d'une narine et celle du bord de l'iris, puis utiliser des comparaisons par simples corrélations entre les gradients de l'image et les gradients à reconnaître. Mais, un calcul de corrélation a une complexité polynomiale et donc un coût trop important, qui peut compromettre le fonctionnement en temps réel du système. Il nous faut donc réaliser un ou plusieurs prétraitements, qui devront être assez efficaces pour, à la fois permettre une exécution rapide mais aussi donner un résultat "rentable" pour l'exploitation de la corrélation. Pour être rentable, le résultat doit permettre de réduire le nombre de données à traiter par le calcul de corrélation. Cette technique doit aussi permettre de rendre le système globalement plus robuste pour la détection des composantes du visage. Car il est évident que les formes de gradient que nous avons observées sur le visage ne sont pas toujours suffisamment discriminantes pour la reconnaissance. Pour augmenter la discrimination, nous utilisons une autre caractéristique commune aux narines et aux yeux : les ensembles compacts de pixels sombres. Après avoir déterminé quels sont ces ensembles dans le visage, il est possible de réaliser localement les calculs de corrélation. Le résultat de ces traitements est validé en

utilisant des contraintes morphologiques tel que celles décrites dans les caractéristiques des composantes du visage ou des contraintes dynamiques.
Afin d'augmenter la robustesse de la détection et du suivi de ces composantes, nous avons mis en place des méthodes permettant à certains seuils ou aux formes de gradients de s'adapter en dynamique aux caractéristiques du visage et de l'image. Ces méthodes sont notamment utilisées lorsque le processus de traitements se trouve dans l'état d'adaptation, mais certaines le sont aussi dans l'état d'initialisation.

Les détections réalisées par cette suite de traitements sont plus robustes pour les narines que pour les yeux. Nous avons donc décidé de détecter dans un premier temps les narines à l'intérieur de la boîte englobante du visage. Puis nous utilisons la localisation des narines pour créer deux zones de traitement rectangulaires au-dessus et de part et d'autre du nez pour y détecter les yeux. Donc, mis à part la zone où est réalisée le traitement dans le visage et l'utilisation d'une caractéristique spécifique du centre de l'iris (cf. page 84), les processus de détection sont identiques pour les narines et pour les yeux.

Après cette présentation générale du fonctionnement de la détection des narines et des yeux, nous allons décrire plus précisément les divers processus mis en œuvre en commençant par l'extraction des zones sombres du visage. Nous préciserons le cas échéant les différences de traitements pour les narines et les yeux, et les différences entre les processus dans l'état d'initialisation et dans l'état d'adaptation.

3.2.3.4.1 Extraction des zones sombres

Dans un premier temps, nous présentons les conditions de traitement pour les processus dans l'état d'initialisation. A partir d'une zone rectangulaire de l'image (la boîte englobante du visage pour les narines ou la zone dans le visage au-dessus et d'un côté ou de l'autre du nez pour un œil), nous devons extraire une ou plusieurs zones englobant chacune un ensemble compact de pixels sombres. Pour réaliser ce traitement nous avons à résoudre deux problèmes : Qu'est-ce qu'un pixel sombre ? Et qu'est-ce qu'un ensemble compact de pixels ?

Seuillage des pixels sombres

On considère qu'un pixel sombre est un pixel dont la luminosité est inférieure à un seuil. Il suffit de réaliser une binarisation de l'image basée sur ce seuil. Seuillage et binarisation font partie des opérations de base en traitement d'image. Le problème est de choisir un seuil permettant de sélectionner un maximum de pixels utiles et un minimum d'autres. Cela n'est possible que si cette caractéristique (pixels sombres) est suffisamment discriminante, ce qui est le cas. Si l'on veut que le seuillage résiste aux variations de luminosité et aux différences de couleur de peau entre personnes, il est nécessaire d'utiliser une méthode d'évaluation dynamique du seuil. Stiefelhagen et al. [Stiefelhagen et al.96] procèdent par seuillages successifs en augmentant progressivement la valeur du seuil jusqu'à obtenir des ensembles de pixels seuillés assez gros. Cette méthode procède par itérations successives

et nécessite de seuiller plusieurs fois l'image avant d'obtenir un résultat. On peut éviter ces itérations en utilisant des méthodes classiques basées sur l'analyse de l'histogramme de l'image à seuiller [Besançon88]. Compte tenu de la complexité d'un visage, il n'est pas possible d'utiliser la méthode de segmentation par histogramme. Nous choisissons donc de fixer un seuil sur le taux de pixels sombres dans l'histogramme de l'image.

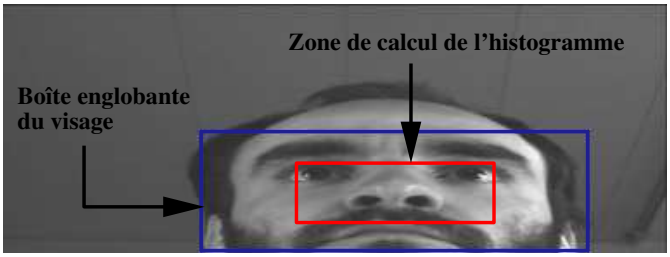


FIG. 3.22 – *Boîte englobante du visage utilisée pour la détection du nez et zone de calcul de l'histogramme.*

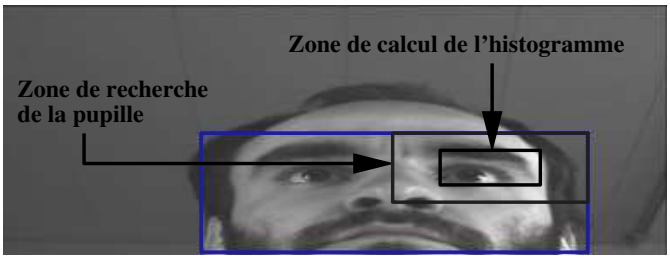


FIG. 3.23 – *Zone de recherche utilisée pour la détection de l'iris et zone de calcul de l'histogramme .*

Les composantes recherchées ont peu de chance de se trouver sur les bords des zones de recherche. Afin de réduire l'influence de la couleur d'autres composantes du visage comme les cheveux ou les sourcils sur le calcul du seuil, nous ne réalisons l'histogramme que sur le quart central de la zone de recherche (Figures 3.22 et 3.23). De plus, quand la composante recherchée est proche du bord de la zone de recherche, cette méthode donne de bons résultats. La détection quant à elle se fait sur la zone du visage réduite d'un tiers pour les narines et sur toute la zone de recherche pour les iris. En faisant varier le seuil de binarisation de l'image et en calculant le pourcentage de pixels retenu par rapport au nombre total de pixels, nous pouvons définir un seuil de taux de pixels sombres. Dans les images suivantes (3.25) nous montrons qu'avec 2,5 % des pixels sombres de l'histogramme (Figure 3.24), nous voyons apparaître les narines de manière suffisamment claire pour qu'elles soient détectées automatiquement. La même méthode est employée pour les deux iris avec un seuil de 5 %.

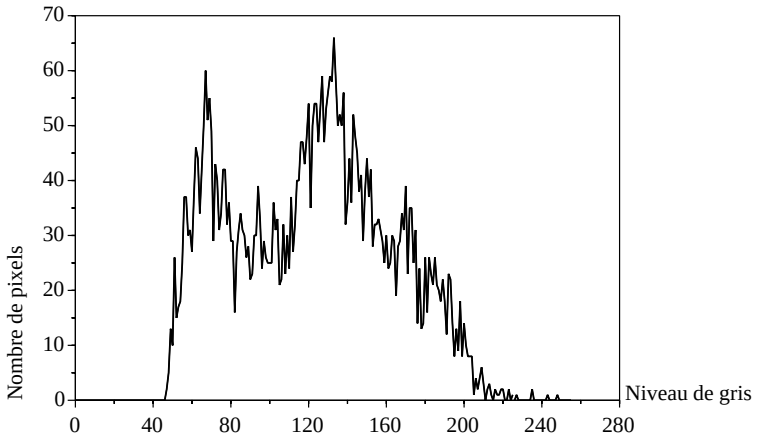


FIG. 3.24 – *Histogramme du quart central de la boîte englobante du visage.*

Détection de zones de pixels sombres

Disposant d'une image de pixels sombres, il nous faut résoudre le second problème. Cela revient tout d'abord à définir comment représenter la forme à reconnaître. À ce stade du traitement, nous avons une image binaire dans laquelle il faut trouver deux groupes de pixels représentant les narines ou un groupe par iris. La représentation de ces groupes de pixels dépend à la fois de la complexité du traitement et sa précision vis-à-vis du traitement réalisé par la suite. C'est la précision qui est prise en compte dans un premier temps. En effet, le traitement suivant (cf. Section 3.2.3.4.2) peut très bien fonctionner en ayant comme données des zones rectangulaires contenant les pixels sombres. Cette représentation suffit aussi pour utiliser des contraintes morphologiques, comme la taille ou la longueur de la composante à reconnaître, pour éliminer les candidats hors normes. Il n'est donc pas nécessaire de calculer précisément des contours ou des régions de ces groupes de pixels. Compte tenu de la simplicité de la représentation requise, nous nous attachons à trouver un algorithme simple lui aussi.

Les algorithmes classiques utilisés en reconnaissance de forme, sont conçu pour fonctionner dans un cadre général. Ils sont de ce fait applicables pour notre cas. Cependant, cette qualité (la généralité) devient un défaut en terme de temps de calcul ou de complexité. Nous avons donc choisi de mettre au point un algorithme spécifique de croissance de zones dirigée par la représentation dont nous avons besoin, c'est-à-dire des zones rectangulaires. Nous montrons que cet algorithme est plus efficace en temps de calcul que les algorithmes classiques dans le cadre de la génération de zones rectangulaires à partir d'une image binaire.

Le principe de l'algorithme consiste à créer et à faire croître des zones rectangulaires au fur et à mesure du parcours de l'image binaire. Cette opération est donc réalisée en

(a) 1 % des pixels sombres.

(b) 2,5 % des pixels sombres.

(c) 5 % des pixels sombres.

(d) 10 % des pixels sombres.

FIG. 3.25 – *Divers seuils du taux de pixels sombres dans l'histogramme.*

une seule passe sur l'image.

On parcourt l'image ligne par ligne dans la zone de recherche et à chaque fois, on tient compte de la ligne courante et de la ligne suivante. On crée ainsi une zone de deux pixels d'épaisseur pour chaque segment ininterrompu de pixels sombres sur les 2 lignes simultanément et une zone d'un pixel d'épaisseur pour chaque segment ininterrompu de pixels sombres sur la ligne courante uniquement. Ces zones sont stockées dans une liste. Afin d'obtenir une liste de zones contenant des groupes compacts de pixels, une opération d'union des zones se chevauchant est réalisée. Le fait de générer des zones de 2 pixels d'épaisseur permet de faire croître ces zones ligne après ligne par chevauchement. On génère des zones d'une ligne d'épaisseur pour le cas où la composante à reconnaître aurait cette forme. Cela arrive notamment pour les narines, lorsque le visage est orienté vers le bas. Cependant, si l'union des zones est faite uniquement sur le critère de chevauchement de celles-ci, on peut générer des aberrations. Par exemple, pour l'union de deux zones ne se chevauchant que sur une petite partie, on obtient une zone ne comportant que très peu de pixels sombres. Cette opération est donc contrainte par un critère de croissance horizontale. En d'autres termes, si deux zones se chevauchent, elles ne peuvent être unies que

si cela n'entraîne pas une croissance horizontale trop importante d'une des deux zones. Par construction, une zone est générée lors du traitement d'une des lignes précédentes, et qui croît par union avec une nouvelle zone sur chaque ligne jusqu'à la ligne courante. Cette zone est appelée zone en croissance. La nouvelle zone générée sur la ligne courante chevauche la zone en croissance avec au moins un pixel. Le critère de croissance horizontale (δ) est calculé proportionnellement à la longueur de la zone en croissance selon un taux de croissance (τ dans 3.11). Ce critère est évalué sur la distance entre les extrémités horizontales (X_d et X_g) de cette zone et de la nouvelle. Si ces deux distances sont inférieures à δ, les deux zones sont réunies.

$$\delta = \tau \times (X_d\_zone\_en\_croissance - X_g\_zone\_en\_croissance) \tag{3.11}$$

Les zones ainsi générées sont cohérentes, dans le sens où elles contiennent une majorité de pixels sombres sous forme compacte. Le problème est donc de sélectionner le taux de croissance (τ) le plus adapté au processus de génération de zones. En effet, si τ est très faible (< 0.5) la zone en croissance ne sera unie qu'avec des zones ayant relativement la même longueur. Cela génère donc beaucoup de petites zones dont certaines seront proches les unes des autres. Si par contre, τ est très élevé (> 1.5), cela risque de générer des zones très grandes incluant plusieurs groupes de pixels sombres.

Pour calculer la complexité de cet algorithme, il faut définir les conditions d'exécution dans le pire des cas. Si toute l'image contient des pixels sombres, l'algorithme traite tous les pixels, mais ne génère qu'une zone. Il n'a pas à effectuer de chevauchement (en fait, il fait un test de chevauchement par ligne). Nous définissons le pire des cas avec l'image d'un damier : pour une image de taille $l \times c$ (nombre de lignes fois nombre de colonnes), un pixel sur deux est sombre et l'algorithme génère une zone un pixel sur deux ($\frac{c}{2}$ pixels par ligne). L'algorithme que nous utilisons a donc une complexité dans le pire des cas de l'ordre de :

$$\underbrace{l \times c}_{scrutation\ image} + \underbrace{(l-1) \times \frac{c}{2}}_{scrutation\ lignes\ suivantes} + \underbrace{(l-1) \times \frac{c}{2}}_{tests\ chevauchement} \leq 2l \times c \tag{3.12}$$

En fait, le nombre de pixels sombres est limité par le pourcentage de pixels utilisé dans le calcul du seuil de binarisation de l'image. On sait que l'on n'aura pas plus de 5 % de pixels de l'image à traiter pour générer des zones.

$$\underbrace{l \times c}_{scrutation\ image} + \underbrace{(l-1) \times 5\%\ c}_{scrutation\ lignes\ suivantes} + \underbrace{(l-1) \times 5\%\ c}_{tests\ chevauchement} \leq l \times c + \frac{(l-1) \times c}{10} \tag{3.13}$$

Cette complexité est proche d'une simple scrutation de l'image ($l \times c$) dans notre cas (3.13) et ne dépasse pas deux scrutations dans le cas général (3.12).

Nous pouvons comparer cette complexité à celle des algorithmes classiques permettant de déterminer une forme dans une image. L'algorithme le plus proche et aussi le plus rapide d'après Horaud [Horaud et al.95a], est celui de chaînage de contours de Giraudon décrit dans [Giraudon87]. En effet, cet algorithme réalise un chaînage de contours sur une image de contours en une seule passe avec une fenêtre 3×3 (complexité de $9 \times l \times c$). Dans notre cas, cela nécessiterait un post-traitement pour séparer les formes unies dans le même contour. Cela peut arriver entre les cils et l'iris ou entre les narines et une moustache. D'autre part, l'algorithme de chaînage de contours est adapté au traitement d'une image de contours. Pour calculer une telle image, il faut avoir recours à des algorithmes complexes mais qui donnent des résultats plus précis que la technique que nous avons utilisée en pré-traitement. L'approche que nous utilisons est plus adaptée à l'imprécision de notre pré-traitement, elle ne cherche pas à délimiter un contour précis des narines ou des iris mais à générer des zones dans lesquelles on peut trouver ces composantes. Notre algorithme est donc plus efficace en terme de complexité mais aussi de résultat parce qu'il s'inclut dans une chaîne de traitements spécifiques.

Extrapolation de la zone de recherche dans l'état d'adaptation

Les traitements décrits ci-dessus sont utilisés par les processus dans l'état d'initialisation. Lorsqu'un processus se trouve dans l'état d'adaptation, ces traitements sont identiques, ce sont les données en entrée qui changent. En effet, les zones de recherche dans lesquelles sont appliqués les traitements, ne sont plus déduites des boîtes englobantes calculées dans les processus précédents, mais extrapolées à partir des zones de la composante dans les images précédentes. Ainsi, ces zones de recherches sont plus réduites et plus sûres, et les traitements sont plus rapides.

Le calcul de la zone de recherche dans l'état d'adaptation, utilise en réalité la zone de la composante détectée dans l'image précédente, ainsi que la vitesse de cette composante. Cela signifie que pour pouvoir passer dans l'état d'adaptation, il est nécessaire d'avoir détecté la composante dans au moins deux images successives. Il est par conséquent nécessaire de disposer de l'intervalle de temps réel qui sépare deux images pour réaliser ces calculs. La vitesse est calculée à partir des localisations du centre de la zone englobant les deux narines ou du centre de l'iris (la pupille). Mais cette vitesse ne permet que d'extrapoler la position de la zone dans la nouvelle image. Cela ne suffit pas, car cette vitesse peut avoir changé entre temps et il faut agrandir la taille de la zone pour ajouter une tolérance de sécurité pour la recherche de la composante. Afin d'adapter cette marge de tolérance à la dynamique de la composante, nous utilisons une marge dont la taille est proportionnelle au temps écoulé depuis l'image précédente. Pour cela nous établissons un seuil d'accélération maximale pour chaque composante. Il suffit ensuite de multiplier ce seuil par le temps écoulé pour disposer d'une vitesse à ajouter ou à retrancher à la vitesse précédente pour obtenir un intervalle de vitesse de déplacement de la zone. Les équations ci-dessous décrivent les calculs utilisés pour extrapoler la zone de recherche de la composante dans l'image $t+1$, avec $x(t)$ et $y(t)$ comme coordonnées du centre de la

zone dans l'image t, $\delta(t)$ pour le temps écoulé entre les images $t-1$ et t, $v_x(t)$ et $v_y(t)$ les vitesses calculées à l'image t, ($Inf_x(t)$, $Inf_y(t)$, $Sup_x(t)$, $Sup_y(t)$) les coordonnées des bornes de la zone dans l'image t, $v'_{x\ max}$ et $v'_{y\ max}$ les seuils d'accélération maximale.

$$v_x(t) = \frac{\partial x}{\partial t} = \frac{x(t-1)-x(t)}{\delta(t)} \tag{3.14}$$

$$v_y(t) = \frac{\partial y}{\partial t} = \frac{y(t-1)-y(t)}{\delta(t)} \tag{3.15}$$

$$\begin{aligned}
Inf_x(t+1) &= Inf_x(t) + (v_x(t) - v'_{x\ max} \times \delta(t+1)) \times \delta(t+1) \\
Inf_y(t+1) &= Inf_y(t) + (v_y(t) - v'_{y\ max} \times \delta(t+1)) \times \delta(t+1) \\
Sup_x(t+1) &= Sup_x(t) + (v_x(t) + v'_{x\ max} \times \delta(t+1)) \times \delta(t+1) \\
Sup_y(t+1) &= Sup_y(t) + (v_y(t) + v'_{y\ max} \times \delta(t+1)) \times \delta(t+1)
\end{aligned}$$

Dans la plupart des cas, ce calcul d'extrapolation garantit de trouver la composante dans la zone de recherche. Cependant, dans le processus de détection du nez, on cherche à localiser deux narines. Si le traitement ne permet la détection que d'une seule dans la zone de recherche, on applique à nouveau ce traitement dans une zone élargie. Cet élargissement est proportionnel au temps écoulé entre les images.

Le calcul présenté ci-dessus peut être défaillant si on l'applique tel quel au suivi de l'iris. En effet, d'une part la vitesse de l'iris peut être très élevée : une saccade oculaire de 40^o d'angle, ce qui correspond à peu près au trajet de l'iris du centre de l'œil vers le coin de l'œil, peut être exécutée en 120 millisecondes [Jacob95]. D'autre part, l'accélération lors de l'exécution d'une saccade est pratiquement infinie. Il faut donc trouver une référence permettant d'extrapoler la localisation non pas de l'iris mais de l'œil d'une manière globale. Ne disposant pas d'information sur la localisation de l'œil, nous utilisons la vitesse du visage et du nez à la place de la vitesse de l'iris. La moyenne de ces deux vitesses permet de rendre compte à la fois des translations et des rotations du visage pour extrapoler la déplacement des yeux.

Enfin, quel que soit l'état dans lequel se trouve le processus, il dispose donc d'une liste de zones contenant les groupes de pixels sombres. A partir de cette liste, on peut éliminer les zones dont les dimensions sont en dehors d'un intervalle correspondant aux dimensions minimales et maximales des formes que nous voulons reconnaître [Stiefelhagen et al.96]. On dispose alors d'une liste réduite de candidats pouvant contenir la composante recherchée. Ceci amène à utiliser un autre critère de reconnaissance pour sélectionner le bon candidat. Pour cela nous appliquons un opérateur connu de reconnaissance de forme : la corrélation.

3.2.3.4.2 Reconnaissance de formes

Dans cette partie du processus, du fait que les données sont réduites en nombre, nous pouvons appliquer la stratégie qui consiste à effectuer des opérations coûteuses en temps de calcul (de complexité non linéaire). En effet, nous disposons d'une liste d'un nombre limité de zones rectangulaires dans l'image, incluant potentiellement la composante du visage cherchée. Sur la base du corpus utilisé (cf. Section 4.3.2), il a été mesuré que le processus détecte jusqu'à 80 zones mais avec une moyenne de 18, lorsqu'il se trouve dans l'état d'initialisation, c'est-à-dire dans le pire des cas. Les traitements précédents se contentent de réduire l'espace de recherche, mais ne permettent pas de déterminer si une zone a plus de chance qu'une autre de contenir la composante recherchée ou si cette zone contient effectivement cette composante.
Le but de cette partie de traitement est d'une part de reconnaître spécifiquement la composante en utilisant un critère de photométrie et d'autre part de calculer une évaluation de cette reconnaissance pour disposer d'un score de détection. A l'issue de cette partie, on doit avoir suffisamment d'informations pour que le système soit capable de sélectionner la zone qui contient la composante recherchée parmi une liste de zones candidates.

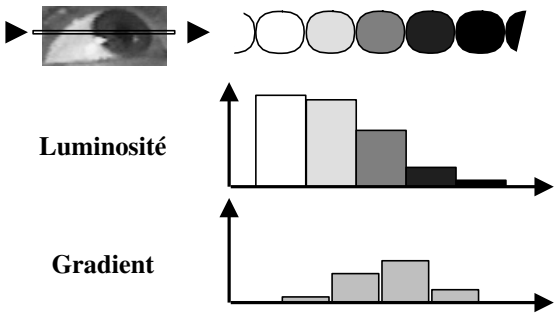


FIG. 3.26 – *Représentation du bord de l'iris par un vecteur de gradients.*

Corrélation de gradients

La technique utilisée est l'appariement d'une forme générique (ou patron) avec les formes candidates dans l'image à l'intérieur des zones. Encore faut-il déterminer quelle forme et quel modèle pour représenter ce patron. L'une des caractéristiques communes aux narines et aux iris est qu'ils sont bordés de pixels clairs qui graduellement s'assombrissent en se rapprochant de la composante (cf. Section 3.2.3.4). Le modèle le plus simple consiste à utiliser un vecteur de gradients. Le gradient permet d'avoir une représentation relativement indépendante de la luminosité de l'image. Le vecteur de gradient permet de représenter l'assombrissement graduel des pixels sur une ligne (Figure 3.26). En pratique, nous n'utilisons qu'une ligne de gradients horizontale car la forme modélisée est plus prononcée horizontalement, notamment pour les yeux. Il existe plusieurs méthodes pour calculer l'appariement entre deux vecteurs. Dans notre cas, le système calcule la

distance euclidienne entre ces vecteurs. Afin de détecter les bords de part et d'autre des composantes, le système est amené à calculer deux distances. Cependant, nous ne tenons compte que de la plus faible distance, puisqu'elle est liée au bord le plus proche. Cela correspond au calcul d'une corrélation entre le vecteur et l'image et une corrélation entre le même vecteur inversé dont les valeurs sont multipliées par -1 et l'image. Ces calculs sont réalisés suivant les équations présentées ci-dessous, où $\mathcal{V}$ est le vecteur de gradients de taille n qui correspond à la forme à reconnaître, $\delta_x(x,y)$ le gradient aux coordonnées x, y dans l'image I (3.16), $D_{\mathcal{V}}(x,y)$ et $D_{-\mathcal{V}}(x,y)$ les distances euclidiennes calculées pour ces coordonnées (équations 3.17 et 3.18) et $D(x,y)$ est la distance qui est utilisée pour ce point (3.19).

$$\delta_x(x,y) = \frac{\partial I(x,y)}{\partial x} \tag{3.16}$$

$$D_{\mathcal{V}}(x,y) = \sqrt{\sum_{i=1}^{n}(\delta_x(x+i-\frac{n}{2},y)-\mathcal{V}_i)^2} \tag{3.17}$$

$$D_{-\mathcal{V}}(x,y) = \sqrt{\sum_{i=1}^{n}(\delta_x(x+i-\frac{n}{2},y)+\mathcal{V}_{n+1-i})^2} \tag{3.18}$$

$$D(x,y) = \min(D_{\mathcal{V}}(x,y), D_{-\mathcal{V}}(x,y)) \tag{3.19}$$

Il faut cependant vérifier si ce modèle et la distance choisis sont suffisamment discriminants pour reconnaître la composante recherchée dans toute l'image du visage. Les images de la figure (3.27) montrent le résultat du calcul de distance pour les formes de gradients des bords des narines et des bords des iris. Dans cette figure, les pixels les plus lumineux correspondent aux distances les plus faibles. On constate que si l'on utilise seulement ce critère pour décider de la reconnaissance de la composante, cela risque de produire des erreurs. En effet, beaucoup de parties du visage ont une forme de gradient proche de celle recherchée et il peut arriver qu'une de ces parties ait une distance plus faible que la composante réelle dans l'image.

Traitement de la liste de zones candidates

En restreignant l'espace de recherche, le pré-traitement décrit dans la section précédente permet d'éliminer énormément de parties ambiguës de l'image (près de 97,5 % de l'image pour les narines et 95 % pour les iris). Nous décidons d'avoir une approche globale pour cette partie de la reconnaissance. On peut s'attendre à ce que la forme de gradient, qui correspond à une ligne sur le bord de la composante, soit reconnu (distance faible) plusieurs fois dans une même zone. Cela s'explique par le fait que cette forme correspond au contour du bord de la composante et que ce contour peut être suivi sur plusieurs lignes superposées dans la zone. L'approche proposée consiste à calculer la moyenne des

meilleurs distances dans chaque zone. Cette moyenne correspond à un score de détection de la composante dans la zone et permet de trier les zones dans l'ordre de la reconnaissance la plus sûre vers la moins sûre. Cette méthode élimine le bruit constitué par une distance faible mais sur un point isolé dans l'image, et permet de disposer d'un score de détection fiable. Ce score est donc calculé à partir des n distances les plus faibles d'une zone en moyennant ces valeurs. Dans l'équation (3.20), $(D_1, D_2, \ldots, D_m)$ est la suite de distances calculées pour une zone de m points, triée par ordre croissant, et n est le nombre de distances utiles pour le calcul du score de détection.

$$score = \frac{\sum_{i=1}^{n} D_i}{n} \tag{3.20}$$

Nous disposons donc à ce niveau du traitement d'une liste de zones candidates triées par ordre croissant sur leur score de détection. Il nous est désormais possible de sélectionner la zone ayant le plus de chance d'être (ou d'englober) la composante recherchée. Afin d'augmenter encore la fiabilité de la détection, nous utilisons un autre critère de discrimination qui est différent pour les narines et pour les iris.

Pour les narines : la méthode de décision est différente selon l'état dans lequel se trouve le processus :

– État d'initialisation : les narines sont généralement visibles toutes les deux (cf. Section 3.2.3.4). On peut considérer que dans cet état, afin de mettre en œuvre une détection robuste des narines, on s'attend à ce que le visage ne soit pas de côté. Cette situation correspond à ce qui se passe nécessairement à partir du moment où l'utilisateur interagit avec la machine. Avec cette caractéristique nous pouvons

(a) (b)

FIG. 3.27 – *Images des distances euclidiennes entre une image et les vecteurs de gradient de la narine (a) et de l'iris (b).*

rechercher, dans la liste des zones candidates, la paire de zones assimilable aux narines. Le système prend en considération la première zone $\mathcal{Z}_1$ de la liste. Il cherche ensuite dans la liste la première zone suivante dont la distance avec $\mathcal{Z}_1$ est dans l'intervalle correspondant à l'espace entre deux narines. S'il trouve la bonne zone, il décide que ces deux zones sont les narines. Sinon, il prend la zone qui suit $\mathcal{Z}_1$ dans la liste et il recommence la recherche dans la suite de la liste, et ainsi de suite. S'il atteint la fin de la liste sans avoir trouvé les deux zones adéquates, il décide de ne détecter qu'une narine et il prend la première zone de la liste ;

– État d'adaptation : si le processus de traitement se trouve dans l'état d'adaptation, le système utilise une approche un peu différente. Il considère que la première zone de la liste est une narine. S'il trouve une zone adéquate dans la liste pour la deuxième narine, il dispose donc des deux narines. Dans le cas contraire, il garde la première sans chercher s'il y a dans la liste deux autres zones qui pourraient être les narines. Cela est possible compte tenu, d'une part du faible nombre de zones candidates dans l'état d'adaptation, et d'autre part de la plus grand discrimination des calculs dû à l'adaptation des paramètres de reconnaissance.

Pour les iris : la caractéristique utilisée est l'opacité de la pupille. Pour cela on détermine la valeur du niveau de gris du pixel le plus sombre dans chaque zone. Cette valeur est considérée au même titre que la distance de la zone par rapport au vecteur de gradients. Ceci sert au calcul du score de détection et donc au choix de la zone reconnue comme étant l'iris.

La procédure **Traitements d'image** des processus est décrite dans les états d'initialisation et d'adaptation. Elle a permis de détecter plusieurs candidats pour la reconnaissance des composantes et d'en sélectionner un seul grâce au score de détection associé aux candidats. Cela est-il suffisant pour garantir une sélection correcte ? Les candidats sont triés par rapport à leur score de détection et c'est le premier qui est reconnu comme étant la composante recherchée. Cela signifie que par rapport aux autres candidats, le premier est le plus proche ou le plus probable, mais pas par rapport à la composante à reconnaître. Il est donc possible de reconnaître toute forme autre que la composante recherchée dans l'image. Cette situation n'est pas satisfaisante notamment si l'on veut garantir la robustesse de la détection dans notre système. La solution peut consister à définir un seuil sur le score de détection en dessous duquel la composante n'est pas reconnue. Cette solution est applicable quand le processus se trouve dans l'état d'adaptation, car on sait que l'on a déjà trouvé la composante et on la suit précisément, notamment en adaptant certains paramètres de reconnaissance. Dans ces conditions, le score de détection est suffisamment discriminant. Pour l'état d'initialisation, nous savons que les mesures sont moins précises et nous devons être sûrs de celles-ci avant de pouvoir transiter vers l'état d'adaptation. La

solution adoptée consiste à mesurer un autre critère permettant de calculer un nouveau score qui, associé au score de détection, permettra de prendre une décision sur des seuils. Cette étape est appelée **Validation de la détection**.

3.2.3.4.3 Validation de la détection

Cette partie du traitement permet de valider les choix faits par le traitement précédant dans le processus. Cette validation est réalisée en vérifiant la cohérence dynamique des mesures. Dans l'état d'initialisation, les mesures réalisées sur l'image ne tiennent pas compte des images ou des mesures précédentes. Les traitements utilisent des caractéristiques locales d'un point de vue temporel (statique). Dans l'état d'adaptation, la dynamique est utilisée à travers la prédiction de la localisation de la zone de recherche par extrapolation des localisations précédentes (cf. page 90). Nous utilisons la même technique, en calculant cette fois l'accélération du centre de la zone englobant les deux narines ou du centre de l'iris. Ces calculs sont réalisés avec les équations suivantes, où $x(t)$ et $y(t)$ sont les coordonnées du centre de la zone dans l'image t, $\delta(t)$ est le temps écoulé entre les images $t-1$ et t, $v_x(t)$ et $v_y(t)$ sont les vitesses calculées à l'image t, v'_x et v'_y sont les accélérations calculées à l'image t.

$$\begin{aligned}
v_x(t) &= \frac{\partial x}{\partial t} = \frac{x(t-1)-x(t)}{\delta(t)} \\
v_y(t) &= \frac{\partial y}{\partial t} = \frac{y(t-1)-y(t)}{\delta(t)} \\
v'_x(t) &= \frac{\partial^2 x}{\partial t^2} = \frac{\mid v_x(t-1)-v_x(t) \mid}{\delta(t)} \\
v'_y(t) &= \frac{\partial^2 y}{\partial t^2} = \frac{\mid v_y(t-1)-v_y(t) \mid}{\delta(t)}
\end{aligned}$$

Le problème est d'assurer un seuillage fin sur l'accélération dans toutes les directions. En effet, si nous n'utilisons qu'un seuil sur l'axe des abscisses et un autre sur l'axe des ordonnées, la zone de seuillage a une forme rectangulaire. Un seuillage sur une forme rectangulaire n'est pas homogène, particulièrement dans les coins du rectangle où le seuil a une valeur supérieure (qui peut aller jusqu'a $\sqrt{2}$ fois la valeur du seuil la plus grande des deux axes). Pour résoudre ce problème, nous calculons un vecteur d'accélération dans le plan et nous établissons la limite sur le bord d'une ellipse. À partir de l'équation d'une ellipse (3.21), orientée selon les axes du système, où a et b sont les bornes en abscisse et en ordonné de l'ellipse, nous pouvons définir l'espace incluant les valeurs inférieures aux seuils (3.22). Nous utilisons les seuils d'accélérations maximales ($v'_{x\ max}$ et $v'_{y\ max}$) comme bornes de l'ellipse. Le seuillage est basé sur l'équation (3.23).

$$b^2x^2 + a^2y^2 - a^2b^2 = 0 \quad (3.21)$$

$$b^2x^2 + a^2y^2 \leq a^2b^2 \quad (3.22)$$

$$v'^{\,2}_x v'^{\,2}_{y\,max} + v'^{\,2}_y v'^{\,2}_{x\,max} \leq v'^{\,2}_{x\,max} v'^{\,2}_{y\,max} \quad (3.23)$$

Si le seuil d'accélération maximale est dépassé, on considère que la composante n'a pas été reconnue. Dans le cas contraire, on normalise la valeur de l'accélération selon l'équation (3.24) et on utilise ce résultat comme score pour la validation (3.25).

$$v'_{x\ normal} = \frac{v'_x}{v'_{x\ max}} \qquad v'_{y\ normal} = \frac{v'_y}{v'_{y\ max}} \quad (3.24)$$

$$v'_{normal} = \sqrt{v'^{\,2}_{x\ normal} + v'^{\,2}_{y\ normal}} \quad (3.25)$$

Finalement, le score de validation correspond à la moyenne entre le score de détection et l'accélération normalisée. Le système peut donc décider s'il a reconnu ou non la composante. De plus, si le processus se trouve dans l'état d'adaptation : soit la composante n'est pas reconnue, soit elle est reconnue mais il peut y avoir des erreurs, soit elle est reconnue à coup sûr et dans ce dernier cas il est possible pour le processus de s'adapter en fonction des mesures réalisées sur cette composante. Le résultat renvoyé par le système contient un score de confiance sur la détection, qui n'est autre que le score de validation.

3.2.3.4.4 Adaptation des paramètres

Lorsqu'un processus est dans l'état d'adaptation, qu'il a fini ses traitements et qu'il a détecté sa composante avec un score de confiance suffisamment élevé, il semble judicieux de profiter de cette situation pour ne plus utiliser les paramètres de reconnaissance généraux. En effet, on peut mesurer les paramètres spécifiques au visage de la personne afin de mieux réaliser la détection de la composante. Pour cela, il faut d'une part, établir quels sont les paramètres qu'il est intéressant d'adapter. D'autre part, la méthode employée pour réaliser cette adaptation peut influer sur le résultat. En effet, si on utilise directement les valeurs mesurées sur l'image courante pour réaliser la reconnaissance dans l'image suivante, on risque de générer des paramètres spécifiques à l'image elle-même mais pas forcément au visage. Dans ce cas, les paramètres ne sont pas assez spécifiques pour un même visage dans des images différentes et la reconnaissance n'est pas améliorée. Si par contre, on utilise une méthode qui généralise à outrance à partir des mesures réalisées, cela n'apporte rien à la reconnaissance.

Nous distinguons deux types de paramètres : d'une part les valeurs de références, que l'on utilise en calculant une distance par rapport aux valeurs mesurées dans l'image (par exemple la forme de gradient, page 92) ; et d'autre part les seuils qui correspondent à des

valeurs maximales ou à des intervalles dans lesquels doivent se trouver les valeurs mesurées ou calculées dans l'image. Dans le premier cas, le problème consiste à préserver le caractère discriminant de ces valeurs de référence. Nous montrons que la méthode de calcul de la distance, utilisant ces références, permet de réaliser une adaptation discriminante avec des valeurs mesurées directement dans l'image. Pour le second cas, le problème est plus compliqué, il faut créer un intervalle à partir d'une seule valeur. La solution pourrait être de calculer des bornes statistiques à partir d'un échantillon de mesures. Nous préférons écarter cette solution (du moins pour le moment) car elle nécessite un échantillon important et surtout représentatif des valeurs mesurées pour être efficace. De telles méthodes sont utilisées dans le traitement automatique de la parole, où l'on utilise des modèles de reconnaissance multi-locuteurs en début de traitement et des modèles mono-locuteur adaptés par apprentissage au cours du traitement. Ces méthodes font appel à des modèles stochastiques de reconnaissance comme les modèles de Markov Cachés ([Lee et al.88]). Cela permet d'avoir un système cohérent entre la reconnaissance, l'apprentissage et l'adaptation. Compte tenu des choix que nous avons fait sur les techniques de reconnaissance, il serait complexe d'y introduire ces méthodes uniquement pour l'adaptation. En effet, nous utilisons des méthodes de reconnaissance par seuil et appariement de forme, alors que les méthodes stochastiques nécessitent un apprentissage et une modélisation des données sous forme statistique. Cependant, nous pensons qu'il est possible d'introduire ces techniques stochastiques dans divers traitements et de les utiliser alors de manière simple pour l'adaptation (cf. Chapitre 5).

Adaptation de la forme de gradients

La première possibilité d'adaptation consiste à exploiter des informations photométriques. Lors de la reconnaissance des formes des bords des composantes, le système utilise un vecteur de gradients horizontaux (cf. Section 3.2.3.4.2). Pour réaliser l'adaptation de ce vecteur, nous voulons simplement le mesurer dans l'image courante. Pour que le vecteur de gradients permette de calculer des distances aussi discriminantes que le vecteur général, il nous faut montrer que la méthode de calcul de cette distance n'est pas sensible à la spécificité d'une forme de gradient. L'évaluation de la distance entre ce vecteur et les vecteurs de gradients dans l'image est réalisée en calculant une distance euclidienne. Cette distance a l'avantage de donner une valeur reflétant l'homogénéité des différences entre les composantes des vecteurs. Ainsi, elle reste discriminante même lorsque les valeurs de vecteurs de gradients sont éloignées de celles du vecteur de la forme à reconnaître. Nous montrons que la proximité globale de la forme est plus importante que la proximité locale en reprenant l'équation donnée dans la section sur la reconnaissance des formes et en remplaçant les valeurs des composantes des vecteurs par les distances correspondantes d_i.

$$D = \sqrt{\sum_{i=1}^{n} d_i^2}$$

Considérons le cas homogène où toutes les distances d_i ont la même valeur $(u+1)$ positive :

$$\sqrt{\sum_{i=1}^{n}(u+1)^2} = \sqrt{n(u+1)^2} = \sqrt{nu^2+2nu+n}$$

Considérons maintenant le cas non homogène avec $\forall i < n, d_i = u$ et $d_n = (u+n)$:

$$\begin{aligned}\sqrt{(u+n)^2+\sum_{i=1}^{n-1}u^2} &= \sqrt{(u+n)^2+(n-1)u^2} \\ &= \sqrt{u^2+2nu+n^2+nu^2-u^2} \\ &= \sqrt{2nu+n^2+nu^2}\end{aligned}$$

$$\sqrt{nu^2+2nu+n} < \sqrt{nu^2+2nu+n^2}$$

Nous constatons que la distance calculée dans le cas homogène est inférieure à celle du cas non-homogène, alors que la moyenne des distances par composante est identique :

$$\begin{aligned}\frac{\sum_{i=1}^{n}u+1}{n} &= u+1 \\ \frac{(u+n)+\sum_{i=1}^{n-1}u}{n} &= \frac{(u+n)+(n-1)u}{n} = u+1\end{aligned}$$

Si la forme de gradients à reconnaître, mesurée pour l'adaptation, est globalement proche de la forme générale, on dispose d'une forme discriminante. Nous sélectionnons donc pour l'adaptation le vecteur de gradients qui dans l'image est le plus proche du vecteur de gradients général. Ce vecteur est utilisé dans l'image suivante pour reconnaître la composante. S'il y a lieu de réaliser de nouveau une adaptation des paramètres, c'est le vecteur de gradients général qui est utilisé pour sélectionner le nouveau vecteur de gradients d'adaptation. Cette méthode présente l'avantage d'éviter que la forme à reconnaître ne diverge au fur et à mesure des adaptations.

Adaptation du seuil des pixels sombres

L'autre paramètre photométrique sur lequel il est possible d'opérer une adaptation, est le seuil des pixels sombres. Nous réalisons déjà une adaptation dynamique de ce seuil en mesurant la valeur limite de luminosité d'un pourcentage de pixels sombres dans l'histogramme de l'image (cf. Section 3.2.3.4.1). Le but de cette nouvelle adaptation est de

donner une mesure plus précise de ce seuil, ceci en l'évaluant à partir de l'histogramme des pixels de la boîte englobante de la composante. La zone de calcul de l'histogramme correspond à celle définie par la boîte englobante élargie de 5mm de chaque côté, car la boîte telle quelle ne comporte en général pas assez de pixels pour permettre une évaluation précise du seuil. Mais la valeur de ce seuil ne peut être utilisée directement dans l'image suivante. En effet, les conditions de luminosité peuvent avoir changé, ne serait-ce que légèrement entre les deux images. On réévalue un seuil des pixels sombres pour l'image courante dans la zone de recherche extrapolée. Puis, on utilise la moyenne du seuil adapté et de celui mesuré dans l'image courante.

Adaptation des paramètres morphométriques

Les paramètres morphométriques sont des seuils ou des intervalles. Pour les adapter en fonction des mesures réalisées pour chaque composante, nous définissons des coefficients de réduction et d'augmentation permettant de calculer les bornes des intervalles. Le principe de base utilise les équations suivantes, où $\mathcal{L}_c(t)$ contient par exemple la longueur mesurée de la composante dans l'image t :

$$\begin{aligned} \mathcal{L}_{min} &= \frac{1}{2}\mathcal{L}_c(t) \\ \mathcal{L}_{max} &= 2\mathcal{L}_c(t) \end{aligned}$$

Les coefficients fixés dans les équations illustrent l'utilisation qui est faite de ces intervalles. Ils servent plutôt à exclure les zones qui ont peu de chance de contenir la composante, qu'à reconnaître cette composante. On calcule ainsi les intervalles suivants : tailles des zones sombres, distance entre les narines et distances entre les narines et les yeux. Ce dernier intervalle correspond à la seule contrainte de cohérence globale des mesures des composantes du visage. En effet, toutes les autres contraintes utilisées, sont des caractéristiques locales de chaque composante. La distance entre les narines et les yeux permet de calculer la taille de la zone de recherche de chaque iris au-dessus et de part et d'autre du nez. Cette contrainte est adaptée au niveau du processus du visage, lorsque celui-ci ainsi que les processus des narines et des iris se trouvent dans l'état d'adaptation. Ces paramètres sont donc évalués en dernier avant le traitement d'une nouvelle image.

Nous montrons l'efficacité de ces adaptations d'un point de vue quantitatif dans le chapitre (4.3). La description de la première partie des traitements du système CapRe s'achève ici. Nous disposons à la sortie de ces processus d'informations sur la localisation des narines et des iris dans l'image. Elles peuvent donc servir d'entrées aux processus de calcul de la direction du regard de l'utilisateur.

3.2.4 Processus de mesure

Nous avons vu dans la description de la structure générale d'un système de capture (Section 3.2.2, page 59) que le calcul de la direction du regard à partir d'une image est un problème complexe. Nous pensons qu'il est important de disposer de résultats des processus de détection et de suivi, qui soient exploitables en termes de précision, de fiabilité et de robustesse. Or, nous montrons dans le chapitre (4.3) que le processus de détection des yeux manque de fiabilité et de robustesse notamment lorsque l'utilisateur porte des lunettes. Il serait donc nécessaire d'améliorer ces traitements si l'on veut mettre au point un système de capture du regard complet et exploitable dans le cadre de l'interaction homme-machine. Nous présentons cependant un état de l'art concernant les processus de mesure et une étude visant à vérifier la validité d'une technique simple de mesure de l'orientation des yeux par rapport au visage.

4e processus : Orientation des yeux par rapport au visage

Ce processus consiste à calculer un vecteur $Y_{(x,y)}$ dans le plan à partir de l'image des yeux. Ce vecteur indique l'orientation d'un œil par rapport au visage. Pour le calculer, il faut localiser un repère fixe par rapport à l'œil et un repère fixe par rapport au visage. La position relative entre ces deux repères peut servir de base à l'évaluation de l'orientation de l'œil par rapport au visage. La mesure de ces repères est réalisée sur une image, qui est la projection du visage sur un plan. Cette projection donne des images différentes selon l'orientation du visage. Pour une même orientation de l'œil par rapport au visage, on obtient donc des mesures différentes entre les deux repères. Les techniques présentées ci-dessous ne suffisent pas à calculer l'orientation "réelle" des yeux par rapport au visage, mais donne une évaluation utile pour réaliser ce calcul.

Baluja et Pomerleau [Baluja et al.94] et Stiefelhagen, Yang et Waibel [Stiefelhagen et al.97] utilisent un réseau de neurones avec des images des yeux en entrée. Baluja et al. donnent en entrée du réseau l'image d'un seul œil dans une fenêtre de 30×15 pixels. Ils obtiennent dans le meilleurs des cas une précision de $1{,}5^o$. Stiefelhagen et al. utilisent les images des deux yeux dans des fenêtres de 20×10 pixels chacune, comme entrée d'un réseau de neurones. Ils ont mesuré $1{,}9^o$ de précision avec un réseau entraîné avec 4 personnes différentes.

Varchmin, Rae et Ritter [Varchmin et al.98] calculent l'orientation d'un œil sur une image de 40×20 pixels grâce à la technique du *eigenspace*, appelée *eigeneye*. Ils ont réalisé un apprentissage dépendant de l'utilisateur sur dix *eigeneye*. Ceux-ci servent de filtres appliqués à l'image de l'œil. Ils obtiennent donc dix valeurs représentatives de l'orientation de l'œil. Ces valeurs sont utilisées par la suite pour calculer la direction du regard par rapport à un écran.

Nous proposons d'évaluer une approche plus simple. Le système CapRe localise la pupille de chaque œil. Cette pupille peut servir de repère fixe par rapport à l'œil. Il reste donc à trouver un repère fixe par rapport au visage. Il faut chercher parmi les éléments

se trouvant autour de l'œil. On ne peut pas utiliser ceux qui ne sont pas fixes par rapport au visage comme les sourcils ou les cils. On peut chercher à localiser les coins de l'œil, qui ne sont pas ou peu mobiles par rapport au visage. Nous avons constaté que compte tenu de la taille des images des yeux [3] et des conditions d'éclairage, il est difficile de réaliser ce type de localisation de manière précise. Nous choisissons donc une caractéristique plus globale qui ne nécessite pas de localisation précise. On peut déterminer "grossièrement" quels sont les pixels qui appartiennent au blanc de l'œil, en calculant un seuil des pixels clairs à partir de l'histogramme. Ensuite, en calculant le vecteur formé des coordonnées de la pupille et de celles du barycentre des pixels blancs, on peut évaluer la rotation de l'œil. Compte tenu du fait que la partie blanche de l'œil est surtout visible de part et d'autre de l'iris et très peu au dessus, on espère discriminer plus facilement des rotations horizontales que verticales. La somme des vecteurs de rotation de chaque œil, donne un vecteur de l'orientation du regard par rapport au visage. Nous montrons les limites de cette solution dans le chapitre consacré à l'évaluation du système (cf. Section 4.3.4).

5e processus : Orientation du visage par rapport à l'écran

Le but recherché est de calculer la localisation de points dans l'espace à partir d'une projection de ces points sur un plan et d'en déduire le vecteur $V_{(x,y,z)}$ d'orientation du visage. Vo et al. [Vo et al.95] proposent d'utiliser un réseau de neurones avec l'image du visage en entrée et des valeurs angulaires en sortie. Ce système est gourmand en temps de calcul et a une précision faible de l'ordre de 12^o sur des rotations autour de l'axe y (axe vertical haut-bas). Stiefelhagen et al. [Stiefelhagen et al.96] utilisent l'algorithme *POSIT* de DeMenthon [DeMenthon et al.92] qui permet de faire une approximation de la position de points dans l'espace. Cet algorithme nécessite de connaître au moins quatre points de correspondances 3D-2D. Ces points ne doivent pas être coplanaires. Stiefelhagen et al. déterminent le meilleur sous ensemble de points grâce à la méthode de Gee et Cipolla [Gee et al.95]. Ils obtiennent une précision de 5^o sur les axes x et y (axes horizontal droite-gauche et vertical haut-bas) et de 1^o sur l'axe z (axe horizontal avant-arrière). Une méthode plus simple pour le calcul de l'orientation du visage, a été développée par Horprasert, Yacoob et Davis [Horprasert et al.96]. Elle permet à partir de cinq points dans l'image, les quatre coins des yeux et le bout du nez, de calculer l'orientation dans l'espace d'un modèle 3D du visage. Ce modèle nécessite de connaître à l'avance la taille du nez et la distance entre les deux coins d'un œil, pour cela les auteurs utilisent des données anthropométriques [Farkas94]. La précision des calculs dépend de la précision des points mesurés dans l'image et de la différence entre les données anthropométriques moyennes et les données réelles du visage. Il serait intéressant de comparer ces deux dernières méthodes dans le même système de capture pour pouvoir juger de leur précision respective.

3. Dans l'image, un œil fait entre 4 et 8 pixels de haut d'une paupière à l'autre, et entre 22 et 30 pixels de large d'un coin à l'autre.

6e processus : Direction du regard par rapport à l'écran

Cette dernière partie est certainement la plus complexe car elle regroupe les problèmes liés aux deux précédentes. Il s'agit à partir des vecteurs $Y_{(x,y)}$ et $V_{(x,y,z)}$ de calculer un vecteur $R_{(x,y,z)}$ de direction du regard. Il faut donc déterminer un polynome F tel que $R = F(Y, V)$. Mais nous avons vu que les valeurs de Y dépendent de l'orientation du visage dans l'espace et donc de V. Les coefficients du polynome F doivent donc être calculés à partir de différentes orientations du visage pour ajuster le vecteur Y selon V. Cela nécessite de réaliser des mesures précises de plusieurs échantillons de R, de V et de Y. De plus, en utilisant les résultats des calculs des processus précédents, on propage leurs erreurs respectives en les combinant. La solution proposée par Varchmin et al. [Varchmin et al.98] consiste à utiliser un outil permettant de réaliser, après apprentissage, des calculs non linéaires : un réseau de projections linéaires locales (*Local Linear Map-network*). Ils donnent en entrée du réseau le résultat des convolutions entre l'image d'un œil et dix *eigeneye*, ainsi que les coordonnées des deux vecteurs formés par la localisation du nez et celle de chaque œil. Cette technique permet de discriminer la direction du regard dans une grille 5×5 sur l'écran, avec une erreur de $1{,}5^o$ horizontalement et $2{,}5^o$ verticalement. Elle intègre l'orientation d'un œil par rapport au visage et celle du visage dans l'espace, mais ne semble pas tenir compte des translations du visage car aucun paramètre relatif à cet aspect n'est utilisé.

Le calcul de la direction du regard dans l'espace, à partir de l'image du visage, reste un problème ouvert. On ne trouve pas d'études réalisant ce calcul de manière complète et précise. Le pré-requis à ce type d'étude est une localisation précise des yeux et des éléments du visage permettant d'évaluer son orientation et sa localisation dans l'espace.

Nous avons exposé une solution permettant de localiser le nez et les yeux dans une séquence d'images, mais il est important d'évaluer la précision, la robustesse et la fiabilité du système. Cette étude est présentée dans le chapitre suivant.

Chapitre 4

Évaluation de CapRe

Les travaux menés, ainsi que les choix et les mises au point présentés dans les chapitres précédents, ne suffisent pas pour juger de l'efficacité du système CapRe et de l'intérêt de son utilisation. Nous avons donné en général des justifications qualitatives lors de la description des divers aspects de l'étude. Cependant, tout au long de cette thèse, il a été fait usage de "moult" évaluations quantitatives, appelées aussi caractérisation des performances (*performance characterization* [Clark et al.97]). En vision par machine ou en interaction homme-machine, l'évaluation est une activité de recherche en soi. En plein développement, la communauté scientifique lui accorde une importance croissante [Bowyer et al.98]. Ceci dit, il n'existe pas de méthode standard d'évaluation, mais on trouve plusieurs méthodes répondant chacune à des objectifs et des moyens précis [AC et al.96].

Le domaine de l'interaction homme-machine a été un cadre pour définir des contraintes de fonctionnement pour CapRe. Ces contraintes sont traduites en termes de traitement d'image et de fonctionnement global du système (cf. Chapitre 3.2). Les méthodes d'évaluation en interaction homme-machine ne sont pas utilisées directement pour l'évaluation du fonctionnement du système, mais elles servent de base pour l'élaboration du corpus de films (cf. Chapitre 4.1), puisque c'est dans ce cadre que doit être utilisé le système. Nous prenons pour référence les méthodes utilisées en vision par machine. Notamment, celles présentées dans des travaux similaires, déjà cités précédemment. Sur cette base, il est possible d'effectuer deux formes d'évaluation : l'évaluation endogène et l'évaluation exogène. L'évaluation endogène permet d'élaborer le système d'une manière progressive. Elle consiste à comparer les résultats obtenus avec différentes méthodes ou différentes valeurs de paramètres utilisés dans ces méthodes, afin de sélectionner la plus performante par rapport aux spécifications du système. L'évaluation exogène permet de comparer des méthodes et des systèmes ayant le même objectif. Elle consiste à utiliser tout ou partie des résultats des évaluations exogènes de chaque système pour les comparer. Pour cela, il est nécessaire de comparer ces systèmes avec les mêmes ensembles de données en entrée [Clark et al.97].

Dans le cadre de ce travail de thèse, nous développons essentiellement l'approche endogène de l'évaluation. L'évaluation exogène est difficile à réaliser parce qu'elle nécessite

de trouver des travaux de recherche similaires ayant été réalisés dans des conditions comparables. Or, les travaux cités précédemment, utilisent leurs propres corpus pour évaluer leurs systèmes.

De manière générale, l'objectif souhaité est de mesurer la capacité du système CapRe à détecter et à suivre le visage, le nez et les yeux de l'utilisateur, ainsi que sa capacité à mesurer la direction de son regard, ceci le long d'une séquence d'images. Il faut donc en premier lieu, décrire une composante importante pour l'évaluation : les séquences d'images. Si les conditions techniques d'acquisition des images vidéo sont décrites dans le chapitre (3.1), il est important de définir le contenu de ces films.

4.1 Définir un corpus de séquences d'images

Pour définir le contenu d'un corpus, il faut préciser d'abord le contexte de son utilisation. Ce travail de thèse se situe à la fois dans l'interaction homme-machine et dans la vision par machine. Il est donc important de s'inspirer des méthodes de conception de corpus de ces domaines pour réaliser un corpus approprié à l'évaluation du système CapRe. Selon le domaine dans lequel on se place, on trouve différents types de corpus.

En interaction homme-machine, Caelen et al. [Caelen et al.97] définissent trois types de corpora différents :

- le corpus-pilote, enregistré en situation réelle, permet de faire une *analyse d'usage* afin de cerner les pratiques et les besoins des utilisateurs ;
- le corpus simulé, enregistré par exemple avec des techniques de magicien d'Oz, permet d'évaluer le système d'interaction dans l'étape de sa conception ;
- le corpus-test, qui permet d'évaluer les performances du système en cours d'élaboration et du système final.

Le corpus simulé s'applique plus à l'élaboration d'un système de dialogue oral ou écrit, voire un système multimodal [AC et al.96] qu'à un système d'interaction par le regard, notamment parce qu'il est difficile dans ce cas d'utiliser la technique du magicien d'Oz. Les mouvements oculaires étant trop rapides et trop fins pour un observateur humain, il n'est possible de les exploiter que de manière grossière. Un problème similaire est rapporté dans [Catinis et al.95] à propos de gestes de pointage avec le doigt sur un écran. Dans le cas présent, on peut construire ce corpus simulé si l'on utilise un oculomètre lors de l'enregistrement du corpus, mais cela est contraire à l'objectif de cette expérience qui est de mettre l'usager dans des conditions réelles d'utilisation du système. Nous allons voir par la suite, comment exploiter un corpus-pilote et un corpus-test.

Dans le domaine de la vision par machine et de manière générale en reconnaissance des formes, on utilise un seul corpus. Lors de la mise en œuvre d'une technique d'apprentissage automatique (réseaux de neurones, modèles de markov cachés ...) ce corpus

est divisé en deux parties (de tailles pas nécessairement égales). Une première partie sert lors de l'apprentissage et la seconde pour tester le système. Clark et Courtney [Clark et al.97] donnent le principe suivant pour la constitution d'un corpus : « *le corpus doit être représentatif du problème traité, en terme de taille, de constituants prélevés et de proportion disponible pour l'apprentissage* ». Nous n'utilisons pas de technique d'apprentissage automatique, nous n'avons donc besoin que d'un corpus de test.

La spécification du système CapRe a été réalisée à partir de résultats des études sur le regard en biologie, en neurologie, en psychologie et en interaction homme-machine (cf. Chapitres 1 et 2). Les informations provenant de ces études peuvent sembler suffisantes, nous n'avons donc pas réalisé de corpus-pilote. Cependant, des études récentes sur la mise au point d'un système de capture et de reconnaissance du geste par caméra [Braffort et al.98], montrent qu'un tel corpus permet de révéler les divers aspects du problème et de réaliser une spécification précise. Pour une étude qui vise à mettre au point un système de ce type (capture du regard mais aussi plus généralement du geste), il convient de réaliser deux corpora :

- le corpus-pilote, comme celui défini par Caelen et al., permet de faire une analyse du comportement de l'utilisateur en train d'interagir avec un système.

- le corpus-test, permet de réaliser les évaluations endogènes et autant que possible exogènes, du système. L'analyse du corpus-pilote et l'utilisation des résultats d'études menées par ailleurs doivent permettre d'établir les conditions d'enregistrement du corpus-test de manière à respecter le principe de Clark et Courtney [Clark et al.97] sur la constitution d'un corpus.

Rappelons l'objectif général fixé pour le corpus de séquences d'images : il doit permettre d'évaluer la capacité du système à détecter et à suivre le visage, le nez et les yeux de l'utilisateur ; et à mesurer la direction de son regard. D'un point de vue de l'interaction homme-machine, nous avons besoin de séquences d'images où l'on voit l'utilisateur en train d'interagir avec la machine. La méthode utilisée classiquement consiste à établir un scénario, qui définit une suite d'actions ou de tâches à réaliser par la personne. La question est donc : quel scénario choisir ? Pour y répondre, commençons par définir quels sont les mouvements oculaires exploitables pour l'interaction.

Quels mouvements oculaires pour l'interaction ?

Nous allons voir parmi les différents mouvements oculaires (cf. Chapitre 2.1), lesquels sont intéressants à exploiter dans le cadre de l'interaction homme-machine :

1. **La vergence**

   Définition : la mesure de la vergence permet d'évaluer la localisation du point de fixation du regard dans l'espace. Les mouvements de vergence sont en général

lents, avec une vitesse maximum de 20 $^{o}/_{s}$, une durée approchant une seconde et une latence de réponse d'approximativement 160 ms ([Shea92], p.259) ;

Intérêt pour l'interaction : la mesure de la vergence peut donc être utilisée dans le cadre d'interactions tridimensionnelles (modeleur 3D, réalité virtuelle ou augmentée) , pour savoir quel est l'objet 3D fixé par le regard de l'utilisateur. Istance et al. [Istance et al.95] montrent les difficultés que cela pose (cf. page 29). Dans le cadre de l'interaction 2D "classique" où les objets sont affichés sur un plan (l'écran), cette information n'est pas nécessaire mais permet d'avoir une redondance rendant plus sûres les mesures du point de fixation du regard ;

Mise en œuvre : la lenteur de ces mouvements fait que la mesure peut être réalisée facilement avec un système évaluant l'orientation et la localisation de l'axe visuel des globes oculaires l'un par rapport à l'autre. Cependant, il faut disposer de valeurs précises pour calculer le point d'intersection des deux axes visuels, ce qui semble difficile à réaliser à partir d'images des yeux.

2. **La poursuite lente et les nystagmus optocinétiques**

Définition : la poursuite d'une cible en mouvement continu est un mouvement lent, généralement entre 30 et 40 $^{o}/_{s}$ mais pouvant aller jusqu'à 100 $^{o}/_{s}$, avec un temps de réaction à un stimulus en mouvement de 125 ms. Ce mouvement permet de garder sur la rétine une image stable de la cible en mouvement. Les yeux bougent à une vitesse proportionnelle à celle de la cible. Si celle-ci excède la vitesse maximum de poursuite des yeux, des saccades correctives viennent parsemer la poursuite. Lorsque tout le champ visuel est en mouvement, la poursuite s'effectue sur une cible qui finit par sortir du champ, les yeux réalisent alors une saccade dans la direction opposée au mouvement du champ visuel pour "attraper" une nouvelle cible à suivre. Cette stratégie est appelée nystagmus optocinétique ([Shea92], p.265) ;

Intérêt pour l'interaction : la poursuite lente est intéressante à capter dans le cadre de l'interaction homme-machine si l'interface est constituée d'objets en mouvement. C'est le cas lorsque l'on fait défiler des objets ou du texte dans une fenêtre ;

Mise en œuvre : la poursuite lente peut être mesurée facilement, ce qui n'est pas le cas des nystagmus optocinétiques trop rapides pour être détectés à partir d'une séquence d'images vidéo. Cependant, il est possible de ne mesurer que les points de départ et d'arrivée des nystagmus et de ne suivre que les poursuites lentes.

3. **Les mouvements vestibulo-oculaires ou coordination yeux tête**

Définition : la coordination du mouvement des yeux avec celui de la tête permet de garder une image stable sur la rétine lors d'une rotation de la tête. Toute

rotation de l'œil de plus de 30^o s'accompagne d'une rotation de la tête ([Charbonnier95], p.17). C'est un mouvement lent pendant lequel l'œil va réaliser, comme pour le mouvement de poursuite, des saccades de correction permettant de diriger l'axe du regard avec précision sur la cible, et cela notamment si le mouvement de la tête se prolonge ([Shea92], p.272) ;

Intérêt pour l'interaction : nous avons pu observer que ce type de mouvement est réalisé lorsque l'on regarde l'écran d'un ordinateur, car le champ de vision dans ce contexte est de l'ordre de 40 à 50^o selon la taille et la distance de l'écran. Il est plus probable qu'il se produise lorsque la scrutation de l'information ou le stimulus affiché à l'écran oblige l'utilisateur à regarder d'un bord à l'autre de celui-ci ;

Mise en œuvre : la mesure de ces mouvements pose les mêmes problèmes que les précédents. Mais Il faut aussi tenir compte des mouvements de la tête qui modifient l'image des yeux comme nous l'avons vu dans le chapitre 3 (page 59).

4. **Les mouvements de saccades oculaires**

Définition : les saccades sont des mouvements rapides conjugués des yeux, utilisés pour scruter et localiser des cibles ou constater leur absence. Elles permettent de centrer l'image de la cible sur la fovéa. La première saccade vers une cible, rate en général celle-ci de peu. Le reste de la distance est parcouru soit par une autre petite saccade corrective, soit par un lent mouvement correctif appelé "glissage", ou alors par une combinaison des deux. La saccade est un mouvement balistique des yeux, c'est-à-dire qu'une fois lancé, il ne peut être interrompu ou modifié avant d'avoir atteint sa cible. En général, la vitesse moyenne de la saccade est proportionnelle à la distance à parcourir : $400\,{}^o/_s$ pour 10^o et $700\,{}^o/_s$ pour 80^o. La plupart des saccades sont inférieures à 15^o. Le temps de latence pour une saccade est d'environ 200 ms ([Shea92], p.276) ;

Intérêt pour l'interaction : mesurer une saccade n'apporte pas d'information pour l'interaction. Par contre, la fixation du regard entre deux saccades est utile pour mesurer où regarde l'utilisateur ;

Mise en œuvre : les saccades comme les nystagmus optocinétiques, sont trop rapides pour être détectés à partir d'une séquence d'images vidéo. Cependant, on peut mesurer les points de départ et d'arrivée, ce qui est utile pour la mesure des fixations.

5. **la fixation et ses micromouvements**

Définition : la fixation a lieu lorsque les yeux sont immobiles. Ils restent entre 100 ms et 1 à 2 secondes sur un point donné et la durée moyenne d'une fixation est entre 200 et 400 ms selon l'individu. Pendant une fixation, les yeux ne sont pas complètement immobiles et effectuent trois types de micromouvements : les micronystagmus, les dérives et les microsaccades. Ces mouvements ont des

fréquences, des vitesses, et des amplitudes caractéristiques différentes. Lors de la fixation, l'œil balaie une plage qui ne dépasse pas 10 minutes d'angle autour de la position moyenne ([Charbonnier95], p.16) ;

Intérêt pour l'interaction : Dans le cadre de l'interaction homme-machine, les micromouvements ne sont pas intéressants à mesurer. Ils risquent de perturber la mesure de la fixation si l'on utilise un outil de mesure extrêmement précis. La fixation elle-même donne une information sur la zone (2^o) d'attention visuelle de l'utilisateur. C'est cette information qui est la plus couramment employée dans les système d'interaction par le regard (cf. Section 1.3.2) ;

Mise en œuvre : l'image des yeux captée par la caméra à une faible résolution, l'amplitude des micromouvements étant inférieure à cette résolution, ils n'y apparaissent pas et ne perturbent donc pas la mesure de la fixation du regard. On peut utiliser la mesure de la fin de la saccade précédente et celle du début de la suivante pour calculer la durée de la fixation. Cette valeur sert de déclencheur pour interagir (cf. Section 1.3.2).

6. **les clignements**

Définition : ce ne sont pas à proprement parler des mouvements oculaires mais ils interviennent de façon régulière dans le champ visuel. Des clignements effectués de manière involontaire sont réalisés en moyenne toutes les 3 secondes et leur durée n'excède pas 160 ms. Les clignements volontaires ont une durée supérieure à 250 ms ([Charbonnier95], p.17) ;

Intérêt pour l'interaction : les clignements volontaires peuvent servir d'actionneur pour l'interaction par le regard, d'autant qu'ils sont suffisamment plus longs que les clignements involontaires pour que l'on puisse les détecter sans ambiguïté ;

Mise en œuvre : le système de capture doit être capable de mesurer si la paupière de l'œil est baissée ou pas. Cette information est relativement facile à mesurer à partir du moment où l'œil a été localisé dans l'image.

Il faut donc un corpus de films de personnes en train d'interagir avec un ordinateur, incluant tout ou partie de ces mouvements. Cependant, ceci n'est pas suffisant, car pour exploiter un tel corpus, il faut aussi disposer des valeurs réelles de la direction du regard, enregistrées en même temps que les films. En effet, il semble difficile de mesurer de telles valeurs avec précision directement sur les images des films.

Il n'existe pas, à notre connaissance, de corpus de ce type. On trouve sur le réseau beaucoup de bases de données de visages captés de face, ce qui ne correspond pas au cadre de capture des images que nous avons défini (cf. Chapitre 3.1). De plus, ces bases ne contiennent pas de films de personnes en train d'interagir avec un ordinateur. Les autres études réalisées sur le même sujet, utilisent soit des techniques de capture différentes, soit des corpus locaux, non disponibles au public. Il a donc fallu constituer un nouveau corpus

de films.

Quels mouvements oculaires dans le scénario ?

Pour réaliser un corpus exploitable lors de l'évaluation, nous avons mis en place un scénario permettant d'amener l'utilisateur à effectuer certains mouvements lors de l'interaction. L'objectif est de mesurer le mouvement des yeux. Les tâches liées à l'interaction avec un ordinateur génèrent souvent de tels mouvements, mais ceux-ci peuvent être plus ou moins complexes. Si par exemple, on demande aux personnes assises devant la machine de réaliser des tâches de lecture, les mouvements des yeux correspondent à une suite de saccades dont les distances et les vitesses dépendent de beaucoup de facteurs ([O'Regan90], p.422), dont la taille des lettres dans le texte [Morrison83], la taille et l'orthographe des mots [Beauvillain et al.98]...Le temps de fixation moyen en lecture est de 250 ms avec un écart type de 100 ms. La distance de progression par saccade est en moyenne de 7 lettres avec un écart type de 3 lettres ([O'Regan90], p.398). Cela donne par exemple, un angle de 4^o pour lire un texte placé à 36 cm des yeux, avec une taille de fonte de 14 points [Morrison83], ce qui correspond à de gros caractères à l'écran.
Utiliser de tels mouvements dans le corpus pose deux problèmes pour l'évaluation : cela n'a d'intérêt que si l'outil de capture peut mesurer des mouvements aussi précis, ce qui n'est pas le cas ; d'autre part, il faut disposer de mesures de références (valeurs réelles de la direction du regard) permettant de calculer l'erreur de mesure de notre système, ce qui nécessite l'utilisation d'un oculomètre réalisant des mesures précises pendant l'enregistrement des films. La mise au point du scénario pour le corpus, tient compte de ces deux contraintes :

- L'utilisateur doit faire des mouvements des yeux faciles à discriminer compte tenu des résolutions spatiale et temporelle du système CapRe : l'alternance saccades-fixations permet cela si les saccades sont réalisées sur des distances assez grandes. En effet, le temps d'une fixation est supérieur à la résolution temporelle du système, on peut donc la détecter et mesurer la direction du regard. Cependant, si nous voulons discriminer deux fixations successives, les saccades qui les séparent doivent être suffisamment espacées, c'est-à-dire l'espace parcouru doit être supérieur à la résolution spatiale du système. Pour réaliser cela, on demande à l'utilisateur de scruter une scène affichée à l'écran. Cette scène représente quelques objets simples répartis sur l'écran de manière équilibrée. Ponsoda et al. [Ponsoda et al.95] montrent que lorsque l'on réalise une tâche de scrutation pour rechercher une icône affichée à l'écran, les mouvements des yeux suivent différentes stratégies. Notamment, deux saccades successives sont réalisées : soit dans la même direction, soit dans des directions opposées, soit dans le sens horaire, soit dans le sens inverse. Cette diversité nous permet de penser qu'une tâche de recherche visuelle simple, engendre des mouvements dans toutes les directions, permettant ainsi d'enregistrer un corpus représentatif dans des conditions simples ;

– On doit enregistrer, en même temps que le film, les localisations des points de fixation du regard. Pour cela, il est demandé à l'utilisateur de cliquer avec la souris sur des endroits précis (cibles) à l'écran. L'utilisateur doit diriger son regard vers la cible pour manipuler la souris de manière à pouvoir cliquer sur cette cible. Ainsi, on dispose des coordonnées à l'écran du point de fixation du regard de l'utilisateur à l'instant du clic souris.

Scénario d'interaction pour le corpus

Le scénario consiste à faire des mouvements des yeux lors de la scrutation à l'écran de dix points caractéristiques. Dix chiffres (de 0 à 9) sont affichés à l'écran, répartis sur les coins, les bords et le centre. Les chiffres sont affichés en blanc et le reste de l'écran est noir. L'affichage des chiffres dans le désordre, oblige l'utilisateur à réaliser une recherche dans l'image pour les lire dans l'ordre croissant (Figure 4.1). Nous demandons au sujet, de cliquer à l'aide de la souris, sur les chiffres dans l'ordre, avant d'appuyer sur la touche q du clavier pour terminer.

Les images commencent à être enregistrées lors du premier clic souris. Ensuite, à chaque clic, on enregistre la position du curseur à l'écran associée au numéro de l'image courante. La pression sur la touche q déclenche l'arrêt de l'enregistrement. Le film et les positions des clics sont enregistrés dans des fichiers différents.

Dans chaque film enregistré, on voit donc une personne déjà assise devant la machine, qui scrute l'écran pour cliquer successivement sur tous les chiffres, puis regarder vers le clavier pour taper sur la touche q. Le curseur de la souris est réglé pour se déplacer rapidement proportionnellement aux déplacements de la souris. Les chiffres affichés étant assez petits par rapport à la taille de l'écran, les personnes doivent apporter une attention particulière pour réaliser un mouvement précis : cliquer sur une petite cible avec une souris rapide. La boucle de coordination visuo-motrice nécessaire pour cette tâche, fait que le sujet doit avoir la cible et le curseur dans sa fovéa à l'instant où il clique. Aussi sommes-nous sûr que les coordonnées relevées par l'enregistrement de la position du curseur lors du clic, sont dans une zone proportionnelle à la taille de la fovéa dans la direction du regard de l'utilisateur. Ces informations ne peuvent pas être évaluées avec précision à posteriori. Il était donc nécessaire de le faire au fur et à mesure de l'enregistrement du film. Cela nous permettra d'évaluer la précision de la mesure de la direction du regard de notre système.

Diversité des visages

La plupart des sujets font partie du laboratoire du LIMSI. Ils ont été choisis dans un objectif de représenter la diversité des visages que l'on peut rencontrer. Cependant, il est bien évident que ces trente trois personnes ne forment pas un corpus de visage exhaustif d'un point de vue de la diversité. Un vrai corpus nécessiterait un travail beaucoup plus important de collecte de films et d'anthropologie. Le trombinoscope présenté page 114, montre une grande partie des différents visages utilisés. Il y a dix femmes et vingt trois

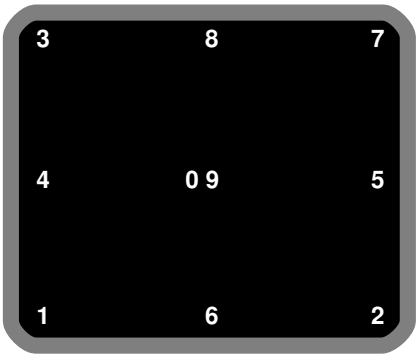


FIG. 4.1 – *Chiffres affichés à l'écran pour le corpus d'interaction.*

hommes, âgés de 21 à 58 ans. On trouve des cheveux de différentes couleurs : blonds, bruns et châtains ; avec diverses coupes et quelques personnes avec une barbe. Les peaux sont en général claires sauf pour deux personnes de peau noire. Les conditions de luminosité sont variables, notamment pour le fond de l'image, du fait de la présence d'une fenêtre sur le côté droit. Un rideau permet d'éviter que la lumière de cette fenêtre ne vienne directement sur le visage. Certains sujets portent des lunettes (15 personnes), nous leur avons demandé quand cela leur était possible, de faire un film avec et un film sans. Cela permet de tester la robustesse du système aux différences intra-personnelle avec sept sujets.

Nous avons demandé aux personnes de s'installer devant la machine de la manière qu'ils jugeaient la plus confortable pour travailler. La plupart d'entre eux, ont l'habitude de travailler avec un ordinateur et savaient déjà manipuler une souris. Toutes les personnes se sont installées bien en face de la machine, mais pas forcement à la même hauteur, ni à la même distance. L'orientation de la caméra a été réglée en fonction de la position du sujet. Après chaque film, nous avons vérifié que le visage se trouvait bien cadré tout le long de la séquence. Pour les premiers films, il y avait une source lumineuse au-dessus de l'écran, orientée vers le mur pour ne pas gêner les utilisateurs. Elle a été retirée parce qu'elle n'apportait que peu de lumière et générait beaucoup de reflets sur les lunettes des sujets.

Les films du corpus

Le résultat donne des films d'une durée de 30 secondes à plus de 2 minutes, avec une moyenne d'environ une minute. En général, les personnes sont pratiquement immobiles au début, puis selon les cas elles bougent plus ou moins la tête. Certaines d'entre elles bougent beaucoup les yeux et font peu de mouvement de tête. Nous observons que les sujets utilisent plus ou moins des saccades oculaires et des mouvements vestibulo-oculaires pour réaliser une même tâche de scrutation. Cette différence de comportement confirme le fait qu'il est important de mesurer à la fois les mouvements des yeux et de la tête, pour calcu-

FIG. 4.2 – *Trombinoscope de la majeure partie du corpus.*

ler l'orientation de ces deux composantes et en déduire la direction du regard (cf. page 59).

Les images des films sont en général exploitables, les narines et les yeux sont visibles par un œil humain, quelle que soit la direction du regard. Cependant, des problèmes apparaissent dans certaines d'entre-elles :

1. les lumières installées donnent des reflets dans les lunettes plus ou moins importants selon l'orientation du visage. Nous avons déjà évoqué la lumière de face qui a été retirée, mais les sources lumineuses de côté génèrent le même problème lorsque la personne oriente son visage vers le côté de l'écran. Ces reflets peuvent cacher tout ou partie de l'œil rendant impossible toute détection même "manuelle" ;

2. les lumières peuvent aussi produire des ombres lorsque la personne est de face. En effet, si la monture des lunettes est épaisse, son ombre se trouve projetée sur l'œil. Cette ombre a moins de conséquences que les reflets, mais elle peut gêner les processus de traitements d'image ;

3. enfin, les personnes qui portent des lunettes et qui ont bien voulu faire un film sans,

ont en général compensé la baisse d'acuité visuelle par un plissement des paupières. Ce phénomène réduit l'image des yeux et gène aussi les traitements d'image ;

4. le dernier problème que nous rencontrons est lié à la caméra. La vitesse de l'obturateur est réglée sur 1/100^{e} de seconde, ce qui n'est pas suffisant lorsque la personne filmée bouge rapidement. C'est rarement le cas, mais cela arrive et a pour conséquence de générer des images floues, plus difficiles à traiter. La solution à ce problème consiste à augmenter la vitesse d'obturation de la caméra. Mais pour cela, il faudrait augmenter aussi la luminosité de la scène soit avec plus de sources lumineuses, soit en ouvrant le diaphragme de l'objectif. Ces problèmes sont décrits dans le chapitre 3 (page 53).

Les séquences d'images sont enregistrées directement sur le disque dur de la machine sur laquelle a lieu l'interaction. C'est sur cette machine que se trouve la carte d'acquisition vidéo à laquelle est connectée la caméra. Les images sont stockées en séquence sans compression, avec pour chacune d'elles le temps écoulé lors de l'acquisition depuis l'image précédente. Ce sont les seules informations dont a besoin le système de capture du regard pour ses traitements. Il n'existe pas de logiciel permettant d'enregistrer des films sous ce format pour la machine utilisée, nous avons donc mis au point un outil d'enregistrement approprié.

Pour pouvoir utiliser le corpus dans l'évaluation de notre système, il faut réaliser un étiquetage de toutes ses images. Nous allons donc détailler cette opération.

4.2 Étiqueter le corpus

L'étiquetage (ou annotation [Caelen et al.97]) d'un corpus consiste à mettre en correspondance des données dites brutes (ici les images) et des valeurs numériques (par exemple les coordonnées du centre de la pupille dans l'image) ou symboliques (l'œil est ouvert ou fermé) d'un plus haut niveau d'abstraction. En l'occurrence nous disposons de films représentés sous forme de séquences d'images codées en niveaux de gris et accompagnées de l'intervalle de temps séparant chaque image de la précédente. Le problème est donc de définir, dans un premier temps, quelles valeurs abstraites nous allons utiliser comme étiquettes. Ensuite, il faut trouver comment réaliser l'étiquetage sur tout ou partie des 16508 images du corpus. Ces opérations sont importantes et nécessitent une certaine

vigilance comme le rappelle Caelen et al. [Caelen et al.97] :

> *«...l'annotation est une opération à mener avec la plus grande circonspection pour deux raisons :*
>
> *a) c'est une méthode lourde et longue qui est très coûteuse en temps et en moyens humains ;*
>
> *b) elle fige les données dans des théories, méthodologies ou principes valables à un instant donné et pour un objectif donné. Un système d'annotation est rarement neutre, peu évolutif et les annotations dépendent en grande partie de l'annotateur lui-même.*
>
> *Nous pensons donc qu'il est important de réfléchir à la généricité d'un corpus...»*

CapRe est structuré fonctionnellement en deux parties : une première partie "Processus de détection et de suivi", qui détecte et suit le visage, le nez et les yeux dans l'image, et une deuxième partie "Processus de mesures" qui évalue la direction du regard (cf. Section 3.2.2). Dans la première partie, chaque processus produit un résultat utilisé par le processus suivant. Il est important d'évaluer ces traitements au fur et à mesure de leur développement. Nous devons évaluer successivement les résultats suivants : la boîte englobante du visage, la boîte englobante des narines, la boîte englobante de chaque œil. Dans la seconde partie, nous avons des résultats moins précis, mais les valeurs de références peuvent être identifiées : un vecteur dans un plan pour l'orientation de chaque œil par rapport au visage, une localisation et trois orientations dans l'espace pour le visage, et la localisation du regard par rapport au plan de l'écran.

Comment étiqueter une boîte englobante dans une image ? Il faut définir les frontières de la composante que doit contenir la boîte englobante. Ces frontières sont soit subjectives et il faut alors les tracer à la main, soit objectives et il faut mettre au point un algorithme capable de les tracer automatiquement. Compte tenu du temps que cela prendrait de réaliser l'étiquetage de toutes les images à la main, il est préférable de choisir l'autre option. Mais celle-ci pose aussi une difficulté, car cela revient à réaliser un système de détection automatique, or c'est dans le but de réaliser un tel système que nous avons besoin d'étiqueter le corpus. Nous adoptons donc une solution intermédiaire, en utilisant une technique appelée *auto-amorçage* ou *boot-strapping*. Dans un premier temps, une partie des algorithmes de détection et de suivi des composantes du visage est mise au point. Le résultat peu fiable permet tout de même de réaliser un étiquetage automatique "sous surveillance". Cela signifie que l'on corrige à la main les erreurs faite par le système au fur et à mesure de l'étiquetage automatique.

Étiquetage de la boîte englobante du visage

On peut définir le visage comme étant la partie de la tête qui contient tout les éléments visibles lorsqu'on la regarde de face : des cheveux jusqu'au menton et d'une oreille à

l'autre [1]. Notre objectif est de disposer de valeurs de référence pour la boîte englobante du visage, afin d'évaluer la précision de localisation de l'algorithme de détection. Compte tenu de la faible précision de cet algorithme, il n'est pas possible de l'utiliser pour l'étiquetage de la boîte englobante du visage. Il est laborieux et difficile de réaliser cet étiquetage à la main. Nous décidons donc d'utiliser d'autres critères pour son évaluation. En fait, ce qui est important dans la boîte englobante du visage, c'est d'être sûr d'y trouver les composantes que l'on va chercher, c'est-à-dire les yeux et le nez. C'est donc ce critère qui permet de réaliser l'évaluation de l'algorithme. Les coordonnées de la boîte formée par les yeux et les narines, sont utilisées comme valeurs limites pour évaluer la boîte englobante du visage. Cette évaluation est plus une mesure de la pertinence de la boîte que de la précision de l'encadrement du visage. Ces étiquettes sont cependant plus génériques que celles correspondant à la boîte englobante du visage. Cette étiquetage est réalisé à partir de l'étiquetage des yeux et du nez sans utiliser l'algorithme de détection de la boîte englobante du visage.

Étiquetage de la boîte englobante du nez et des yeux

Nous avons donné une définition de ces deux composantes en énumérant leurs caractéristiques visuelles (page 84). Compte tenu de la complexité de ces formes, il est difficile de choisir un étiquetage précis. En effet, comme pour le visage, l'algorithme de détection de la boîte englobante ne réalise pas de mesures précises. Afin de réduire cette imprécision deux méthodes différentes sont utilisées pour étiqueter le nez et les yeux. Pour le nez, c'est le centre de la boîte englobante qui est calculé. Pour les yeux nous utilisons le centre de la pupille. Ainsi on dispose d'un étiquetage moins dépendant de l'algorithme utilisé et plus précis. Il permettra en outre d'évaluer facilement d'autres algorithmes par la suite.

Au cours de cet étiquetage, deux difficultés ont été rencontrées avec quelques films. Ces difficultés sont liées à la simplicité des techniques utilisées pour l'étiquetage automatique. Notamment, nous n'avons pas pu étiqueter :

- un film dans lequel plusieurs personnes sont présentes dans l'image car le système n'est pas prévu pour fonctionner dans ce cas. Nous avons constaté par la suite que le système une fois mis au point gère correctement cette situation et fonctionne bien, même si nous n'avons pas pu faire d'évaluation quantitative ;
- deux films d'une personne de peau noir car il était difficile de détecter les composantes du visage. En effet, les techniques de détection des composantes du visage utilisent la luminosité relative du visage (la forme du gradient, cf. Section 3.2.3.4.2). Dans ce cas, la forme est la même mais la dynamique est différente et cela nécessiterait une adaptation de la méthode de reconnaissance.

Nous avons décidé de ne pas utiliser ces films pour l'évaluation, mais de les garder pour des études ultérieures. Il reste donc 39 films, soit 14830 images en tout dans le corpus.

1. Le dictionnaire donne la définition suivante : « *partie antérieure de la tête de l'homme* »

Étiquetage de la direction du regard

Cet étiquetage est réalisé pendant l'enregistrement des films du corpus comme nous l'avons expliqué dans le chapitre 4.1. On peut remarquer que seule la valeur finale du traitement, c'est-à-dire le point de fixation du regard à l'écran, est étiquetée. Il manque donc l'orientation des yeux par rapport au visage et l'orientation du visage dans l'espace.

orientation des yeux par rapport au visage : pour réaliser cet étiquetage il faudrait des outils de mesures montés sur la tête de la personne, tel que nous le décrivons dans le section 2.2.1. De tels outils ne laissent pas toute leur liberté de mouvement aux personnes. Il est donc difficile de les utiliser dans les conditions que nous avons définies, notamment le fait d'utiliser un système non-intrusif. Cependant il serait intéressant de disposer de quelques films enregistrés dans ces conditions, ne serait-ce que pour réaliser une mise au point efficace de cette partie du système ;

orientation du visage dans l'espace : le problème est identique au problème précédent. Il est nécessaire d'utiliser un outil de mesure de l'orientation de la tête monté sur la personne pour réaliser cet étiquetage. Il existe des outils de ce type, comme par exemple le *flock of bird*® [ATC96], suffisamment petits et qu'il est possible d'attacher sur la tête d'une personne à l'aide d'un casque léger. Des mesures peuvent donc être réalisées avec un appareillage un peu moins encombrant que ci-dessus, mais toujours intrusif. Là encore, il serait intéressant de disposer de mesures précises pour la mise au point de cette partie du système.

Dans ces conditions, nous savons que nous pouvons faire une évaluation partielle de notre système. Cela permet de le mettre au point dans son ensemble et de réaliser une première validation des choix qui ont été fait. D'autre part, cela permet de définir les besoins matériels et les conditions d'élaboration du corpus permettant de mettre au point et d'évaluer ce type de système de capture, pour les études à venir.

Ayant défini les étiquettes qu'il a été possible d'associer au corpus de films, nous allons montrer comment elles ont été exploitées pour réaliser l'évaluation du système CapRe.

4.3 Résultats

L'objectif souhaité est de réaliser une évaluation quantitative du système. On définit l'évaluation quantitative comme la partie de la recherche qui consiste à calculer à partir d'un résultat ou un ensemble de résultats $\mathcal{X}_r$ d'un traitement et de valeurs de référence $\mathcal{X}_e$, une ou plusieurs valeurs $\mathcal{Y}$ représentatives de la différence entre $\mathcal{X}_r$ et $\mathcal{X}_e$, en utilisant la fonction : $\mathcal{Y} = \mathcal{F}(\mathcal{X}_r, \mathcal{X}_e)$. Toutes les composantes de cette équation sont à définir :

$\mathcal{X}_r$ contient les mesures que nous réalisons dans les images, notamment les localisations

des composantes, ou des informations sur le comportement du système, comme nombre de transitions entre les états initialisation et adaptation...

$\mathcal{X}_e$ contient les étiquettes correspondant aux valeurs "réelles" des mesures que nous réalisons dans les images ou au cours de l'enregistrement à l'aide d'autres outils de mesure.

$\mathcal{Y}$ contient des valeurs statistiques : moyenne, écart type, taux, maximum, minimum... comme le préconisent Clark et Courtney [Clark et al.97], afin de permettre de comparer ces résultats avec ceux d'autres systèmes.

$\mathcal{F}$ dépend des données en entrée et du résultat escompté. Cependant, les fonctions dont on peut avoir besoin sont des fonctions de statistiques classiques.

Ensuite, il faut interpréter ces résultats et notamment décider de ce que l'on considère comme étant satisfaisant. Pregibon [Pregibon86] propose des règles générales, qu'il applique à l'analyse statistique de données. Ces règles sont reprises par Förstner [Förstner94] pour l'évaluation des outils de vision par machine. On peut considérer l'outil comme satisfaisant si :

- il peut "manipuler" X % de tous les nouveaux problèmes qu'il rencontre ;
- pour les $(100 - X)$ % qu'il ne peut "manipuler", il le sait ;
- vous êtes content avec X.

Pregibon précise qu'une description raisonnable à donner aux pourcentages X, suit une progression en puissance de 2 partageant le nombre de nouveaux problèmes non résolus :

< 50 %	: incomplet
50 % $- 75$ %	: médiocre
75 % $- 88$ %	: bon
88 % $- 94$ %	: exceptionnel
> 94 %	: complet ou presque

Förstner ajoute une présentation plus fine des données, notamment pour évaluer la capacité du système à "s'auto-évaluer" (*selfdiagnosis*). Cela nous intéresse, car notre système réalise cette fonction à travers les scores de confiance calculés après la détection de chaque composante du visage. Le tableau de présentation est le suivant :

		selon l'auto-évaluation : le résultat est	
		correct	erroné
en réalité	correct	**1** décision correcte	**2** décision erronée
	erroné	décision erronée	décision correcte

Les cas **1** et **2** sont importants dans l'évaluation globale du système, car ils reflètent son comportement vis-à-vis de l'extérieur. En effet, le système renvoie comme résultat des valeurs justes dans le cas **1** ou des valeurs fausses dans le cas **2**. Les deux autres cas, sont intéressants lors du développement du système, mais ne renvoient pas de résultats exploitables et n'ont donc pas d'incidence directe vis-à-vis de l'extérieur.

Nous présentons dans les chapitres suivants, les résultats commentés de l'évaluation des processus du système qui ont été implémentés. Il serait fastidieux de présenter et de lire une évaluation exhaustive de tous les paramètres du système. Aussi, nous nous contentons de présenter les points les plus significatifs sur les performances des techniques employées et du système dans son ensemble.

4.3.1 Évaluation du calcul de la boîte englobante du visage

Le processus de détection de la boîte englobante du visage calcule trois valeurs : les bornes horizontales x_d et x_g, et la borne verticale supérieure y_h, de la boîte (page 71). Pour évaluer la pertinence des calculs nous utilisons les coordonnées de la zone rectangulaire englobant les deux yeux et les narines, appelée zone de référence. Il suffit de prendre comme bornes horizontales rx_d et rx_g, respectivement l'abscisse du centre de la pupille de l'œil droit et celui de l'œil gauche. Ces valeurs sont celles étiquetées dans le corpus. Pour la borne supérieure ry_h, on prend l'ordonnée la plus élevée des centres de pupilles des deux yeux. Si l'une des bornes de la boîte englobante du visage est égale à la même borne de la zone de référence, cela signifie que plus de la moitié de l'œil se trouve dans la boîte englobante du visage, ce qui correspond au minimum d'information nécessaire pour détecter l'œil, compte tenu du traitement réalisé pour cela. Pour réaliser l'évaluation, on commence par compter le nombre de fois où l'une ou plusieurs des bornes de référence sont dépassées dans les images (cf. Tableau 4.1).

Bornes	Mesures Brutes		Mesures Filtrées	
	Nb images	Taux	Nb images	Taux
$x_d \geq rx_d$	1747	11,81 %	9	0,06 %
$x_g \leq rx_g$	968	6,54 %	61	0,41 %
$y_h \geq ry_h$	2977	20,13 %	116	0,78 %
$x_d \geq rx_d$ et $x_g \leq rx_g$	453	3,06 %	0	0 %
$x_d \geq rx_d$ et $y_h \geq ry_h$	976	6,6 %	0	0 %
$x_g \leq rx_g$ et $y_h \geq ry_h$	542	3,66 %	0	0 %
$x_d \geq rx_d$ et $x_g \leq rx_g$ et $y_h \geq ry_h$	310	2,09 %	0	0 %
Total dépassements	4031	27,26 %	186	1,25 %

TAB. 4.1 – *Table des dépassements de bornes lors de la détection de la boîte englobante du visage.*

Ceci est fait pour les boîtes englobantes du visage calculées par le processus de

détection du mouvement (cf. Section 3.2.3.3.1) et pour les mêmes boîtes après utilisation du filtre récursif (cf. Section 3.2.3.3.2). On constate que le filtrage réduit considérablement le nombre de dépassements des bornes de références, mais aussi permet de ne jamais dépasser plusieurs bornes sur la même image. De ce fait, si l'on ne considère que les bornes horizontales, on peut s'attendre à ce qu'au moins un des deux yeux soit toujours détectable, soit dans 99,22 % des cas ($y_h < ry_h$).

Distances entre les Bornes	Mesures Brutes		Mesures Filtrées	
	Moyenne	Écart type	Moyenne	Écart type
x_d et rx_d	22,6	20,7	30,4	15,4
x_g et rx_g	28,5	18,8	33,6	14,5
y_h et ry_h	24,2	28,2	33,9	15,1

TAB. 4.2 – *Table des distances moyennes et des écarts types entre bornes lors de la détection de la boîte englobante du visage.*

Pour avoir une idée des variations globales des bornes de la boîte englobante du visage, on évalue la distance moyenne (en mm) entre celles-ci et les bornes de la zone de référence. Le tableau 4.2 permet de voir ces valeurs pour les bornes de la boîte englobante sans et avec filtrage. On constate qu'après filtrage, l'écart type des distances entre les bords du visage et les bornes de référence, permet de présager qu'il existe une marge de plus de 15 mm dans 84 % des cas [2], ce qui est suffisant pour détecter l'œil en entier par la suite. Le filtrage permet en outre d'obtenir des valeurs de moyenne et d'écart type entre les différentes bornes très proches, ce qui semble plus cohérent. Si l'on ajoute une marge de

Bornes	Nb images	Taux
$x_d \geq r'x_d$	37	0,25 %
$x_g \leq r'x_g$	115	0,77 %
$y_h \geq r'y_h$	319	2,15 %
$x_d \geq r'x_d$ et $x_g \leq r'x_g$	0	0 %
$x_d \geq r'x_d$ et $y_h \geq r'y_h$	0	0 %
$x_g \leq r'x_g$ et $y_h \geq r'y_h$	0	0 %
$x_d \geq r'x_d$ et $x_g \leq r'x_g$ et $y_h \geq r'y_h$	0	0 %
Total dépassements	471	3,18 %

TAB. 4.3 – *Table des dépassements de bornes après filtrage et avec une marge de 5 mm, lors de la détection de la boîte englobante du visage.*

2. La répartition des distances est proche de celle de la loi Normale ([Spiegel81], p.71). Dans une distribution normale 68,27 % des cas sont compris entre les bornes de l'écart type de part et d'autre de la moyenne. On ne s'intéresse qu'à un côté de la distribution donc $68,27/2 + 50 = 84,135$ % des cas.

5 mm sur les bornes de références (notées r') pour être sûr d'y inclure les iris [3] des deux yeux, on constate qu'il n'y a des erreurs que dans 3 % des images et que l'on ne dépasse jamais plus d'une borne par image (cf. Tableau 4.3).

En résumé, le processus de détection de la boîte englobante du visage renvoie une information sûre et exploitable pour la suite des traitements (elle contient l'image complète des deux yeux) dans 96,82 % des cas, une information peu sûre mais toujours exploitable (elle contient plus de la moitié de l'image de chaque œil) dans 1,93 % des cas et une information incomplète (il manque plus de la moitié de l'image d'un œil) dans 1,25 % des cas. Ces résultats montrent que ce processus est assez fiable et qu'il constitue un pré-traitement rentable pour la suite des traitements.

4.3.2 Évaluation de la détection des narines

L'évaluation est faite à différents niveaux des traitements. Le principe est de réaliser la détection complète du nez, en ajoutant à chaque évaluation une composante ou information complémentaire pour les traitements. Cela permet de montrer l'efficacité des divers composantes du système pour la détection. Les évaluations sont faites sur toutes les combinaisons des composantes suivantes :

- on cherche le nez dans toute l'image sans tenir compte de la boîte englobante du visage / on cherche le nez dans la boîte englobante du visage ;
- on utilise toutes les zones sombres détectées / on n'utilise que les zones sombres dont les dimensions sont morphologiquement plausibles ;
- le système reste toujours dans l'état d'initialisation / on autorise le système à fonctionner avec ses deux états : l'initialisation et l'adaptation.

On juge de l'efficacité du traitement en mesurant la distance euclidienne, appelée distance d'erreur, entre les coordonnées mesurées du nez et celles de l'étiquetage. On considère comme satisfaisant une distance inférieure ou égale à 5 mm.

Le tableau 4.4 permet de mettre en évidence le gain apporté par les différentes composantes du traitement. On constate l'importance de la sélection des zones sombres selon des critères morphologiques, notamment lorsque le traitement est réalisé dans toute l'image. Ce résultat avec un taux de localisations satisfaisantes de 80 %, montre la capacité de reconnaissance des processus de traitement d'image, qui servent de base à la suite de la localisation. On peut voir aussi, que la stratégie mise en place pour localiser et suivre le nez, avec les états d'initialisation et d'adaptation, donne un résultat très satisfaisant même lorsque le traitement est appliqué dans toute l'image.

3. L'iris mesure environ 8 mm de diamètre. La borne de référence étant son centre, il suffit d'ajouter 5 mm aux coordonnées de celle-ci.

			Distances		Distance $\leq$ 5 mm	
			Moyenne	Écart type	Nb images	Taux
init.	toute l'image	ttes. zones	27,62	35,30	8303	56,17 %
		zones morph.	13,47	28,44	11900	80,50 %
	boîte visage	ttes. zones	8,23	19,32	12603	85,26 %
		zones morph.	6,82	17,44	12992	87,89 %
init. + adapt.	toute l'image	ttes. zones	11,30	25,99	12500	84,56 %
		zones morph.	4,70	15,26	13892	93,98 %
	boîte visage	ttes. zones	4,15	13,35	13987	94,62 %
		zones morph.	3,20	10,63	14179	95,92 %

TAB. 4.4 – *Table des distances d'erreur de localisation du nez, pour différents niveaux des traitements.*

La distribution des distances d'erreur dans la configuration obtenant le meilleur taux de localisation satisfaisante (Figure 4.3) permet de voir que la majeure partie de ces distances se trouve proche de la moyenne (3,2 mm) et donc inférieure à 5 mm.

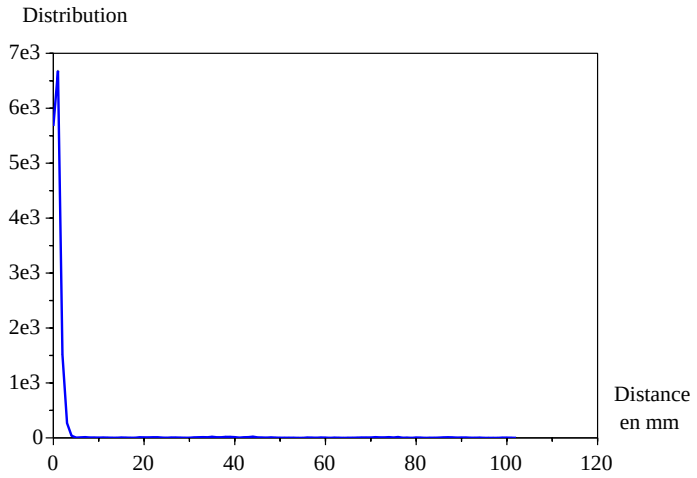


FIG. 4.3 – *Distribution des distances d'erreur de localisation du nez.*

Afin d'analyser plus en détails le comportement du système, nous présentons des statistiques concernant des objets utilisés lors de la détection des narines :

- Le nombre total de zones sombres détectées ;
- Le nombre de zones sombres après élimination des zones hors contraintes morphologiques (essentiellement la taille) ;

- Le nombre de zones sélectionnées comme narines candidates, rangées dans la liste par ordre de scores de détection ;
- Le nombre de narines reconnues (en général le système en trouve deux, mais il arrive qu'il n'y en ait qu'une) ;
- Le numéro d'ordre de la première narine reconnue dans la liste des candidates ;
- Le numéro d'ordre de la seconde narine reconnue dans la liste des candidates.

Pour chacune de ces données, nous indiquons le nombre moyen, l'écart type (ρ) et les valeurs minimum et maximum. De plus, ces valeurs sont indiquées pour toutes les images et pour les cas où la distance d'erreur est inférieure ou égale à 5 mm.

		Tous les cas			Distance $\leq$ 5 mm		
		moy.	ρ	min ; max	moy.	ρ	min ; max
nb zones sombres	total	4,32	5,58	[1; 85]	3,67	4,04	[1; 85]
	filtre morph.	4,05	5,07	[0; 80]	3,47	3,70	[0; 80]

TAB. 4.5 – *Table de statistiques sur le nombre de zones sombres détectées pour les narines, dans tous les cas et pour une distance d'erreur inférieure ou égale à 5 mm.*

Le tableau 4.5 permet de constater que le nombre de zones générées par le traitement est au maximum 85, mais aussi que le traitement le plus gourmand en temps de calcul (cf. Section 3.2.3.4.2) n'est appliqué que dans 80 zones (c.-à.-d. après filtrage sur la morphologie) dans le pire des cas et 4 zones en moyenne. Le fait que la quantité de données à traiter soit extrêmement faible, permet d'assurer un traitement rapide et donc un fonctionnement du système en "temps réel", ce qui est un de nos objectifs.
On note que le fait de réaliser une localisation satisfaisante, n'a pas d'incidence sur le nombre de zones générées au maximum et très peu en moyenne.

	Tous les cas			Distance $\leq$ 5 mm		
narines	moy.	ρ	min; max	moy.	ρ	min; max
nb candidates	4,04	5,07	[0; 80]	3,46	3,69	[0; 80]
n^o 1^e	1,03	0,36	[0; 16]	1,01	0,12	[0; 8]
n^o 2nd	2,29	1,72	[2; 41]	2,08	0,57	[2; 20]

TAB. 4.6 – *Table de statistiques sur les narines candidates lors de la détection, pour tous les cas et pour les cas où la distance d'erreur est inférieure ou égale à 5 mm.*

Le tableau 4.6 présente le résultat des traitements de reconnaissance des narines (cf. Section 3.2.3.4.2). On voit, que le nombre de zones après reconnaissance ne varie pas par

rapport au nombre en entrée du traitement (nombre de zones filtrées sur la morphologie, tableau 4.5). En effet, ce traitement a pour but principal de donner un score de détection à chaque zone sans pour autant prendre de décision sur la reconnaissance des narines. Ainsi la liste des narines "candidates" contient des zones ordonnées selon le score de détection de chacune. Cela explique que la plupart des zones sélectionnées comme étant l'une ou l'autre des narines, le soient en moyenne dans les deux premières positions dans la liste des candidates. L'importance de la dissociation des traitements de reconnaissance et de sélection des narines est illustré par le fait qu'il est parfois nécessaire d'aller jusqu'à la huitième zone pour sélectionner correctement une première narine et la vingtième pour la seconde. Cela n'est possible que parce que l'on a retardé le moment de la prise de décision dans la chaîne des traitements.
Un point peut paraître surprenant, il est possible qu'aucune narine soit détectée mais que la localisation soit satisfaisante (ça n'arrive qu'une fois dans le corpus). Cela est rendu possible par le fait que dans ce cas, on utilise la localisation mesurée dans l'image précédente.

	Distance $\leq$ 5 mm			Distance $>$ 5 mm		
score de	moy.	ρ	min; max	moy.	ρ	min; max
confiance	0,86	0,15	[0; 0,99]	0,33	0,28	[0; 0,99]

TAB. 4.7 – *Table de statistiques sur le score de confiance lorsque la distance d'erreur est inférieure ou égale à 5 mm et lorsqu'elle est supérieure à 5 mm.*

Une fois que le système a décidé s'il a reconnu le nez, il renvoie les coordonnées de celui-ci associées à un score de confiance. Pour savoir si cette valeur est fiable et si la décision du système est correcte, il faut vérifier qu'il est possible de discriminer deux classes de score de confiance : les scores calculés lorsque la distance d'erreur est inférieure ou égale à 5 mm, et ceux calculés lorsque la distance d'erreur est supérieure à 5 mm (cf. tableau 4.7). On établit un seuil permettant de discerner ces deux classes : la moyenne des deux moyennes des scores de confiance dans les deux classes. Ce seuil est utilisé par le système pour décider si le nez a été détecté ou non.

	Distance $\leq$ 5 mm		Distance $>$ 5 mm	
Détection	Nb images	Taux	Nb images	Taux
Acceptée	13427	90,83 %	122	0,83 %
Rejetée	752	5,09 %	481	3,25 %

TAB. 4.8 – *Table de statistiques sur les décisions du système selon le score de confiance.*

Il est ainsi possible de déterminer quatre informations importantes dans l'évaluation (cf. tableau 4.8) :

- le système pense avoir détecté le nez et c'est le cas ;
- le système pense avoir détecté le nez et c'est une erreur ;
- le système pense ne pas avoir détecté le nez et c'est le cas ;
- le système pense ne pas avoir détecté le nez et c'est une erreur ;

Ainsi, nous pouvons remarquer qu'avec le seuil défini ci-dessus (en l'occurrence 0,6), le système fonctionne bien dans 90 % des cas et renvoie un résultat complètement faux dans seulement 0,83 % des cas.

	Distance $\leq$ 5 mm		Distance $>$ 5 mm	
Détection	Nb images	Taux	Nb images	Taux
Acceptée	10758	76,31 %	22	0,16 %
Rejetée	3102	22,00 %	216	1,53 %

TAB. 4.9 – *Table de statistiques sur l'adaptation du système selon le score de confiance.*

La même analyse est menée pour les scores de confiance à chaque fois que le système se trouve dans l'état d'adaptation. En effet, dans cet état, le système peut décider toujours selon le score de confiance, de réaliser une adaptation des paramètres de reconnaissances en fonction des caractéristiques spécifiques des narines qu'il vient de détecter (cf. Section 3.2.3.4.4). Il est important qu'il ne réalise l'adaptation que dans les cas où il est sûr de ne pas commettre d'erreur. Si l'on utilise un seuil plus élevé (0,8), le système réalise une adaptation correcte dans 76 % des cas où il se trouve dans l'état d'adaptation et ne se trompe que dans 0,16 % des cas (cf. tableau 4.9).

Pour finir avec l'évaluation de la détection du nez, nous présentons des informations sur le comportement du système vis-à-vis de ses deux états de fonctionnement : l'initialisation et l'adaptation. Le système se trouve dans 95,37 % des cas dans l'état d'adaptation. Il faut en moyenne 8 images pour que le système transite une première fois de l'état d'initialisation vers l'état d'adaptation, ce qui correspond à un temps inférieur à une seconde (2/3 de seconde). On considère donc qu'il réalise une détection du nez très sûre dès la première seconde du traitement. Le nombre de transitions par personne est en moyenne de 3,5. Si on regarde plus en détails les causes de ces transitions, on constate qu'en général, le système "perd" le nez si la personne effectue un mouvement de tête rapide, générant ainsi une image floue. On constate que de manière générale, la mise au point de la stratégie de fonctionnement en deux états, est assez performante et permet d'exploiter de façon judicieuse les algorithmes de détection et de suivi.

En conclusion, nous considérons que le processus de détection et de suivi du nez donne des résultats suffisamment fiables et justes, pour être exploités dans la suite des traitements, notamment dans la détection des yeux.

4.3.3 Évaluation de la détection des yeux

Nous suivons le même schéma de présentation que pour l'évaluation de la détection du nez. Cependant, nous faisons la différence entre les films où la personne porte des lunettes et ceux où elle n'en porte pas. En effet, le fait de porter des lunettes représente une difficulté sérieuse pour les traitements, ce qui a une incidence sur les résultats comme nous le verrons. D'autre part, nous n'évaluons pas les capacités du système à trouver un œil dans toute l'image, car il est évident que cela donne de moins bons résultats que lorsque l'on restreint la zone de recherche comme nous l'avons fait. Compte tenu des résultats des processus de détection précédents dont dépend le calcul de la zone de recherche des yeux, nous considérons que le système n'a pas besoin d'être performant en dehors de cette zone.

Les évaluations sont donc faites sur des combinaisons des composantes du traitement suivantes :

- on utilise toutes les zones sombres détectées / on n'utilise que les zones sombres dont les dimensions sont morphologiquement plausibles ;
- le système reste toujours dans l'état d'initialisation / le système fonctionne avec ses deux états : l'initialisation et l'adaptation / le système fonctionne avec ses deux états, avec en plus une adaptation "globale" réalisée lorsque le visage est dans l'état d'adaptation (cf. Section 3.2.3.4.4, page 100).

La distance d'erreur est toujours une distance euclidienne, entre les coordonnées mesurées du centre de la pupille et celles de l'étiquetage. La limite pour considérer qu'une distance est satisfaisante est de 4 mm, car cela correspond en moyenne au rayon de l'iris. Les valeurs sont données pour les deux yeux en même temps, le nombre d'images est donc multiplié par deux.

On constate grâce aux valeurs présentées dans le tableau 4.10, que le gain le plus significatif dans les différentes composantes du traitement, vient de l'élimination des zones en dehors des contraintes morphologiques. Le meilleur résultat, avec un taux de localisation satisfaisante de 83 %, même s'il est élevé n'est pas suffisant pour permettre une utilisation du système dans le cadre de l'interaction homme-machine. Il nous permet de valider des choix de techniques de reconnaissance et de constater que celles-ci sont insuffisante pour réaliser une détection fiable.

Cependant, le tableau 4.11 montre qu'il est surtout difficile de détecter les yeux lorsque la personne filmée porte des lunettes. Il y a plusieurs raisons à cela :

- si la monture des lunettes est épaisse et de couleur sombre, elle gène les traitements

		Distances		Distance $\leq$ 4 mm	
		Moyenne	Écart type	Nb images	Taux
init.	ttes. zones	8,76	15,63	17966	60,77 %
	zones morph.	5,07	15,43	24256	82,05 %
init. + adapt.	ttes. zones	8,41	15,23	18265	61,78 %
	zones morph.	4,77	15,24	24604	83,22 %
init.+adapt. +adapt. visage	zones morph.	4,35	14,88	24798	83,88 %

TAB. 4.10 – *Table des distances d'erreur de localisation des pupilles, pour différents niveaux des traitements.*

	Distances		Distance $\leq$ 4 mm	
	Moyenne	Écart type	Nb images	Taux
total	4,35	14,88	24798	83,88 %
avec lunettes	7,28	16,46	8411	72,11 %
sans lunettes	2,44	13,42	16387	91,55 %

TAB. 4.11 – *Table des distances d'erreur de localisation des pupilles, avec et sans lunettes.*

qui détectent les iris. En effet, ces traitements peuvent confondre des parties de la monture avec l'iris ;

- cette monture peut aussi produire une ombre sur l'œil et modifier ainsi la photométrie localement dans l'image ;
- enfin, selon l'orientation du visage, les lumières génèrent des reflets sur les verres des lunettes qui cachent tout ou partie des yeux, les rendant impossible à détecter.

C'est cette dernière qui est majoritairement la cause des problèmes de détection des yeux. On voit cependant, que dans 72 % des cas, la localisation des yeux est correcte malgré les lunettes, ce qui nous permet de penser que les techniques que nous avons mises en œuvre sont insuffisantes mais peuvent servir de base pour l'évaluation de techniques plus complexes. La même remarque peut être faite sur la détection des yeux dans les cas où il n'y a pas de lunettes. En effet, le résultat de 91 % de localisations satisfaisantes, est prometteur mais pas suffisamment fiable.

Si on compare la distribution des distances d'erreur pour la localisation des yeux (Figure 4.4) avec celle du nez (Figure 4.3), on constate que les yeux sont localisés avec beaucoup plus de précision. Cela s'explique par le fait que cette localisation est caractérisée par un repère visible dans l'image, le centre de la pupille, alors que la localisation du nez ne correspond à aucune marque ce qui la rend moins précise et plus subjective. On remarque aussi que la majeur partie des distances d'erreur pour les yeux, est inférieure à

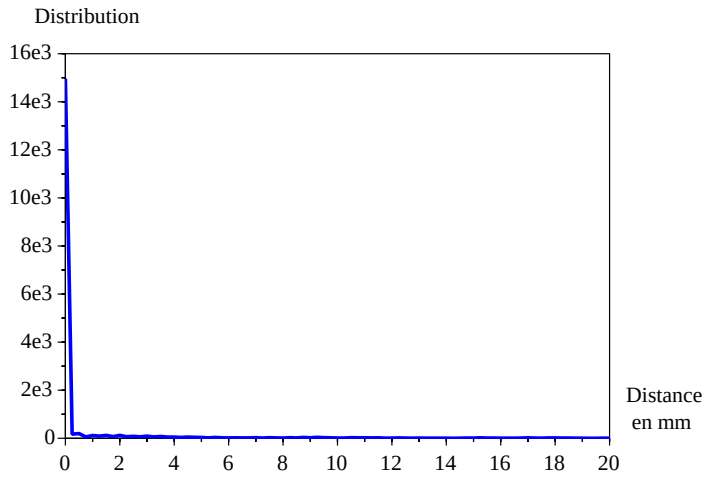


FIG. 4.4 – *Distribution des distances d'erreur de localisation du nez.*

1 mm. Cette précision est importante car on désire utiliser cette information pour mesurer la direction du regard.

Comme pour l'évaluation de la détection des narines, nous exposons des statistiques concernant des objets utilisés lors de la détection des yeux :

- Le nombre total de zones sombres détectées ;
- Le nombre de zones sombres après élimination des zones hors contraintes morphologiques (essentiellement la taille) ;
- Le nombre de zones sélectionnées comme iris candidates, rangées dans la liste par ordre de scores de détection ;
- Le numéro d'ordre de l'iris reconnue dans la liste des candidates ;

Pour chacune de ces données, nous indiquons le nombre moyen, l'écart type (ρ) et les valeurs minimum et maximum. De plus, ces valeurs sont indiquées pour toutes les images et pour les cas où la distance d'erreur est inférieure ou égale à 4 mm.

Les données présentées dans le tableau 4.12 sont issues des traitements de tous les films, car il y a peu de différences entre les données issues des films avec lunettes et celles des films sans. Le nombre maximum de zones sombres est 30, ce qui semble assez élevé si l'on tient compte du fait que le traitement n'est réalisé que dans une partie très restreinte de l'image. Cela s'explique par le fait qu'il arrive que les sourcils, les cils, les cheveux ou les montures des lunettes, génèrent beaucoup de zones sombres en plus de l'iris. Mais ce phénomène reste assez rare, en effet le nombre moyen de zones sombres est de 4. Le filtrage sur la morphologie permet de réduire cette moyenne à 1,6, ce qui illustre l'efficacité de ce

		Tous les cas			Distance $\leq$ 4 mm		
		moy.	ρ	min ; max	moy.	ρ	min ; max
nb zones sombres	total	4,29	3,11	[0; 30]	3,81	2,39	[1; 30]
	filtre morph.	1,64	1,13	[0; 27]	1,64	0,99	[0; 27]
iris	nb candidates	1,50	0,89	[0; 4]	1,54	0,81	[0; 4]
	numéro d'ordre	1,28	0,71	[0; 4]	1,33	0,65	[0; 4]

TAB. 4.12 – *Table de statistiques sur le nombre de zones sombres et d'iris candidates détectées, pour tous les cas et pour une distance d'erreur inférieure ou égale à 4 mm.*

traitement. Le nombre d'iris candidates est volontairement limité à 4 et l'on voit qu'en moyenne il est proche de 1,5, comme le numéro d'ordre de l'iris sélectionnée dans la liste des candidates. Les traitements réalisés pour la détection des iris sont donc suffisamment discriminants pour classer l'iris parmi les deux premières dans la liste des candidates. Si l'on veut améliorer le processus de détection de l'iris et le rendre plus performant, il est donc possible de partir de ce résultat et d'ajouter un traitement permettant de choisir de manière plus sûre l'iris parmi les premières candidates de la liste.
Comme lors de la détection du nez, on note que le fait de réaliser une localisation satisfaisante, n'a pas d'incidence sur le nombre de zones générées au maximum et très peu en moyenne.

		Distance $\leq$ 4 mm			Distance $>$ 4 mm		
		moy.	ρ	min; max	moy.	ρ	min; max
score de confiance	total	0,76	0,22	[0; 1]	0,40	0,34	[0; 1]
	avec lunettes	0,75	0,24	[0; 1]	0,44	0,35	[0; 1]
	sans lunettes	0,76	0,21	[0; 1]	0,31	0,31	[0; 1]

TAB. 4.13 – *Table de statistiques sur le score de confiance lorsque la distance d'erreur est inférieure ou égale à 4 mm et lorsqu'elle est supérieure à 4 mm.*

Les scores de confiance calculés après détection de l'iris sont présentés dans le tableau (4.13). L'objectif est le même que pour le nez, il s'agit de déterminer deux classes dans ces scores, l'une pour l'acceptation et l'autre pour le rejet de l'œil détecté. On note que le score moyen des mauvaises localisations, est plus faible et donc plus facile à discriminer, dans les films où la personne ne porte pas de lunettes que dans les films où elle en porte. Le seuil utilisé dans le système pour décider si l'œil a bien été reconnu, est déterminé en calculant la moyenne des scores des deux classes : environ 0,6.

Comme cela était prévisible, les performances du système au niveau de la décision de détection des yeux, sont meilleurs lorsqu'il n'y a pas de lunettes (cf. tableau 4.14). Le système considère à juste titre qu'il a détecté un œil dans les trois quarts des cas. Il se

Détection		Distance ≤ 4 mm		Distance > 4 mm	
		Nb images	Taux	Nb images	Taux
total	Acceptée	20196	68,31 %	1568	5,3 %
	Rejetée	4602	15,57 %	3198	10,82 %
avec lunettes	Acceptée	6680	57,27 %	1244	10,67 %
	Rejetée	1731	14,84 %	2009	17,22 %
sans lunettes	Acceptée	13516	75,51 %	324	1,81 %
	Rejetée	2871	16,04 %	1189	6,64 %

TAB. 4.14 – *Table de statistiques sur les décisions du système selon le score de confiance.*

trompe en acceptant de mauvaise détection dans presque 2 % des cas. Ces résultats sont une fois de plus, encourageants mais insuffisants. En effet, ils ne permettent pas d'avoir un système fiable et surtout ils montrent que le système ne peut pas *a priori* renvoyer des résultats à la fréquence escomptée de 12 Hz, mais au mieux à 9 Hz en moyenne.

		Distance ≤ 4 mm			Distance > 4 mm		
		moy.	ρ	min; max	moy.	ρ	min; max
Nombre de localisations successives	total	39,49	52,27	[1; 433]	7,54	17,70	[1; 184]
	avec lunettes	33,16	52,97	[1; 433]	12,44	25,31	[1; 184]
	sans lunettes	43,93	51,30	[1; 299]	4,13	7,57	[1; 55]

TAB. 4.15 – *Table de statistiques sur le nombre de localisations successives pour lesquelles la distance d'erreur est inférieure ou égale à 4 mm et celles supérieures à 4 mm.*

En fait, ces erreurs sont rarement ponctuelles. Si on observe les nombres de localisations correctes successives et le nombre d'erreurs de localisation successives (cf. tableau 4.15), on constate qu'en moyenne il se passe environ 3 secondes au cours desquelles la détection est correcte et entre une demi et une seconde au cours de laquelle la détection est erronée.

On évalue aussi la pertinence du système dans l'état d'adaptation, lorsqu'il décide d'utiliser l'œil détecté pour réaliser une adaptation des paramètres de reconnaissances (cf. Section 3.2.3.4.4). En utilisant un seuil d'acceptation plus élevé (0,9), le système réalise une adaptation correcte dans environ 30 % des cas où il se trouve dans l'état d'adaptation et ne se trompe que dans 0,48 % des cas pour les films sans lunettes et 3,78 % des cas avec lunettes (cf. tableau 4.16). On remarque que l'adaptation des paramètres de reconnaissance, est peu fréquente mais correcte lorsqu'il n'y a pas de lunettes. Par contre, cette adaptation n'est pas fiable lorsqu'il y a des lunettes, car elle est réalisée sur des détections erronées plus d'une fois sur dix.

		Distance $\leq$ 4 mm		Distance $>$ 4 mm	
Détection		Nb images	Taux	Nb images	Taux
total	Acceptée	6726	30,02 %	374	1,67 %
	Rejetée	13885	61,98 %	1419	6,33 %
avec lunettes	Acceptée	2321	28,75 %	305	3,78 %
	Rejetée	4440	54,99 %	1008	12,49 %
sans lunettes	Acceptée	4405	30,74 %	69	0,48 %
	Rejetée	9445	65,91 %	411	2,87 %

TAB. 4.16 – *Table de statistiques sur l'adaptation du système selon le score de confiance.*

	État		Transitions	
	Init.	Adapt.	N^o 1^e	Nombre / pers.
total	24,22 %	75,78 %	8,81	37,86
avec lunettes	30,78 %	69,22 %	8,17	39,0
sans lunettes	19,94 %	80,06 %	9,21	35,90

TAB. 4.17 – *Table des distances d'erreur de localisation des pupilles, avec et sans lunettes.*

Les informations concernant le nombre de fois où le système est dans l'un des deux états de fonctionnement (l'initialisation et l'adaptation)(cf. tableau 4.17), conduisent aux mêmes remarques que ci-dessus. Le système se trouve moins souvent dans l'état d'adaptation que lors de la détection du nez. Compte tenu des problèmes de détection que nous avons montrés précédemment, cela semble normal. Par contre, la première transition de l'état d'initialisation vers l'état d'adaptation a lieu en moyenne vers la neuvième image, ce qui correspond comme pour la détection du nez à un temps inférieur à une seconde. On sait que cette détection est sûre mais peu stationnaire. En effet, le nombre de transitions par personne est en moyenne de 37 (dix fois plus que pour le nez). Les problèmes qui génèrent tant de transitions, sont d'une part liées aux images floues comme pour la détection du nez, mais surtout ils sont dus aux techniques employées lors des traitements d'images. En effet, celles-ci sont insuffisantes pour réaliser des détections correctes dans des images aussi petites et complexes [4]. Il faudrait soit changer la taille des images, soit ajouter d'autres techniques pour affiner les détections.

En conclusion, nous considérons que le processus de détection et de suivi des yeux nécessiterait des améliorations pour que les informations qu'il produit puissent être exploitées dans la suite des traitements, notamment pour le calcul de la direction du regard.

4. Dans l'image, un œil fait entre 4 et 8 pixels de haut d'une paupière à l'autre, et entre 22 et 30 pixels de large d'un coin à l'autre.

4.3.4 Évaluation du calcul de la direction du regard

Dans un premier temps, nous évaluons le résultat de la détection des deux yeux en même temps. Le fait d'exploiter les deux yeux pour le calcul de la direction du regard doit permettre une mesure plus sûre. Le tableau (4.18) permet de constater que lorsque l'on croise les résultats de détection des deux yeux, on n'obtient pas de meilleurs résultats que dans l'évaluation séparée. On voit qu'il n'est possible d'exploiter les informations renvoyées par le processus de détection des yeux, que dans 3 images sur 5, dans le meilleur des cas.

		Deux détections correctes		Une ou deux détections erronées	
Détection		Nb images	Taux	Nb images	Taux
total	Acceptée	8045	54.42 %	745	5.04 %
	Rejetée	3239	21.91 %	2753	18.62 %
avec lunettes	Acceptée	2441	41.86 %	532	9.12 %
	Rejetée	1097	18.81 %	1762	30.21 %
sans lunettes	Acceptée	5604	62.61 %	213	2.38 %
	Rejetée	2142	23.93 %	991	11.07 %

TAB. 4.18 – *Table de statistiques sur les décisions du système selon le score de confiance.*

Compte tenu de ces difficultés, l'évaluation du calcul de la direction du regard n'est réalisée que dans les cas où la détection des deux yeux est correcte. Cela permet d'évaluer ce processus indépendamment des autres. Nous cherchons à savoir si la technique utilisée pour calculer la direction du regard permet de discriminer neuf directions différentes. Pour cela, nous avons vu que lors de l'enregistrement du corpus de films, le système enregistrait aussi le numéro de l'image captée au moment du clic souris, ainsi que les coordonnées du curseur à l'écran. Il est donc possible de trouver dans toutes les directions calculées par le système, les neuf directions par film dont on connaît la valeur réelle. On élimine celles calculées à partir de détections erronées, puis on les partage en trois fois trois classes :

- Gauche / Milieu / Droite ;
- Haut / Milieu / Bas.

Ainsi, on peut déterminer les précisions horizontale et verticale du calcul, l'une en fonction de l'autre, pour les deux coordonnées x et y (cf. tableau 4.19).

On s'attend à ce que les valeurs moyennes soient très proches verticalement pour x et horizontalement pour y, et assez éloignées dans l'autre sens. On voit qu'il est possible de distinguer trois classes sur x, mais que le centre se mélange avec le milieu-haut sur y. Cependant, si l'on tient compte des écarts types, il semble difficile de discriminer précisément les neuf classes escomptées. Ces valeurs montrent que les mesures sont imprécises et ne

		Gauche		Milieu		Droite	
		moy.	ρ	moy.	ρ	moy.	ρ
x	Haut	2,13	1,39	0,21	0,94	-1,65	2,26
	Milieu	1,84	1,52	0,22	1,15	-2,43	1,36
	Bas	1,10	1,64	0,57	1,27	-1,05	1,56
y	Haut	2,12	1,30	1,91	1,77	1,93	1,28
	Milieu	1,49	1,42	2,11	1,32	1,22	1,15
	Bas	0,50	1,54	0,07	1,84	0,12	1,63

TAB. 4.19 – *Table de statistiques sur neuf classes de directions du regard, pour tous les films.*

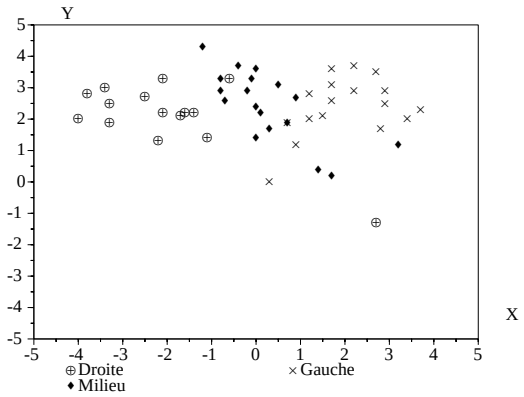
suffisent pas à distinguer ne serait-ce que neuf directions différentes. On peut trouver la cause de ces imprécisions dans le fait qu'il y a des images où la personne porte des lunettes, ce qui peut tromper le traitement qui calcul la direction du regard même quand les yeux ont été correctement détectés. Le tableau (4.20) montre que les mesures sont un peu plus précises quand il n'y a pas de lunettes, mais il est toujours difficile de discriminer les différentes classes de direction du regard. Plus précisément, lorsque la direction est vers le haut ou à mi-hauteur, la discrimination horizontale est possible même si elle reste entachée d'erreurs. Par contre si l'utilisateur regarde vers le bas, les mesures ont tendance à être mélangées, ne permettant pas de réaliser une classification de la direction du regard. Cela est dû en partie au fait que les paupières ont tendance à se refermer, modifiant ainsi l'image de l'œil sur laquelle se base le calcul de la direction du regard.

		Gauche		Milieu		Droite	
		moy.	ρ	moy.	ρ	moy.	ρ
x	Haut	2,17	1,19	0,26	1,03	-2,03	1,60
	Milieu	1,90	1,20	-0,04	0,81	-2,32	1,44
	Bas	0,95	0,77	0,55	1,28	-1,09	1,66
y	Haut	2,42	0,88	2,43	1,09	2,11	1,08
	Milieu	1,67	1,26	2,29	1,21	1,38	0,78
	Bas	0,19	1,65	-0,01	1,69	0,45	1,38

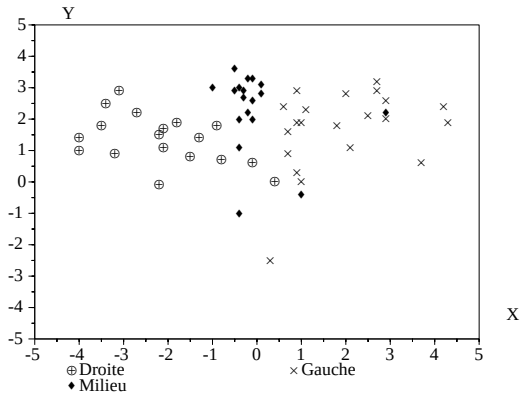
TAB. 4.20 – *Table de statistiques sur neuf classes de directions du regard, pour les films où il n'y a pas de lunettes.*

Ces remarques sont confirmées par l'observation des différentes directions représentées séparément selon la hauteur, dans la figure (4.5).

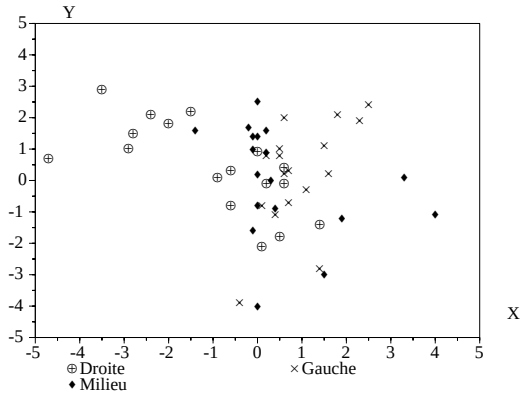
En conclusion, les techniques utilisées pour réaliser le calcul de la direction du regard ne sont pas suffisantes pour donner un résultat précis. Le fait que les paupières se ferment lorsque l'on regarde vers le bas, n'est pas pris en compte et perturbe notablement



(a) vers le haut.



(b) vers la mi-hauteur.



(c) vers le bas.

FIG. 4.5 – *Mesures de direction du regard, regroupées par classe selon la hauteur.*

les calculs. Il est à noter que ce type de problème est rencontré dans d'autres formes d'oculomètres basés sur des systèmes optiques. Il est donc nécessaire d'utiliser des techniques tenant compte de la modification de l'image de l'œil selon l'orientation de celui-ci. D'autre part, l'exploitation d'une image de l'œil de plus grande taille, devrait aussi permettre d'améliorer la précision de la mesure de la direction du regard.

Conclusion

À partir des méthodes d'évaluation utilisées en interaction homme-machine et en vision par ordinateur, il a été possible de définir le protocole permettant d'évaluer le système CapRe. Un corpus de test a été élaboré de manière à rendre compte des conditions dans lesquelles le système doit fonctionner. Après étiquetage du corpus, les différentes composantes logicielles du système ont été évaluées. Les résultats de cette évaluation per-

mettent de valider des choix réalisés lors de l'élaboration du système et de proposer des améliorations de certaines des techniques utilisées. L'évaluation du système CapRe permet de valider les choix suivants :

détection de la boîte englobante du visage

- la technique de détection de mouvements entre deux images successives, permet de calculer un résultat en une passe sur deux images sous-échantillonnées. Ce résultat est trop bruité pour être exploité tel quel pour la suite des traitements. Cette technique est efficace d'un point de vue du temps de calcul mais pas suffisante pour réaliser le traitement escompté ;

- l'utilisation d'un simple filtre récursif après la détection de mouvements, permet d'obtenir un résultat fiable dans 96,82 % des images. Cette suite de traitements est donc appropriée tant d'un point de vue du temps de calcul que de la fiabilité et de la précision du traitement qu'elle réalise.

détection des narines

- la technique de détection des zones sombres permet de réduire considérablement la quantité de données à traiter pour chercher les narines (4 zones en moyenne). Elle nécessite trois opérations de base en vision (calcul de l'histogramme de l'image, seuillage de l'image et génération des zones de pixels), chacune réalisée en une passe sur l'image. Cette technique est donc efficace en temps de calcul et pour sélectionner les données pour le traitement suivant ;

- l'élimination des zones hors contraintes morphologiques rend la détection plus robuste notamment dans les situations "difficiles", comme lorsque la recherche du nez est effectuée dans toute l'image. Cette technique est donc efficace pour la robustesse du système ;

- la détection des narines dans les zones sombres, par un algorithme de corrélation de gradients, fonctionne de manière précise et fiable dans 95,92 % des images. Étant calculée sur une proportion réduite de donnée, elle n'altère pas la rapidité globale des traitements. Cette suite de traitements est donc satisfaisante vis-à-vis du fonctionnement en temps réel du système, de la fiabilité, de la précision et de la robustesse des résultats ;

- la méthode d'auto-évaluation du processus de détection des narines permet à la fois de renvoyer des résultats fiables, de gérer les transitions entre les deux états de fonctionnement (initialisation et adaptation) et de réaliser une adaptation des paramètres de reconnaissance au moment opportun. Elle apporte au système plus de fiabilité et de robustesse ;

détection des yeux

- comme lors de la détection des narines, la technique de détection des zones sombres permet de réduire la quantité de données à traiter (1,6 zones en moyenne). Elle est efficace en temps de calcul et pour sélectionner les données pour le traitement suivant ;

- l'élimination des zones hors contraintes morphologiques augmente considérablement la robustesse de la détection (un tiers de détections correctes en plus).

Cette évaluation permet aussi de mettre à jour l'insuffisance des techniques suivantes :

détection des yeux

- la détection des iris dans les zones sombres, par un algorithme de corrélation de gradients, fonctionne de manière précise et fiable dans 91,55 % des images sans lunettes et dans 72,11 % des images avec. Il serait nécessaire de réaliser d'autres traitements à la place ou en plus de celui-ci pour augmenter la fiabilité et la robustesse du système ;

- le problème exposé ci-dessus se propage dans l'auto-évaluation du processus de détection des yeux. Ce qui réduit la fiabilité des résultats renvoyés, et l'efficacité du fonctionnement en deux états du processus et de l'adaptation des paramètres de reconnaissance.

calcul de la direction du regard

- l'utilisation du blanc de l'œil comme repère pour calculer la rotation du globe oculaire par rapport au visage n'est exploitable que lorsque les paupières sont levées. Cette méthode n'est pas assez fiable pour réaliser un calcul précis, il serait donc nécessaire d'avoir recours à d'autres techniques.

Cette étape de l'étude a permis, en plus de l'évaluation du système CapRe, de poser les problèmes liés à l'évaluation des systèmes de vision appliqués à l'interaction homme-machine. Notamment, elle fait ressortir l'importance des corpus tant pour la spécification des systèmes que pour leur évaluation. Le problème de l'évaluation des systèmes est complexe et mériterait aussi d'être approfondi.

Les résultats présentés dans ce chapitre apportent, par ailleurs, beaucoup d'informations sur le comportement du système et permettent d'envisager les évolutions possibles du système CapRe. Celles-ci sont présentées dans le dernier chapitre de ce mémoire.

Chapitre 5

Perspectives et Conclusion

Plusieurs logiciels ont été développés au cours de cette thèse : *CapFilm*, un logiciel d'enregistrement de films sur disque dur pour réaliser le corpus ; *CapReFilm*, un logiciel de visualisation, d'analyse et d'étiquetage des images des films du corpus, qui contient toutes les fonctionnalités de *CapRe* ; *CapRe*, un logiciel de capture du regard à partir des images provenant de la carte d'acquisition vidéo de la machine. Ce développement a nécessité l'équivalent de dix-huit hommes/mois pour écrire plus de 25 000 lignes en C++. De plus, des programmes ont été développés sous l'environnement de calcul numérique Scilab [1]. Ils permettent d'analyser le comportement de CapRe, de tester le filtrage de certaines données et de réaliser l'évaluation du système. Ce travail a nécessité l'équivalent de trois hommes/mois pour écrire plus de 4 000 lignes en langage scripte de Scilab. Certains de ces logiciels ou une partie de leurs procédures, ont été exploités au sein du LIMSI, pour de récentes études sur la reconnaissance de gestes de la main [Braffort et al.98].

Plusieurs exemples d'applications où le regard est exploité pour l'interaction avec la machine ont été présentées dans la section (1.3). Nous envisageons de tester l'utilisation de CapRe dans d'autres types applications, dont les principes sont présentés ci-dessous :

La navigation 3D : le regard peut être exploité comme outil de navigation dans un environnement virtuel en trois dimensions. La méthode mise au point par Charlier et al. [Charlier et al.92] pour commander le déplacement d'un microscope par le regard, peut être adaptée pour réaliser des translations et des rotations en 3D. Il est cependant nécessaire de prévoir un "levier" d'activation de la commande (de rotation ou de translation) pour éviter le problème du *Midas Touch* (cf. page 24). L'activation de la navigation par le regard peut être faite par une commande vocale ou manuelle. Ce type de navigation devrait être plus naturelle que la manipulation des outils conventionnels comme le *3D tracker*, la *3D trackball* ou le manche à balai 3D ([Buxton87] [Shneiderman87]). Il serait nécessaire de réaliser une étude ergonomique de l'utilisation d'un tel système ;

L'interaction multimodale : les interfaces multimodales sont adaptées pour intégrer

1. Scilab est disponible sur le serveur de l'*INRIA* à l'adresse suivante : http://www-rocq.inria.fr/scilab/

de nouveaux dispositifs d'entrée comme CapRe. La capture du regard seule peut sembler limitée pour réaliser toutes les commandes nécessaires à l'interaction. Une interface multimodale permettra d'utiliser le regard en collaboration avec d'autres canaux de communication comme la parole, le geste ou le clavier. Cette technique rend plus naturel le dialogue entre l'homme et la machine, car elle se rapproche du dialogue entre humains. L'idée principale consiste à lever des ambiguïtés liées à la commande vocale. Le simple fait que l'ordinateur ne sait pas si l'utilisateur lui parle ou bien parle à une autre personne, peut être décidé à partir du moment où la machine sait si l'utilisateur regarde l'écran ou pas. L'exemple peut être étendu dans l'interface de plusieurs applications affichées simultanément à l'écran. Ainsi la machine sait, selon la fenêtre d'interaction qui est regardée par l'utilisateur, à quelle application sont destinés ses ordres qu'ils proviennent de la parole ou du clavier ;

La communication médiatisée et le travail coopératif : ces applications permettent ou nécessitent un dialogue entre plusieurs personnes par le biais d'ordinateurs. Ce dialogue est facilité si les interlocuteurs peuvent se voir, mais cela n'est pas toujours possible car il faut alors transmettre des images en temps réel entre plusieurs machines souvent à travers un réseau. Plusieurs solutions existent pour ce problème, dont l'utilisation d'avatars animés à partir de dispositifs de capture des mouvements du visage [Saulnier et al.95]. Cette solution ne donne pas toutes les informations visuelles dont on dispose lorsque l'on discute en face d'une personne. Par exemple, elle ne permet pas de savoir où la personne est en train de regarder (dans l'écran ou en dehors). Cette information permet de déduire l'activité sur laquelle la personne focalise son attention, ce qui peut être utile dans le cadre d'un travail coopératif ou d'une discussion. CapRe permet d'apporter ce genre d'informations ;

Toutes ces applications ne nécessitent pas une grande précision des mesures de la direction du regard. Les informations renvoyées par CapRe, devront permettre de discriminer entre plusieurs fenêtres d'interaction, ce qui correspond à séparer de 9 à 16 zones à l'écran. On peut imaginer d'autres applications dans lesquelles CapRe pourra être intégré, mais il sera important de poursuivre le développement du système, pour augmenter sa précision, sa fiabilité et sa robustesse.

Cet travail de thèse a permis d'exposer une partie des problèmes liés à la conception d'un système de capture du regard pour l'interaction homme-machine. Des solutions ont été proposées et évaluées au travers du système CapRe. Cependant, certaines de ces solutions ne sont pas satisfaisantes et le système de capture n'est pas complet.

Exploiter la couleur : avec une caméra couleur, on dispose de plus d'informations pour le traitement d'image. Elle peut être utile pour discriminer le visage du fond de l'image notamment lorsqu'il n'y a pas de mouvement, ou pour discriminer le blanc

des yeux et la peau pour la détection de ceux-ci ou le calcul de la direction du regard ;

Implémenter des filtres : il serait intéressant de comparer les résultats obtenus par le filtre récursif utilisé dans cette étude avec ceux d'un filtre de Kalman. En effet, ce filtre est utilisé dans d'autre études pour des problèmes proches ([Ridder et al.95] [Wren et al.96] [DeCarlos et al.97] [Crowley et al.97]). Pour l'adapter à notre situation, il est nécessaire de pondérer certains paramètres du filtre de Kalman avec la mesure du taux de mouvement (page 74). Ce filtre peut être utilisé dans d'autres processus du système, par exemple pour estimer la localisation d'une composante dans la nouvelle image ;

Utiliser des méthodes stochastiques : beaucoup de seuils dans le système pourraient être remplacés par des lois de probabilités calculées à partir d'un corpus d'apprentissage. Ainsi, il serait possible de définir par exemple la probabilité qu'un pixel soit sombre, ce qui ajoute une certaine tolérance utile pour le regroupement en zone des pixels adjacents. De même, lors de la reconnaissance des composantes du visage avec des vecteurs de gradients, on peut remplacer le calcul de la distance euclidienne par une distance de Mahalanobis [Yow et al.97] ;

Intégrer des techniques avec apprentissage : il est intéressant de tester l'intégration de méthodes exploitant l'apprentissage automatique comme les réseaux de neurones ou les *eigenspace*, déjà présentés dans ce mémoire. Le système CapRe réalise un certain nombre de prétraitements visant à réduire l'espace de recherche des composantes et par la même le nombre de données à traiter. Il est donc possible d'exploiter des techniques gourmandes en temps de calcul suite aux pré traitements, sans que cela n'altère le fonctionnement en temps réel. Ces techniques peuvent résoudre certains problèmes de détection des yeux et de calcul de la direction du regard ;

Ajouter un modèle du visage : CapRe utilise des informations sur les composantes pour réaliser les processus de détection. Il est possible de renforcer la robustesse du système en ayant une approche plus globale qui utilise des informations sur les positions relatives entre les composantes, par exemple avec un modèle du visage. Ainsi, la décision de détection des composantes à partir de plusieurs candidates qui est réalisée pour chaque composante indépendamment les unes des autres, pourrait se faire au niveau du visage.

L'étude d'un système de capture du regard dans le contexte de l'interaction homme-machine est une étape dans le développement de nouvelles interfaces plus naturelles et plus efficaces pour la communication entre l'homme et la machine. Cette thèse s'inscrit dans un cadre théorique pluridisciplinaire et s'appuie sur une expérimentation réalisée

dans des conditions réelles d'interaction. Elle a permis la mise au point d'outils logiciels et méthodologiques pour la réalisation d'un système de capture du regard.

Table des figures

Liste des tableaux

Bibliographie

[AC et al.96] Aublet-Cuvelier (Laurent), Carraux (Eric), Coutaz (Joëlle), Nigay (Laurence), Portolan (N.), Salber (Daniel) et Zanello (Marie-Laure). – NEIMO, un laboratoire d'utilisabilité numérique : Leçons de l'expérience. *actes de ERGO.IA 96, Ergonomie et Informatique avancée.* – Biarritz, France, 9–11 octobre 1996.

[ATC96] Ascension Technology Corporation, POB 527, Burlington, Vermont, USA. – *The Flock of Birds®, installation and operation guide*, 22 juin 1996.

[Bahan et al.95] Bahan (Benjamin J.) et Supalla (Samuel J.). – Line Segmentation and Narrative Structure : A Study of Eyegaze Behavior in American Sign Language. *Language, Gesture and Space*, éd. par Emmorey (Karen) et Reilly (Judy S.), chap. 9, pp. 171–191. – Hillsdale, New Jersey, Lawrence Erlbaum Associates, Publishers, 1995.

[Baluja et al.94] Baluja (Shumeet) et Pomerleau (Dean). – *Non-Intrusive Gaze Traking Using Artificial Neural Networks.* – Rapport technique, Pittsburgh, PA 15213, USA, ISL, Carnegie Mellon University, 1994.

[Beauvillain et al.98] Beauvillain (Cécile) et Doré (Karine). – Orthographics Codes are Used in Integrating Information from the Parafovea by the Saccadic Computation System. *Vision Research*, vol. 38, n° 1, 1998, pp. 115–123.

[Bellanger81a] Bellanger (Maurice). – Cellules de filtres à réponse impulsionnelle infinie (RII). *Traitement Numérique du Signal, Théorie et Pratique* [Bellanger81c], chap. 6, pp. 174–199.

[Bellanger81b] Bellanger (Maurice). – Les filtres à réponse impulsionnelle finie (RIF). *Traitement Numérique du Signal, Théorie et Pratique* [Bellanger81c], chap. 5, pp. 133–173.

[Bellanger81c] Bellanger (Maurice). – *Traitement Numérique du Signal, Théorie et Pratique.* – MASSON, 1981.

[Bérard et al.96] Bérard (François) et Coutaz (Joëlle). – Coopération de techniques Sensorielles pour une Interaction écologique. *Actes de IHM'97, 8es Journées sur*

l'Ingénierie des Interfaces Homme-Machine. – Grenoble, France, septembre 1996.

[Besançon88] Besançon (Jacques E.). – Seuillage d'histogramme. *Vision par ordinateur, en deux et trois dimensions*, chap. A4-3.2, pp. 83–87. – EYROLLES, 1988.

[BG et al.94] Brock-Gunn (Simon A.), Dowling (Geoff R.) et Ellis (Tim J.). – Tracking using Colour Information. *Proc. of 3rd Intl. Conf. on Automation, Robotics and Computer.* – Singapore, 1994.

[Birchfield97] Birchfield (Stan). – An Elliptical Head Tracker. *Proc. of 31st Asilomar Conf. on Signals, Systems & Computers.* – Pacific Grove, California, novembre 1997.

[Birchfield98] Birchfield (Stan). – Elliptical Head Tracking using Intensity Gradients and Color Histograms. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition.* – Santa Barbara, California, juin 1998.

[Black et al.95] Black (Michael) et Yacoob (Yaser). – Tracking and Recognizing Facial Expressions in Image Sequences, using Local Parameterized Models of Image Motion. *Proc. of Intl. Conf. on Computer Vision*, pp. 374–381. – 1995.

[Bouzouita et al.96] Bouzouita (Ali), Uhl (Claude), Luciani (Annie) et Cadoz (Claude). – Manipulations complexes d'objets virtuels en présence de retour d'effort. *Actes d'Interface des Mondes Réels et Virtuels*, pp. 459–470. – Montpellier, France, 21–24 mai 1996.

[Bowyer et al.98] Bowyer (K. W.) et Phillips (P. J.) (édité par). – *Empirical Evaluation Techniques in Computer Vision*, IEEE Computer Society. – Santa Barbara, USA, IEEE CS Press, 21–22 juin 1998. Workshop on Empirical Evaluation of Computer Vision Algorithms.

[Braffort et al.98] Braffort (Annelies) et Gherbi (Rachid). – Video-Tracking and recognition of pointing gestures using Hidden Markov Models. *Proc. of IEEE Intl. Conf. on Intelligent Engineering Systems, INES'98.* – Vienne, septembre 1998.

[Braffort96] Braffort (Annelies). – *Reconnaissance et compréhension de gestes, application à la langue des signes.* – Orsay, France, Mémoire de doctorat, Université de Paris-XI Orsay, 28 juin 1996.

[Bruce et al.84] Bruce (Vicki) et Green (Patrick). – Perceiving Depth and Movement. *Visual Perception : Physiology, Psychology and Ecology*, chap. 6, pp. 129–161. – London, U.K., Lawrence Erlbaum Associates, Publishers, 1984.

[Buxton87] Buxton (Willian). – There's More to Interaction Than Meets the Eye : Some Issues in Manual Input. *Readings in Human-Computer Interaction*, éd. par Baecker (R.) et Buxton (W.), pp. 366–375. – Morgan Kaufmann Publishers, 1987.

[Cadoz93] Cadoz (Claude). – Le geste canal de communication homme/machine. *Cours de l'école d'été ARC/PRC CHM*, pp. 35–67. – Bonas, Gers, Association pour la Recherche Cognitive, 4–16 juillet 1993.

[Caelen et al.97] Caelen (J.), Zeiliger (J.), Bessac (M.), Siroux (J.) et Perennou (G.). – Les corpus pour l'évaluation du dialogue homme-machine. *Actes du 1^er^ JST 1997 FRANCIL*. AUPELF•UREF, pp. 215–222. – Avignon, France, 15–16 avril 1997.

[Caelen92] Caelen (Jean). – Compte-rendu du "workshop" Interaction Homme-Machine Multimodales. *Actes de IHM'92, 4^es^ Journées sur l'Ingénierie des Interfaces Homme-Machine*, éd. par Télécom (Paris). GDR-PRC CHM, pp. 213–228. – Dourdan, France, 13–14 avril 1992.

[CAFGR96] IEEE Computer Society. – *Intl. Conf. on Automatic Face & Gesture Recognition*. – Killington, Vermont, USA, 1996.

[Calbris85] Calbris (Geneviève). – Espace-temps : Expression gestuelle du temps. *Semiotica*, vol. 55, n° 1, 1985, pp. 43–73.

[Card et al.80] Card (Stuart K.), Moran (Thomas P.) et Newell (Allen). – The Keystroke-Level Model for User Performance Time with Interactive Systems. *Communication of the ACM*, vol. 23, n° 7, juillet 1980, pp. 396–410.

[Cassell98] Cassell (Justine). – A framework for gesture generation and interpretation. *Computer Vision in Human-Machine Interaction*, éd. par Cipolla (R.) et Pentland (A.). – Cambridge University Press, 1998.

[Catinis et al.95] Catinis (Lian) et Caelen (Jean). – Analyse du comportement multimodal de l'usager humain dans une tâche de dessin. *Actes de IHM'95, 7^es^ Journées sur l'Ingénierie des Interfaces Homme-Machine*, pp. 123–129. – Toulouse, France, 1995.

[Charbonnier et al.94] Charbonnier (Colette) et Massé (Dominique). – écriture par commande visuelle. *Actes d'Interface des Mondes Réels et Virtuels*, pp. 185–191. – Montpellier, France, 1994.

[Charbonnier95] Charbonnier (Colette). – *La commande oculaire : étude et validation exprérimentale d'interfaces homme-machine contrôlées par la direction du regard*. – Grenoble, France, Mémoire de doctorat, Université Joseph Fourier / LETI-CEA, octobre 1995.

[Charlier et al.92] Charlier (Jacques), Sourdille (Philippe), Behague (Maurice) et Buquet (Cathy). – Commande par le Regard d'un Système de Visualisation 2D : exemple du Microscope Opératoire. *Actes d'Interface des Mondes Réels et Virtuels*, pp. 659–666. – Montpellier, France, 1992.

[Chekaluk et al.92] Chekaluk (Eugene) et Llewellyn (Keith) (édité par). – *The role of Eye Movements in Perceptual Processes.* – NORTH–HOLLAND, Elsevier Science Publishers B.V., 1992, *ADVANCES IN PSYCHOLOGY 88 (Ed. G. E. Stelmach and P. A. Vroon).*

[Clark et al.97] Clark (Adrian F.) et Courtney (Patrick). – On Databases for Performance Characterization. *Proc. of Workshop on Performance Characteristics and Quality of Computer Vision Algorithms.* – Braunschweig, Germany, 18–19 septembre 1997.

[Cleveland et al.92] Cleveland (Dixon) et Cleveland (Nancy). – Eyegaze Eyetracking system. *Proc. of Imagina'92: Images Beyond Imagination, 7th Monte-Carlo Intl. Forum on New Images*, pp. 11.15–11.23. – Monte-Carlo, janvier 1992.

[Cleveland92] Cleveland (Dixon). – *The Eyegaze Development System® : a tool for human factors applications.* – LC Technologies, 4415 Glenn Rose Street, Fairfax, Virginia 22032, U.S.A, janvier 1992.

[Collet et al.97a] Collet (Christophe), Finkel (Alain) et Gherbi (Rachid). – CapRe: un système de Capture du Regard dans un contexte d'Interaction Homme-Machine. *Actes des 6es Journées Internationales sur l'interface des mondes réels et virtuels*, pp. 36–39. – Montpellier, France, 28–30 mai 1997.

[Collet et al.97b] Collet (Christophe), Finkel (Alain) et Gherbi (Rachid). – Gaze Capture system in Human-Machine Interaction. *Proc. of IEEE International Conference on Intelligent Engineering Systems, INES'97*, éd. par IEEE, pp. 557–581. – Budapest, Hongary, 15–17 septembre 1997.

[Collet et al.97c] Collet (Christophe), Finkel (Alain) et Gherbi (Rachid). – Prise en compte dynamique des Attitudes Perceptive de l'Usager. – Rapport de synthèse version IV de l'Action Inter-PRC 10.2 GDR-PRC ISIS & CHM: "Interaction Système/Environnement pour l'Interprétation des Signaux et des Images", 1997.

[Collet et al.98a] Collet (Christophe), Finkel (Alain) et Gherbi (Rachid). – CapRe: a gaze tracking system in man-machine interaction. *Journal of Advanced Computational Intelligence*, vol. 2, n° 3, juin 1998, pp. 77–81.

[Collet et al.98b] Collet (Christophe) et Gherbi (Rachid). – Visual Perception Tools for Natural Interaction: a Gaze capture and tracking system. *Proc. of IEEE Southwest Symposium on Image Analysis and Interpretation, SSIAI'98*, éd. par IEEE, pp. 91–96. – Tucson, Arizona, USA, 5–7 avril 1998.

[Collobert et al.96] Collobert (M.), Feraud (R.), Le Tourneur (G.), Bernier (O.), Viallet (J. E.), Mahieux (Y.) et Collobert (D.). – LISTEN : a System for Locating and Tracking Individual Speakers. In CAFGR [CAFGR96], pp. 283–288.

[Cook84] Cook (Mark). – Regard et regard réciproque dans les interactions sociales. *La communication non verbale*, pp. 125–144. – Neuchâtel, Suisse, Delachaux & Niestlé, 1984.

[Cotin et al.96] Cotin (Stéphane), Delingette (Hervé), Clément (Jean-Marie), Tassetti (Vincent), Marescaux (Jacques) et Ayache (Nicholas). – Simulation de chirurgie hépatique avec système de retour de forces. *Actes d'Interface des Mondes Réels et Virtuels*, pp. 139–147. – Montpellier, France, 21–24 mai 1996.

[Courtney et al.97] Courtney (Patrick) et Lapresté (J. T.). – Performance Evaluation of a 3D Tracking System for Space Applications. *Proc. of Workshop on Performance Characteristics and Quality of Computer Vision Algorithms.* – Braunschweig, Germany, 18–19 septembre 1997.

[Coutaz et al.91] Coutaz (Joëlle) et Gourdol (Arnaud). – Communication Homme-Machine Multimodale : perspectives pour la recherche. *Actes des 2es Journées Nationales du GRECO-PRC-Communication Homme-Machine.* – Toulouse, France, 29–30 janvier 1991.

[Crowley et al.97] Crowley (James L.) et Bérard (François). – Multi-Modal Tracking of Faces for Video Communications. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition.* – Puerto Rico, 17–19 juin 1997.

[Cuxac93] Cuxac (Christian). – Iconicité des langues des signes. *Cours de l'école d'été ARC/PRC CHM.* – Bonas, Gers, Association pour la Recherche Cognitive, 4–16 juillet 1993.

[Darrell et al.96] Darrell (Trevor), Moghaddam (Baback) et Pentland (Alex P.). – *Active Face Tracking and Pose Estimation in an Interactive Room.* – Research Result n° 356, Cambridge, MA 02139, USA, M.I.T. Media Laboratory, 1996.

[DeCarlos et al.96] DeCarlos (Douglas) et Metaxas (Dimitris). – The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 231–238. – San Francisco, USA, juin 1996.

[DeCarlos et al.97] DeCarlos (Douglas) et Metaxas (Dimitris). – *Optical Flow Constraints and Deformable Models with Applications to Face Tracking.* – Rapport technique n° MS-CIS-97-23, Philadelphia, PA 19104-6389, CIS, University of Pennsylvania, 1997.

[DeCarlos et al.98] DeCarlos (Douglas) et Metaxas (Dimitris). – Deformable Model-Based Shape and Motion Analysis from Images using Motion Residual Error. *Proc. of Intl. Conf. on Computer Vision*, pp. 113–119. – Bombay, India, 1998.

[DeMenthon et al.92] DeMenthon (D. F.) et Davis (Larry S.). – Model based object pose in 25 lines of code. *Proc. of 2nd European Conference on Computer Vision*, éd. par Sandini (G.). pp. 335–343. – Santa Margherita, Ligure, mai 1992.

[DJ et al.98] Daly-Jones (Owen), Monk (Andrew) et Watts (Leon). – Some advantages of video conferencing over high-quality audio conferencing : fluency and awareness of attentional focus. *International Journal of Human Computer Studies*, vol. 49, n° 1, juillet 1998, pp. 21–58.

[Doval98] Doval (Boris). – Démonstration des conditions de stabilité d'un filtre récursif non invariant de la forme : $y_n = a_n x_n + b_n y_{n-1}$. – Communication personnelle, manuscrit, 1998.

[Ekman et al.72] Ekman (Paul) et Friesen (Wallace V.). – Hand Movements. *The Journal of Communication*, vol. 22, décembre 1972, pp. 353–374.

[Essa et al.95] Essa (Irfan A.) et Pentland (Alex). – Facial Expression Recognition using Virtually Extracted Facial Action Parameters. *Proc. of Intl. Work. on Automatic Face & Gesture Recognition*, éd. par Bichsel (M.), pp. 35–40. – Zurich, Switzerland, 26–28 juin 1995.

[Fabiani et al.96] Fabiani (Lionel), Richard (Paul), Gomez (Daniel), Coiffet (Philippe) et Burdea (Grigore). – Performance humaine lors d'interactions avec des objets virtuels : évaluation du Rutgers Master II. *Actes d'Interface des Mondes Réels et Virtuels*, pp. 105–113. – Montpellier, France, 21–24 mai 1996.

[Farkas94] Farkas (Leslie G.). – *Anthropometry of the Head and Face.* – RAVEN PRESS, 1994, 2nd édition.

[Flanagan et al.97] Flanagan (James) et Marsic (Ivan). – Issues in measuring the benefits of multimodal interfaces. *Proc. of ICASSP'97, Intl. Conf. on Acoustic, Speech, and Signal Processing.* IEEE, pp. 163–166. – Munich, Germany, 21–24 avril 1997.

[Foley87] Foley (James). – Les communications entre l'Homme et l'ordinateur. *Pour la Science*, décembre 1987, pp. 74–82.

[Förstner94] Förstner (Wolfgang). – Diagnostics and Performance Evaluation in Computer Vision. *Proc. of NSF/ARPA Workshop on Performance versus Methodology in Computer Vision*, pp. 11–25. – Seattle, juillet 1994.

[Frey et al.90] Frey (Lisa A.), K. Preston White (JR.) et Hutchinson (Thomas E.). – Eye-Gaze Word Processing. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, n° 4, juillet/août 1990, pp. 944–950.

[Gavrila et al.96] Gavrila (Dariu M.) et Davis (Larry S.). – 3-D Model-based Tracking of Humans in Action: a Multi-view proach. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*. – San Francisco, USA, juin 1996.

[Gee et al.95] Gee (Andrew H.) et Cipolla (Roberto). – *Fast visual tracking by temporal consensus*. – Rapport technique n° CUED/F:INFENG/TR207, University of Cambridge, Department of Engineering, février 1995.

[Gibet et al.96] Gibet (Sylvie), Braffort (Annelies), Collet (Christophe), Forest (Françoise), Gherbi (Rachid) et Lebourque (Thierry). – Gesture in Human-Machine Communication : capture, analysis-synthesis, recognition, semantics. *Proc. of Gesture Workshop'96, Progress in Gestural Interaction*, éd. par Harling (P.A.) et Edwards (A.D.N.). University of York, pp. 89–95. – York, UK, 1996.

[Gips et al.93] Gips (James), Olivieri (Peter) et Tecce (Joseph). – Direct Control of the Computer through Electrodes Placed Around the Eyes. *Proc of HCI'93, 5th Intl. Conf. on Human-Computer Interaction*. pp. 630–635. – Orlando, Florida, 1993.

[Giraudon87] Giraudon (Gérard). – An efficient edge following algorithm. *Proc. of 5th Scandinavian Conference on Image Analysis*, pp. 547–554. – Stockholm, 1987.

[Glenstrup et al.95] Glenstrup (Arne John) et Engell-Nielsen (Theo). – *Eye Controlled Media : Present and Future State*. – Universitetsparken 1, DK-2100 Denmark, Bachelor thesis, Laboratory of Psychology, University of Copenhagen, juin 1995.

[Godaux et al.89] Godaux (Emile) et Chéron (Guy). – Les mouvements oculaires. *Le mouvement*, chap. 9, pp. 229–264. – Paris, Medsi/McGraw-Hill, 1989.

[Graf et al.95] Graf (Hans Peter), Chen (Tsuhan), Petajan (Eric) et Cosatto (Eric). – Locating Faces and Facial Parts. *Proc. of Intl. Work. on Automatic Face & Gesture Recognition*, éd. par Bichsel (M.), pp. 41–46. – Zurich, Switzerland, 26–28 juin 1995.

[Hallett86] Hallett (Peter E.). – Eye Movements. *Handbook of Perception and Human Performance, Vol I : Sensory processes and perception*, éd. par Boff (Kenneth), Kaufman (Lloyd) et Thomas (James), chap. 10, pp. 10.1–10.112. – New York, Wiley-Interscience Publication, 1986.

[Herpers et al.96] Herpers (R.), Michaelis (M.), Lichtenauer (K.-H.) et Sommer (G.). – Edge and keypoint Detection in Facial Regions. In CAFGR [CAFGR96], pp. 212–213.

[HFT] HFourward Technologies, 1939 Friendship Drive, Ste. E, El Cajon, CA 92020 USA. – *DPI EyeTracker*®.

[Horaud et al.95a] Horaud (Radu) et Monga (Olivier). – Chaînage de contours. *Vision par ordinateur, outils fondamentaux* [Horaud et al.95c], chap. 2.12.1, pp. 99–100.

[Horaud et al.95b] Horaud (Radu) et Monga (Olivier). – Récursivité et implantation des filtres. *Vision par ordinateur, outils fondamentaux* [Horaud et al.95c], chap. 2.4.4, pp. 52–57.

[Horaud et al.95c] Horaud (Radu) et Monga (Olivier). – *Vision par ordinateur, outils fondamentaux*. – HERMES, 1995, 2^{nd} édition.

[Horprasert et al.96] Horprasert (Thanarat), Yacoob (Yaser) et Davis (Larry S.). – Computing 3-D Head Orientation from a Monocular Image Sequence. In CAFGR [CAFGR96], pp. 242–247.

[Hubel94] Hubel (David). – *L'Œil, le Cerveau et la Vision, les étapes cérébrales du traitement visuel*. – Pour la science, Diffusion Belin, 1994.

[Hunke et al.94] Hunke (H. Martin) et Waibel (Alex). – Face Location and Tracking for Human-Computer Interaction. *Proc. of 27^{th} Asilomar Conf. on Signals, Systems & Computers*. – Monterey, California, novembre 1994.

[Hutchinson et al.89] Hutchinson (Thomas E.), K. Preston White (JR.), Martin (Worthy N.), Reichert (Kelly C.) et Frey (Lisa A.). – Human-Computer Interaction Using Eye-Gaze Input. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, n° 6, novembre/décembre 1989, pp. 1527–1534.

[HVS Image] HVS Image, Ormond Crescent, Hampton TW12 2TH, U.K. – *SP140 Photobeam Eye Tracker*®.

[Iida et al.89] Iida (M.), Tomono (A.) et Kobayashi (Y.). – A study of Human Iteraction using an Eye-Movement Detection System. *Work with Computers : Organizational, Management, Stress and Health Aspects*, éd. par Smith (Michael) et Salvendy (Gavriel), pp. 666–673. – Amsterdam, North-Holland, Elsevier Science Publishers B.V., 1989.

[Ishii et al.94] Ishii (Hiroshi), Kobayashi (Minoru) et Arita (Kazuho). – Iterative Design of Seamless Collaboration Media. *Communication of the ACM*, vol. 37, n° 8, août 1994, pp. 83–97.

[Istance et al.94] Istance (Howell) et Howarth (Peter). – Keeping an eye on your interface : The potential for eye-based control of graphical user interfaces (GUI's). *Proc. of HCI'94*. – Cambridge University Pres, août 1994.

[Istance et al.95] Istance (Howell), Lindqvist (Martin) et Howarth (Peter). – The feasibility of eye-based interaction with objects displayed in the viewing volume created by a stereoscopic display. *Proc. of HCI'95*. – 1995.

[Istance et al.96] Istance (Howell), Spinner (Christian) et Howarth (Peter). – Providing motor-impaired users with access to standard Graphical User Interface (gui) software via eye-based interaction. *Proc. of 1st European Conf. on Disability, Virtual Reality and Associated Technologies*, éd. par Sharkey (Paul M.). pp. 109–116. – Maidenhead, UK, 8–10 juillet 1996.

[Jacob95] Jacob (Robert J.K.). – Eye Tracking in Advanced Interface Design. *Virtual Environments and Advanced Interface Design*, éd. par Barfield (W.) et Furness (T.A.), pp. 258–288. – New York, USA, Oxford University Press, 3rd édition, 1995.

[Jeannerod88] Jeannerod (Marc). – The role of visual feedback in movement control. *The Neural and Behavioural Organization of Goal-Directed Movements*, éd. par Broadbent (D. E.), McGaugh (J. L.), Mackintosh (N. J.), Posner (M. I.), Tulving (E.) et Weiskrantz (L.), chap. 9, pp. 84–131. – Clarendon Press, Oxford, 1988.

[Kaczmarek93] Kaczmarek (Richard). – Quand il ne reste que le regard. *La Recherche*, vol. 24, n° 252, mars 1993, p. 254.

[Kendon94] Kendon (Adam). – Do Gestures Communicate? a review. *Research on Language and Social Interaction*, vol. 27, n° 3, 1994, pp. 175–200.

[Khan et al.95] Khan (Arup S.), Korde (Rohan) et Muehlen (Ralf). – Use of Pupil Diameter Change for Enhancing Key Selection in Eye-Gaze Systems. *Proc. of Conf. on Intelligent Human Computer Interfaces.* – University of Delaware, 15 mai 1995.

[Krus et al.97] Krus (Mike), Bourdot (Patrick), Guisnel (Françoise) et Thibault (Guillaume). – Levels of Detail & Polygonal Simplification. *ACM's Crossroads*, vol. 3, n° 4, Summer 1997.

[Lee et al.88] Lee (Kai-Fu) et Hon (Hsiao-Wuen). – Large-Vocabulary Speaker-Independent Continuous Speech Recognition using HMM. *Proc. of IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing'88*, pp. 123–126. – 1988.

[Leroy et al.96] Leroy (Bertrand), Chouakria (Ahlame), Herlin (Isabelle L.) et Diday (Edwin). – Approche géométrique et classification pour la reconnaissance de visage. *Actes du 10^{e} Congrès de Reconnaissance des Formes et d'Intelligence Artificielle, RFIA'96.* – Rennes, 16–18 janvier 1996.

[Loftus83] Loftus (Geoffrey R.). – Eye Fixations on Text and Scenes. In Rayner [Rayner83], chap. 21, pp. 359–376.

[Lusted et al.96] Lusted (Hugh) et Knapp (Benjamin). – Des signaux nerveux pour commander les ordinateurs. *Pour la Science*, n° 230, décembre 1996, pp. 78–84.

[Machin96] Machin (David). – Real-Time facial Motion Analysis for Virtual Teleconferencing. In CAFGR [CAFGR96], pp. 340–344.

[Mackay et al.97] Mackay (Wendy E.) et Fayard (Anne-Laure). – HCI, Natural Science and Design : A Framework for Triangulation Across Disciplines. *Proc. of ACM DIS'97, Designing Interactive Systems*, pp. 223–234. – Amsterdam, the Netherlands, 18–20 août 1997.

[McConkie83a] McConkie (George W.). – Eye Movements and Perception during Reading. In Rayner [Rayner83], chap. 5, pp. 65–96.

[McConkie83b] McConkie (George W.). – The Perceptual Span and Eye Movement Control during Reading. In Rayner [Rayner83], chap. 6, pp. 97–120.

[McKenna et al.96] McKenna (Stephen) et Gong (Shaogang). – Tracking Faces. In CAFGR [CAFGR96], pp. 271–276.

[Metrovision] Metrovision, Parc du Technopole, 2 rue Archimède, 59650 Villeneuve d'Ascq, France. – *Visioboard*®.

[Moghaddam et al.95] Moghaddam (Baback) et Pentland (Alex). – Maximum Likelihood Detection of Faces and Hands. *Proc. of Intl. Work. on Automatic Face & Gesture Recognition*, éd. par Bichsel (M.), pp. 122–128. – Zurich, Switzerland, 26–28 juin 1995.

[Moody83] Moody (Bill). – *La langue des signes. Histoire et grammaire.* – Ellipses, Paris, 1983. volume 1.

[Morrison83] Morrison (Robert E.). – Retinal Image Size and the Perceptual Span in Reading. In Rayner [Rayner83], chap. 2, pp. 31–40.

[MS90] Morrel-Samuels (Palmer). – Clarifying the distinction between lexical and gestural commands. *Intl. Journal of Man-Machine Studies*, vol. 32, 1990, pp. 581–590.

[Myers98] Myers (Brad A.). – A brief history of human-computer interaction technology. *Interactions, new vision of human-machine interaction*, vol. 2, mars – avril 1998, pp. 44–54.

[NAC] NAC Europe, Wellington House, Thame, Oxon. OX9 3BU, England. – *EMR-7*®.

[Nielsen90] Nielsen (Jakob). – Trip report : CHI'90. *SIGCHI Bulletin*, vol. 22, n° 2, octobre 1990, pp. 20–25.

[Nielsen93] Nielsen (Jakob). – Noncommand User Interfaces. *Communication of the ACM*, vol. 36, n° 4, avril 1993, pp. 83–99.

[Oliver et al.96] Oliver (Nuria), Pentland (Sandy), Bérard (François) et Coutaz (Joëlle). – *LAFTER : Lips and Face Tracker.* – Research Result n° 396, Cambridge, MA 02139, USA, M.I.T. Media Laboratory, 1996.

[O'Regan90] O'Regan (J. Kevin). – Eye movements and reading. *Eye movements and their role in visual and cognitive processes*, éd. par Kowler (E.), chap. 9, pp. 395–453. – Elsevier Science Publishers B.V., 1990.

[Pappu et al.98] Pappu (Ravikanth) et Plesniak (Wendy). – Haptic interaction with holographic video images. *Proc. of IS&T/SPIE's Symposium on Electronic Imaging, Practical Holography XII.* – janvier 1998.

[Permobil inc.] Permobil inc., 6B Gill Street, Woburn MA, 01801, USA. – *ober2*®.

[Petajan et al.96] Petajan (Eric) et Graf (Hans Peter). – Robust Face Feature Analysis for Automatic Speechreading and character animation. In CAFGR [CAFGR96], pp. 357–362.

[Ponsoda et al.95] Ponsoda (V.), Scott (D.) et Findlay (J. M.). – A probability vector and transition matrix analysis of eye movements during visual search. *Acta Psychologica*, vol. 88, n° 2, 1995, pp. 167–185.

[Pregibon86] Pregibon (Daryl). – A DIY Guide to Statistical Strategy. *Artificial Intelligence & Statistics*, éd. par Gale (W. A.), chap. 17, pp. 389–399. – Addison Wesley, 1986.

[Rayner83] Rayner (Keith) (édité par). – *Eye Movements in Reading : perceptual and language processes.* – ACADEMIC PRESS, New York, 1983, *PERSPECTIVES IN NEUROLINGUISTICS, NEUROPSYCHOLOGY AND PSYCHOLINGUISTICS.*

[Reinders et al.96] Reinders (M. J. T.), Koch (R. W. C.) et Gerbrands (J. J.). – Locating Facial Features in Image Sequences using Neural Networks. In CAFGR [CAFGR96], pp. 230–235.

[Reuchlin86] Reuchlin (Maurice). – Psychophysiologie de la Perception Visuelle. *Psychologie*, part 2, pp. 46–63. – Paris, Presses Universitaires de France, 6^e édition, août 1986.

[Ridder et al.95] Ridder (Christof), Munkelt (Olaf) et Kirchner (Harald). – Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. *Proc. of Intl. Conf. on Recent Advances in Mechatronics, ICRAM'95*, éd. par Kaynak (Okyay) et Özkan (Memed). UNESCO Chair on Mechatronics, pp. 193–199. – Bogazici University, 80815 Bebek, Istanbul, TURKEY, 14-16 août 1995.

[Rubine91] Rubine (Dean). – *The automatic recognition of gestures.* – Pittsburgh, PA 15213, USA, Phd thesis, Carnegie Mellon University, 1991.

[Samaria93] Samaria (Ferdinando). – Face Segmentation for Identification using Hidden Markov Models. *Proc. of 4th British Machine Vision Conference.* pp. 399–408. – Springer–Verlag, septembre 1993.

[Saulnier et al.95] Saulnier (Agnès), Viaud (Marie-Luce) et Geldreich (David). – Real-Time Facial Analysis and Synthesis Chain. *Proc. of Intl. Work. on Automatic Face & Gesture Recognition*, éd. par Bichsel (M.), pp. 86–91. – Zurich, Switzerland, 26–28 juin 1995.

[Shea92] Shea (Sandra L.). – Eye movements : developmental aspects. In Chekaluk et Llewellyn [Chekaluk et al.92], part 8, pp. 239–306.

[Shneiderman87] Shneiderman (Ben). – *Designing the User Interface : Strategies for Effective Human-Computer Interaction.* – Addison-Wesley Publishing Company, mai 1987. 2nd edition en 1992.

[Sobottka et al.96] Sobottka (Karin) et Pitas (Ioannis). – Segmentation and Tracking of Faces in Color Images. In CAFGR [CAFGR96], pp. 236–241.

[Spiegel81] Spiegel (Murray R.). – *Théorie et Applications de la Statistique.* – Paris, McGraw-Hill, 1981, 10^{e} édition, *Schaum.*

[Starker et al.90] Starker (India) et Bolt (Richard A.). – A Gaz-Responsive Self-Disclosing Display. *Proc. of ACM CHI'90, Human Factors in Computing Systems Conference.* pp. 3–9. – Addison-Wesley/ACM Press, avril 1990.

[Stiefelhagen et al.96] Stiefelhagen (Rainer), Yang (Jie) et Waibel (Alex). – A Model-Based Gaze Tracking System. *Proc. of IEEE Intl. Joint Symposia on Intelligence & Systems - Image, Speech & Natural Language Systems*, pp. 304–310. – Washington DC, USA, 1996.

[Stiefelhagen et al.97] Stiefelhagen (Rainer), Yang (Jie) et Waibel (Alex). – Tracking Eyes and Monitoring Eye Gaze. *Proc. of Workshop on Perceptual User Interfaces.* – Banff, Alberta, Canada, 20-21 octobre 1997.

[Torres et al.97] Torres (O.), Cassell (J.) et Prevost (S.). – Modeling Gaze Behavior as a Function of Discourse Structure. *Proc. of 1st Intl. Workshop on Human-Computer Conversations.* – Bellagio, Italy, 1997.

[Varchmin et al.98] Varchmin (Axel Christian), Rae (Robert) et Ritter (Helge). – Image Based Recognition of Gaze Direction Using Adaptive Methods. *Gesture and Sign Language in Human-Computer Interaction*, éd. par Wachsmuth (Ipke) et Fröhlich (Martin), pp. 245–257. – Springer–Verlag, 1998.

[Vo et al.95] Vo (Minh Tue), Houghton (Ricky), Yang (Jie), Bub (Udo), Meier (Uwe), Waibel (Alex) et Duchnowski (Paul). – Multimodal Learning Interfaces. *Proc. of ARPA Spoken Language Technology Workshop.* – Barton Creeks, janvier 1995.

[Ware et al.87] Ware (Colin) et Mikaelian (Harutune H.). – An Evaluation of an Eye Tracker as Device for Computer Input. *SIGCHI Bulletin ACM, Special Issue 'CHI+GI'87 Human Factors in Computer Systems Conference Proceedings'*, 1987, pp. 183–188.

[Wren et al.96] Wren (Christopher R.), Azarbayejani (Ali), Darrell (Trevor) et Pentland (Alex). – Pfinder: Real-Time Tracking of the Human Body. In CAFGR [CAFGR96], pp. 51–56.

[Wu et al.95] Wu (Haiyan), Chen (Qian) et Yachida (Masahiko). – An Application of Fuzzy Theory : Face Detection. *Proc. of Intl. Work. on Automatic Face & Gesture Recognition*, éd. par Bichsel (M.), pp. 314–319. – Zurich, Switzerland, 26–28 juin 1995.

[Yang et al.98] Yang (Jie), Stiefelhagen (Rainer), Meier (Uwe) et Waibel (Alex). – Visual Tracking for Multimodal Human Computer Interaction. *Proc. of CHI'98*, pp. 140–147. – 1998.

[Yow et al.95] Yow (Kin Choong) et Cipolla (Roberto). – Finding Initial Estimates of Human Face Location. *Proc. of 2nd Asian Conf. on Computer Vision*, pp. 514–518. – Singapore, 1995.

[Yow et al.97] Yow (Kin Choong) et Cipolla (Roberto). – Feature-Based Human Face Detection. *Image and Vision Computing*, vol. 15, n° 9, 1997, pp. 713–735.

[Yow et al.98] Yow (Kin Choong) et Cipolla (Roberto). – Enhancing Human Face Detection using Motion and Active Contours. *Proc. of 3rd Asian Conf. on Computer Vision*, pp. 515–522. – Hong Kong, 1998.