



HAL
open science

Utilisation de l'eye-tracking pour l'interaction mobile dans un environnement réel augmenté

Qinjie Ju

► **To cite this version:**

Qinjie Ju. Utilisation de l'eye-tracking pour l'interaction mobile dans un environnement réel augmenté. Autre. Université de Lyon, 2019. Français. NNT : 2019LYSEC011 . tel-02166965

HAL Id: tel-02166965

<https://theses.hal.science/tel-02166965>

Submitted on 27 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre NNT : 2019LYSEC011

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de l'Ecole Centrale de Lyon
Spécialité : Informatique

Utilisation de l'Eye-tracking pour l'Interaction
Mobile dans un Environnement Réel Augmenté

dans le cadre de l'École Doctorale InfoMaths
présentée et soutenue publiquement par

Qinjie JU

9 Avril 2019

JURY

Christophe Kolski	Université Polytechnique Hauts-de-France	Rapporteur
Jean Vanderdonckt	Université Catholique de Louvain	Rapporteur
Patrick Girard	Université de Poitiers	Examineur
Francis Jambon	Université Grenoble Alpes	Examineur
Christine Michel	INSA de Lyon	Examineur
Stéphane Derrode	École Centrale de Lyon	Directeur de thèse
René Chalon	École Centrale de Lyon	Co-directeur de thèse

Table des matières

Résumé	v
Abstract	vii
1 Introduction	1
1.1 L'œil et ses Mouvements	1
1.2 Définition et Histoire de l'Eye-tracking	3
1.3 Eye-tracker	5
1.4 Domaine d'Application de l'Eye-tracking	7
1.5 Eye-tracking comme Modalité d'Entrée	9
1.6 Problématique et Approche	11
1.7 Objectifs	12
1.8 Plan de la Thèse	12
2 Etat de l'Art	15
2.1 Application de l'Eye-tracking en IHM	15
2.2 Interaction Mobile et Ubiquitaire par le Regard	19
2.2.1 Utilisation de la Caméra Frontale	20
2.2.2 Applications à l'Extérieur	21
2.2.3 Interprétation d'Informations	24
2.3 Problèmes liés à l'Utilisation de l'Eye-tracking pour les IHM	26
2.4 Solutions Envisagées	28
2.4.1 Mouvements Volontaires des Yeux	29
2.4.2 Poursuite Lisse	32
2.4.3 Couplage avec un autre dispositif	37
2.4.3.1 Couplage avec un dispositif traditionnel	37
2.4.3.2 Couplage avec les gestes 3D	39
2.4.4 Commandes Implicites	44
2.5 Conclusion du Chapitre	47

3	Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe	49
3.1	Etat de l'Art du Mouvement de la Tête à Regard Fixe	49
3.2	Le Choix du Mouvement de la Tête à Regard Fixe pour Déclencher les Commandes	51
3.3	Mouvements Volontaires de la Tête à Regard Fixe Choisis	54
3.4	Techniques de Reconnaissance des Mouvements de la Tête à Regard Fixe	55
3.4.1	Reconnaissance des Mouvements Verticaux et Horizontaux	56
3.4.2	Reconnaissance des Dodelinements de la Tête	61
3.4.3	Algorithme Global	63
3.5	Conclusion du Chapitre	63
4	EyeMusic : Apprentissage Interactif de la Musique	65
4.1	Choix du Scénario	65
4.2	Réalisation	67
4.2.1	Reconnaissance de la Partition et des Notes	67
4.2.2	Calcul de la Position Relative du Regard par rapport à la Partition	68
4.2.3	Détection des Mouvements de la Tête à Regard Fixe	69
4.3	Prototype Expérimental	70
4.4	Test d'Utilisabilité	71
4.4.1	Configuration Expérimentale	72
4.4.2	Procédure	73
4.4.2.1	Ouverture	73
4.4.2.2	Phase d'apprentissage	73
4.4.2.3	Phase Opérationnelle	74
4.4.2.4	Clôture	75
4.4.3	Informations Collectées	75
4.4.4	Participants	77
4.5	Résultats Expérimentaux et Discussion	78

Table des matières

4.5.1	Performance de l'Application	78
4.5.1.1	Résultats Bruts	78
4.5.1.2	Résultats Ajustés	79
4.5.1.3	Analyse des Mauvaises Détections	81
4.5.2	Performance des Algorithmes de Reconnaissance de Mouvement de la Tête	82
4.5.3	Performance des Participants avec Différents Niveaux de Connaissance de la Notation Musicale	84
4.5.4	Capacité à Mémoriser Rapidement la Correspondance entre les Commandes et les Mouvements, et à Déclencher 6 Commandes Différentes par ces Mouvements	86
4.5.5	Erreurs des Algorithmes de Reconnaissance des Mouvements de la Tête et Solutions Possibles	88
4.5.6	Préférence de Mouvement	89
4.5.7	Résultat du SUS (<i>System Usability Scale</i>)	90
4.5.8	Utilité des Commandes	93
4.5.9	Supposition sur l'Utilité de EyeMusic	94
4.5.10	Implications sur la Conception du Prototype	95
4.6	Conclusion du Chapitre	97
5	EyeRecipe et EyePay	99
5.1	EyeRecipe	99
5.1.1	Choix du Scénario	99
5.1.1.1	Consultation d'une Recette	100
5.1.1.2	Prise des Images et de la Vidéo	101
5.1.1.3	Contrôle des Équipements Intelligents	102
5.1.2	Réalisation	102
5.1.2.1	Choix de la Correspondance entre les Mouvements et les Commandes	102
5.1.2.2	Calcul de la Position Relative du Regard par rapport à l'Écran	104

5.1.2.3	Reconnaissance des Informations Regardées	104
5.1.2.4	Reconnaissance de l’Objet à Contrôler	106
5.1.2.5	Détection des Mouvements Verticaux et Horizon- taux de la Tête à Regard Fixe	107
5.1.2.6	Détection des Inclinaisons de la Tête à Regard Fixe	107
5.1.3	Algorithme Global d’EyeRecipe	108
5.1.4	Conclusion de EyeRecipe	110
5.2	EyePay	110
5.2.1	Paiement Rapide sans Besoin de Libérer les Mains	111
5.2.2	Entrée de Code Privé par le regard	111
5.2.3	Scanlib Version Eye-tracker	112
5.2.4	Affectation des Commandes	113
5.2.5	Conclusion de EyePay	113
5.3	Conclusion du Chapitre	113
6	Conclusion et perspectives	115
6.1	Travaux Réalisés	115
6.2	Contributions	117
6.3	Perspectives	118
A	Annexe Relative à l’Evaluation de EyeMusic	121
A.1	Présentation Générale d’EyeMusic	121
A.2	Formulaire de Consentement	122
A.3	Post-test Questionnaire	124
A.4	SUS : System Usability Scale	127
A.5	Fiche de Recueil pour l’Evaluation du Prototype EyeMusic	128
A.6	Participation à l’Evaluation du Prototype EyeMusic	130
	Bibliographie	139

Résumé

Les dispositifs d'eye-tracking ont un très fort potentiel en tant que modalité d'entrée en IHM (Interaction Homme-Machine), en particulier en situation de mobilité. Dans cette thèse, nous nous concentrons sur la mise en œuvre de cette potentialité en mettant en évidence les scénarios dans lesquels l'eye-tracking possède des avantages évidents par rapport à toutes les autres modalités d'interaction. Au cours de nos recherches, nous avons constaté que cette technologie ne dispose pas de méthodes pratiques pour le déclenchement de commandes, ce qui réduit l'usage de tels dispositifs. Dans ce contexte, nous étudions la combinaison d'un eye-tracker et des mouvements volontaires de la tête lorsque le regard est fixe, ce qui permet de déclencher des commandes diverses sans utiliser les mains ni changer la direction du regard. Nous avons ainsi proposé un nouvel algorithme pour la détection des mouvements volontaires de la tête à regard fixe en utilisant uniquement les images capturées par la caméra de scène qui équipe les eye-trackers portés sur la tête, afin de réduire le temps de calcul. Afin de tester la performance de notre algorithme de détection des mouvements de la tête à regard fixe, et l'acceptation par l'utilisateur du déclenchement des commandes par ces mouvements lorsque ses deux mains sont occupées par une autre activité, nous avons effectué des expériences systématiques grâce à l'application EyeMusic que nous avons conçue et développée. Cette application EyeMusic est un système pour l'apprentissage de la musique, capable de jouer les notes d'une mesure d'une partition que l'utilisateur ne comprend pas. En effectuant un mouvement volontaire de la tête pendant qu'il fixe son regard une mesure particulière d'une partition, l'utilisateur obtient un retour audio. La conception, le développement et les tests d'utilisabilité du premier prototype de cette application sont présentés dans cette thèse. L'utilisabilité de notre application EyeMusic est confirmée par les résultats expérimentaux : 85% des participants ont été en mesure d'utiliser tous les mouvements volontaires de la tête à regard fixe que nous avons

implémentés dans le prototype. Le taux de réussite moyen de cette application est de 70%, ce qui est partiellement influencé par la performance intrinsèque de l'eye-tracker que nous utilisons. La performance de notre algorithme de détection des mouvements de la tête à regard fixe est 85%, et il n'y a pas de différence significative entre la performance de chaque mouvement de la tête testé. Nous avons également exploré deux scénarios d'applications qui reposent sur les mêmes principes de commande, EyeRecipe et EyePay, dont les détails sont également présentés dans cette thèse.

Abstract

Eye-tracking has a very strong potential in human-computer interaction (HCI) as an input modality, particularly in mobile situations. In this thesis, we concentrate in demonstrating this potential by highlighting the scenarios in which the eye-tracking possesses obvious advantages comparing with all the other interaction modalities. During our research, we find that this technology lacks convenient action triggering methods, which can scale down the performance of interacting by gaze. In this instance, we investigate the combination of eye-tracking and fixed-gaze head movement, which allows us to trigger various commands without using our hands or changing gaze direction. We have proposed a new algorithm for fixed-gaze head movement detection using only scene images captured by the scene camera equipped in front of the head-mounted eye-tracker, for the purpose of saving computation time. To test the performance of our fixed-gaze head movement detection algorithm and the acceptance of triggering commands by these movements when the user's hands are occupied by another task, we have implemented some tests with the EyeMusic application that we have designed and developed. The EyeMusic system is a music reading system, which can play the notes of a measure in a music score that the user does not understand. By making a voluntary head movement when fixing his/her gaze on the same point of a music score, the user can obtain the desired audio feedback. The design, development and usability testing of the first prototype for this application are presented in this thesis. The usability of our EyeMusic application is confirmed by the experimental results, as 85% of participants were able to use all the head movements we implemented in the prototype. The average success rate of this application is 70%, which is partly influenced by the performance of the eye-tracker we use. The performance of our fixed-gaze head movement detection algorithm is 85%, and there were no significant differences between the performance of each head movement. Apart from the EyeMusic application, we have explored

two other scenarios that are based on the same control principles : EyeRecipe and EyePay. The details of these two applications are also presented in this thesis.

Introduction

1.1 L'œil et ses Mouvements

L'œil humain est un organe complexe commandé par 3 paires de muscles donnant à l'œil 3 degrés de liberté : une paire est responsable des mouvements horizontaux, une paire contrôle des mouvements verticaux, et la troisième paire permet des mouvements de rotation autour de la direction du regard.

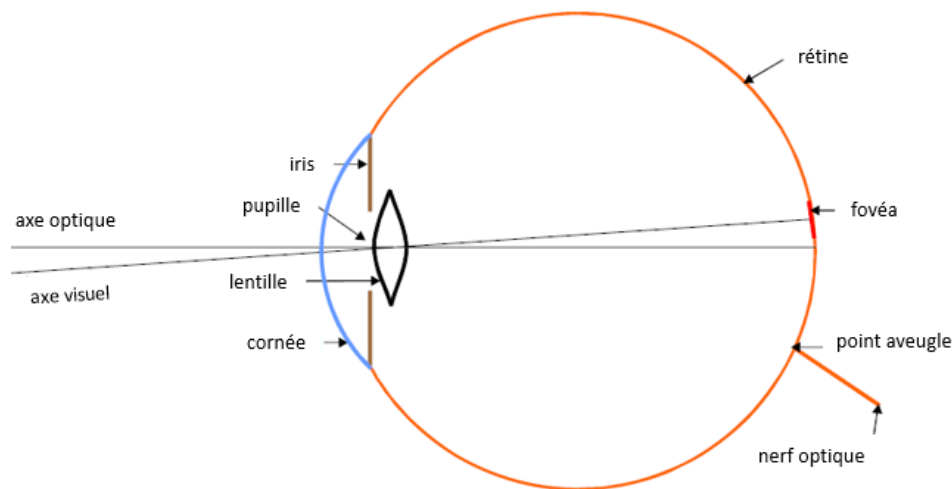


FIGURE 1.1 : Le schéma simplifié de l'œil.

La construction de l'œil est similaire à celle d'une caméra. La Figure 1.1 montre un schéma simplifié de l'œil. L'iris sert à régler l'ouverture de l'œil, le trou dans l'iris s'appelle pupille. La lentille de l'œil peut changer la distance focale en modifiant sa forme. La zone sensible à la lumière s'appelle rétine, elle se situe sur la surface interne arrière de l'œil. La fovéa est une petite surface d'environ 1 degré de taille angulaire au centre de la rétine. C'est la zone de la rétine qui a une bonne

résolution et qui correspond au champ visuel 'principal'. Le champ visuel extérieur de la fovéa sert à la perception du mouvement ambiant car sa résolution est faible [25]. Chaque fois que nous voulons voir clairement quelque chose, nous bougeons l'œil pour obtenir la projection de cet objet sur la fovéa. Si la taille angulaire de cet objet est inférieure à 1 degré, il suffit que la projection se trouve quelque part sur la fovéa, pas nécessairement au centre. Par conséquent, l'œil n'a pas besoin de positionner son regard avec plus de précision. De plus, les micromouvements oculaires (d'une magnitude de 0,1 degré) contribuent à l'inexactitude du pointage de l'œil. En conséquence, tous ces comportements de l'œil (principalement la taille de la fovéa) limite sa précision à environ $\pm 0,5$ degré [20].

L'œil est capable de faire plusieurs mouvements différents à l'aide de ses 6 muscles, ces mouvements sont étudiés en détail par les psychologues et les neurologistes. Globalement, il y a 4 types de mouvement : la fixation, la saccade, la poursuite lisse (*smooth pursuit*) et la réflexe vestibulo-oculaire (VOR). Lors de l'observation de la scène, le mouvement des yeux consiste généralement en une série de fixations et de saccades.

La **fixation** représente l'état dans lequel l'œil est relativement immobile et fixé sur un objet d'intérêt. C'est le temps pendant lequel le cerveau fait la reconnaissance des objets et l'acquisition des informations visuelles. Normalement la durée de celles-ci se situe entre 100ms et 600ms [61]. Comme l'œil bouge constamment, la vraie fixation n'existe pas, les mouvements involontaires qui se produisent pendant les fixations sont les microsaccades, les tremblements (*tremors*) et les déviations (*drifts*) [11, 27]. Ces micromouvements mesurent moins d'un degré [118] et servent à recentrer une image sur la fovéa et à garantir que de nouvelles informations sont transmises constamment sur la rétine.

Les fixations sont séparées par les **saccades**, qui sont des sauts rapides entre les points de fixation. La saccade est un type de mouvement des yeux qui a lieu très fréquemment et très rapidement, il dure de 30 à 120 millisecondes pour une vitesse moyenne de 400 à 600 degrés par seconde [41]. La durée de la saccade, la vitesse moyenne et la vitesse maximale augmentent toutes lorsque la distance angulaire de la saccade augmente [4]. C'est-à-dire, plus l'amplitude de la saccade est élevée,

Chapitre 1. Introduction

plus la saccade est longue mais également rapide [25, 117]. Les saccades de grandes amplitudes peuvent atteindre une vitesse maximale de 700 degrés par seconde [12], les petites saccades de 5-10 degrés ne prennent qu'entre 30 et 40 ms, les grandes saccades de plus de 30 degrés durent également moins de 100ms [89]. Une fois qu'une saccade a été initiée, elle ne peut pas être interrompue, sa direction ne peut pas changer non plus [48].

La **poursuite lisse** est un type de mouvement des yeux moins fréquent que la saccade et la fixation, elle peut être décrit comme une fixation en mouvement et se produit lorsque les yeux de la personne suivent une cible qui se déplace avec une vitesse dans un cadre spécifique (100-200 degrés par seconde) [117]. Les poursuites lisses nécessitent une cible en mouvement et une combinaison des saccades et des fixations : la fixation stabilise l'objet sur la rétine afin de récupérer des informations, les saccades effectuent des corrections mineures [85]. Sans l'objet en mouvement à regarder, les yeux ne peuvent pas effectuer la poursuite lisse. Les poursuites lisses peuvent être décomposées en deux étapes : tout d'abord la poursuite en boucle ouverte qui correspond au mouvement initial vers la cible, cela ressemble à une saccade et dure normalement moins de 100 ms ; ensuite la poursuite en boucle fermée, pendant laquelle le système visuel tente en permanence de s'ajuster à la vitesse de la cible en mouvement [59, 73].

La **réflexe vestibulo-oculaire** n'a lieu que pendant les mouvements de la tête lorsque le regard de la personne est fixé sur un objet immobile. Pendant ce processus, les 3 paires de muscles de l'œil compensent le mouvement de la tête en faisant un mouvement de l'œil dans le sens inverse avec la même vitesse afin de stabiliser le regard. La réflexe vestibulo-oculaire se passe avec une latence très courte (de 7 à 15 ms), en conséquence, il peut être considéré comme simultané au mouvement de la tête [89].

1.2 Définition et Histoire de l'Eye-tracking

L'eye-tracking (suivi du regard) est un ensemble de technologies qui permet d'enregistrer les mouvements des yeux et ainsi fournir une estimation de la direction

du regard. Cela signifie non seulement l'estimation de la direction du regard par rapport à la tête, mais aussi l'identification de l'objet sur lequel porte le regard. Pour les interactions avec un ordinateur traditionnel, l'objet regardé est identifié par les coordonnées du regard sur l'écran. L'interprétation de la direction du regard devient plus difficile lors des interactions avec l'environnement physique.

On fait référence à la mémoire de thèse de Drewes et au travail de Jacob et al. pour l'aperçu de l'histoire de l'eye-tracking [20, 49].

Les recherches sur les mouvements des yeux commencent au 18ème siècle. Mise à part l'observation visuelle, les méthodes initiales d'eye-tracking étaient assez invasives et impliquaient un contact mécanique direct avec la cornée de l'œil. Dodge et Cline ont développé la première technique d'eye-tracking précise et non invasive [19]. Ils utilisent la réflexion de la lumière sur la cornée pour enregistrer la position horizontale du regard, mais ils imposent que la tête du participant soit immobile. Quelques années plus tard, Judd et al. appliquent la photographie cinématographique dans ce domaine afin d'enregistrer les mouvements des yeux en deux dimensions [53]. L'invention de la photographie cinématographique donnait la possibilité d'analyser trame par trame le mouvement des yeux et permettait des recherches quantitatives sur des bases solides. Durant la première moitié du 20ème siècle, les chercheurs ont fait des avancées sur les systèmes d'eye-tracking en combinant la technique de réflexion de la cornée et la technique de la cinématographie de manières différentes [66].

En 1939, Jung a mesuré simultanément les mouvements verticaux et horizontaux des yeux avec des électrodes appliquées sur la peau proche des yeux [20]. Cette méthode, appelée électrooculographie (EOG), donnait la première possibilité du traitement des données du regard en temps réel.

Le premier eye-tracker porté sur la tête a été inventé par Hartridge et Thompson en 1948 [40]. Cette innovation a permis de libérer les utilisateurs en enlevant les contraintes strictes sur le mouvement de la tête. Le concept d'eye-tracker porté sur la tête a été amélioré dans les années 1960, en diminuant davantage les contraintes sur le mouvement de la tête. Et en 1958 Mackworth et Mackworth ont conçu un système qui enregistre les mouvements des yeux en superposant le regard sur la

scène vue par l'utilisateur [66].

Les techniques d'eye-tracking fleurissent dans les années 1970. Les eye-trackers sont devenus moins intrusifs et plus précis ; il devient possible de dissocier les mouvements des yeux de celui de la tête par des réflexions multiples de l'œil [13]. En plus, l'avènement des mini-ordinateurs fournit les ressources nécessaires au traitement rapide des données. Cette innovation était un prérequis essentiel pour l'utilisation de l'eye-tracking en tant que modalité d'interaction, car presque tous les travaux de l'eye-tracking avant cela utilisaient les données de manière rétrospective plutôt qu'en temps réel.

Dans les années 1980, les mini-ordinateurs sont devenus assez performants pour suivre les mouvements des yeux en temps réel, ce qui marque le début de l'utilisation de l'eye-tracking en temps réel dans les interactions homme-machine [20, 49]. A partir des années 1990, l'utilisation de l'eye-tracking a eu une croissance modeste. Avec des améliorations sur plusieurs points critiques (par exemple la précision, le délai, le poids, etc.) et la baisse du prix, cette technologie est de plus en plus utilisée dans les études de marché et les recherches en interaction homme-machine.

1.3 Eye-tracker

Comme nous l'avons déjà mentionné, l'outil qui sert à nous fournir la direction du regard et à nous indiquer l'objet regardé s'appelle eye-tracker ou oculomètre.

Il existe trois méthodes différentes pour suivre le mouvement des yeux. La méthode la plus directe consiste à fixer un capteur (par exemple une lentille) sur l'œil. Une autre méthode est l'électrooculographie (EOG), pour laquelle les électrodes sont fixées sur la peau autour des yeux afin de mesurer le champ électrique (voir la Figure 1.2). Cette méthode fonctionne bien mais elle est sensible aux interférences électromagnétiques. Ces deux méthodes sont importunes et ne conviennent pas à l'interaction par le regard. La troisième méthode utilisée dans la majorité des eye-trackers actuels est basée sur des caméras, l'estimation de la direction du regard est faite par le traitement des images fournies (la détection de la pupille ou la réflexion de la lumière sur la cornée par exemple).

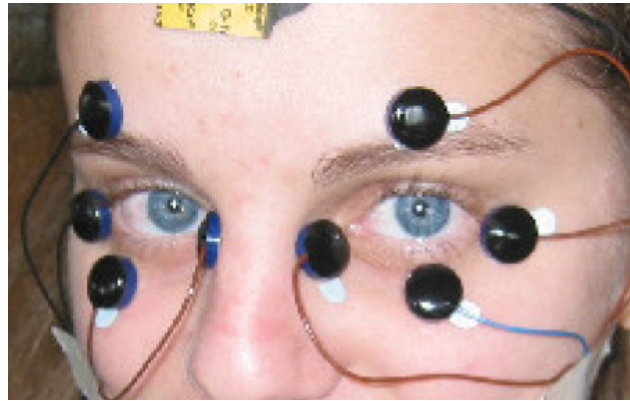


FIGURE 1.2 : Exemple de EOG : les électrodes sont fixées sur la peau autour des yeux.

Avec le développement de la science et de la technologie, les eye-trackers sont devenus de plus en plus légers. Actuellement les 2 types d'eye-tracker les plus utilisés sont les eye-trackers portables sous forme d'une paire de lunettes et les eye-trackers stationnaires (voir la Figure 1.3). L'eye-tracker sous forme d'une paire de lunettes peut être porté sur la tête des utilisateurs, permettant ainsi à l'utilisateur de bouger sa tête comme il le souhaite. L'eye-tracker stationnaire, par contre, est placé à côté des objets avec lesquels les utilisateurs ont envie d'interagir. Cela peut paraître plus confortable que l'eye-tracker porté sur la tête, mais il ne peut suivre que les regards dans la région à proximité de l'eye-tracker.

Le dispositif que nous avons utilisé tout au long de la thèse est le Eye Tracking Glasses 2 Wireless produit par SMI (SensoMotoric Instruments)¹. C'est un eye-tracker porté sur la tête sous la forme d'une paire de lunettes (voir la Figure 1.4). Deux caméras internes filment les mouvements des yeux à un taux d'échantillonnage de 30Hz afin de calculer la direction du regard, tandis qu'une troisième caméra située devant les lunettes capture des images de la scène à une fréquence de 24 trames par seconde pour une résolution de 1280*960. Cet eye-tracker pèse 176 grammes et sa plage de suivi du regard est de 80 degrés horizontalement et 60 degrés verticalement. Le champ de vision de la caméra de la scène est de 60 degrés horizontalement et 46 degrés verticalement. C'est un eye-tracker robuste et avec une grande précision, car il est utilisable sous conditions de luminance différentes et, d'après les données

1. http://www.smivision.com/wp-content/uploads/2017/05/smi_prod_ETG_120Hz_asgm.pdf

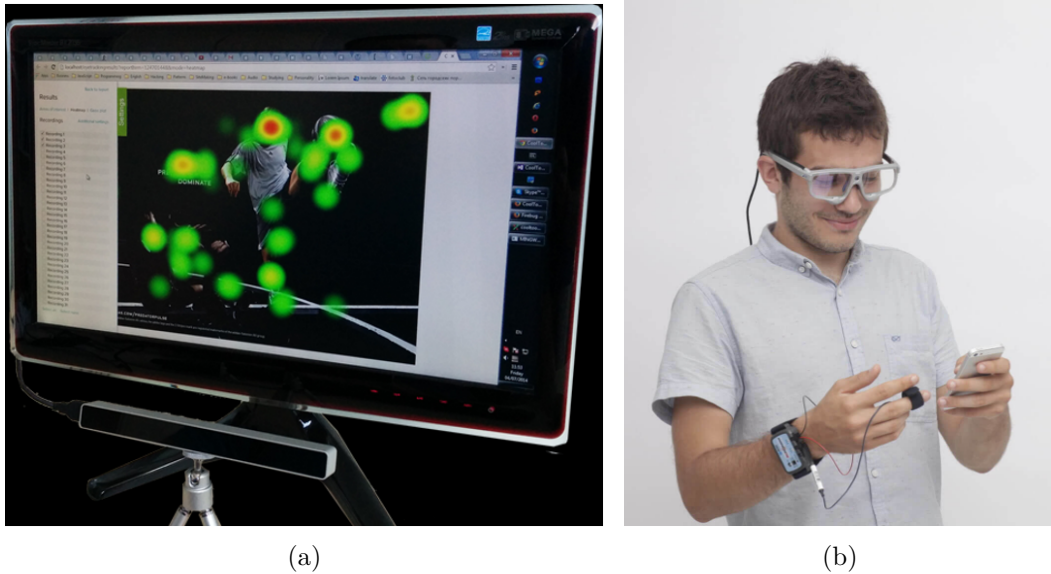


FIGURE 1.3 : Exemples d'eye-trackers : (a) eye-tracker stationnaire placé en-dessous d'un écran, (b) eye-tracker portable sous forme d'une paire de lunettes.

statistiques de SMI, la déviation n'est que de 0,5 degrés pour toutes les distances. De plus, cet eye-tracker nous permet de suivre le regard en mobilité car il peut être connecté à un téléphone portable avec les logiciels spécifiques (le *smart recorder*). Cet outil peut fournir la puissance nécessaire au calcul de la direction du regard en remplaçant un ordinateur ; il est ainsi possible de suivre le regard de l'utilisateur à l'extérieur, en branchant l'eye-tracker avec le smart recorder. Par ailleurs, le kit de développement logiciel de cet équipement donne un accès direct à des informations précises du regard (les images des yeux et les coordonnées du regard sur l'image de la scène) et des images de la scène avec le point du regard superposé.

1.4 Domaine d'Application de l'Eye-tracking

La technologie de l'eye-tracking peut être utilisée dans plusieurs domaines, les domaines d'application typiques sont :

- Etude du Design

Un domaine d'application de l'eye-tracking concerne l'étude du design. En analysant la direction du regard des personnes lorsqu'elles regardent un produit ou une publicité, il est possible de comprendre quelles sont les infor-



FIGURE 1.4 : L'Eye Tracking Glasses 2 Wireless de SMI. Les deux caméras des yeux filment les mouvements des yeux tandis que la caméra de la scène capture l'image regardée.

mations ou bien les couleurs qui les attirent le plus. En travaillant sur la trajectoire du regard des personnes lorsqu'ils utilisent un site Internet ou bien un système en fonctionnement (*operating system*), il est possible de trouver les défauts de design et ainsi l'améliorer.

— Recherche dans le Domaine de la Psychologie et le Domaine Médical

Un autre domaine d'application pour l'eye-tracking est la recherche psychologique ou médicale. Avec la trajectoire du mouvement des yeux fournie par l'eye-tracker, les médecins arrivent à mieux comprendre les symptômes des patients avec des blessures aux yeux, les psychologues sont capables de mieux comprendre le processus d'une certaine action, comme par exemple le processus de lecture d'un article.

— Interaction par le Regard

L'eye-tracking peut également servir à faciliter la vie des personnes à mobilité réduite. Pour les gens qui ont perdu le contrôle des muscles, le regard devient une alternative pour interagir avec le monde. Plusieurs systèmes sont déjà créés pour leur permettre de dactylographier une phrase, de diriger un fauteuil roulant, etc. Ce n'est cependant pas le seul contexte d'utilisa-

tion de l’eye-tracking dans les interactions homme-machine. Ce dispositif est également capable d’assister tout le monde en tant que modalité d’entrée lorsqu’on a envie d’interagir avec un objet intelligent. Comme par exemple, une télévision qui s’allume automatiquement lorsque l’utilisateur la regarde. Ce domaine d’application de l’eye-tracking n’est pas encore commercialisé et nécessite des recherches supplémentaires.

1.5 Eye-tracking comme Modalité d’Entrée

Le regard est un moyen prometteur pour les interactions homme-machine qui peut s’avérer plus pratique que ceux que nous utilisons actuellement. Les avantages potentiels d’interaction par le regard sont les suivants [20] :

— Facilité d’utilisation

Un avantage d’utilisation de l’eye-tracking est la réduction du stress des muscles des mains et des bras parce que l’entrée des données est réalisée par les yeux au lieu des mains. Cela ne demande pas nécessairement une charge supplémentaire des muscles des yeux car, la plupart du temps, les yeux bougent quand même lorsqu’on réalise des interactions à l’aide des modalités d’entrée traditionnelles.

— Rapidité

L’eye-tracking est capable d’accélérer l’interaction car les yeux sont plus rapides que les autres parties du corps et la grande majorité des utilisateurs ont l’habitude (bien souvent inconsciente) de regarder l’objet avec lequel ils souhaitent interagir.

— Contrôle à distance

Un autre avantage de l’eye-tracking est la possibilité de contrôle à distance. Les caméras à haute résolution permettent la détection du regard et la reconnaissance d’objets à plusieurs mètres de distance, ce qui est beaucoup long que la longueur du bras.

— Libération des deux mains

L’eye-tracking permet de réaliser des interactions sans besoin de manipuler

un dispositif avec la main. Il est donc utilisable lorsque les deux mains de l'utilisateur ne sont pas disponibles (elles sont soit sales, soit prises par une autre activité).

— Sécurité

A l'aide de l'eye-tracking, il est possible de garantir la sécurité des informations privées des utilisateurs lorsqu'on interagit en public. Par exemple, les personnes autour de soi ne pourront plus voir notre code bancaire si l'entrée du code est faite par le regard. En plus, le regard peut garantir l'attention de l'utilisateur, du coup cela peut éviter les interactions erronées comme appeler quelqu'un accidentellement alors que le téléphone portable est dans la poche.

— Pas besoin d'entretien

L'eye-tracking basé sur la vidéo fonctionne sans contact, ce qui signifie qu'il n'est pas nécessaire de nettoyer l'appareil. En plus, la caméra peut être protégée en la mettant derrière un matériau transparent solide. Ce n'est pas possible pour les modalités d'entrée traditionnelles comme le clavier et la souris.

— Interface hygiénique

L'eye-tracking permet de réaliser des interactions sans rien toucher, rendant le dispositif très utile dans les environnements avec des exigences d'hygiène très élevées, comme par exemple une salle d'opération chirurgicale. Pour les interfaces publiques, une interaction hygiénique est également préférable.

— Détection des informations supplémentaires

Les yeux peuvent fournir beaucoup d'informations liées à l'activité de l'utilisateur. Par exemple, le point du regard indique l'attention de l'utilisateur, la condition physique ou émotionnelle de l'utilisateur peut également être révélée par des analyses sophistiquées sur les données fournies par l'eye-tracker [17]. Ainsi l'eye-tracking est capable de fournir des informations utiles pour les systèmes sensibles au contexte [82].

1.6 Problématique et Approche

Depuis plusieurs années, notre équipe travaille sur les techniques d'interaction adaptées pour des utilisateurs mobiles évoluant dans des environnements réels augmentés au sens de l'informatique pervasive et ubiquitaire. Des travaux antérieurs ont permis de dégager 3 types d'interactions [121] :

- IEI (In-environment interfaces) : un utilisateur nomade utilise des dispositifs d'interaction qui sont présents dans l'environnement ;
- EDI (Environment dependant interfaces) : l'utilisateur porte un dispositif d'interaction qui lui permet d'interagir avec les objets présents dans l'environnement ;
- EII (Environment independant interfaces) : l'utilisateur porte un dispositif d'interaction autonome.

Nos travaux s'inscrivent plus particulièrement dans le cadre des EDI où nous cherchons à étudier les potentialités de l'eye-tracking pour assister les interactions. Les systèmes conçus pour les personnes à mobilité réduite qui permettent aux gens de diriger un ordinateur uniquement par les yeux fonctionnent bien et fournissent une bonne assistance pour les personnes qui en ont besoin. Mais ils sont moins efficaces que les systèmes contrôlés par le clavier et la souris et ils ne sont donc pas convenables pour tout le monde. Ainsi nous nous intéressons à l'application de l'eye-tracking dans les interactions avec les objets de l'environnement réel tout en laissant les deux mains de l'utilisateur libres pour accomplir des gestes liés à son activité principale afin de bien bénéficier des avantages de l'eye-tracking. Notre problématique de recherche principale est donc :

- de mettre en évidence des cas d'usage où l'eye-tracking apporte une réelle plus-value par rapport à d'autres dispositifs d'interaction dans des situations où l'utilisateur est mobile et n'a pas l'usage de ses mains malgré les limitations connues de l'eye-tracking pour l'interaction par le regard,
- d'analyser et de proposer des solutions qui peuvent contribuer à l'amélioration de la performance des systèmes d'interactions par le regard.

Notre démarche se veut principalement expérimentale par le développement de pro-

totypes applicatifs mettant en œuvre ces concepts et leurs évaluations par des études empiriques.

1.7 Objectifs

Cette thèse a été réalisée au sein du laboratoire LIRIS de l'Ecole Centrale de Lyon et se situe dans le domaine de l'IHM (Interaction Homme-Machine). Ce domaine étudie les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine. Pour cette thèse, nous travaillons sur l'utilisation des informations récupérées par un eye-tracker dans les systèmes d'interaction homme-machine, surtout pour des interactions avec l'environnement physique. Contrairement aux systèmes existants pour les personnes à mobilité réduite, dans lesquels l'eye-tracking est la seule modalité d'entrée disponible, notre recherche se concentre sur les scénarios pour le public où l'eye-tracking a des avantages par rapport aux autres modalités d'entrée existantes (par exemple la souris, le clavier, etc.). Dans cette thèse, on ne travaille pas sur le matériel du suivi du regard ni sur les technologies pour la détection de la position du regard. Par contre, on cherche à utiliser directement les informations fournies par les eye-trackers existants. Ainsi, ces travaux sont basés sur la supposition qu'il y aura des équipements d'eye-tracking fiables et peu coûteux dans un futur proche. Le but de cette thèse est d'étudier la potentialité de l'eye-tracking pour interagir avec des objets de l'environnement réel en laissant les mains de l'utilisateur libres pour accomplir des gestes liés à son activité principale.

1.8 Plan de la Thèse

La thèse comporte six chapitres : une introduction, un état de l'art concernant l'application du suivi du regard dans le domaine des IHM, un chapitre sur la solution que nous proposons afin d'améliorer la performance des interactions par le regard, deux chapitres sur les scénarios qu'on a conçus ou réalisés, et finalement une conclusion.

Chapitre 1. Introduction

L'introduction fournit des connaissances élémentaires liées au sujet de thèse, y compris la présentation des mouvements des yeux, la présentation de la technologie eye-tracking et l'équipement qu'on utilise, les domaines d'application et les avantages de l'eye-tracking en tant que modalité d'entrée. On y décrit également la problématique et les objectifs de cette thèse à la fin de ce chapitre.

L'état de l'art commence par des applications de l'eye-tracking dans le domaine de l'IHM proposées par les chercheurs, à partir desquelles il est possible d'esquisser la tendance des scénarios choisis pour ce dispositif. Et puis, on conclut sur les problèmes et les difficultés rencontrés dans ce domaine. Enfin, on présente précisément les solutions envisageables proposées, en prenant en compte le contexte d'utilisation et la performance.

Une comparaison des solutions envisagées présentées dans l'état de l'art est fournie au troisième chapitre. En comparant ces solutions sous différents aspects, nous avons découvert que le couplage des mouvements volontaires de la tête à regard fixe convient mieux aux interactions avec des objets de l'environnement réel, tout en laissant les mains de l'utilisateur libres. Dans ce chapitre, nous présentons également le nouvel algorithme que nous avons proposé au niveau de la détection de ces mouvements de la tête. Dans cet algorithme, seules sont utilisées les images de la scène avec le point du regard superposé fournies par l'eye-tracker porté sur la tête équipée d'une caméra frontale.

Le chapitre 4 décrit le scénario EyeMusic conçu et réalisé durant la thèse. Il présente le choix du scénario, les points essentiels au niveau de la réalisation et le prototype réalisé pour cette application. Le test d'utilisabilité conduit pour évaluer ce prototype (ou plus spécifiquement notre algorithme de détection des mouvements volontaires de la tête à regard fixe) et les résultats expérimentaux sont également présentés dans ce chapitre.

Le chapitre 5 discute deux autres scénarios (le EyeRecipe et le EyePay) conçus afin de montrer davantage le potentiel de l'eye-tracking dans le domaine des IHM en tant que modalité d'entrée. Ces deux scénarios possèdent une cible beaucoup plus vaste et ils sont à usage quotidien. A l'aide de l'eye-tracking et des mouvements volontaires de la tête à regard fixe, les utilisateurs pourront réaliser des interactions

plus pratiques et plus rapides lors d'une consultation d'une recette ou lors d'un paiement.

La conclusion résume les résultats obtenus et discute le potentiel de l'eye-tracking pour l'IHM. Il présente également les perspectives de ce travail de thèse.

Etat de l'Art

Afin de concevoir des solutions capables de résoudre certaines limitations de l'eye-tracking dans le domaine de l'interaction homme-machine et de trouver les scénarios dans lesquels l'eye-tracking peut apporter une réelle plus-value par rapport à d'autres dispositifs, nous avons conduit une vaste étude documentaire sur les précédentes recherches, principalement au niveau des applications possibles, des avantages et des limitations connues de l'eye-tracking, et des solutions proposées pour améliorer la performance des interactions par le regard. Nos découvertes via cette étude sont présentées dans ce chapitre.

2.1 Application de l'Eye-tracking en IHM

Des progrès notables dans le domaine de l'eye-tracking ont été réalisés à la fin du 20ème siècle, avec l'augmentation importante de la puissance de calcul des ordinateurs qui permet un traitement de la vidéo en temps réel : c'est le début de l'utilisation de l'eye-tracking pour les IHM [20].

Un des intérêts majeurs de l'eye-tracking dans le domaine de l'IHM est de faciliter la vie des personnes à mobilité réduite (surtout pour ceux qui n'ont que l'usage de leurs yeux). Cela leur a apporté la possibilité de réaliser les actions qu'ils n'arrivaient pas à faire à l'aide d'un système spécifique piloté uniquement par le mouvement des yeux. Beaucoup de recherches sont destinées à leur permettre de dactylographier une phrase [68, 115]. La plupart des systèmes de saisie oculaire sont réalisés à l'aide d'un clavier sur l'écran, les utilisateurs peuvent taper une phrase en fixant leur regard sur les touches virtuelles, les lettres correspondantes apparaîtront dans le champ de texte au-dessous ou en-dessus du clavier sur l'écran. La dispo-

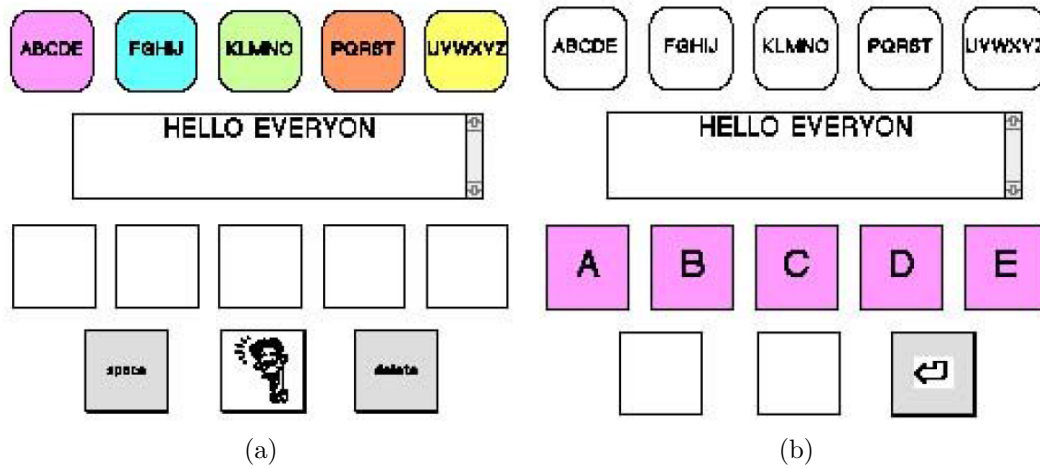


FIGURE 2.1 : Le système EagleEyes : (a) le premier écran des 2 niveaux , (b) l'écran après la sélection du groupe 'ABCDE'.

tion des touches du clavier est aussi étudiée afin d'améliorer la performance de ces systèmes pour les personnes qui s'en servent. La disposition QWERTY ou AZERTY n'est sans doute pas la meilleure disposition pour les personnes à mobilité réduite qui ne sont pas familières avec celle-ci ; dans ce cas là l'ordre alphabétique est plus facile à apprendre. Une autre idée proposée par les chercheurs est d'arranger les lettres selon leur fréquence d'utilisation dans une langue spécifique, comme ce qui est fait par Kate et al. dans l'application de communication eye-switch [103]. La méthode hiérarchique est également étudiée afin de laisser une plus grande surface pour chaque touche, la précision de l'eye-tracker étant limitée. Gips et Olivieri ont proposé dans le système EagleEyes un programme à 2 niveaux : les utilisateurs sélectionnent d'abord un groupe de 5 ou 6 lettres comme dans la Figure 2.1a. Une fois le groupe sélectionné ('ABCDE' par exemple), les lettres de ce groupe apparaissent sur l'écran (comme dans la Figure 2.1b) et les utilisateurs peuvent donc sélectionner la lettre voulue [35]. Les applications de l'eye-tracking pour assister les personnes à mobilité réduite ne sont pas limitées à la saisie d'une phrase. Hornof et al. ont conçu et développé une application pour les enfants qui s'appelle EyeDraw (voir la Figure 2.2) afin qu'ils puissent dessiner sur l'écran [42]. A l'aide de l'ERICA (Eye-gaze Response Interface Computer Aid), il est possible pour eux de passer des commandes diverses en fixant leur regard sur l'écran [43]. Certains produits dans ce

Chapitre 2. Etat de l'Art

domaine sont déjà commercialisés, comme par exemple, la société Tobii¹ qui propose des écrans augmentés d'un eye-tracker (comme la Figure 2.3), spécialement conçus pour les personnes qui ont perdu la capacité de bouger leurs corps afin de leur permettre d'interagir avec le monde extérieur par les yeux.

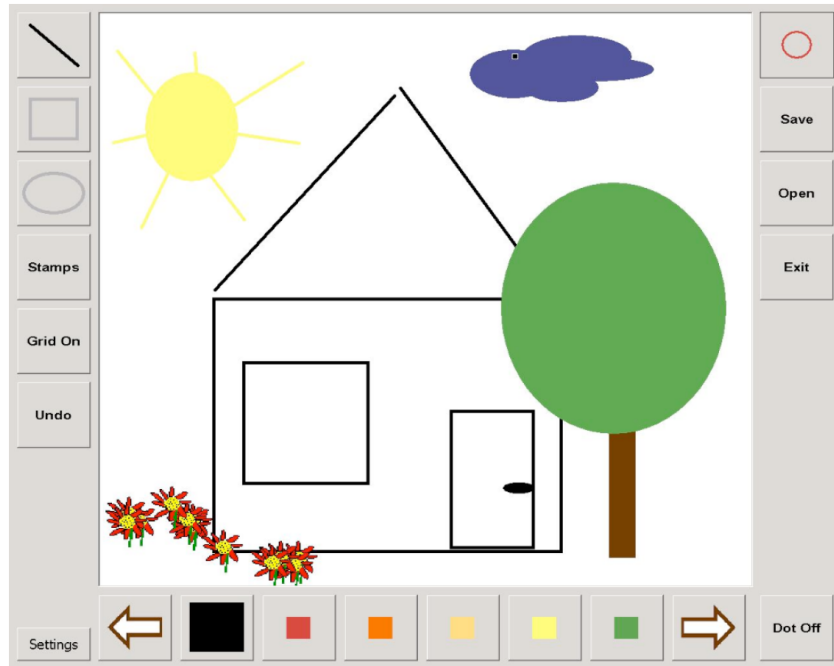


FIGURE 2.2 : Application EyeDraw avec un dessin fait par l'un des auteurs.

Ce n'est cependant pas le seul contexte d'utilisation de l'eye-tracking ; ce dispositif est aussi capable d'améliorer l'expérience utilisateur de tous lors de l'interaction avec un système traditionnel comme un ordinateur puisque cela nous permet d'effectuer des interactions à distance avec les mains libres. Le regard peut aussi être considéré comme une modalité d'entrée naturelle, car il est assez facile de se concentrer sur les éléments uniquement en les regardant. Les expériences de Stampe et Reingold ont prouvé que cette méthode est intuitive et qu'elle ne nécessite aucun d'entraînement [98]. De plus, l'utilisation de l'eye-tracking dans l'interaction avec un écran en tant que modalité de pointage peut également faire gagner du temps [96, 113]. En effet, le pointage du regard de l'utilisateur est systématiquement proche de l'icône qu'il a envie de saisir. Ainsi à l'aide de l'eye-tracking il n'est

1. <https://www.tobiidynavox.com/>



FIGURE 2.3 : Un système d'eye-tracking proposé par Tobii pour les personnes à mobilité réduite.

plus nécessaire de chercher la position actuelle du curseur et ainsi de le déplacer [20]. David Fono et Roel Vertegaal ont présenté et évalué l'idée d'utiliser le point du regard comme une indication de l'attention parmi de multiples fenêtres d'application et de zoomer ce que l'utilisateur est en train de regarder. Le résultat montre que cela peut être 2 fois plus rapide que la souris ou le clavier pendant des tâches de dactylographie [30]. Stellmach et al. ont quant à eux appliqué l'eye-tracking dans le système GazeGalaxy qui sert à la recherche d'un fichier multimédia désiré dans une collection d'images ou de musiques [99]. Ils utilisent une lentille fisheye à la position du regard pour agrandir localement l'affichage des fichiers multimédia en préservant la résolution périphérique, afin que les utilisateurs puissent obtenir à la fois les détails et le contexte (voir la Figure 2.4).

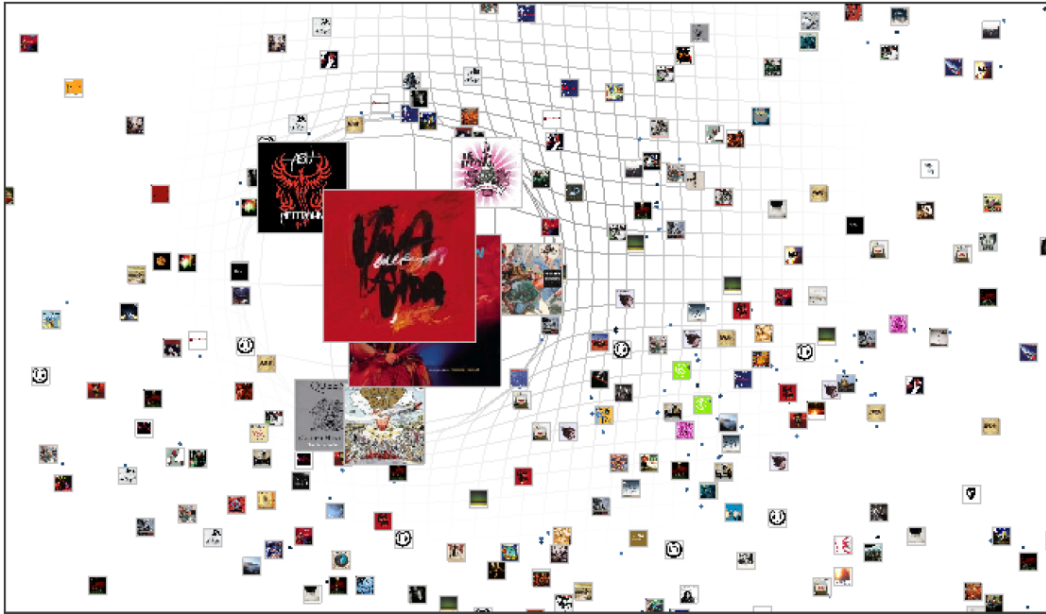


FIGURE 2.4 : Une capture d'écran de GazeGalaxy.

2.2 Interaction Mobile et Ubiquitaire par le Regard

Les recherches sur les environnements intelligents commencent à la fin des années 90, et produisent de nouvelles possibilités d'application de l'eye-tracking pour l'IHM. Plusieurs études sont basées sur l'utilisation de l'eye-tracking afin de sélectionner un appareil et de déclencher l'interaction avec celui-ci en disposant des caméras à côté de chaque appareil [93, 111]. Par exemple, Vertegaal et al. ont créé l'EyeTuner en combinant un haut-parleur avec un ECS (Eye Contact Sensor) digital; les utilisateurs peuvent sélectionner cet appareil en fixant leur regard sur celui-ci (voir la Figure 2.5). Selon les expérimentations de Maglio et al., la plupart des utilisateurs ont rapidement pris l'habitude de regarder l'appareil avec lequel ils vont faire des interactions [67]; cela ne demande pas d'efforts supplémentaires pour les utilisateurs qui peuvent alors interagir à distance avec une vitesse accélérée. Ce résultat a été vérifié dans les recherches de Oh et al. [79] : parmi les 3 méthodes choisies dans les expérimentations pour déclencher une interaction avec l'ordinateur (appuyer sur un bouton, tourner la tête vers l'écran avec lequel interagir, prononcer le mot « computer »), la plupart des participants utilisent le regard avant d'interagir (look-to-talk) lorsqu'on leur demande de commencer l'interaction comme ils

veulent.



FIGURE 2.5 : EyeTuner composé d'un haut-parleur et d'un ECS digital.

2.2.1 Utilisation de la Caméra Frontale

L'eye-tracker porté sur la tête possédant une caméra frontale est également utilisé au profit d'interactions avec l'environnement physique. Parce que ce dispositif peut capturer des images de la scène, il est ainsi capable de servir d'outil de sélection sans nécessiter de nombreuses caméras posées dans l'environnement. Grâce aux méthodes de traitement d'images et de reconnaissance de formes, les chercheurs ont réussi à détecter et reconnaître l'objet regardé en calculant une correspondance entre l'image de la scène en temps réel avec un dispositif pré-connu d'images enregistrées à l'avance dans une base de données. Bonino et al. ont présenté un paradigme d'interaction pour la domotique basée sur le regard qui permet aux utilisateurs d'interagir pleinement avec l'environnement domestique [8] : les caractéristiques SIFT (Scale-Invariant Feature Transform) [65] ont été utilisées pour représenter les périphériques domotiques, le système ART (Attention Responsive Technology) [95] surveille en permanence les mouvements des yeux de l'utilisateur et constate l'attribution de l'attention visuelle dans l'environnement afin de déterminer si le regard de l'utilisateur est fixé sur un appareil contrôlable. Si un appareil est identifié comme étant regardé, l'interaction avec celui-ci commence. Ruminski et al. ont

développé la plateforme eGlasses qui a le même but mais en utilisant différentes technologies pour la reconnaissance des objets, la performance est satisfaisante sous un angle d'observation et des conditions d'éclairage similaires [90].

2.2.2 Applications à l'Extérieur

Comme l'eye-tracker est devenu de plus en plus mobile et robuste, le champ d'application de l'eye-tracking n'est plus contraint à une simple pièce. Anagnostopoulos et al. ont envisagé d'appliquer l'eye-tracking à l'extérieur afin de réaliser des interactions basées sur la position et le regard de l'utilisateur à l'aide de la vision par ordinateur ou d'outils servant à détecter l'orientation de la tête [2]. Ils ont classifié des interactions du regard dans 9 catégories selon 2 dimensions : le mouvement de l'utilisateur pendant l'interaction et le mouvement de l'objet avec lequel interagir (voir le Tableau 2.1). D'après eux, l'objet à interagir peut

- être immobile (par exemple : un bâtiment, un écran collectif),
- se déplacer indépendamment de l'utilisateur (par exemple : un animal, une voiture qui passe), ou
- se déplacer avec l'utilisateur (par exemple : un téléphone portable, un écran porté sur la tête).

Au niveau du mouvement de l'utilisateur, il faut distinguer 3 cas différents :

- l'utilisateur reste dans un endroit prédéfini pendant cette interaction,
- l'utilisateur reste n'importe où pendant cette interaction,
- l'utilisateur se déplace pendant l'interaction.

TABLE 2.1 : Classification des interactions du regard.

	Mouvement de l'utilisateur lorsque l'interaction a eu lieu		
Mouvement de l'objet à interagir		sans déplacement, l'interaction disponible à un certain endroit	sans déplacement, l'interaction disponible n'importe où
	objet immobile	interaction avec des objets immobiles basée sur le regard avec contrainte de localisation Exemple : un écran collectif	interaction avec des objets immobiles basée sur le regard et la localisation Exemple : exploration de la ville
	objet se déplaçant indépendamment de l'utilisateur	interaction avec des objets en mobilité basée sur le regard avec contrainte de localisation	interaction avec des objets en mobilité basée sur le regard et la localisation Exemple : exploration des animaux dans un zoo
objet se déplaçant avec l'utilisateur	interaction avec des objets portables basée sur le regard avec contrainte de localisation	interaction avec des objets portables basée sur le regard	interaction avec des objets portables basée sur le regard pendant un déplacement Exemple : interagir avec un HMD (Head-Mounted Display) par le regard lors de la recherche d'une route

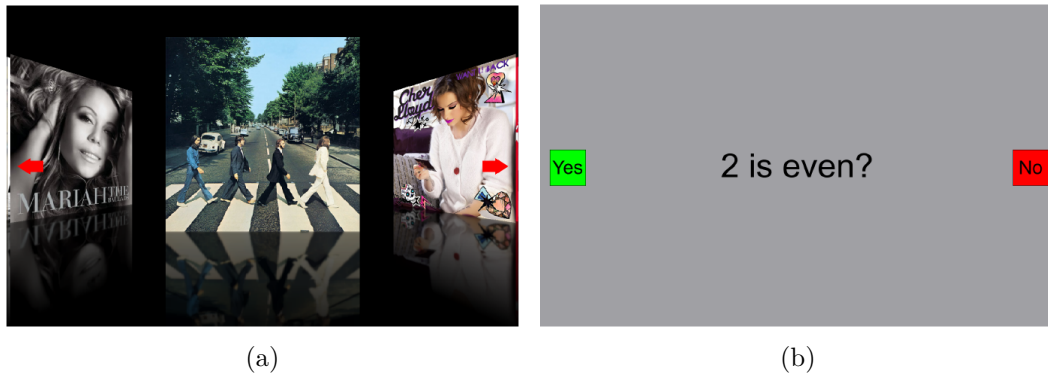


FIGURE 2.6 : Deux applications de SideWays : (a) navigateur de couverture d'albums, (b) quiz du regard.

Il existe des scénarios intéressants pour chaque catégorie, et pour le moment les chercheurs ont déjà développé ou proposé quelques applications qui servent à interagir avec les espaces extérieurs à l'aide de l'eye-tracking. Anagnostopoulos et al. ont proposé un guide touristique basé sur le regard, qui est capable de présenter à l'utilisateur les endroits qui l'intéressent [2]. Drewes et al. ont travaillé sur l'interaction du regard pour les téléphones portables [21]. L'interface SideWays développé par Zhang et al. permet des interactions spontanées avec un écran collectif sans besoin de calibrage ni d'entraînement [120]. Sur cette interface, les utilisateurs peuvent commencer l'interaction en regardant le centre de cet écran, et ils sont capables de réaliser la sélection, le défilement et le glissement à l'aide d'un coup d'œil (à gauche ou à droite). La Figure 2.6 montre deux exemples d'applications de SideWays utilisés dans leur recherche. Eaddy et al. ont proposé une application qui permet de réaliser une interaction dans la même catégorie (interaction avec un objet immobile avec la contrainte de localisation) [26]. Au lieu d'interagir avec un écran collectif, l'EyeGuide qu'ils ont proposé sert à interagir avec une carte (à la station de métro par exemple). Giannopoulos et al. ont réalisé une application (GazeNav) qui sert à indiquer si l'utilisateur est orienté vers la bonne route lorsqu'il est en face d'une intersection [34]. Lorsque le regard de l'utilisateur est fixé sur la bonne route, l'application l'informe par un feedback vibro-tactile du téléphone portable. Dans ce cas-là, la navigation est réalisée en laissant les deux mains de l'utilisateur libres, et l'utilisateur interagit avec l'environnement au lieu d'un appareil pendant

ce processus.

2.2.3 Interprétation d'Informations



FIGURE 2.7 : EyeGuide assiste l'utilisateur en station de métro pour la recherche de la bonne route.

En plus, avec une caméra qui filme la scène regardée, il devient possible de fournir aux utilisateurs une aide pour l'interprétation de ce qu'ils n'arrivent pas à comprendre. En appliquant cela dans le contexte d'un musée, Toyama et al. ont créé une application appelée Museum Guide 2.0, capable de fournir à l'utilisateur des informations complémentaires sur l'objet exposé d'intérêt [105]. Lorsque le regard de l'utilisateur est fixé sur un des objets exposés, l'application joue le fichier audio préenregistré correspondant pour lui décrire précisément cet objet. L'application EyeGuide que nous avons évoqué dans le paragraphe ci-dessus est un exemple dans le contexte de la recherche de la route [26]. Lorsque l'utilisateur est en face d'une carte, l'EyeGuide est capable de lui fournir des assistances par un feedback audio afin qu'il puisse trouver plus rapidement et plus facilement la bonne route (voir la Figure 2.7). En même temps, avec un écouteur il est possible de protéger l'intimité de

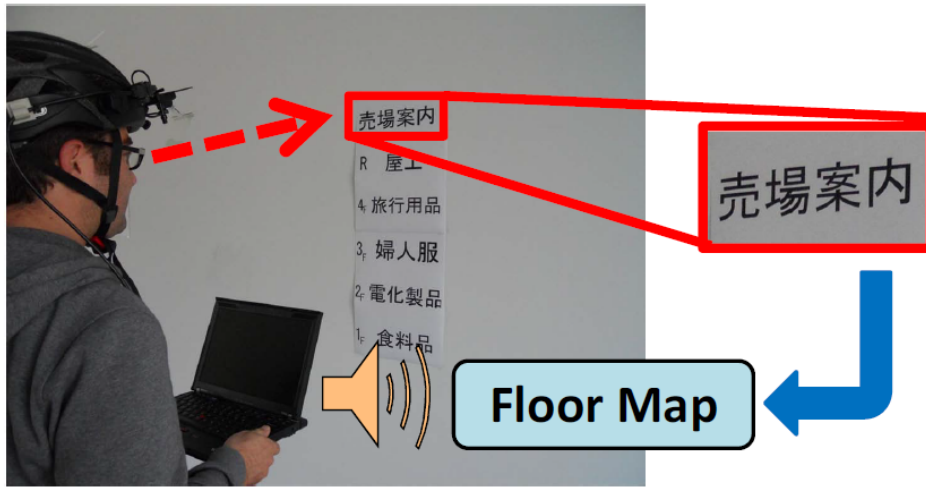


FIGURE 2.8 : Système de traduction à l'aide d'un eye-tracker porté sur la tête avec une caméra frontale.

l'utilisateur si l'endroit qu'il cherche contient des informations privées. Kobayashi et al. ont, quant à eux, travaillé sur la combinaison entre un eye-tracker et un système de reconnaissance de caractères en réalisant un dispositif de reconnaissance (Optical Character Recognition) [75] autour de la région pointée par le regard : cela permet de donner des informations complémentaires aux mots reconnus sous de multiples formes (traduction, image, etc.) [57]. Comme l'eye-tracker est porté sur la tête de l'utilisateur, il n'a pas besoin de prendre une caméra à la main, il a donc les deux mains libres pour réaliser la tâche principale. En plus, comme le processus de reconnaissance n'est appliqué qu'autour de la région pointée par le regard, le temps de calcul est considérablement réduit. D'après les résultats expérimentaux, le temps de calcul pour la reconnaissance des mots est en-dessous d'une seconde en moyenne. La Figure 2.8 montre le système de traduction qu'ils ont réalisé. Biedert et al. ont présenté un système pour les livres augmentés capables de produire des effets liés au contenu que l'utilisateur est en train de lire [6]. Cet effet peut être soit jouer des sons ambients (le hurlement du loup par exemple), soit afficher des images. La Figure 2.9 montre un exemple du livre augmenté de « Le Petit Prince » ; un événement est exécuté (l'image change) lorsque l'utilisateur lit l'annotation correspondante.

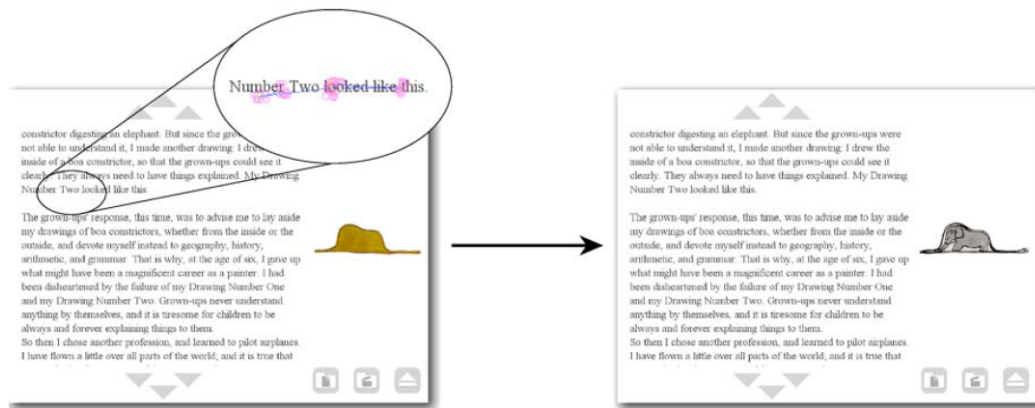


FIGURE 2.9 : Exemple de EyeBook : lorsque l'utilisateur lit une annotation, l'événement correspondant se déclenche.

2.3 Problèmes liés à l'Utilisation de l'Eye-tracking pour les IHM

Le mémoire de thèse de Drewes [20] présente une comparaison entre le regard et les autres modalités de pointage traditionnelles. Le résultat est illustré dans le Tableau 2.2.

TABLE 2.2 : Propriétés des modalités de pointage d'après [20].

	Souris	Trackpoint	Pavé tactile	Écran tactile	Regard
Vitesse	rapide	moyen	rapide	rapide	très rapide
Exactitude	précis	précis	précis	taille d'un doigt	surface de la fovéa
Demande d'espace	grand	petit	petit	aucun	aucun
Retour d'information	oui	oui	oui	non	non
Méthode	indirect	indirect	indirect	direct	direct
Pointage multiple	2 mains	2 mains	10 doigts	10 doigts	1 paire d'yeux
Clic intrinsèque	oui	non	oui	oui	non

Bien que l'utilisation de l'eye-tracking en IHM en tant que modalité de pointage procure de nombreux bénéfices, il pose néanmoins encore un certain nombre de

problèmes [24] :

- La fixation du regard manque souvent de précision : les modalités d'entrée traditionnelles comme la souris nous permettent d'interagir avec l'écran quasiment à la précision du pixel. La fixation du regard par contre, même avec le meilleur eye-tracker existant, peut générer un écart de 0,5 degré à cause de la surface de la fovéa [119].
- Le regard ne permet pas facilement de cliquer : généralement, la sélection d'un élément par l'interface traditionnelle est finie par un 'clic' souris, mais cette action n'est pas facilement réalisable par les yeux. Simuler le clic souris par un clignement de l'œil n'est pas une solution souhaitable, puisqu'il est difficile de différencier un clignement naturel d'un clignement volontaire. De plus le clignement des paupières peut entraîner une dérivation du regard vers le bas et une absence temporaire de données sur le regard. Une autre alternative est de simuler le clic souris par un temps d'arrêt (*dwell time*), comme dans l'EyeDraw proposé par Hornof et al. [42]. Mais cette méthode est très consommatrice en temps (600-1000ms [68]) et ainsi réduit les gains potentiels liés à la rapidité de l'œil. En plus, afin de compenser le manque de précision apporté par les micromouvements ou les tremblements (*tremor* en anglais) des yeux, l'objet à sélectionner (objet physique ou bouton sur l'écran) doit être assez grand ; en conséquence, pour les interactions avec un écran, la quantité d'informations affichable sur l'écran est limitée [73].
- L'œil est également utilisé pour inspecter la scène : la tâche principale de l'œil est d'observer les informations dans l'environnement. Dans le cas d'eye-tracking il n'est pas évident de différencier si l'utilisateur souhaite simplement observer un élément ou bien interagir avec celui-ci. Ce problème est identifié par Jacob et s'appelle le problème du touché de Midas (*Midas touch*) [47].

2.4 Solutions Envisagées

Les chercheurs ont alors proposé de multiples solutions afin de résoudre un ou plusieurs de ces problèmes.

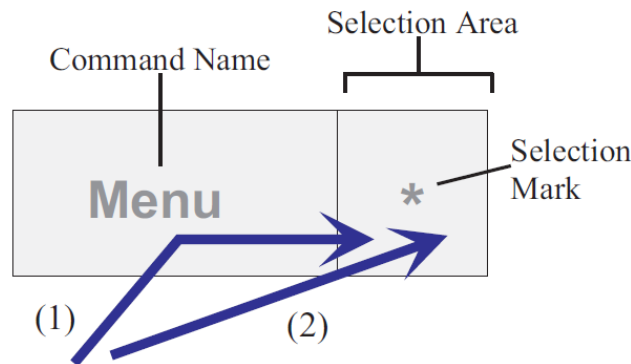


FIGURE 2.10 : Exemple de séparation de la cible à sélectionner en 2 parties : le nom de la commande et la zone de sélection.

Deux méthodes de sélection par le regard sont proposées afin de diminuer le temps consommé par le temps d'arrêt et d'éviter le problème du touché de Midas. La première consiste à séparer la cible en 2 parties : le nom de la commande et la zone de sélection (voir la Figure 2.10) [80]. Pour sélectionner une commande, l'utilisateur peut regarder la zone de sélection correspondante au lieu du nom de la commande. Cela peut être réalisé de 2 manières : les nouveaux utilisateurs qui ne connaissent pas encore l'emplacement des commandes peuvent rechercher d'abord dans la liste des noms de commande puis regarder la zone de sélection correspondante afin de finir la sélection. Pour les utilisateurs qui connaissent très bien l'emplacement des commandes, ils peuvent également regarder directement la zone de la sélection de la cible. Dans la deuxième méthode, il s'agit de limiter les activations erronées potentielles en exigeant deux fixations successives via la visualisation dynamique de la cible. Pour l'interface NeoVisus présenté par Tall, la fixation initiale provoque l'extension de la cible et la révélation d'une seconde cible sur laquelle il faut ensuite fixer pour finir la sélection [101]. Ce processus est illustré dans la Figure 2.11.

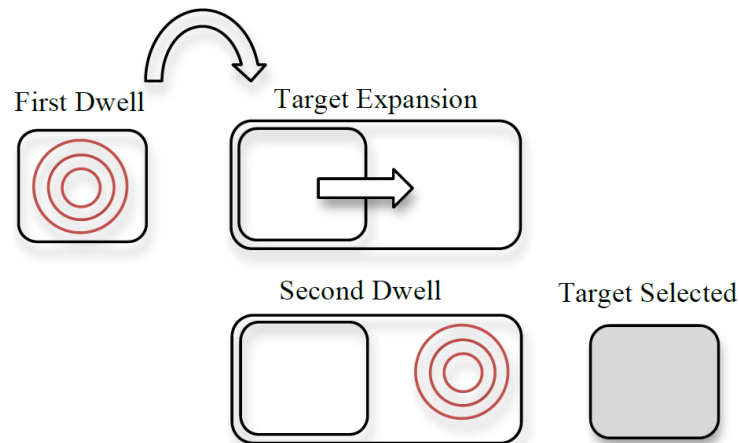


FIGURE 2.11 : Sélection par deux fixations successives adoptée dans NeoVisus.

2.4.1 Mouvements Volontaires des Yeux

Au lieu de fixer le regard sur le bouton à cliquer, les mouvements volontaires des yeux sont étudiés dans le but de passer diverses commandes. Le mouvement volontaire des yeux est défini comme « Un modèle des mouvements oculaires définissables effectués dans une période de temps limitée, qui peut ou ne peut pas être contraint à une plage ou une zone particulière, qui peut être identifié en temps réel, et utilisé pour signifier une commande ou une intention particulière » par Instance et al. [46]. C'est une modalité potentielle pour améliorer les interactions du regard car cela peut potentiellement être très rapide (les saccades peuvent couvrir un angle visuel de 1 à 40 degré et durer entre 30 et 120 ms) [25], et en plus les mouvements volontaires des yeux peuvent être différenciés des mouvements naturels des yeux, évitant le problème du touché de Midas.

La commande peut être passée par un simple coup d'œil (le mouvement entre deux fixations prévues, *single stroke gaze gesture* en anglais) ou un mouvement volontaire complexe des yeux (composé par plusieurs coups d'œil). Mollenbach et al. ont travaillé sur la performance d'un simple coup d'œil lors d'interactions avec un écran [73]. D'après leur expérience, les coups d'œil courts peuvent être nettement plus rapides que les coups d'œil longs, et il n'y a pas de différence significative entre leurs nombres d'erreurs de sélection. Ils ont ainsi présenté la possibilité d'utiliser les coups d'œil de longueurs différentes dans une interface d'interaction. En plus, leur

résultat expérimental montre que l'assistance sur un écran d'une grille par exemple n'influence pas le nombre d'erreurs de sélection ; elle n'est donc pas nécessaire pour l'utilisation de simples coups d'œil.

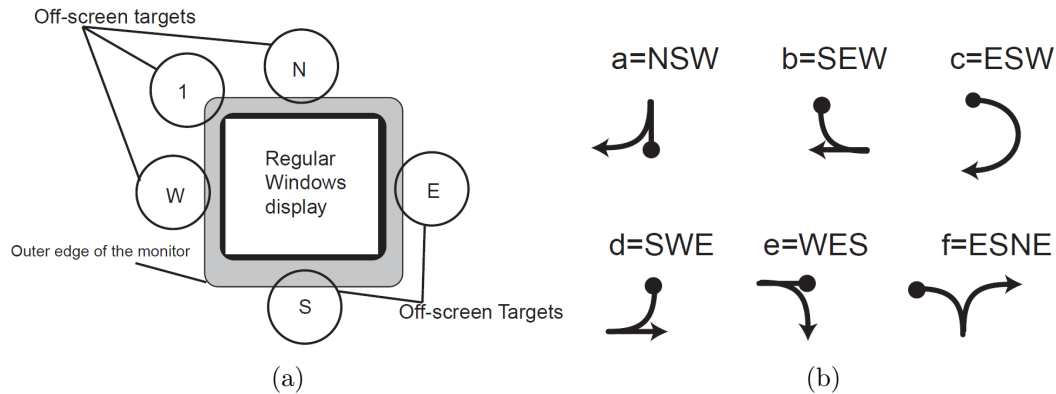


FIGURE 2.12 : Système MDITIM : (a) emplacement des cibles, (b) exemples des interprétations 2D des caractères.

Plusieurs études ont été faites au niveau de l'utilisation des mouvements volontaires complexes des yeux. En 2000, Isokoski et al. ont présenté un système de saisie de texte qui s'appelle MDITIM, dans lequel les mouvements volontaires des yeux entre 5 grandes cibles en dehors de l'écran sont utilisés pour taper du texte [44]. Quatre des cibles sont mises en correspondance aux quatre directions principales : Nord, Est, Sud et Ouest (ou N, E, S et W) afin de laisser aux caractères une interprétation en deux dimensions ; la cinquième cible est utilisée comme un modificateur pour la saisie de caractères majuscules ou d'autres interprétations secondaires des caractères [45]. L'emplacement des cibles et des exemples des interprétations 2D des caractères MDITIM sont montrés dans la Figure 2.12. Wobbrock et al. ont présenté EyeWrite, un système pour la saisie de texte basé sur le regard [115]. C'est un système basé sur le système EdgeWrite qui sert à la saisie du texte pour les PDA (Personal Digital Assistant) [114]. Au lieu d'utiliser les cibles en-dehors de l'écran, EyeWrite fournit aux utilisateurs une fenêtre dédiée à l'exécution et à l'interprétation des gestes (voir la Figure 2.13a). Cette fenêtre est divisée en 5 parties : les 4 coins et le centre. Le texte est entré en passant par les coins au départ du centre et puis en revenant au centre à la fin de la saisie de chaque caractère. Dans ce système,

Chapitre 2. Etat de l'Art

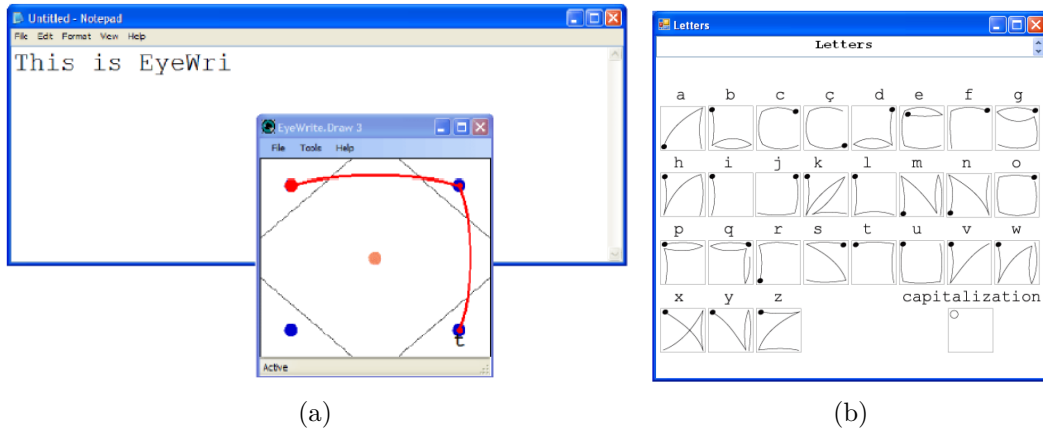


FIGURE 2.13 : Système EyeWrite : (a) fenêtre dédiée à l'exécution et à l'interprétation des gestes, (b) interprétations des caractères.

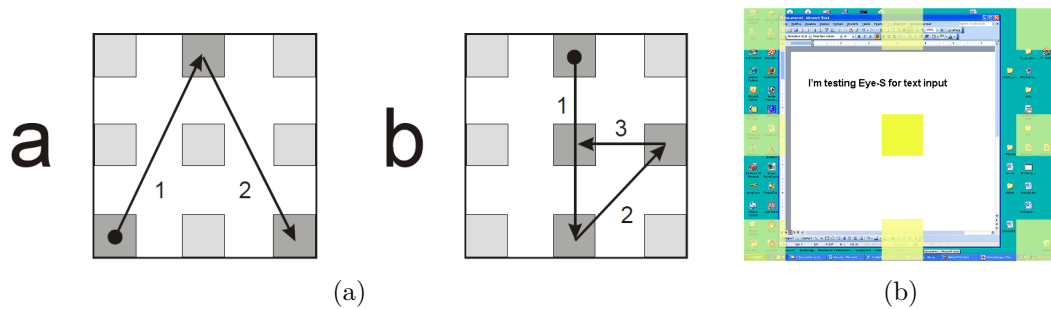


FIGURE 2.14 : Système Eye-S : (a) exemples des séquences des mouvements des yeux pour 'a' et 'b', (b) affichage explicite des cibles.

le croisement est utilisé au lieu du pointage parce que, d'après la loi de Fitts, le croisement est plus facile que le pointage pour les grandes cibles proches l'une de l'autre. En plus, l'alphabet de ce système ressemble à la langue naturelle (voir la Figure 2.13b) afin d'améliorer la mémorisation. De même, inspiré par le concept de 'eye graffiti' présenté par Milekic [72], le système Eye-S proposé par Porta et Turina utilise les séquences des mouvements des yeux qui ressemblent aux formes des lettres pour la saisie des caractères [83], cf. caractères 'a' et 'b' illustrés dans la Figure 2.14a. Au lieu d'une fenêtre spécifique, les séquences des mouvements des yeux sont assistées par 9 cibles distribuées sur l'écran (voir la Figure 2.14b). Ces cibles sont affichées sous forme d'une grille uniformément répartie, invisible mais implicitement identifiable. Ainsi l'Eye-S ne demande aucune interface spécifique sur l'écran, écran qui est ainsi complètement disponible pour des applications. De plus,

leurs études ne sont pas limitées à la saisie des lettres, l'Eye-S peut également servir à passer diverses commandes générales. Dans les autres recherches, les mouvements volontaires des yeux non symboliques composés par des mouvements des yeux vers les directions fondamentales sont étudiés (voir la Figure 2.15)[10, 22, 21]. Les mouvements vers la gauche, la droite, le haut et le bas sont représentés par les symboles 'L', 'R', 'U' et 'D', les mouvements diagonaux sont représentés par les symboles '1', '3', '5' et '7' conformément au pavé numérique du clavier. Drewes et Schmidt ont fait plusieurs tests pour évaluer la performance et l'acceptabilité de cette approche, comme par exemple : l'acceptabilité des mouvements avec les difficultés différentes et la performance de cette approche avec des arrière-plans différents [22]. Drewes et al. ont également évalué la possibilité d'appliquer cette approche sur un appareil mobile avec un petit écran (un téléphone portable) [21].

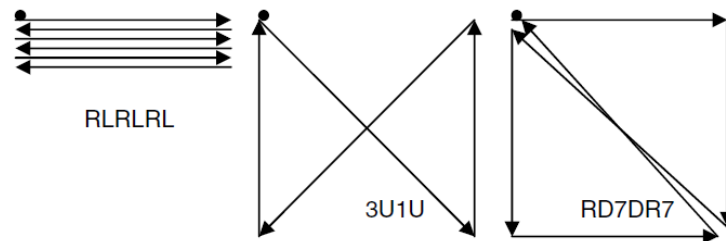


FIGURE 2.15 : Exemples de séquences d'un mouvement d'œil non symbolique.

Bien qu'avec les mouvements volontaires des yeux, la précision du point du regard n'est plus aussi importante, ces mouvements peuvent engendrer de la fatigue oculaire puisque le contrôle intentionnel des mouvements oculaires n'est pas naturel. En plus, il faut éviter le problème du chevauchement potentiel entre les mouvements volontaires et les mouvements naturels des yeux lors du choix des mouvements utilisés [73].

2.4.2 Poursuite Lisse

Toujours dans le même but, la poursuite lisse des yeux (*smooth pursuit*) est étudiée dans le but de sélectionner une cible.

Dans le brevet américain 6,243,076 B1, Hatfield a présenté une invention qui détermine les commandes que l'utilisateur souhaite passer en comparant le mouve-

Chapitre 2. Etat de l'Art

ment de ses yeux avec les mouvements connus d'objets graphiques mobiles générés par ordinateur [41]. Dans son application, chaque objet graphique en déplacement est lié avec une série de commandes, l'utilisateur peut ainsi passer les commandes en suivant le mouvement de l'objet correspondant par ses yeux. D'après les recherches, le suivi des objets qui se déplacent dans la plage de 1 à 30 degrés d'angle visuel par seconde (la poursuite lisse) est plus naturel et plus facile à effectuer que la fixation. Du coup c'est une modalité potentielle pour le processus de sélection en combinant l'étape de concentration (déplacer le curseur sur un bouton) et l'étape de consentement (le clic souris).



FIGURE 2.16 : Un utilisateur est intéressé par un album, il suit le mouvement de cet album sur l'écran par ses yeux; un extrait de la musique est alors joué automatiquement.

Vidal et al. ont présenté à la conférence CHI 2013 le principe de la poursuite qui est naturelle et utilisable sans processus de calibration [112]. Ils ont évalué cette approche en développant 3 applications interactives implémentées sur un système composé par un grand écran et un eye-tracker au-dessous de l'écran. Une de ces applications est illustrée dans la Figure 2.16. D'après leurs expériences, le déplacement des objets est flexible : cela peut être linéaire ou par suivi d'une trajectoire plus complexe; les objets peuvent partager la même trajectoire globale et encore être sélectionnés en fonction de leur mouvement local; et la taille de la cible n'a pas d'importance.

La poursuite lisse est aussi appliquée dans l'interface StarGazer développée par Hansen et al. [39]. Le StarGazer est une interface 3D pour l'interaction basée sur le

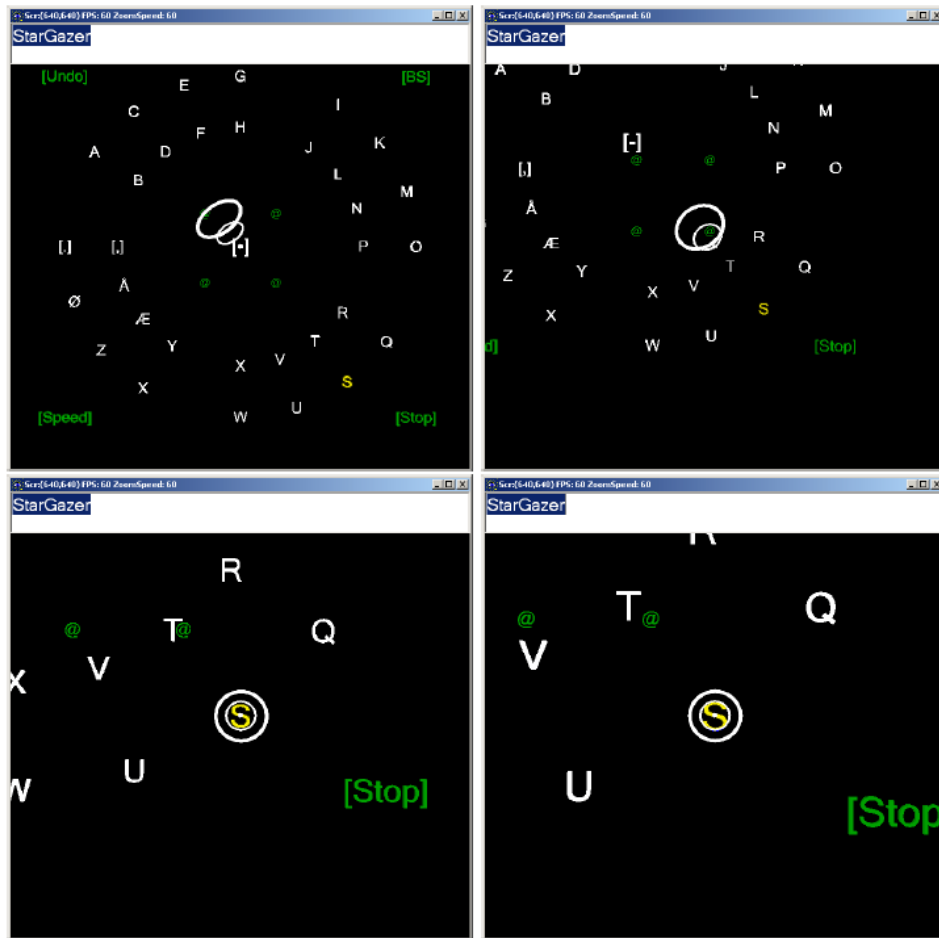


FIGURE 2.17 : Les zooms et les pans lors de la sélection de la lettre 's'.

regard et la sélection de la cible à l'aide des pans et des zooms en continu. La Figure 2.17 illustre les zooms et les pans lors de la sélection d'un objet ; les trois anneaux concentriques au centre de l'écran montrent la direction du regard actuel. Lorsque le point du regard de l'utilisateur est détecté sur une lettre, celle-ci se déplace au centre ; les yeux de l'utilisateur réalisent automatiquement la poursuite lisse en se déplaçant radialement vers le centre afin de suivre l'objet. En même temps, le zoom est fait pour que la séparation spatiale entre les objets soit suffisamment grande ; l'utilisateur peut changer la direction de son regard s'il veut suspendre la sélection. Lorsque l'objet est assez 'gros' et assez proche du centre de l'écran, la sélection est réalisée.

Esteves et al. ont travaillé sur l'utilisation de la poursuite lisse pour sélectionner

Chapitre 2. Etat de l'Art

l'objet cible sur une montre intelligente [28]. Dans leur recherche, plusieurs orbites circulaires sur l'écran sont utilisées en même temps, et plusieurs cibles se déplacent sur chaque orbite. L'utilisateur peut passer une commande en suivant la cible correspondante. Les auteurs ont essayé de différencier les multiples orbites de plusieurs façons : 1) varier la phase de leurs cibles en ajoutant un décalage à la position initiale de chaque cible ; 2) varier la vitesse angulaire des cibles ; 3) varier la direction du mouvement, où une partie des cibles se déplacent dans le sens des aiguilles d'une montre, alors que les autres se déplacent dans le sens antihoraire ; 4) varier la taille des orbites (le diamètre de la trajectoire). Plusieurs expériences ont été conduites afin d'évaluer la performance de leur approche, et ils ont développé trois applications pour l'illustrer en pratique. L'application 'lecteur de la musique' est présentée dans la Figure 2.18 en tant qu'exemple.

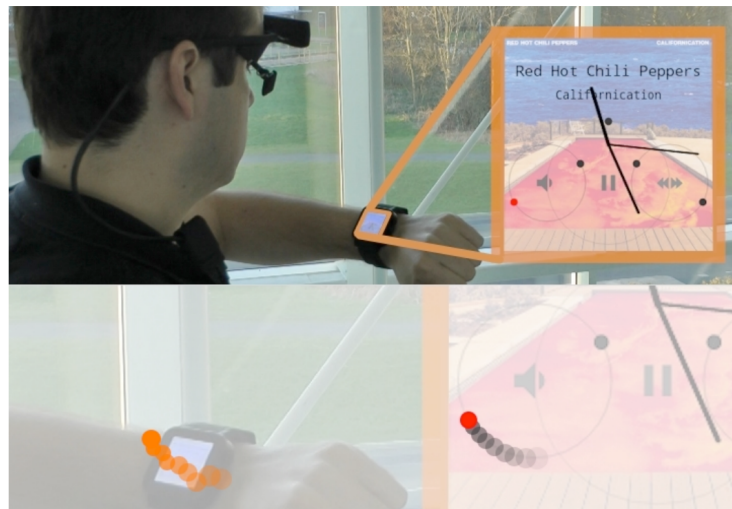


FIGURE 2.18 : Exemple de l'augmentation du volume d'un lecteur de musique d'une montre intelligente.

Au lieu d'interagir avec un écran, Velloso et al. s'intéressent à appliquer la poursuite lisse dans l'environnement physique. Ils ont proposé l'environnement intelligent AmbiGaze qui utilise l'animation des cibles pour fournir aux utilisateurs un contrôle direct des appareils uniquement par la poursuite lisse des yeux [110]. La difficulté d'interagir avec des objets physiques par la poursuite lisse est d'exposer les contrôles sous forme animée. Pour cela, ils ont proposé un espace de conception

dans lequel ils ont considéré plusieurs façons pour présenter le mouvement : virtuelle par une animation graphique versus mécanique par un mouvement physique, et généré localement par l'appareil versus projeté extérieurement sur l'appareil. Ils ont ainsi créé quatre prototypes (voir la Figure 2.19), un pour chaque catégorie. Pour le prototype du lecteur de la musique, l'animation est projetée sur les haut-parleurs afin que l'utilisateur puisse interagir avec celui-ci. La lampe est contrôlée via un moulin dont les quatre pales sont de couleurs différentes ; l'utilisateur peut modifier la couleur de la lampe en suivant le mouvement de la pale correspondant. Le quatrième prototype est réalisé en projetant un point laser sur une plaque acrylique montée autour d'un ventilateur. Le point du laser se déplace autour du ventilateur, l'utilisateur peut ainsi allumer ou éteindre cet appareil par la poursuite lisse.



FIGURE 2.19 : Quatre prototypes d'AmbiGaze : (1) lecteur de la musique (virtuel+externe), (2) interface vidéo-sur-demande (virtuel+interne), (3) lampe (mécanique+interne), (4) ventilateur (mécanique+externe).

Comme la poursuite lisse des yeux n'est pas utilisée pour l'observation de la scène, cela nous permet d'éviter le problème du touché de Midas. En plus, le calibrage précis de l'eye-tracker n'est plus obligatoire, car la sélection est faite par la vitesse et la direction du mouvement de l'objet au lieu de la position exacte de celui-ci [73]. Par contre, la poursuite lisse est difficile à effectuer en l'absence d'un

objet en déplacement même avec des entraînements [41]. Donc il faut que la cible soit en mouvement pendant le processus de sélection. Dans ce cas-là, la poursuite lisse est surtout convenable pour les interactions avec un écran collectif comme il n'y a pas de difficulté pour afficher des cibles mobiles, et les utilisateurs n'ont plus besoin de faire le calibrage avant d'interagir avec celui-ci.

2.4.3 Couplage avec un autre dispositif

En-dehors de l'utilisation de la poursuite lisse et des mouvements volontaires des yeux, la majorité des solutions proposées par les chercheurs est basée sur le couplage du regard avec un autre dispositif. Ce dispositif secondaire peut être issu d'une modalité d'entrée traditionnelle (comme par exemple une souris [23, 119], un écran sensible au touché [94, 107] ou un pavé tactile [99]) ou d'une nouvelle technologie (comme par exemple l'interface cerveau-ordinateur [102] ou les dispositifs pour la capture de mouvement [109]).

2.4.3.1 Couplage avec un dispositif traditionnel

Dans le système EyeWindows, Fono et Vertegaal ont combiné l'eye-tracking avec une touche de déclenchement de commande (la barre d'espace) pour sélectionner la fenêtre de focus sur l'écran [30]. Dans ce cas, ils sont capables de profiter des avantages de l'eye-tracking en évitant le problème du touché de Midas. D'après leur expérience, cette méthode est environ 72% plus rapide que la souris et les touches de raccourci, et elle est préférée par la majorité des participants.

Dans l'étude de pointage MAGIC (*Manuel And Gaze Input Cascaded*), Zhai et al. utilisent une souris pour déplacer le curseur dans la région à côté du point du regard afin de compenser le manque de précision de la fixation du regard et réaliser l'action de clic après [119]. Avec une telle approche, le pointage apparaît à l'utilisateur comme une tâche manuelle, utilisée pour une manipulation fine. Cependant, une grande partie du mouvement du curseur est éliminée en affichant le curseur dans la zone du regard qui entoure la cible, l'effort physique et la durée de manipulation sont ainsi réduits. Ils ont implémenté 2 techniques spécifiques de

pointage MAGIC : une, plus libre, pour laquelle le curseur s'affiche à proximité de (ou sur) chaque cible potentielle ; dans l'autre, plus contrainte, le curseur n'apparaît à proximité de la position du point du regard qu'après activation de la souris afin d'éliminer la perturbation du curseur. Drewes et al. ont conçu un mécanisme qui est la combinaison du pointage MAGIC de Zhai et d'une souris sensible au touché : le curseur s'affiche à la position du point du regard lorsque le doigt est posé sur le capteur (sur le bouton gauche de la souris) pour éliminer la perturbation du curseur [23]. La souris, sensible au touché, se comporte comme une souris ordinaire tant que le doigt reste sur le capteur tactile. Grâce à cet appareil, le pointage MAGIC est activé sans besoin d'efforts supplémentaires : il suffit de mettre le doigt sur le bouton à cliquer, et il est possible de repositionner le curseur immédiatement en levant le doigt et puis toucher à nouveau le bouton.

Au niveau de la sélection d'objet hors de portée physique sur un grand écran de bureau, Mauderer et al. ont proposé de combiner l'eye-tracking avec le geste pichenette, qui est une imitation de l'action de glisser un objet sur la table présenté par Reetz et al. [88], pour réaliser la sélection à distance et augmenter l'exactitude de la sélection [70]. La geste produit une ligne (f) qui traverse la surface de l'écran. Idéalement cette ligne devrait couper la ligne (g) entre la position des yeux et celle du regard au moment de la sélection de la cible souhaitée (voir la Figure 2.20). Comme ce cas ne peut pas être supposé, ils calculent la ligne (h) qui minimise la distance entre (f) et (g), et considèrent que l'objet le plus proche de cette ligne est sélectionné. Turner et al. ont également travaillé sur l'interaction avec un objet hors de portée physique sur un écran [107]. Ils ont présenté trois techniques qui utilisent la combinaison de l'eye-tracking et des gestes sur un écran sensible au touché pour déplacer l'objet d'intérêt : couper/coller, glisser/déposer et récupérer/retourner. A l'aide de ces trois techniques, il leur est possible de mettre l'objet à manipuler à portée de l'utilisateur et de le placer dans un endroit plus éloigné à la fin de la manipulation.

Dans le projet de recherche ART (Attention Responsive Technology), Shi et al. ont couplé l'eye-tracking avec un écran sensible au touché afin d'interagir en direct avec les objets physiques dans l'environnement [94]. La fixation sur un objet

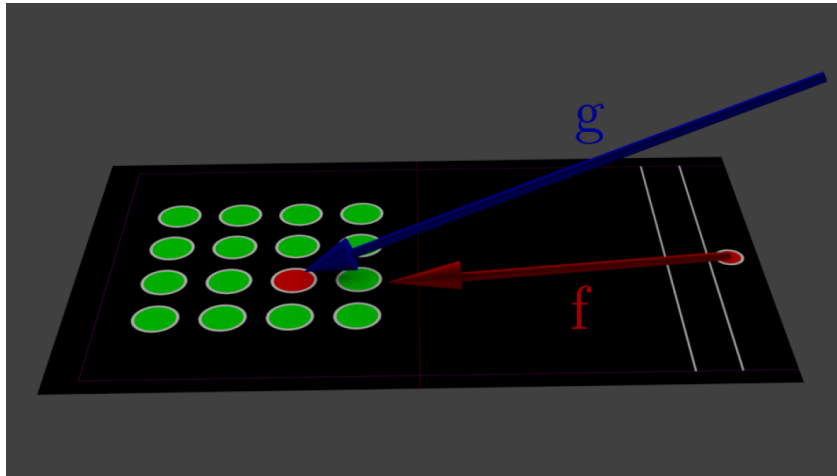


FIGURE 2.20 : Représentation schématique des informations issues du geste pichelette et du regard, la ligne (f) représente la direction du geste, la ligne (g) représente le regard de l'utilisateur.

physique est confirmée via un menu affiché sur l'écran. Ainsi ils peuvent éviter les commandes erronées déclenchées par un regard fait au hasard.

2.4.3.2 Couplage avec les gestes 3D

Le corps humain est capable de faire énormément de gestes qui peuvent être utilisés pour contrôler des objets choisis. Récemment, le domaine de l'interaction gestuelle 3D a motivé de nombreuses études sur la taxonomie des gestes, les mécanismes de capture et de suivi des gestes, les algorithmes de reconnaissance de gestes et l'utilisation des gestes 3D en tant que modalité d'interaction [54]. Les gestes 3D comme modalité d'entrée ont été adoptés dans le domaine du jeu vidéo, et cela intéresse de plus en plus de systèmes commerciaux, comme par exemple la Kinect de Microsoft et la télécommande WiiMote. Elle devient ainsi une modalité envisageable à coupler avec l'eye-tracking.

L'interaction gestuelle concerne la totalité du corps humain qui peut demander l'utilisation du corps entier ou bien des micromouvements effectués avec un seul ou plusieurs membres du corps.

Le poignet est un membre du corps astucieux et facile à utiliser. Le mouvement du poignet possède 6 degrés de liberté : les translations et les rotations respective-

ment selon les 3 axes (voir la Figure 2.21). La flexion/extension est le mouvement de pliage pour diminuer ou augmenter l'angle entre la paume et le bras, la pronation/supination qui est souvent appelé la rotation du poignet signifie le changement de l'orientation de la paume, la déviation ulnaire/radiale est le mouvement du poignet vers le haut (ou vers le bas) lorsque la paume est orientée vers la côté [36]. D'après Feldman et al., la rotation du poignet est la plus apte pour passer les commandes puisque c'est minimal en termes de quantité de mouvement dans l'espace, et peut être parfaitement reconnue par les capteurs inertiels [29].

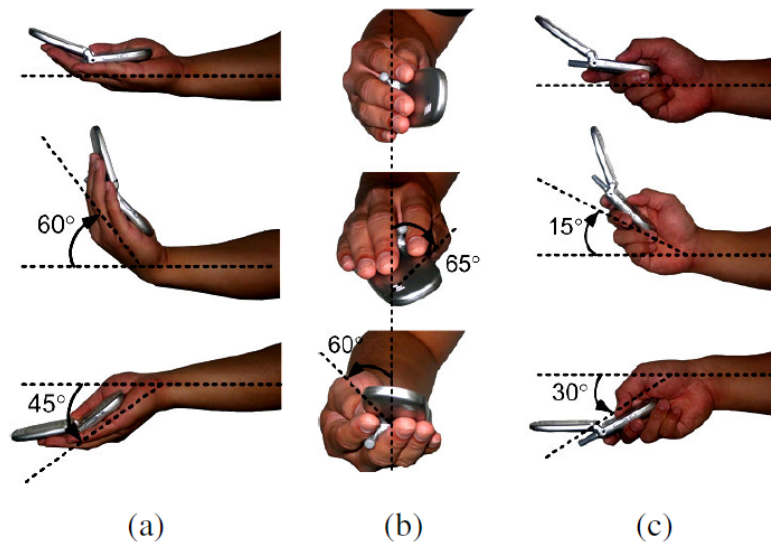


FIGURE 2.21 : Rotations du poignet selon les 3 axes.

Les recherches de Crossan et al. ont essayé de contrôler la position latérale du curseur sur un écran par la rotation du poignet. Le résultat exprime que les utilisateurs peuvent bien sélectionner les cibles avec une largeur correspondant à un angle de plus que 9 degrés lorsqu'ils sont stationnaires [16]. Oakley et O'Modhrain ont contribué au travail de sélection d'un objet par la rotation du poignet en comparant la performance pour deux techniques de correspondance [77] :

- la correspondance basée sur la vitesse utilise l'angle de rotation pour contrôler la vitesse de transition d'un objet à l'autre,
- la correspondance basée sur la position attribue une plage d'angles pour chaque objet dans la liste.

Chapitre 2. Etat de l'Art

Selon leurs études, le contrôle basé sur la position marche mieux pour les listes avec un nombre limité d'objets. Par contre le contrôle basé sur la vitesse n'a pas de contraintes sur le nombre d'objets. Rahman et al. ont discrétisé l'espace angulaire par 3 méthodes différentes (discrétisation linéaire, discrétisation sigmoïde et discrétisation quadratique) pour tester la performance des rotations du poignet selon les 3 axes [84]. Leurs expérimentations montrent que les utilisateurs sont capables de sélectionner confortablement la cible parmi 12 (ou 16) choix par la flexion/extension (ou la pronation/supination) du poignet lorsque l'espace angulaire est discrétisé de manière quadratique. Oakley et Park ont proposé de diviser une gamme angulaire de 90 degrés en 3 zones de taille identique, et transmettre les commandes par une séquence de mouvements entre les 3 zones [78]. Avec la différence entre la zone de départ, la zone terminale et les zones parcourues, il y a au total 19 commandes utilisables pour ce système. Tsukada et Yasumura ont travaillé sur le contrôle des objets physiques par les gestes qui correspondent aux métaphores des commandes [106]. Ils proposent de modifier le volume d'un haut-parleur par la rotation du poignet car normalement cette commande est réalisée en tournant un bouton.

Les gestes effectués avec le pied ouvrent un grand espace de conception pour les problèmes d'interaction lorsque les mains des utilisateurs sont occupées ou indisponibles. Plusieurs gestes ont été étudiés pour passer des commandes diverses. Crossan et al. ont travaillé sur la possibilité d'interagir avec un téléphone portable en tapant des pieds afin que l'utilisateur puisse passer les commandes sans sortir le téléphone de sa poche. Le résultat est plutôt satisfaisant au niveau de l'exactitude alors que l'efficacité est acceptable pour les gestes de moins de cinq tapotements [14]. Mise à part le tapotement du pied qui se produit lorsque l'utilisateur est stationnaire, le pas (*foot-step*) a également du succès en tant que modalité d'entrée quand l'utilisateur est en train de faire du jogging. Tetsuya et al. sont arrivés à identifier quatre types de pas (*step right/left*, *step twice right/left*) avec une grande exactitude à l'aide des capteurs d'accélération sur les chaussures [116]. Les recherches de Scott et al. ont porté sur quatre types de rotations de la cheville : dorsiflexion, flexion plantaire (*plantar flexion*), rotation du talon (*heel rotation*) et rotation des orteils (*toe rotation*) qui sont détectables par un accéléromètre et ont une amplitude de mouvement

raisonnable (voir la Figure 2.22) [92]. Les résultats expérimentaux montrent que la sélection à l'aide de la flexion plantaire, de la rotation du talon ou de la rotation des orteils ont des taux d'erreur et des vitesses acceptables ; ils sont donc utilisables pour les fonctions comme par exemple la navigation dans un menu. Schöning et al. ont essayé d'interagir avec un système d'information géographique en utilisant la pression du pied (*foot pressure*) capturée par un Wii Balance Board [91]. Ils ont découvert que la pression du pied a des avantages dans les tâches spatiales puisque cela peut nous fournir une modalité d'entrée intuitive pour les données continues et elle ne fatigue pas les utilisateurs. Han et al. ont proposé de passer les commandes en utilisant les coups de pied (*kicking*) car cela permet un grand degré de liberté [38]. D'après leur recherche, les coups de pied sont assez précis pour distinguer jusqu'à 5 directions en face de l'utilisateur et au moins deux vitesses différentes. En complément de leur travail, Alexander et al. ont évalué la performance de cette approche avec des techniques de correspondance différentes dans les missions de navigation [1]. Les résultats expérimentaux montrent que le contrôle basé sur la vitesse est plus efficace et plus précis que le contrôle basé sur le déplacement.

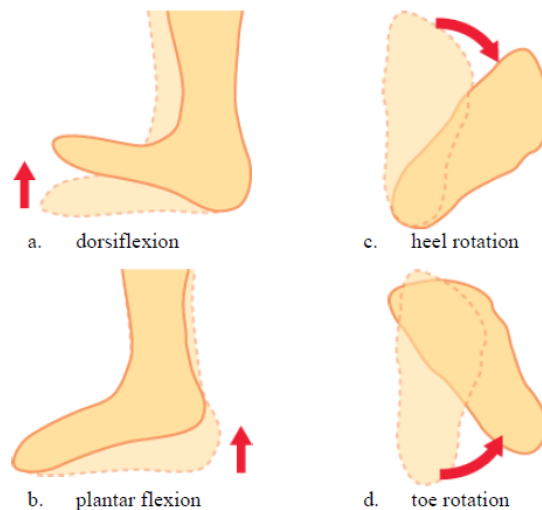


FIGURE 2.22 : Quatre types de rotation de cheville.

Les mouvements effectués par la tête peuvent également nous servir à réaliser plusieurs tâches. Les études de Crossan et al. essayent de contrôler la position ho-

horizontale d'un curseur sur l'écran à l'aide de l'inclinaison de la tête (à gauche et à droite) respectivement par la position (l'angle d'inclinaison) et la vitesse angulaire du pliement [15]. Leurs études montrent que le contrôle basé sur la position a des avantages au niveau du temps de réaction et est plus exact lorsque l'utilisateur est stationnaire. Par contre le contrôle basé sur la vitesse est plus précis lorsque l'utilisateur est en train de marcher. Kjeldsen a également travaillé sur les possibilités offertes par l'inclinaison de la tête. Il pense que cela doit être utilisé à une amplitude et d'une fréquence limitées, puisque la tête inclinée fatigue les muscles du cou [56]. A part l'inclinaison, les rotations de la tête (à gauche/droite et en avant/arrière) sont aussi utilisables avec une bonne exactitude. Brewster et al. ont développé un menu radial auditif en 3 dimensions (3D auditory radial pie menu) qui utilise le hochement de la tête vers des directions différentes pour la sélection [9]. D'après leur expérimentation, le nombre de cibles dans le menu est limité afin de garantir l'exactitude, et les participants trouvent que le hochement en arrière n'est pas très confortable à effectuer. Morency et Darrell ont travaillé sur l'utilisation du hochement et le secouement de la tête pour accepter ou rejeter les boîtes de dialogue en remplaçant la souris et le clavier [74]. Le résultat expérimental montre que les gestes de la tête sont efficaces et les participants trouvent cette approche naturelle.

La combinaison du regard et des gestes 3D a été étudiée dans plusieurs recherches. Koons et al. ont proposé d'intégrer les informations qui viennent de la voix, du regard et des gestes 3D de la main lors d'interactions avec une interface d'ordinateur afin d'augmenter la bande passante entre les deux parties de la communication [58]. Un paradigme d'interaction en couplant ces 2 techniques est celui de l'interaction soutenue par le regard : le regard suggère et puis les gestes 3D le confirment. Velloso et al. ont essayé de manipuler les objets dans une 3DUI (*3D user interface*) en utilisant le regard et les gestes de la main en même temps : l'utilisateur regarde l'objet qu'il a envie de sélectionner, sa main fait le geste du pincement et bouge afin de sélectionner et déplacer cet objet, et puis relâche le pincement pour désengager cette interaction [109]. D'après leur expérience, cette méthode est beaucoup plus rapide qu'en utilisant seulement les gestes de la main, et cette technique est préférée par les participants.

Dans la plupart des études au niveau de l'interaction par les gestes 3D, les gestes sont capturés soit par un capteur électrique embarqué (comme par exemple un bracelet ou une pédale), soit à l'aide de la vision par ordinateur en traitant les images capturées par une (ou un groupe de) caméras. Dans tous ces cas, un équipement supplémentaire spécialement pour la capture des gestes est obligatoire si on veut interagir avec les objets physiques dans l'environnement. Les eye-trackers sous forme d'une paire de lunettes équipées d'une caméra de la scène ont apporté une autre possibilité : cette caméra frontale peut nous servir pour la détection d'une partie des gestes 3D, du coup aucun appareil supplémentaire n'est nécessaire. Dans l'étude de Hales et al., l'algorithme de la reconnaissance des gestes de la main est activé sur l'image de la scène pour une durée courte lorsqu'un marqueur qui représente un objet à interagir est détecté [37]. Les gestes de la main qu'ils ont essayé de détecter sont composés de 2 parties : la partie statique en tenant la main avec un nombre prédéfini de doigts pendant un temps d'arrêt prédéfini de 1 s, la partie dynamique en bougeant la main vers une direction (le haut, le bas, la gauche ou la droite).

Malgré les avantages, le contrôle gestuel 3D à son propre frein : les utilisateurs n'ont pas de moyen de connaître les commandes disponibles et la correspondance entre les gestes et les commandes [18]. Donc il faut soit avoir une conception des jeux de gestes utilisés dans un contexte bien précis [60, 108] soit fournir une technique de guidage à l'utilisateur [3].

2.4.4 Commandes Implicites

Normalement, les commandes d'interaction sont passées d'une façon explicite. Mais une nouvelle tendance de l'utilisation de l'eye-tracking pour les IHM est de prédire l'intention de l'utilisateur grâce au mouvement de ses yeux et ainsi de contrôler les objets de manière implicite.

Biswas a utilisé la vitesse et l'accélération des mouvements des yeux pour prédire la cible prochaine lors d'interactions avec une interface sur un écran d'ordinateur [7]. Tong et Fisher ont proposé de simuler un grand écran avec une résolution ultra haute en fournissant une haute résolution sur la zone que l'observateur est en train

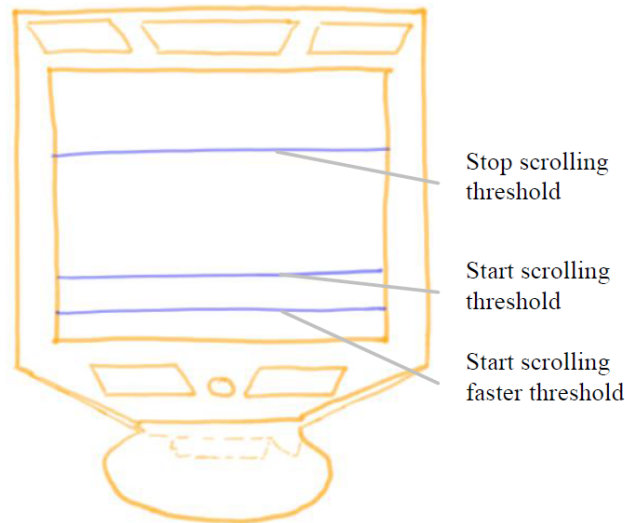


FIGURE 2.23 : Lignes de seuil pour le défilement de la page.

de regarder et une résolution plus basse à la périphérie [104]. Kumar et al. ont proposé plusieurs manières implicites pour le défilement d'un fichier sur un écran [62]. D'après leur recherche, au lieu de cliquer sur le bouton qui permet le défilement de la page, il est possible d'ajuster dynamiquement la vitesse du défilement afin de garantir que le regard de l'utilisateur se situe dans le tiers médian de l'écran, grâce aux algorithmes d'estimation de la vitesse de la lecture [5]. Cette méthode ne marche que pour la lecture du contenu textuel parce que le modèle de balayage pour les images incluses dans le texte peut varier. Une autre manière est d'utiliser plusieurs lignes de seuil sur l'écran comme sur la Figure 2.23. Lorsque le regard de l'utilisateur tombe en-dessous du seuil de début du défilement, le document commence à défiler lentement. Si le regard de l'utilisateur tombe en-dessous du seuil du défilement rapide, le système commence à défiler le texte plus rapidement. La vitesse du défilement est réglée légèrement plus vite que la vitesse de lecture de l'utilisateur, de manière à déplacer lentement la position du regard de l'utilisateur vers le haut de l'écran. Lorsque le regard de l'utilisateur atteint le seuil de l'arrêt du défilement, le défilement s'arrête et l'utilisateur peut continuer à lire la page normalement. Dans l'étude de Kumar et al., les auteurs ont essayé de faire défiler la page, de raffiner le résultat d'une recherche d'image et d'éditer un fichier multimédia implicitement pour réaliser des interactions plus naturelles [61]. Le défilement de la

page est fait implicitement en analysant la position du regard actuelle : la direction du défilement est déterminée par le quadrant du regard, alors que la vitesse est proportionnelle à la distance du centre d'écran. Au niveau de la recherche d'image, les feedbacks sont collectés implicitement lorsque l'utilisateur est engagé dans la navigation, les images fixées ou zoomées impliquent la région d'intérêt et peuvent ainsi servir au raffinement de la recherche. Et puis, les images importantes identifiées par le modèle du regard de l'utilisateur peuvent ainsi être sauvegardées pour faciliter la rédaction du fichier ultérieur.

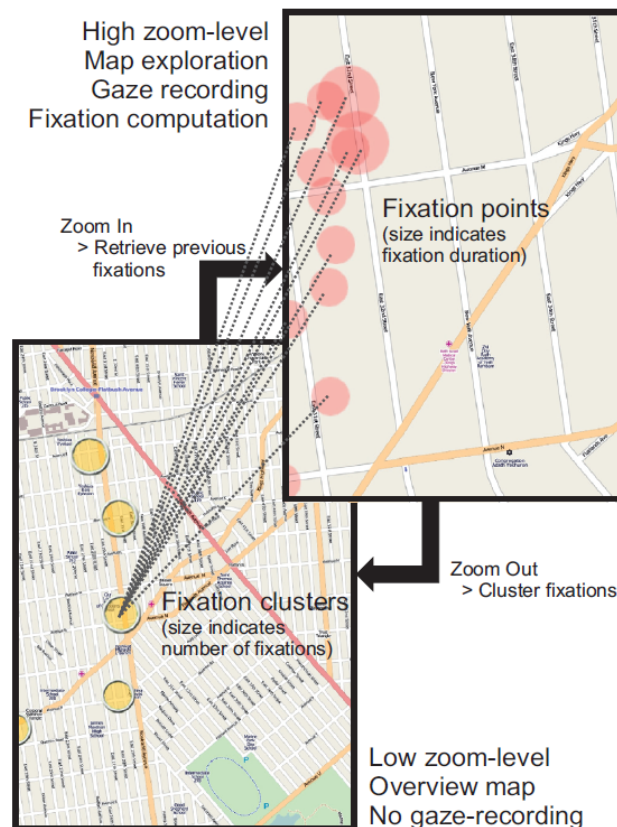


FIGURE 2.24 : La conception de l'application GeoGazemarks.

Dans l'application de GeoGazemarks, Giannopoulos et al. proposent de fournir l'historique du regard sur la carte à l'utilisateur en tant que référence de l'orientation pour qu'il puisse retrouver facilement les lieux d'intérêts sur un petit écran [33]. Cette application est composée de 2 phases : la phase d'enregistrement et la phase de récupération. La phase d'enregistrement est activée lorsque l'utilisateur interagit

avec la carte avec un niveau de zoom élevé ; la fixation de l'utilisateur est enregistrée et agrégée au point d'intérêt le plus proche. La phase de récupération est activée lorsque l'utilisateur effectue un zoom arrière en-dessous d'un certain seuil de niveau de zoom. Le regard n'est plus enregistré et les fixations notées précédemment sont groupées et visualisées (voir la Figure 2.24).

Les commandes implicites ne sont pas limitées aux interactions traditionnelles avec un écran. Kiefer et al. ont proposé de prédire le moment auquel l'utilisateur perd son intérêt lors de l'observation d'un panorama, car ce moment pourrait être un bon moment pour offrir des informations d'assistance [55]. Ils ont conduit une expérience sur 12 voyageurs en leur demandant de regarder le panorama de la ville aussi longtemps qu'ils veulent. D'après leur recherche, la durée de l'exploration du panorama de la ville est liée avec le nombre de AOI (area of interest) revisité.

2.5 Conclusion du Chapitre

Dans ce chapitre, nous avons présenté un état de l'art pour l'application de l'eye-tracking dans le domaine de l'IHM. Nous avons présenté les applications proposées par les chercheurs, y compris les applications servant à faciliter la vie des personnes à mobilité réduite, les interactions avec un écran d'ordinateur utilisable pour tous, les applications mobiles pour contrôler des objets intelligents, et les interactions externes diverses. Ces applications peuvent nous inspirer pour la conception de nos propres scénarios. Puis, nous avons examiné les problèmes et les difficultés rencontrés dans ce domaine, et nous avons trouvé que l'une des difficultés concerne l'absence de méthodes pratiques pour déclencher des commandes. Nous avons ensuite étudié les solutions envisageables proposées (mouvements volontaires des yeux, poursuite lisse, couplage avec un autre dispositif, etc). La comparaison de tous ces solutions envisagés sera présentée au chapitre qui suit afin de souligner les avantages de la solution proposée.

Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

L'étude bibliographique nous a permis de découvrir que l'une des difficultés majeures de l'Eye-tracking dans le domaine de l'IHM concerne l'absence de méthodes pratiques pour déclencher des commandes. Ce chapitre explore la solution que nous proposons pour fluidifier les interactions par le regard. En comparant les avantages et les inconvénients des différentes solutions proposées par les chercheurs, nous avons décidé de travailler sur la combinaison de l'eye-tracking et du mouvement de la tête à regard fixe parce que cela nous permet de déclencher des commandes diverses sans utiliser les mains ni changer la direction du regard [50].

3.1 Etat de l'Art du Mouvement de la Tête à Regard Fixe

Les mouvements naturels de la tête lors de l'observation de l'environnement sont toujours accompagnés d'un changement de la direction du regard. D'autre part, grâce aux réflexes vestibulo-oculaires des yeux qui compensent les mouvements de la tête par un mouvement des yeux dans la direction opposée, un mouvement volontaire de la tête avec le regard fixé sur un point précis est possible.

La combinaison des mouvements de la tête et du regard pour interagir avec un écran en utilisant uniquement un eye-tracker a été étudiée depuis 2012. Ces

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

recherches permettent de coupler le regard avec les mouvements de la tête sans besoin d'un appareil supplémentaire (à part l'eye-tracker).

Mardanbegi et al. ont essayé de détecter les mouvements des yeux à regard fixe uniquement à l'aide d'un eye-tracker monté sur la tête et ont proposé d'utiliser ce type de mouvement pour interagir avec un écran [69]. Dans leur recherche, les mouvements des yeux à regard fixe sont détectés en traitant les images des yeux capturés par les caméras des yeux de l'eye-tracker. Ils ont essayé de détecter 8 mouvements élémentaires des yeux : 2 horizontaux (à gauche et à droite), 2 verticaux (vers le haut et vers le bas), 2 diagonaux vers le bas, et 2 rotations (à gauche et à droite). Les gestes utilisés pour déclencher les commandes sont composés de 2 mouvements élémentaires des yeux à regard fixe : cela peut être un geste de balayage (2 mouvements élémentaires dans le sens l'inverse) ou un geste répétitif (2 mouvements élémentaires dans le même sens séparé par une pause courte). D'après leur expérience, les gestes diagonaux ne sont pas réalisables pour tous les participants, les utilisateurs trouvent que les gestes répétitifs sont plus difficiles à effectuer par rapport aux gestes de balayage, alors que les gestes rotationnels sont un peu plus fatigants que les autres gestes.

De même, Spakov et al. ont étudié la même technologie en utilisant un eye-tracker à distance [97]. Les auteurs détectent 5 gestes différents en comparant la direction des deux yeux capturée avec les plages de direction prédéfinies. Les plages utilisées pour chaque geste sont estimées dans un test préliminaire, si le modèle du mouvement tombe dans les plages prédéfinies, le geste correspondant est identifié. Comme les plages appliquées dans l'algorithme peuvent être influencées par la position de l'utilisateur par rapport à l'écran, le champ d'application de cette méthode est contraint.

Les travaux de Nukarinen et al. ne sont pas limités au déclenchement d'une action simple ; ils ont étudié l'optimisation de la performance des mouvements de la tête dans des interactions complexes et continues [76]. Ils utilisent la rotation de la tête (à gauche ou à droite) pour diminuer ou augmenter le nombre affiché sur un écran. Une fois que l'angle de rotation de la tête dépasse un seuil prédéfini, le nombre affiché augmente ou diminue en continu. Ils ont testé plusieurs taux de

changement dans leur expérience afin de trouver un compromis entre la vitesse de la sélection et le problème de surréaction.

3.2 Le Choix du Mouvement de la Tête à Regard Fixe pour Déclencher les Commandes

Nous avons comparé les solutions exposées dans la Section 2.4 pour résoudre les problèmes au niveau de l'application de l'eye-tracking dans les interactions sous plusieurs aspects. Les avantages et les inconvénients pour chaque modalité sont présentés dans le Tableau 3.1. Le déclenchement de commande par un temps d'arrêt (*dwell time*) est très consommateur en temps, et réduit ainsi les gains potentiels liés à la rapidité de l'œil, donc cette solution n'est pas prise en compte dans notre comparaison.

On s'aperçoit que les mouvements volontaires des yeux et la poursuite lisse nécessitent un changement de la direction du regard. Dans ce cas, le regard ne peut plus servir en tant que modalité de pointage comme le curseur sur un écran. S'il existe plusieurs objets sur la scène, ils peuvent être utilisés pour sélectionner un objet ou pour déclencher une commande sur l'objet sélectionné, au lieu de sélectionner un objet et de déclencher des commandes simultanément. Dans notre recherche, on s'intéresse plutôt à l'interaction soutenue par le regard afin de profiter de la rapidité du regard en tant que modalité de pointage, du coup ces deux modalités ne nous conviennent pas. En plus, l'affichage des cibles en déplacement est obligatoire pour la réalisation de la poursuite lisse ; des appareils supplémentaires sont nécessaires lors d'une interaction avec les objets physiques dans l'environnement.

Les gestes 2D de la main (souris/pavé tactile/écran sensible au touché), par contre, peuvent être utilisés pour déclencher les commandes sur l'objet fixé. Cependant, comme ils exigent que les utilisateurs lâchent leurs mains afin d'exécuter ces actions, ils ne sont pas pratiques si leurs deux mains sont occupées par une autre tâche, et c'est un gaspillage de l'avantage de l'eye-tracking : la possibilité d'effectuer des interactions avec les deux mains libres. De plus, ces outils demandent aux utilisateurs de les regarder pour poser leur main dessus, ce qui est gênant.

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

TABLE 3.1 : Comparaison des différentes modalités pour le déclenchement des commandes en combinaison avec un eye-tracker porté sur la tête.

	Besoin d'un appareil supplémentaire	Besoin de libérer les deux mains	Besoin de modifier la direction du regard	Capacité à déclencher de nombreuses commandes	Durée d'exécution de la commande
Mouvements volontaires de la tête à regard fixe	Non	Non	Non	Moyenne	Moyenne
Mouvements volontaires des yeux	Non	Non	Oui	Bonne	Moyenne
Poursuite lisse	Oui	Non	Oui	Moyenne	Moyenne
Souris/pavé tactile /écran sensible au touché	Oui	Oui	Oui	Moyenne	Courte
Mouvements 3D de la main/du bras (capteur portable ou caméra supplémentaire)	Oui	Oui	Non	Bonne	Moyenne
Mouvements 3D de la main (la caméra frontale de l'eye-tracker)	Non	Oui	Oui	Moyenne	Moyenne
Mouvements du pied (pédale ou capteur portable ou caméra supplémentaire)	Oui	Non	Non	Faible (pédale) ou moyenne	Courte à moyenne
Contrôle par la voix	Oui	Non	Non	Bonne	Moyenne à longue

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

Les gestes 3D de la main ou du bras sont également capables de déclencher des commandes sur l'objet fixé, mais ils demandent aussi de lâcher la main pour effectuer les mouvements correspondants. En plus, si les gestes sont détectés par un capteur portable ou par le traitement des images filmées par une caméra spécifique, un équipement supplémentaire est exigé ; si les gestes sont identifiés à l'aide des images capturées par la caméra de la scène de l'eye-tracker, aucun appareil supplémentaire n'est nécessaire, mais la fixation du regard est perturbée par les gestes de la main.

Par conséquent, les mouvements volontaires de la tête à regard fixe, les mouvements du pied et le contrôle par la voix sont plutôt compétitifs si nous voulons déclencher des commandes sur un objet dans l'environnement physique tout en gardant les deux mains libres. Parmi ces trois modalités, la capacité à déclencher de nombreuses commandes pour les mouvements du pied est plutôt faible. Le contrôle vocal est capable de déclencher un grand nombre de commandes, mais son temps d'exécution lors du déclenchement des commandes peut être plus long que les autres modalités, car l'utilisateur doit prononcer ces commandes. Ces modalités peuvent être choisies en fonction des besoins de l'application. Lorsque nous n'avons besoin que d'un nombre limité de commandes et que nous ne voulons pas que l'exécution des commandes prennent trop de temps, le mouvement de la tête à regard fixe sera le meilleur choix. Cependant, si nous devons déclencher un plus grand nombre de commandes, le contrôle vocal pourrait être un choix potentiel intéressant. En plus, ces 2 modalités pourraient être combinées pour satisfaire à la fois le besoin de l'efficacité et de la variété : les mouvements volontaires de la tête à regard fixe servent à déclencher les commandes principales ou les commandes qui ont plus d'exigence au niveau de l'efficacité, alors que le contrôle vocal est appliqué pour passer des commandes complémentaires.

Par ailleurs, les mouvements du pied et le contrôle vocal nécessitent tous les deux un appareil supplémentaire (une pédale, un capteur portable ou une caméra pour les mouvements du pied ; un microphone pour le contrôle vocal). Les mouvements volontaires de la tête à regard fixe, par contre, peuvent être détectés en utilisant seulement les images capturées par la caméra de la scène de l'eye-tracker. Par conséquent, l'avantage de coupler les mouvements de la tête à regard fixe avec

le regard pour déclencher des commandes sur l'objet cible est davantage mis en évidence.

3.3 Mouvements Volontaires de la Tête à Regard Fixe Choisis

Il y a beaucoup de mouvements de la tête disponibles. Dans notre recherche, nous travaillons sur la détection des mouvements combinés par des mouvements élémentaires et le retour à la position naturelle à la fin de l'exécution : ces mouvements sont faciles à effectuer, et comme la tête est à la position naturelle après chaque mouvement, c'est pratique pour continuer l'observation ou pour faire un mouvement ultérieur. La Figure 3.1 montre les 6 mouvements qu'on a choisis de détecter dans notre travail.

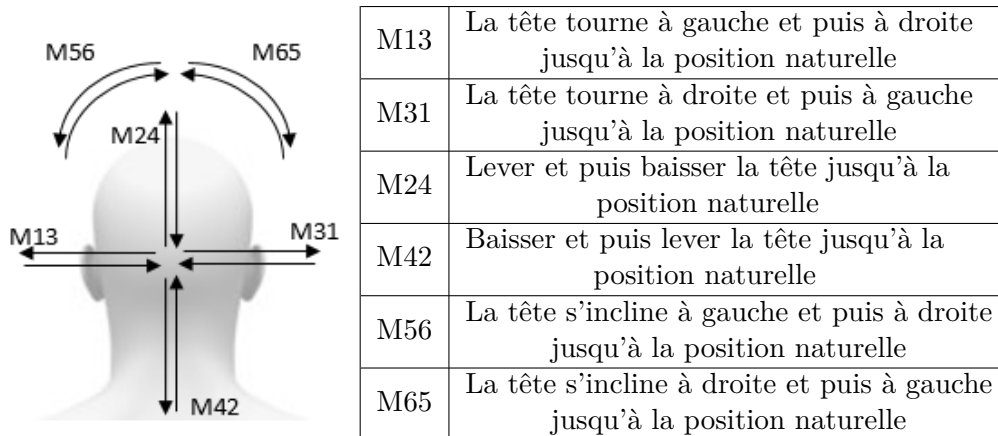


FIGURE 3.1 : Les mouvements volontaires de la tête à regard fixe choisis.

Même si les mouvements de la tête à regard fixe qu'on a choisis de détecter ne sont pas aussi nombreux que ceux utilisés par Mardanbegi et al. [69], ils sont en fait les mouvements préférés par les participants selon leurs résultats expérimentaux. En effet les gestes diagonaux ne sont pas utilisables par tout le monde et les gestes répétitifs sont plus difficiles à effectuer que ces gestes de balayage. De plus, en comparant avec les travaux de Spakov et al. [97] et Nukarinen et al. [76] qui utilisent respectivement 5 et 2 mouvements volontaires de la tête à regard fixe, 6 mouvements est un nombre relativement important.

3.4 Techniques de Reconnaissance des Mouvements de la Tête à Regard Fixe

Un mouvement de la tête à regard fixe doit respecter deux conditions : la première est que le regard est fixe, la seconde est qu'il y a un mouvement de la tête volontaire. Dans nos travaux, l'idée est de se limiter aux mouvements simples et minimaux selon les 3 axes de la tête afin de laisser le reste du corps le plus libre possible. Mais les mouvements indirects de la tête entraînés par les mouvements du haut du corps sont aussi acceptables, car certains utilisateurs ont tendance à bouger aussi le haut du corps (en particulier pour les gestes de dodelinement), et que dans tous les cas on aura bien un 'mouvement de la tête à regard fixe'.

La fixation du regard est identifiée par la variation des coordonnées relatives du regard par rapport à l'environnement, et ce dernier est calculable à l'aide de la matrice d'homographie (une matrice 3*3 qui relie la transformation des coordonnées entre 2 surfaces planes dans le domaine de la vision par ordinateur). Avec les objets pré-enregistrés détectés, on est capable de calculer la matrice d'homographie en utilisant les coordonnées des objets détectés sur l'image de la scène. Il est donc facile de calculer les coordonnées relatives du regard par rapport à l'environnement et ainsi de savoir si le point du regard est fixé sur la même région pendant un mouvement de la tête.

Comme les yeux bougent dans la direction opposée et à la même vitesse que la tête lors des mouvements de la tête à regard fixe, il est possible de mesurer les mouvements de la tête par les mouvements des yeux. La Figure 3.2 illustre le mouvement des yeux lors d'un mouvement de la tête à regard fixe. Mardanbegi et al. ont étudié la détection des mouvements des yeux à l'aide des images oculaires capturées par les caméras des yeux [69]. Dans notre recherche, nous étudions la détection des mouvements de la tête à regard fixe en utilisant uniquement des images de la scène avec le point du regard superposé afin de gagner en temps de calcul, en particulier pour les mouvements verticaux et horizontaux.

Nous avons implémenté deux algorithmes différents en parallèle pour détecter les 6 mouvements de la tête à regard fixe que nous avons choisis : un pour les

mouvements verticaux et horizontaux (M13, M31, M24 et M42), l'autre pour les deux dodelinements de la tête (M56 et M65) [51].

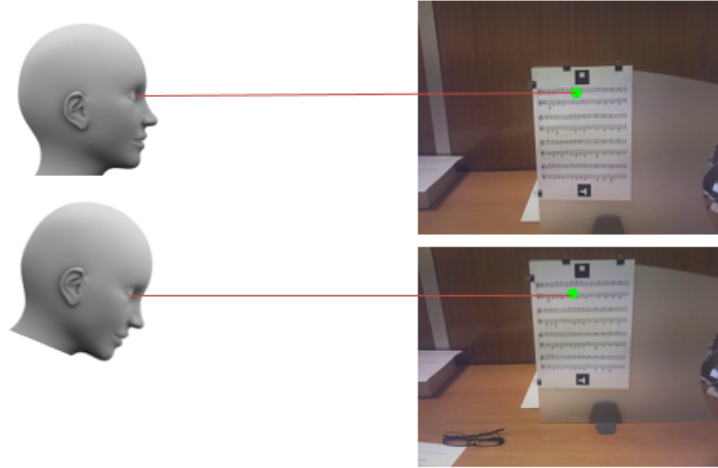


FIGURE 3.2 : Variation du point du regard lors des mouvements verticaux de la tête à regard fixe.

3.4.1 Reconnaissance des Mouvements Verticaux et Horizontaux

Lors des mouvements verticaux ou horizontaux de la tête à regard fixe, les yeux de l'utilisateur se déplacent verticalement ou horizontalement, et les coordonnées du point du regard sur l'image de la scène changent également de la même manière. Ainsi, M13, M31, M24 et M42 sont identifiables par la variation des dérivées des coordonnées du point de regard. Par exemple, lorsque l'utilisateur effectue le mouvement M24 avec son regard fixé sur le même point, la coordonnée y de son point du regard devrait avoir une variation similaire à celle de la Figure 3.3(a). Ce type de variation des coordonnées peut être détecté en calculant sa dérivée : si la dérivée de la coordonnée y possède successivement un pic positif et un pic négatif qui ont dépassé les seuils dans un temps limité, comme dans la Figure 3.3(b), nous admettons qu'il existe un mouvement requis des yeux. Les seuils ont été choisis à l'aide de tests préliminaires menés avec nos collègues du laboratoire. Au cours de ces tests, nous leur avons demandé de faire des mouvements de la tête à regard fixe rapidement et confortablement, car nous ne voulons pas que notre approche prenne beaucoup de temps ou soit fatigante. Nous avons constaté que la dérivée

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

des coordonnées du point du regard possède toujours des pics (un pic négatif et un pic positif) avec une valeur absolue supérieure à 15 pixels toutes les 1/12 seconde (180 pixels/seconde puisque nous traitons 12 images par seconde) lorsqu'ils font les mouvements horizontaux ou verticaux. Cela implique que la vitesse angulaire maximum peut être supérieure à 10 degrés par seconde. En outre, la contrainte de la durée est définie en exigeant que les deux pics ont lieu dans un délai de 10 trames traitées (0,8 s), car cela ne prend pas beaucoup de temps. En effet, nos tests préliminaires montrent que le mouvement rapide de la tête à regard fixe satisfaisant cette contrainte est réalisable par tous les participants.

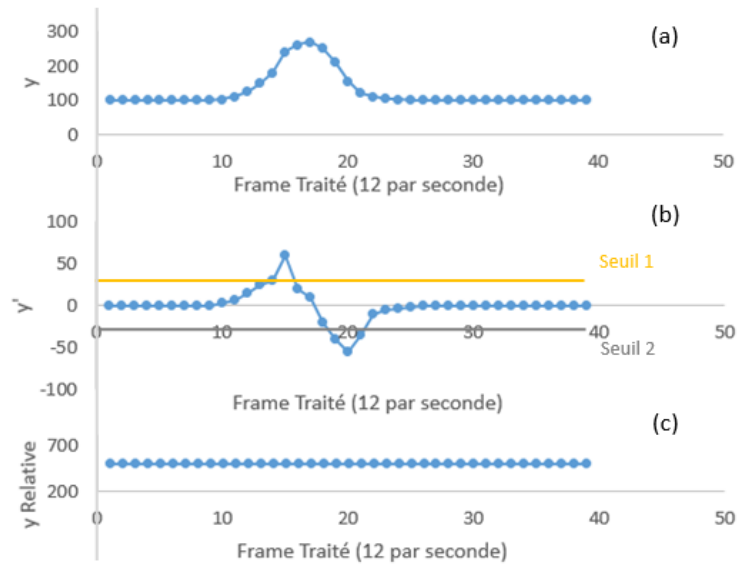


FIGURE 3.3 : Variations des coordonnées du point du regard pendant un mouvement M24 à regard fixe : (a) variation de la coordonnée y , (b) dérivée de la coordonnée y , (c) variation de la coordonnée y relative.

Idéalement, si les coordonnées relatives du point du regard par rapport à l'environnement restent au même point ou changent légèrement dans la région du même objet pendant le mouvement des yeux (comme dans la Figure 3.3(c)), nous pouvons être sûrs que l'utilisateur a effectué un mouvement de la tête à regard fixe. Malheureusement, lors des mouvements de la tête, la netteté des images de la scène n'est pas garantie. Il y a donc plusieurs trames sans objet détectable, ce qui signifie que nous perdons les coordonnées relatives du regard pendant les mouvements. La Figure 3.4(a) montre le signal de la coordonnée y relative que nous obtenons réelle-

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

ment pendant un mouvement M24 à regard fixe : la partie cachée par un carré gris est la période pendant laquelle nous pouvons perdre les coordonnées relatives du regard. En conséquence, dans le pire des cas, nous n'obtenons que des coordonnées relatives du regard au début et à la fin de chaque mouvement. Par conséquent, nous cherchons à distinguer différentes situations qui pourraient provoquer un tel mouvement oculaire et exclure tous les autres cas.

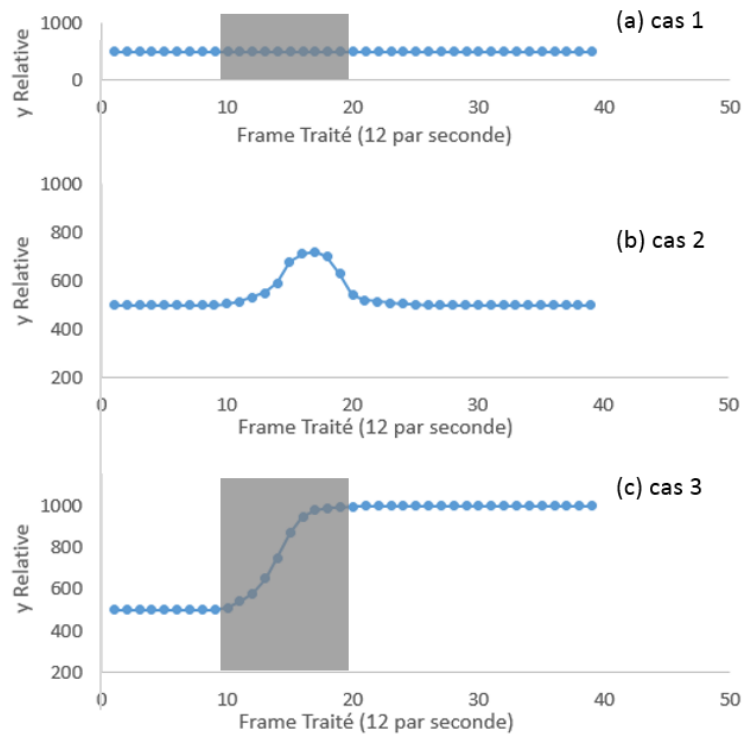


FIGURE 3.4 : Différences sur la coordonnée y relative entre les 3 cas qui peut entraîner une variation des coordonnées du regard souhaitée : (a) coordonnée y relative pendant un mouvement de la tête à regard fixe, (b) coordonnée y relative au cours d'un pur mouvement oculaire, (c) coordonnée y relative lorsque les yeux et la tête de l'utilisateur bougent en même temps.

Normalement, un mouvement oculaire tel que celui illustré dans la Figure 3.3(a) a lieu dans 3 situations différentes :

Cas 1 : mouvement de la tête à regard fixe,

Cas 2 : mouvements oculaires naturels lorsque la tête ne bouge pas,

Cas 3 : Les yeux et la tête de l'utilisateur bougent en même temps dans la même direction et les yeux bougent plus vite que la tête.

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

Le premier cas est le mouvement de la tête à regard fixe que nous voulons détecter via les dérivées des coordonnées du point du regard sur l'image de la scène, ainsi le cas 2 et le cas 3 doivent être exclus.

Exclusion du cas 2 : Pour le deuxième cas, le signal de la coordonnée y relative doit ressembler à ce qui est représenté dans la Figure 3.4(b) si la variation de la coordonnée y est semblable au signal dans la Figure 3.3(a). Lors du mouvement naturel des yeux, les coordonnées relatives du point du regard possèdent une variation significative par rapport aux coordonnées du point du regard sur l'image de la scène. De plus, bien que ces deux variations suivent la même direction, le problème au niveau de la perte d'informations du point du regard n'existe pas car la tête de l'utilisateur ne bouge pas. Ainsi, il est possible d'exclure les mouvements oculaires purs naturels en fixant une contrainte sur le rapport entre la dérivée des coordonnées relatives du point du regard par rapport à l'environnement regardé et la dérivée des coordonnées du point du regard sur l'image de la scène. Nous avons choisi de fixer un seuil sur le rapport de ces deux dérivées au lieu de limiter l'amplitude de la variation des coordonnées relatives du point du regard, car ce dernier ne peut exclure que les mouvements de grande envergure des yeux, tandis que le premier est également capable d'éliminer les petits mouvements oculaires et les tremblements. Lors des mouvements horizontaux et verticaux de la tête avec le regard fixe (cas 1), ce rapport doit soit ne pas exister (à cause du manque de netteté de l'image de la scène) soit être inférieur au seuil (car le regard reste à la même position) tandis que ce seuil est dépassé pour les mouvements oculaires naturels purs. Par conséquent, ce seuil peut être utilisé pour exclure les mouvements oculaires purs naturels sans ignorer les mouvements de la tête à regard fixe désirés.

Exclusion du cas 3 : Pour le troisième cas, il existe une énorme différence entre les coordonnées relatives du point regard avant et après le mouvement, comme ce qui est montré dans la Figure 3.4(c). Également, le point du regard est fixé sur une autre région après le mouvement. Bien que nous puissions perdre les coordonnées relatives du point du regard par rapport à l'environnement pendant le mouvement, il est facile de différencier les mouvements de la tête avec le regard fixé pour lesquels les coordonnées relatives du regard sont stables avant et après le mouvement.

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

En conclusion, s'il existe 2 pics (un pic positif et un pic négatif) sur la dérivée de la coordonnée x (ou y) qui ont dépassé 15 (THRESHOLD_PEAK) dans 10 images de la scène traitées (THRESHOLD_DURATION), au cours desquelles le rapport entre la dérivée de la coordonnée relative x (ou y) et la dérivée de la coordonnée x (ou y) sur l'image de la scène ne dépasse pas un seuil (THRESHOLD_RATIO), et le regard est fixé sur la même région avant et après le début du mouvement (premier pic), nous admettons qu'il existe un mouvement vertical (ou horizontal) de la tête à regard fixe. Bien que nous séparions les conditions pour les mouvements verticaux et les mouvements horizontaux dans l'algorithme ci-dessous afin de le rendre plus clair, ces mouvements sont détectés simultanément dans notre algorithme.

Algorithme 1 : Reconnaissance des mouvements horizontaux et verticaux de la tête à regard fixe

<pre> 1 def reconnaissance des mouvements horizontaux et verticaux de la tête à regard fixe (numéro du frame traitée n) 2 begin 3 obtenir les coordonnées du regard → x(n), y(n) 4 calculer les dérivées des coordonnées → x'(n), y'(n) 5 isSharpEnough = FALSE 6 if object détecté 7 begin 8 isSharpEnough = TRUE 9 obtenir les coordonnées de l'objet 10 calculer la position relative du regard → xR(n), yR(n) 11 calculer les dérivées des coordonnées relatives du regard → x'R(n), y'R(n) 12 chercher le numéro de la région correspondante → nRegion(n) 13 end 14 if x'(n) > THRESHOLD_PEAK (ou < -THRESHOLD_PEAK) 15 if isSharpEnough = FALSE ou x'R(n)/ x'(n) < THRESHOLD_RATIO 16 begin 17 peak_x(n) = 1 (ou -1) 18 for i = 0 to THRESHOLD_DURATION 19 if peak_x(n-i) = -1 (ou 1) 20 for j = 0 to i 21 if nRegion(n- THRESHOLD_DURATION) = nRegion(n-j) 22 return M13 (or M31) detected 23 end 24 return 0 25 end </pre>	<pre> reconnaissance des mouvements verticaux y'(n) y'R(n)/ y'(n) peak_y(n) peak_y(n-i) M24 (ou M42) </pre>
---	--

3.4.2 Reconnaissance des Dodelinements de la Tête

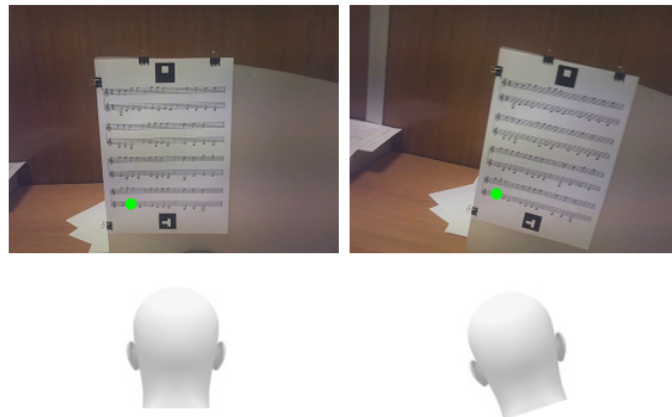


FIGURE 3.5 : Exemple de la variation du point du regard pendant le dodelinement de la tête.

Les dodelinements de la tête, (M56 et M65) par contre, ne peuvent pas être détectés par la variation des coordonnées du point du regard sur l'image de la scène, car celle-ci est liée avec la direction du regard lorsque la tête de l'utilisateur est à la position naturelle (sans effectuer des mouvements de la tête). Il peut y avoir des variations importantes sur les coordonnées x et les coordonnées y en même temps si le regard est fixé sur le coin de l'image de la scène, comme illustré dans la Figure 3.5. Mais si le regard est fixé sur un point situé près du centre de l'image la scène, les coordonnées peuvent être presque stables. Par conséquent, au lieu de calculer les dérivées des coordonnées du point du regard sur l'image de la scène, nous détectons les dodelinements de la tête par le traitement de l'image de la scène. Avec la matrice d'homographie calculée, nous pouvons obtenir facilement l'angle de rotation en 3 dimensions pour chaque image à l'aide des fonctions de la librairie openCV. Nous utilisons 2 seuils pour identifier les dodelinements de la tête à partir d'une série d'angles de rotation : le premier est sur la dérivée de l'angle de rotation z (axe perpendiculaire au plan de l'image de la scène), tandis que le second concerne la valeur absolue maximale de la différence entre l'angle de rotation z pendant le dodelinement de la tête et l'angle de rotation z avant ce mouvement. Nous utilisons simultanément ces deux limites afin d'éliminer les variations infimes de l'angle de rotation z qui peuvent être causées par la déviation du traitement

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

d'image et afin d'exclure le cas dans lequel la tête de l'utilisateur reste dans une position avec un angle de rotation significatif pour un but inconnu (par exemple, la tête de l'utilisateur peut être inclinée vers la gauche lorsqu'il joue du violon). Ces seuils ont été choisis par des tests préliminaires réalisés avec nos collègues du laboratoire, et la contrainte sur la durée est la même que celle pour les mouvements horizontaux et verticaux de la tête (10 trames traitées).

L'algorithme de reconnaissance des dodelinements de la tête à regard fixe est présenté ci-dessous. S'il existe un pic positif et un pic négatif sur la dérivée de l'angle de rotation z qui dépasse 1,9 (THRESHOLD_ROTATION) dans un délai de 10 frames traitées (THRESHOLD_DURATION), et que la valeur absolue maximale de la différence entre l'angle de rotation z avant le dodelinement de la tête et pendant ce mouvement est supérieure à 10 degrés (THRESHOLD_DIFFERENCE), nous admettons qu'il existe un dodelinement de la tête volontaire.

Algorithme 2 : Reconnaissance des dodelinements de la tête à regard fixe

```
1 def reconnaissance des dodelinements de la tête à regard fixe (numéro du frame traitée n)
2 begin
3   calculer le numéro de la région correspondante  $\rightarrow$  nRegion(n)
4   calculer l'angle de rotation  $z \rightarrow rZ(n)$ 
5   calculer la dérivée de l'angle de rotation  $z \rightarrow r'Z(n)$ 
6   potentialHeadTilt = FALSE
7   if  $r'Z(n) > \text{THRESHOLD\_ROTATION}$  (ou  $< -\text{THRESHOLD\_ROTATION}$ )
8     begin
9       peak_z(n) = 1 (ou -1)
10      for  $i = 0$  to THRESHOLD_DURATION
11        if peak_z(n-i) = -1 (or 1)
12          for  $j = 0$  to  $i$ 
13            if  $|rZ(n-j) - rZ(n-\text{THRESHOLD\_DURATION})| > \text{THRESHOLD\_DIFFERENCE}$ 
14              potentialHeadTilt = TRUE
15      end
16      if potentialHeadTilt = TRUE
17        for  $i = 1$  to THRESHOLD_DURATION
18          if nRegion(n-i)  $\neq$  nRegion(n)
19            return 0
20        return M56 (ou M65) détecté
21      return 0
22 end
```

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

3.4.3 Algorithme Global

Comme les dodelinements de la tête peuvent également entraîner des variations sur les coordonnées du point du regard sur l'image de la scène, nous identifions d'abord les dodelinements de la tête. Les variations des coordonnées du point du regard sont examinées pour détecter les mouvements horizontaux et verticaux de la tête uniquement s'il n'y a pas de dodelinement de la tête en cours d'après la variation de l'angle de rotation z . L'algorithme global est présenté ci-dessous.

Algorithme 3 : Algorithme global

```
1 begin
2   for chaque trame traitée (numéro n)
3     begin
4       run l'algorithme de la reconnaissance des dodelinements de la tête à regard fixe (n)
5       for i = 0 to THRESHOLD_DURATION
6         if peak_z(n-i) = -1 (ou 1)
7           for j = 0 to i
8             if |rZ(n-j) - rZ(n-THRESHOLD_DURATION)| > THRESHOLD_DIFFERENCE/2
9               goto trame n+1
10      run l'algorithme de la reconnaissance des mouvement horizontaux et verticaux de
                                     la tête à regard fixe (n)
11   end
12 end
```

3.5 Conclusion du Chapitre

Dans ce chapitre, nous avons comparé les solutions envisagées présentées dans l'état de l'art pour régler le problème de l'absence de méthode pratique pour déclencher des commandes, et nous avons mis en évidence les avantages de coupler les mouvements volontaires de la tête à regard fixe avec l'eye-tracking. Ensuite, nous avons détaillé nos algorithmes, permettant de reconnaître, en temps-réel, 6 gestes réalisés avec le regard fixe, en utilisant uniquement les images de la scène avec le point du regard superposé, fournies par l'eye-tracker porté sur la tête équipé d'une caméra frontale. Dans le chapitre suivant, nous présentons un scénario d'application conçu et développé, dans lequel la performance de nos algorithmes et l'acceptation

Chapitre 3. Déclenchement des Commandes par les Mouvements de la Tête à Regard Fixe

du déclenchement des commandes par ces mouvements sont évaluées.

EyeMusic : Apprentissage Interactif de la Musique

Le premier scénario que nous avons réalisé est lié à l'apprentissage de la musique et plus particulièrement à l'apprentissage d'un instrument de musique. Cette application est destinée à aider les débutants de la musique à interagir avec la partition de manière rapide et pratique en gardant les deux mains libres pour jouer l'instrument de musique [52, 51].

4.1 Choix du Scénario

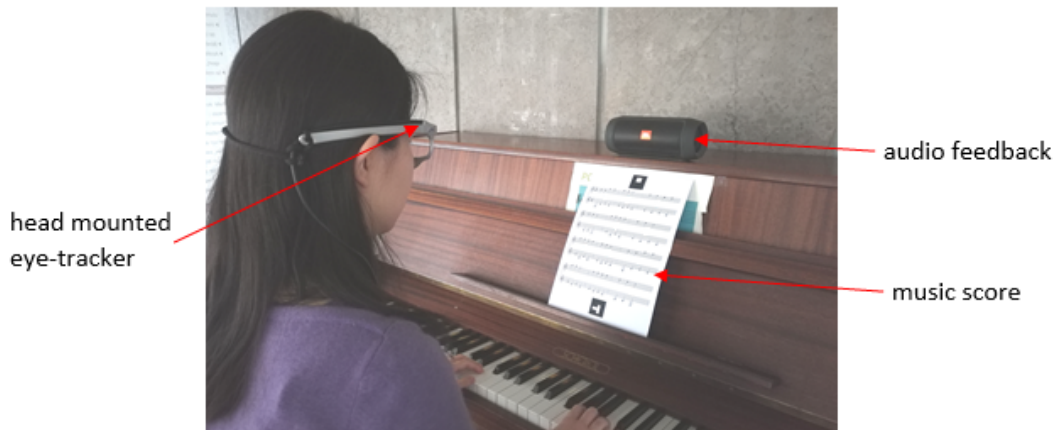


FIGURE 4.1 : Configuration typique d'EyeMusic pour apprendre à jouer du piano.

L'apprentissage de la musique procure de nombreux bénéfices, tant intellectuel que personnel et culturel [63]. La lecture de la partition est incontournable pour les musiciens et nécessite un temps certain d'apprentissage. Apprendre à lire les partitions n'est pas une mission simple pour les débutants de la musique, cela est encore

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

beaucoup plus compliqué qu'apprendre une nouvelle langue, puisque la langue est unidimensionnelle alors que la musique est bidimensionnelle (la hauteur et la longueur des notes). Du coup la création d'un système d'assistance pour l'apprentissage de la musique peut s'avérer très significatif.

Normalement, lorsqu'un débutant de la musique apprend à jouer une mélodie sur un instrument de musique, il a souvent besoin d'imiter l'exécution d'un autre musicien comme il ne maîtrise pas très bien la lecture de la musique. Afin d'entendre la mélodie pour une mesure spécifique, il lui faut mettre son instrument de musique de côté, ouvrir le fichier audio correspondant et puis chercher la bonne mesure. Ce processus n'est pas pratique pour le musicien débutant et consomme du temps. Le système EyeMusic que l'on réalise dans ce projet est un système d'apprentissage de la musique capable de jouer les notes d'une mesure d'une partition musicale que l'utilisateur n'arrive pas à comprendre, à l'aide d'un eye-tracker. Lorsque l'utilisateur rencontre des difficultés sur la lecture des partitions, par un simple mouvement de la tête à regard fixe, il pourra déclencher le feedback audio de la mesure qu'il fixe du regard sans avoir à libérer ses mains occupées par son instrument de musique (flute, guitare, piano, etc.). La Figure 4.1 montre un exemple en lien avec le fait d'apprendre à jouer du piano à l'aide de EyeMusic.

Le processus de la lecture de la partition se fait par des fixations et des saccades oculaires. Si la partition est à portée simple, il y a généralement une série de saccades horizontales suivies d'un mouvement vertical pour passer à la rangée suivante, et puis à nouveau des saccades horizontales. Lorsque la partition est à double portée, les musiciens ont tendance à utiliser des mouvements verticaux alternativement, de la première portée à l'autre portée, parallèlement aux saccades horizontales [86]. Les mouvements naturels de la tête lors de la lecture de la partition sont toujours accompagnés des variations de la direction du regard. Donc, dans cette application, on a choisi de déclencher des commandes par les mouvements volontaires de la tête avec le regard fixé qui sont possibles grâce au réflexe vestibulo-oculaire des yeux. Par rapport aux autres modalités pour le déclenchement des commandes, les mouvements de la tête à regard fixe possèdent plusieurs avantages évidents dans ce contexte :

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

1. Les mouvements de la tête à regard fixe peuvent être détectés par la variation de l'image de la scène et le changement du point du regard : aucun équipement supplémentaire (à l'exception d'un eye-tracker porté sur la tête équipé d'une caméra de la scène) n'est requis dans ce système (*cf.* Chapitre 3).
2. Les mouvements de la tête à regard fixe peuvent être effectués rapidement, sans libérer les mains.
3. Pendant les mouvements de la tête à regard fixe, le point du regard de l'utilisateur ne change pas, donc ils ne vont pas perturber le processus de lecture de l'utilisateur.
4. Par rapport aux mouvements naturels de la tête qu'on utilise tout le temps, les mouvements de la tête à regard fixe ne sont pas beaucoup utilisés pendant le processus de lecture, bien qu'il existe des gens qui ont l'habitude de battre le tempo par le hochement de la tête. On peut ainsi éviter le problème du touché de Midas.

4.2 Réalisation

4.2.1 Reconnaissance de la Partition et des Notes

Afin de reconnaître les notes regardées par l'utilisateur, le moyen le plus direct consiste à appliquer la reconnaissance optique de la musique (OMR) [71, 87] à l'image de la scène capturée. Toutefois, comme la partition de la musique est beaucoup plus compliquée que le langage écrit, les performances des algorithmes OMR utilisés dans les logiciels commerciaux et les recherches actuelles ne sont pas satisfaisantes. Alternativement, nous avons décidé de lier chaque partition physique avec sa version numérique (fichier MIDI) à l'aide de marqueurs graphiques différents et d'utiliser la correspondance entre l'emplacement sur la page et la position temporelle dans le fichier MIDI pour obtenir les notes correspondantes dans la mesure désirée. Dans ce prototype, nous avons choisi d'utiliser des marqueurs d'ArUco [32] qui sont assez nombreux et facilement détectables à l'aide des fonctions de la

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

librairie de traitement d'images openCV. Un exemple de la partition musicale où nous avons superposé des marqueurs ArUco est présenté à la Figure 4.2a. Dans notre application, 2 marqueurs sont attachés sur chaque page de la partition (1 en haut, 1 en bas) pour assurer qu'il y a au moins un marqueur visible sur l'image de la scène lorsque l'utilisateur lit la partition. En effet, à cause de la limite du champ de vision de la caméra de la scène, une partie de la partition peut être en dehors de l'image de la scène, comme illustré dans la Figure 4.2b. Chaque ensemble de marqueurs est utilisé pour identifier trois éléments différentes : la partition musicale lue par l'utilisateur, quelle page de la partition en cours est lue, et la position relative de chaque mesure sur cette page.

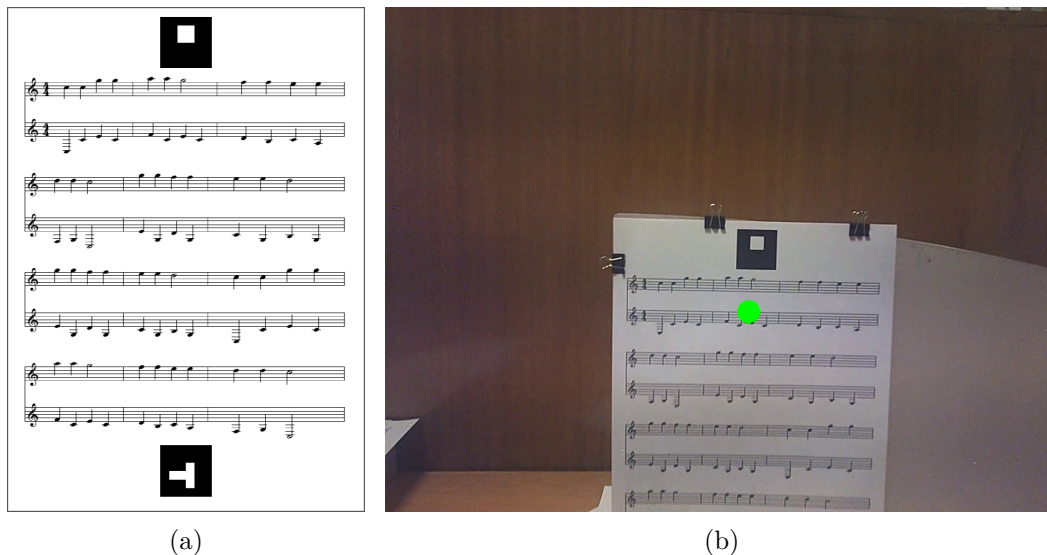


FIGURE 4.2 : Exemple de la partition et de sa vue sur l'image de scène : (a) exemple de la partition avec 2 marqueurs ArUco, (b) exemple de l'image de la scène avec le point du regard superposé alors qu'une partie de la partition se situe en dehors du champs de vision de la caméra de la scène.

4.2.2 Calcul de la Position Relative du Regard par rapport à la Partition

La position du regard que nous obtenons via l'API (application programming interface) de l'eye-tracker est donnée dans le repère des coordonnées de l'image de la scène. Par contre, ce qui est utile dans notre application est la position relative

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

du regard par rapport à la partition. Dans ce cas, afin de savoir quelle mesure de la partition intéresse l'utilisateur, nous calculons la matrice d'homographie en utilisant les coordonnées des marqueurs ArUco détectés. Puis, à l'aide de cette matrice d'homographie, nous sommes en mesure de transformer les coordonnées du regard en pixels sur l'image de la scène en coordonnées relatives du regard par rapport à la feuille (la partition).

La direction du regard obtenue via l'API fournie par SMI a un taux d'échantillonnage de 30Hz, mais la fréquence de la caméra de la scène est de 24 trames par seconde. Nous devons trouver une solution pour obtenir les informations du regard correspondant à chaque frame. Comme le logiciel de l'eye-tracker de SMI ajoute automatiquement un point vert sur chaque image de la scène (comme celui sur la partition dans la Figure 4.2b), nous avons décidé de détecter le point vert pour obtenir les coordonnées du regard sur chaque image de la scène. Ceci est facilement réalisable à l'aide de fonctions de la librairie de traitement d'images. Contraint par la puissance de calcul de l'ordinateur et la performance de nos algorithmes, les coordonnées relatives du regard sont calculées toutes les 2 trames : ce qui implique que la fréquence du traitement d'images est de 12 trames par seconde.

4.2.3 Détection des Mouvements de la Tête à Regard Fixe

Les algorithmes de détection des mouvements de la tête à regard fixe que l'on a développés dans cette application sont présentés dans la Section 3.3. Pour la détection des mouvements verticaux/horizontaux, le rapport entre la dérivée des coordonnées relatives du point du regard par rapport à la partition et la dérivée des coordonnées du point du regard sur l'image de la scène (`THRESHOLD_RATIO`) est fixé à 3,0 (*cf.* Algorithme 1), ce qui équivaut au rapport entre la variation des coordonnées relatives du point du regard et la variation des coordonnées du point du regard lorsque la distance entre la partition et l'utilisateur est à 35 cm (distance minimale de lecture). Normalement, la distance entre les yeux de l'utilisateur et la partition doit être supérieure à 35 cm, donc ce rapport doit être supérieur à 3,0 pour les purs mouvements oculaires. Lors des mouvements horizontaux et verticaux de la tête avec le regard fixe (cas 1), ce rapport doit, soit ne pas exister (à cause

du manque de netteté de l'image de la scène), soit être inférieur à 3,0 (car le regard reste à la même position). Ceci a été vérifié par des tests préliminaires avec nos collègues de laboratoire. Par conséquent, ce seuil peut être utilisé pour exclure les mouvements oculaires purs naturels sans ignorer les mouvements de la tête à regard fixe désirés.

4.3 Prototype Expérimental

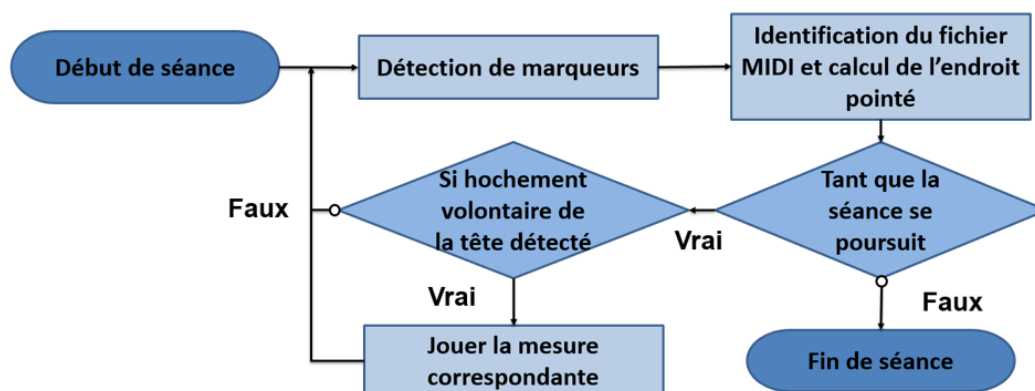


FIGURE 4.3 : Vue d'ensemble de l'algorithme principal d'EyeMusic.

La Figure 4.3 montre l'algorithme principal de notre application. Le prototype que nous avons développé pour les tests d'utilisabilité est capable de détecter les 6 mouvements (M13, M31, M24, M42, M56, M65) et 6 commandes différentes leur sont attribuées. À l'aide de ce prototype, l'utilisateur est capable de jouer une seule mesure, de jouer à partir d'une mesure spécifique jusqu'à la fin de la partition, de changer de volume, de s'arrêter et de reprendre. La correspondance entre les mouvements et les commandes est illustrée dans la Figure 4.4. Cette correspondance a été pensée pour que les mouvements apparaissent naturels à l'utilisateur. Les inclinaisons de la tête sont utilisées pour modifier le volume car ces mouvements sont moins naturels et plus fatiguants que les autres et sont donc affectés à des commandes secondaires. En outre, ils correspondent bien à la métaphore du bouton rotationnel du volume : la rotation dans le sens horaire pour augmenter le volume et la rotation dans le sens antihoraire pour baisser le volume. Les hochements verticaux sont utilisés pour déclencher les commandes principales car ils sont faciles à exécuter.

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

L'arrêt et la reprise sont affectés aux mouvements qui suivent le même axe (les deux hochements horizontaux) car ils sont complètement opposés. En plus le hochement à droite est utilisé pour relancer le fichier audio car cela correspond bien à la direction de la flèche affichée sur le bouton de reprise.

Nous avons également préparé une version simplifiée de notre prototype, qui ne possède qu'une seule commande : jouer la seule mesure spécifique que l'utilisateur souhaite entendre. Dans cette configuration, les 6 mouvements peuvent être utilisés pour déclencher cette commande. Cette version simplifiée est principalement utilisée dans la procédure d'apprentissage du test afin d'aider les participants ; elle est également utilisée pour évaluer la performance de la détection des mouvements de la tête à regard fixe de notre application.

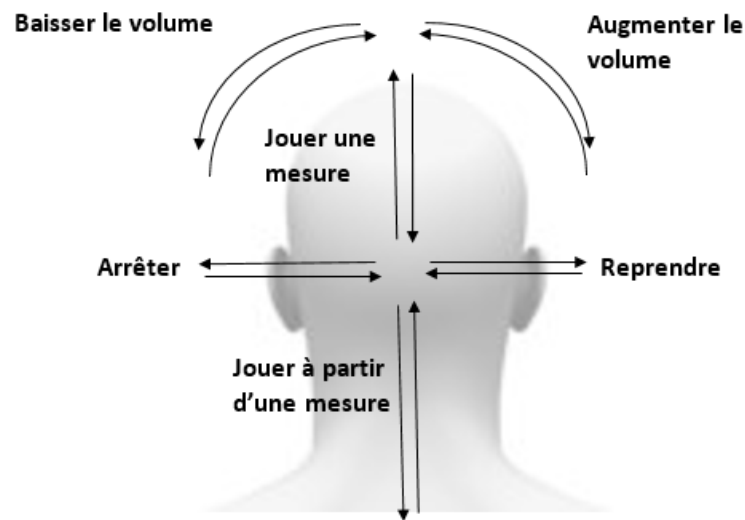


FIGURE 4.4 : Diagramme schématisé de la correspondance entre les mouvements et les commandes.

4.4 Test d'Utilisabilité

Nos expériences sont divisées en 2 parties. La première partie est la procédure d'apprentissage, durant laquelle les participants doivent apprendre les 6 mouvements de la tête à regard fixe différents à l'aide de la version simplifiée du prototype. Les données expérimentales capturées durant cette procédure sont utilisées

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

pour calculer et comparer le taux de réussite de chaque mouvement de la tête et ainsi déterminer quels sont les mouvements les plus appropriés pour déclencher des commandes. La deuxième partie est la procédure opérationnelle : nous demandons aux participants de manipuler la version complète de notre prototype afin de déterminer si les utilisateurs sont capables de mémoriser rapidement la correspondance entre les commandes et les mouvements de la tête, et de diriger un système qui utilise plusieurs mouvements simultanément.

4.4.1 Configuration Expérimentale

Le dispositif que nous utilisons est l'eye-tracker porté sur la tête sous forme d'une paire de lunettes équipé d'une caméra de scène (produit par SMI) que nous avons présenté dans la Section 1.3. La partition que nous utilisons pour les tests est fixée sur un tableau perpendiculaire au bord de la table. Les participants portant l'eye-tracker sont assis devant le tableau. Ils étaient libres de se déplacer ou de modifier la distance entre le tableau et la chaise pendant l'expérience. La Figure 4.5 montre la configuration de notre expérience.

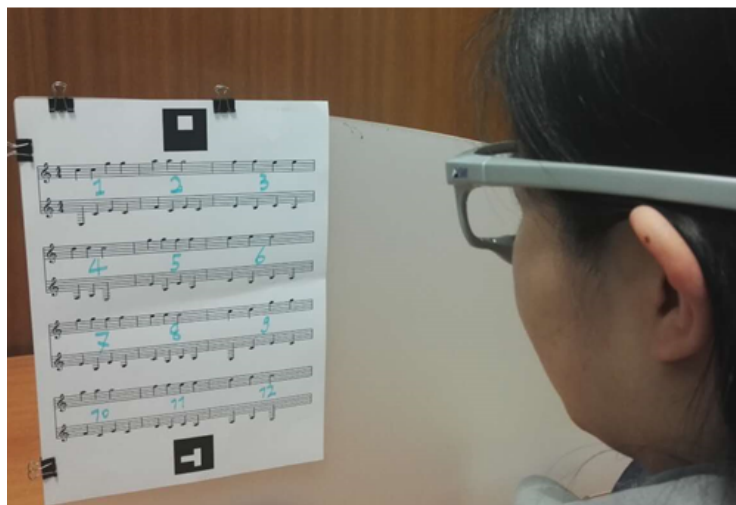


FIGURE 4.5 : Configuration expérimentale.

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

4.4.2 Procédure

L'expérience dure entre 40 et 90 minutes pour chaque participant. La durée varie d'une personne à l'autre, car certaines personnes ont besoin de plus de temps pour s'exercer et pour trouver la vitesse attendue pour chaque mouvement. Les participants ont été autorisés à s'arrêter au cours de l'expérience car celle-ci prend un temps certain.

4.4.2.1 Ouverture

Trois étapes composent le début du test :

- Accueillir le participant, lui montrer la présentation générale d'EyeMusic pré-écrite (voir l'Annexe A.1).
- Lui demander de signer le formulaire de consentement (voir l'Annexe A.2).
- Vérifier si l'eye-tracker utilisé est capable de capturer la direction du regard du participant. Si l'eye-tracker n'est pas capable de détecter les mouvements des yeux du participant, il s'agit de le remercier et d'interrompre le test.

4.4.2.2 Phase d'apprentissage

Nous utilisons dans cette phase la version simplifiée de notre prototype qui joue la mesure d'intérêt si l'un des 6 mouvements de la tête à regard fixe est détecté. Nous testons un par un les 6 mouvements de la tête. Pour chaque mouvement, nous présentons au participant ce mouvement au début, lui laissons plusieurs minutes pour s'exercer et se familiariser avec celui-ci. Ensuite, nous demandons au participant de jouer 3 mesures spécifiques par ce mouvement. Si la musique n'est pas jouée correctement, nous demandons au participant de réessayer sauf s'il a échoué 5 fois de suite sur la même mesure. Pour simplifier la tâche pour les participants qui ne savent pas lire la musique, le numéro correspondant de la mesure est écrit au milieu de chaque mesure, comme illustré dans la Figure 4.5. Les 3 mesures sont choisies soigneusement pour qu'elles soient réparties sur la partition, c'est-à-dire que les 3 mesures sont toujours réparties dans 3 lignes et 3 colonnes différentes. Le numéro correspondant à chaque mesure est présenté un par un par le modérateur et les

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

participants ne connaissent pas le numéro de la mesure pour l'essai suivant jusqu'à la fin de l'essai en cours. Il y a au total 12 séquences différentes (2 groupes de 6). Le premier groupe contient 1, 5, 9; 3, 8, 10; 2, 7, 12; 6, 11, 1; 7, 5, 3 et 11, 9, 4. Le deuxième groupe contient 12, 4, 2; 8, 10, 3; 2, 4, 12; 1, 6, 8; 9, 5, 10 et 11, 7, 6. Les 2 groupes sont utilisés alternativement, et les 6 séquences de chaque groupe sont affectées aux 6 mouvements de la tête différents en rotation afin de maintenir une composition de mesures équilibrée pour chaque mouvement.

4.4.2.3 Phase Opérationnelle

La version complète du prototype possédant 6 commandes différentes est utilisée dans cette phase. Nous montrons le diagramme schématique de la correspondance entre les mouvements et les commandes (Figure 4.4) au participant et lui laissons 1 minute pour mémoriser cette correspondance. Ensuite, nous lui demandons de déclencher plusieurs commandes sans consulter le diagramme schématique. La série de commandes que nous utilisons dans ce test est la suivante :

- jouer à partir de la mesure 3,
- arrêter,
- reprendre,
- jouer à partir de la mesure 2,
- baisser le volume,
- augmenter le volume,
- jouer la mesure 1,
- jouer à partir de la mesure 7,
- arrêter,
- reprendre,
- jouer la mesure 5,
- jouer à partir de la mesure 4,
- baisser le volume,
- augmenter le volume et
- jouer la mesure 6.

Cela inclut 4 fois l'action de jouer à partir d'une mesure, 2 fois d'arrêter, 2

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

fois de reprendre, 2 fois de 'baisser le volume', 2 fois d''augmenter le volume' et 3 fois de jouer une seule mesure. La commande 'arrêter', la commande 'reprendre' et les changements de volume sont toujours déclenchés après la commande 'jouer à partir de la mesure', tandis que la commande 'reprendre' est déclenchée lorsque la lecture avait été arrêtée auparavant. Comme la commande 'arrêter', la commande 'reprendre' et les changements de volume sont identiques pour toutes les mesures ; ils peuvent être déclenchés en fixant le regard sur n'importe quelle mesure. Nous demandons aux participants de réessayer si la commande n'a pas été déclenchée successivement sauf s'ils ont échoué 5 fois de suite pour la même commande.

4.4.2.4 Clôture

Il s'agit de demander au participant de compléter le questionnaire post-test (voir l'Annexe A.3) et le System Usability Scale (voir l'Annexe A.4).

4.4.3 Informations Collectées

Lors du test d'utilisabilité, nous avons collecté beaucoup de données en tant que traces pour des analyses qui suivent. Il y a principalement 3 catégories d'informations collectées¹.

La première partie est constituée des logs générés par nos algorithmes. Pour chaque image de la scène traitée, les données brutes comprenant les coordonnées du point du regard sur l'image de la scène détectées, les coordonnées relatives du point du regard par rapport à la partition, les dérivées des coordonnées du point du regard, les angles de rotation, etc., sont conservés dans un fichier log. Ainsi, en cas de mouvement non détecté, par l'analyse de ces fichiers de log, nous pouvons comprendre quelle exigence de nos algorithmes n'a pas été satisfaite en analysant ces traces ultérieurement. En outre, les images originales de la scène sont également conservées afin de vérifier si le point du regard capturé est bien fixé sur la mesure désirée pendant les mouvements de la tête.

1. Certaines des données collectées (la fiche de recueil, le SUS, le questionnaire, les logs générés et les images traitées pour la phase d'apprentissage d'un des participants) sont disponibles *online* à l'adresse suivante : <https://github.com/EyetrackingHCI/EyeMusic>

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

Nous prenons également des notes pendant le test à l'aide d'une fiche de recueil (voir l'Annexe A.5). Pour la procédure d'apprentissage, nous enregistrons le numéro de la mesure qui a été jouée réellement pour chaque mouvement de la tête à regard fixe effectué. S'il n'y a pas eu de mesure jouée après un mouvement volontaire effectué, nous notons un 0. Dans ce cas-là, nous sommes capable de compter le nombre de tentatives utilisées pour chaque essai. Pour chaque participant, nous avons collecté 18 métriques (3 mesures * 6 mouvements). Pour la procédure opérationnelle, nous enregistrons le mouvement effectué, le mouvement détecté et le numéro de la mesure (uniquement pour la commande "jouer une mesure" et la commande "jouer à partir d'une mesure", car pour les 4 autres commandes présentées dans la Section 4.3, le numéro de la mesure n'a pas d'importance). Ainsi, nous pouvons savoir si les participants ont pu mémoriser la correspondance entre les mouvements de tête et les commandes et si les mouvements effectués ont été correctement détectés. Pour chaque participant, nous avons collecté au maximum 15 métriques.

La troisième partie est constituée des informations collectées via le post-test questionnaire et le SUS. Dans le post-test questionnaire, nous avons demandé aux participants de répondre à 10 questions, y compris :

- genre,
- âge,
- niveau de connaissance de la notation musicale,
- niveau d'étude,
- leur préférence sur le mouvement de la tête à regard fixe,
- l'utilité de chacune des 6 commandes que nous avons appliquées dans le prototype,
- l'adéquation de la correspondance entre les commandes et les mouvements que nous utilisons,
- les autres commandes qu'ils souhaiteraient ajouter dans notre application,
- si cette application serait utile pour leur apprentissage de la musique et
- s'ils sont intéressés pour participer aux autres tests de notre application.

4.4.4 Participants

Nous avons testé un panel de 41 utilisateurs (30 hommes/11 femmes) âgés de 14 à 57 ans (médiane = 24) qui ont participé à la procédure d'apprentissage du test d'utilisabilité du prototype d'application EyeMusic. 34 d'entre eux (28 hommes/6 femmes) ont également participé à la procédure opérationnelle de l'expérience. 39 participants (30 hommes/9 femmes) ont rempli le post-test questionnaire et le SUS après les tests. La participation des utilisateurs pour les phases d'expérience différentes est illustrée dans l'Annexe A.6. Aucun d'entre eux n'avait essayé ce type d'eye-tracker auparavant. Deux autres volontaires étaient incapables de participer à notre expérience car l'équipement qu'on utilise ne pouvait pas suivre correctement les mouvements de leurs yeux car il n'y avait pas de support de nez qui leur convenait. L'un parmi les 41 participants a accompli l'expérience en portant ses propres lunettes de vision à l'intérieur de l'eye-tracker, car il était trop fatigué pour lui de se concentrer sur la partition sans ses lunettes. Comme les verres de ses lunettes sont minces et sans monture, l'eye-tracker a fonctionné normalement pendant l'expérience.

Pour les 39 participants qui avaient rempli le post-test questionnaire, nous avons collecté leur auto-évaluation du niveau de connaissance en lecture musicale. 13 participants affirment qu'ils n'ont jamais acquis aucune connaissance, 7 participants affirment qu'ils possèdent une connaissance élémentaire, 5 participants affirment qu'ils ont une connaissance moyenne, 10 participants affirment qu'ils ont une bonne connaissance et 4 participants affirment qu'ils possèdent une connaissance excellente de la notation musicale. Comme le nombre de participants pour 2 niveaux de connaissance de la notation musicale (moyen et excellent) sont faibles, nous avons décidé de les regrouper en 3 catégories équilibrées : 13 participants avec aucune connaissance de la notation musicale (le groupe cible de l'application), 12 participants possédant une connaissance élémentaire/moyenne de la notation musicale (qui peuvent également avoir besoin de cette application), et 14 participants avec une bonne/excellente connaissance de la notation musicale (qui n'ont certainement pas besoin de cette application). Lorsque l'audio de la mesure joué est er-

roné (suite à une mauvaise détection du numéro de mesure), les participants ayant une bonne/excellente connaissance de la notation musicale peuvent identifier cette erreur immédiatement et avec confiance, les participants ayant une connaissance élémentaire/ moyenne de la notation musicale sont capables de trouver l'erreur dans un délai court ou hésitent sur l'erreur, tandis que les participants qui n'ont jamais acquis aucune connaissance de la notation musicale sont incapables de trouver l'erreur par eux-mêmes. En ce qui concerne leur niveau d'étude, 1 participant est diplômé de collège, 1 participant est diplômé de lycée, 18 participants déclarent qu'ils ont obtenu une licence, 17 participants déclarent qu'ils possèdent un diplôme de master, et 2 participants affirment qu'ils ont obtenu un doctorat.

4.5 Résultats Expérimentaux et Discussion

4.5.1 Performance de l'Application

Nous évaluons la performance du prototype de l'application par les résultats expérimentaux de la procédure d'apprentissage de notre test d'utilisabilité. Cette évaluation est faite à l'aide de deux mesures principales : le taux de réussite et le nombre de tentatives nécessaires pour chaque geste/mouvement.

4.5.1.1 Résultats Bruts

Comme nous avons pris des notes pendant le test à l'aide d'une fiche de recueil, nous pouvons connaître le numéro de la mesure réellement jouée pour chaque mouvement de la tête à regard fixe effectué, et ainsi compter le nombre de tentatives utilisées pour chaque mouvement. Lors de la procédure d'apprentissage du test d'utilisabilité avec la version simplifiée de notre prototype, nous avons demandé à tous les participants de jouer 3 mesures différentes pour chaque mouvement de la tête à regard fixe testé. Si le participant ne réussissait pas, nous lui demandions de réessayer sauf s'il avait déjà essayé 5 fois sans succès sur la même mesure, ce qui signifiait que la procédure avait échoué. Ainsi, chaque essai peut être effectué 1, 2, 3, 4, 5 fois ou échoué. Il y a au total $41 * 3 = 123$ essais pour chaque mouvement. Le résultat est illustré à la Figure 4.6.

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

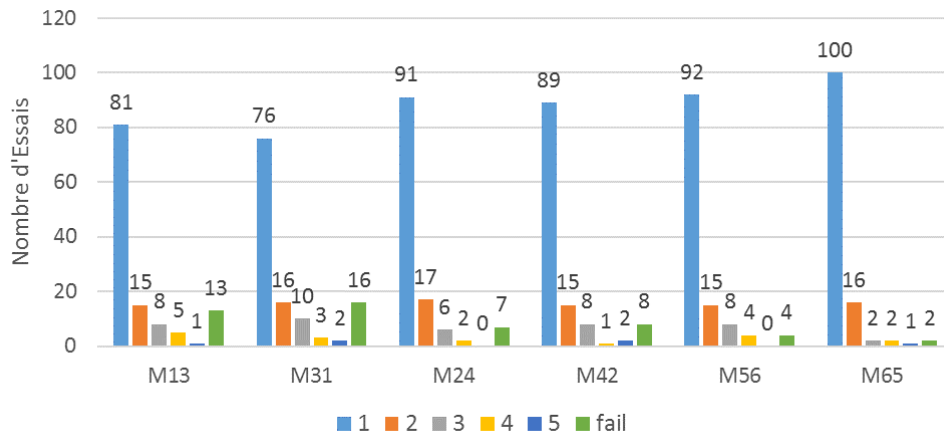


FIGURE 4.6 : Nombre d'essais réussis après 1, 2, 3, 4, 5 tentatives ou échoués pour chaque mouvement de la tête à regard fixe.

Le taux de réussite de l'application est calculé en faisant la moyenne du taux de réussite de chaque participant, tandis que le taux de réussite individuel est compté en faisant la moyenne de son taux de réussite pour chaque mouvement de la tête à regard fixe testé. Nous avons calculé le taux de réussite personnel de chaque mouvement de la tête à regard fixe en divisant le nombre d'essais réussis par le nombre de mouvements effectués au total. Par conséquent, le résultat n'est pas égal au pourcentage d'essais réussis par le premier mouvement effectué. Le taux de réussite calculé est de 74,1%, alors que l'écart type est de 16%. L'histogramme du taux de réussite est illustré à la Figure 4.7. 43,9% des participants ont accompli les tâches avec un taux de réussite supérieur à 80% ; 82,9% des participants ont accompli les tâches avec un taux de réussite supérieur à 60%.

4.5.1.2 Résultats Ajustés

Comme les 6 mouvements de la tête à regard fixe correspondent à la même commande (jouer la mesure) dans la version simplifiée de notre prototype, si un mouvement est détecté comme un autre, nous pouvons toujours entendre le retour audio. Ce problème se produit parfois et affecte le taux de réussite réel de notre application. Par conséquent, nous avons vérifié manuellement les traces capturées lors du test afin de détecter ce type d'erreur, et nous avons recalculé le taux de

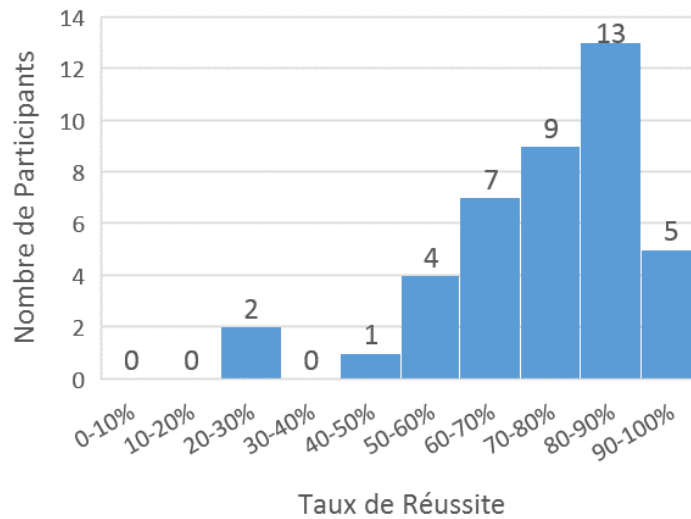


FIGURE 4.7 : Histogramme du taux de réussite pour jouer la bonne mesure.

réussite et le nombre de mouvements nécessaires pour chaque essai.

Il y a au total 47 essais (sur 738) dans lequel le retour audio est déclenché par un autre mouvement détecté : plus précisément, il y a 8 fois (sur 123) pour M13, 9 fois pour M31, 8 fois pour M24, 3 fois pour M42, 6 fois pour M56 et 13 fois pour M65. Le taux de réussite ajusté pour jouer la bonne mesure avec le bon mouvement détecté pour tous les participants est de 70,0% (Figure 4.8), l'écart type est de 20%. Comme nous avons effectué ces 47 essais comme s'ils avaient réussi, nous ne savons pas combien de tentatives nous avons vraiment besoin pour jouer la mesure avec le bon mouvement détecté. Ainsi, ils ne sont pas comptabilisés dans la Figure 4.9.

Selon le test du chi-carré de Pearson, il n'y a pas de différence significative entre les 6 mouvements ($p > 0,05$) lorsque nous les séparons en 6 catégories (essais accomplis 1 à 5 fois ou échoués). Cependant, si nous ne séparons que les essais réussis et les essais échoués, $p < 0,01$ selon le test du chi-carré de Pearson, du coup le nombre d'essais échoués pour M56 et M65 est nettement inférieur à celui des autres mouvements.

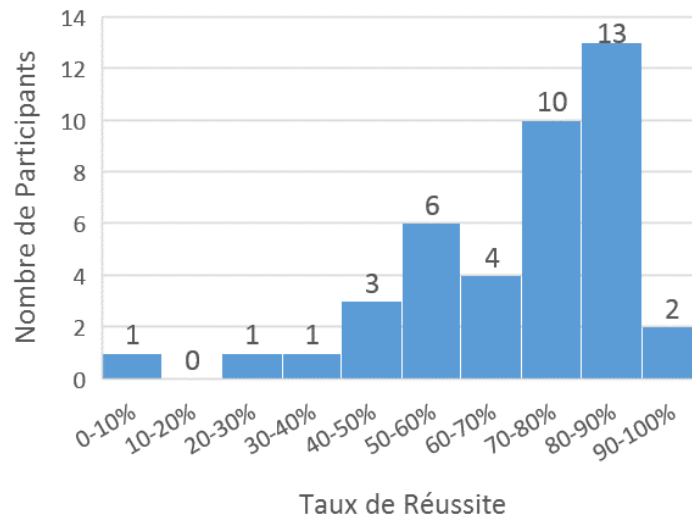


FIGURE 4.8 : Histogramme du taux de réussite pour jouer la bonne mesure avec le bon mouvement détecté.

4.5.1.3 Analyse des Mauvaises Détections

À l'exception du taux de réussite et du nombre de mouvements nécessaires pour chaque essai, nous avons compté et analysé la confusion entre les mouvements afin de déterminer s'il est préférable de combiner plusieurs mouvements pour éviter le problème de mauvaise détection. La matrice de confusion est illustrée dans le Tableau 4.1. Dans ce tableau, nous observons qu'une grande partie de mauvaises détections s'est produite entre des mouvements de la tête suivant le même axe, c'est-à-dire entre M13 et M31, entre M24 et M42, ou entre M56 et M65. Ces erreurs sont faciles à justifier : parfois les participants font un micro-mouvement dans la direction opposée, avant ou après le mouvement désiré, afin de laisser une plus grande amplitude pour le mouvement désiré ou pour compenser le mouvement qui a légèrement dépassé la position naturelle. Il y a en plus des mauvaises détections entre M56, M65 et les autres mouvements de la tête. Cela est dû au fait que les coordonnées du regard peuvent varier en même temps lorsque la tête s'incline à gauche ou à droite, et que les participants peuvent effectuer des mouvements de la tête horizontaux en même temps que les inclinaisons de la tête, car les inclinaisons pures de la tête sont plus fatigantes que les autres mouvements. Ce tableau nous

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

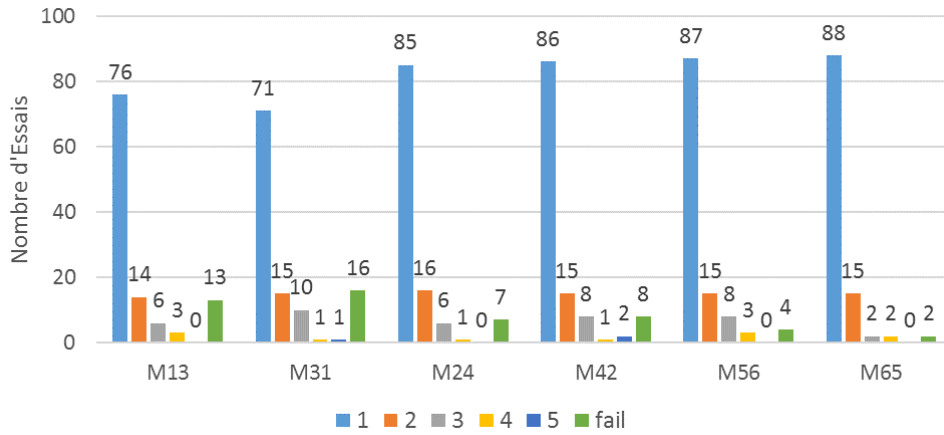


FIGURE 4.9 : Nombre d'essais réussis après 1, 2, 3, 4, 5 tentatives ou échoués pour chaque mouvement de la tête à regard fixe.

montre que le pourcentage de mauvaise détection pour les mouvements testés sont tous inférieurs à 11% et qu'il n'est donc pas nécessaire de combiner les mouvements.

TABLE 4.1 : Matrice de Confusion des 6 Mouvements.

	Mouvement Détecté					
	M13	M31	M24	M42	M56	M65
M13	92,5%	2,8%	3,7%	0%	0,9%	0%
M31	4,7%	91,6%	1,9%	1,9%	0%	0%
M24	0%	0%	93,1%	5,2%	1,7%	0%
M42	0%	0%	2,6%	97,4%	0%	0%
M56	1,7%	0%	0,8%	0,8%	95,0%	1,7%
M65	2,5%	3,3%	0%	0,8%	4,2%	89,2%

4.5.2 Performance des Algorithmes de Reconnaissance de Mouvement de la Tête

Si une erreur se produit lorsqu'on joue une mesure où la commande est déclenchée avec un autre mouvement de la tête, le problème peut provenir :

- soit de l'eye-tracker que nous avons utilisé dans l'expérience (informations sur le regard perdues, manque de précision), ou
- soit d'un mauvais jugement de la capacité des utilisateurs (il est difficile pour quelques participants de fixer leurs yeux sur la mesure pendant les mouvements ou d'effectuer correctement le mouvement attendu : combinaison de

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

- 2 mouvements, un mouvement supplémentaire au début ou à la fin) ou
- soit d'un problème de nos algorithmes de détection des mouvements de la tête à regard fixe.

En analysant manuellement les traces collectées, en particulier les images de scène traitées au cours des 555 défaillances, nous avons pu exclure les erreurs causées par les autres problèmes et nous concentrer sur les erreurs causées par nos algorithmes, c'est-à-dire les détections défaillantes des mouvements de la tête à regard fixe bien effectués par le participant et correctement captés par l'eye-tracker. Le pourcentage de mouvements de la tête à regard fixe correctement détectés est de 84,7% avec un écart type de 0,11 lorsque nous excluons toutes les sources d'erreur, à l'exception de celles de nos algorithmes de reconnaissance du mouvement de la tête (Figure 4.10). La Figure 4.11 montre que presque tous les essais ont été réalisés avec moins de 3 mouvements effectifs. Selon le test du chi-carré de Pearson, il n'y a pas de différence significative entre la performance de détection des six mouvements différents. On peut également constater que la différence entre le nombre d'essais échoués présenté à la Section 4.5.1.2 n'est pas due aux algorithmes que nous utilisons pour différents mouvements de la tête, car ils ne sont plus pris en compte. Cela peut être dû à la différence de degré de difficulté lorsqu'on effectue différents mouvements de la tête avec le regard fixé, ou bien l'eye-tracker est moins précis lorsque les yeux de l'utilisateur sont dans les coins.

En analysant l'origine des mouvements ineffectifs, nous avons constaté qu'il y a 6 participants au total qui n'ont pas réussi à effectuer des mouvements qui satisfont notre exigence pour un certain mouvements de tête :

- U11, U13, U25 et U44 n'ont pas réussi à effectuer M13,
- U11, U13, U25 et U35 n'ont pas réussi à effectuer M31,
- U26 et U44 n'ont pas réussi à effectuer M24,
- U13 n'a pas réussi à faire M42.

M56 et M65 sont réalisables pour tous les participants, au moins pour une fois. Cela montre que le problème de l'eye-tracker ou la capacité de fixation existe pour 15% des participants, et que la grande majorité des participants sont à l'aise avec tous les mouvements de la tête choisis pour cette application.

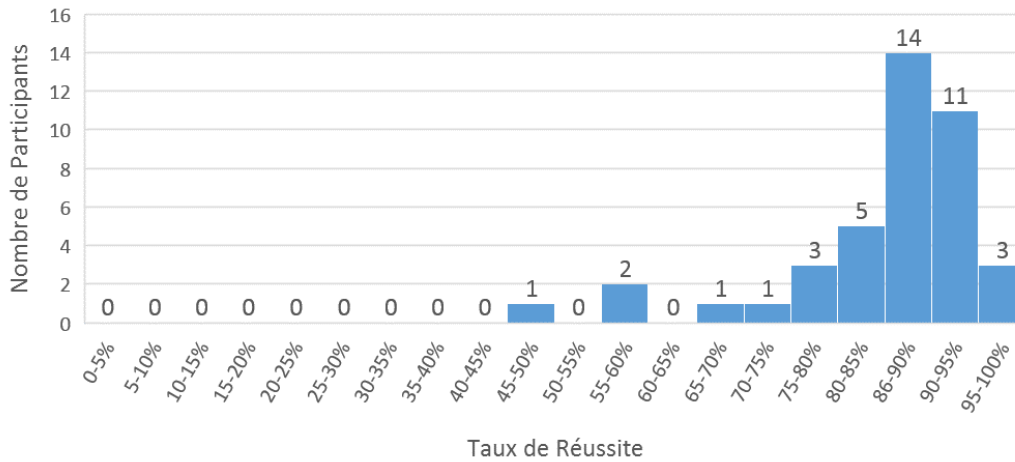


FIGURE 4.10 : Histogramme du taux de réussite des algorithmes de la reconnaissance des mouvements de la tête.

4.5.3 Performance des Participants avec Différents Niveaux de Connaissance de la Notation Musicale

Le taux de réussite ajusté de la procédure d'apprentissage pour les participants avec différents niveaux de connaissance de la notation musicale est illustré dans la Figure 4.12. On constate que le taux de réussite ajusté pour les participants avec une connaissance élémentaire/moyenne de la notation musicale est légèrement inférieur au taux de réussite ajusté moyen de tous les participants (70,0%), tandis que les taux de réussite ajustés des 2 autres catégories sont légèrement supérieurs au taux de réussite ajusté moyen de tous les participants. La distribution du taux de réussite ajusté est indiquée dans le Tableau 4.2. Nous observons que, bien que le nombre de participants et le taux de réussite ajusté de chaque catégorie soient différents, les répartitions sont similaires : il existe des participants avec un bon taux de réussite ajusté (supérieur à 80%) et des participants avec un taux de réussite ajusté faible (inférieur à 60%) pour chaque catégorie. De plus, les distributions pour ces 3 catégories sont normales selon le test de Lilliefors [64].

En effectuant un test d'analyse de la variance (ANOVA), nous constatons qu'il n'y a pas de différence significative entre le taux de réussite ajusté des participants dans différentes catégories ($p > 0,05$). Par conséquent, bien que les participants que

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

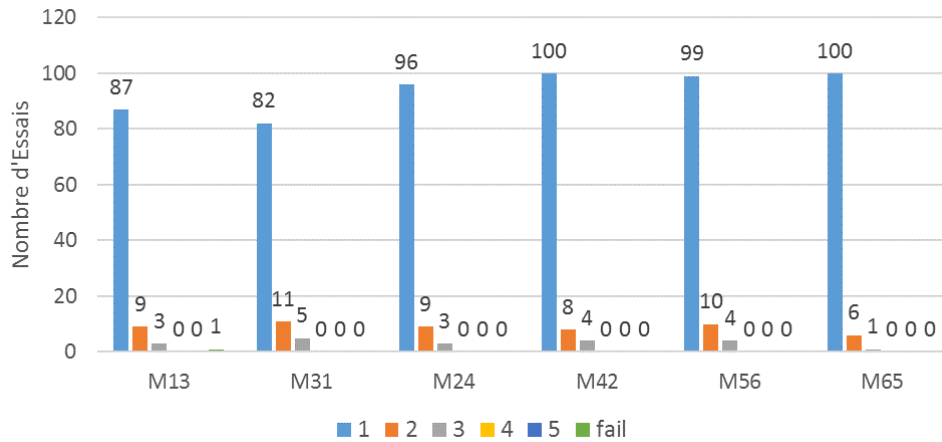


FIGURE 4.11 : Nombre d'essais réussis après 1, 2, 3, 4, 5 mouvements effectifs ou échoués pour chaque mouvement de la tête à regard fixe.

nous avons recrutés pour ce test d'utilisabilité ne soient pas tous des musiciens novices, cela n'affecte pas la performance de notre application. Cela est compréhensible car nous ne testons que l'utilisabilité de notre prototype, et aucune compétence spécifique au niveau de la connaissance de la notation musicale n'est requis.

TABLE 4.2 : Distribution du taux de réussite ajusté de la phase d'apprentissage pour les participants avec niveaux de connaissance de la notation musicale différents.

	aucun	élémentaire/moyen	bon/excellent
0%-10%	0	0	1
10%-20%	0	0	0
20%-30%	1	0	0
30%-40%	0	1	1
40%-50%	0	1	0
50%-60%	2	1	2
60%-70%	1	3	1
70%-80%	3	3	4
80%-90%	6	3	3
90%-100%	0	0	2
TOTAL	13	12	14

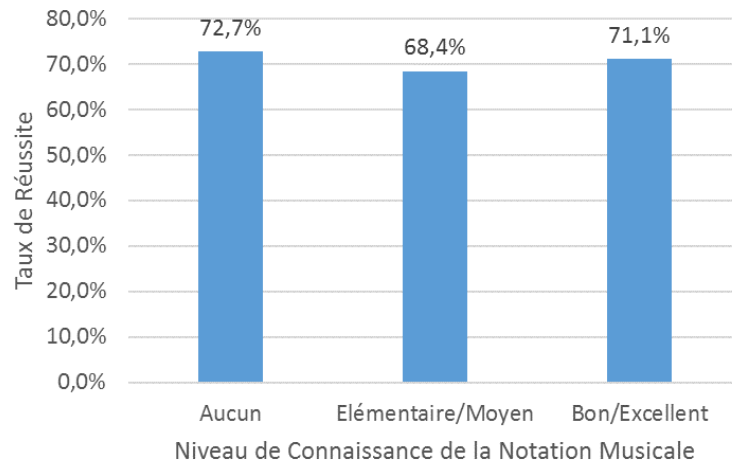


FIGURE 4.12 : Taux de réussite ajusté de la phase d'apprentissage pour les participants avec différents niveaux de connaissance de la notation musicale.

4.5.4 Capacité à Mémoriser Rapidement la Correspondance entre les Commandes et les Mouvements, et à Déclencher 6 Commandes Différentes par ces Mouvements

Dans le contexte de la phase opérationnelle (utilisant la version complète de notre prototype), nous avons demandé à chaque participant de déclencher 15 commandes au total : 4 fois pour 'jouer à partir d'une mesure', 2 fois pour 'arrêter', 2 fois pour 'reprendre', 2 fois pour 'augmenter le volume', 2 fois pour 'baisser le volume' et 3 fois pour 'jouer une mesure', sauf pour 3 participants qui n'ont pas exécuté la commande 'reprendre' car ils n'ont pas réussi à déclencher la commande 'arrêter' en premier.

Le taux de réussite moyen est de 69,3%, tandis que l'écart type est de 16%. Ce taux est légèrement inférieur au taux de réussite de jouer une mesure avec le bon mouvement détecté pour la procédure d'apprentissage (présenté dans la Section 4.5.1.2) bien qu'il soit beaucoup plus difficile d'utiliser les 6 mouvements en même temps que de les tester un par un. Comme la différence entre ces deux tâches est acceptable, il n'y a donc aucun problème pour les participants à déclencher 6 commandes différentes à l'aide des mouvements de la tête. La Figure 4.13 et la Figure 4.14 affichent les résultats de la phase opérationnelle du test d'utilisabilité

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

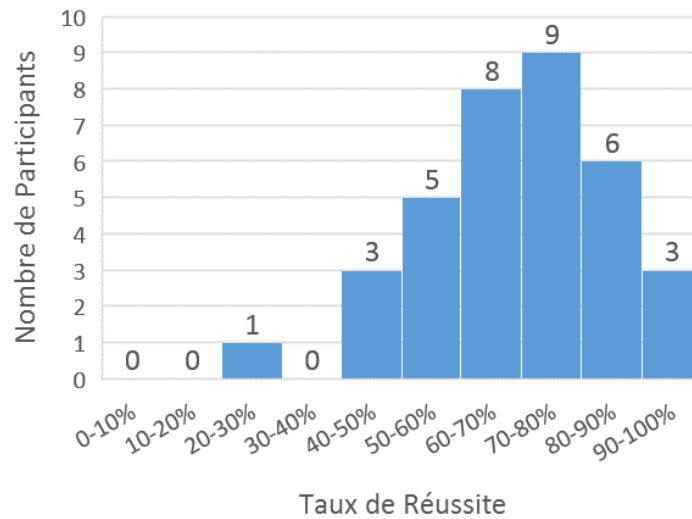


FIGURE 4.13 : Histogramme du taux de réussite pour la phase opérationnelle.

sur la version complète de notre prototype.

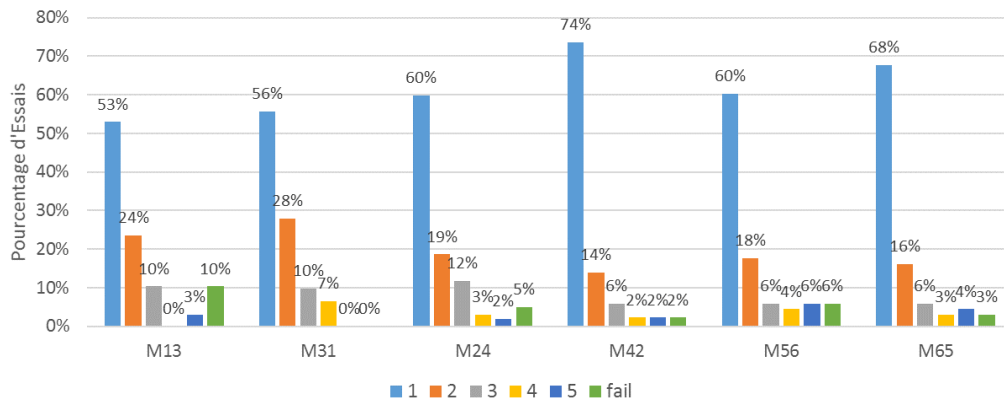


FIGURE 4.14 : Pourcentage d'essais réussis après 1, 2, 3, 4, 5 tentatives ou échoués pour chaque mouvement de la tête à regard fixe pour la phase opérationnelle.

Deux participants ont commis 2 erreurs au niveau de la correspondance entre les mouvements et les commandes, 10 participants ont commis 1 erreur, tandis que les autres 22 participants n'ont commis aucune erreur bien que nous ne leur ayons laissé qu'une minute pour mémoriser ces commandes. Il semble que, pour tous les participants, il est facile de mémoriser la correspondance entre les commandes et les mouvements avec peu d'effort. L'erreur la plus courante est la confusion entre 'jouer à partir d'une mesure' (M42) et 'jouer une mesure' (M24).

4.5.5 Erreurs des Algorithmes de Reconnaissance des Mouvements de la Tête et Solutions Possibles

Malgré les tests préliminaires que nous avons effectués avec nos collègues de laboratoire, plusieurs problèmes sont apparus au cours de ce test d'utilisabilité. Pour les erreurs causées par nos algorithmes de reconnaissance des mouvements de la tête lors de la procédure d'apprentissage du test d'utilisabilité, nous avons analysé le problème de chaque erreur et essayé de trouver des solutions possibles pour améliorer la performance de nos algorithmes.

Les 6 problèmes qui peuvent nous apporter des faux négatifs ou des faux positifs sont les suivants :

- Les coordonnées relatives du regard par rapport à la partition ne sont pas correctement calculées : il peut exister une différence significative entre la position réelle du regard et la position calculée du regard, en particulier pour la coordonnée y (cela a eu lieu 39 fois sur 1191 mouvements effectués).
- Parfois, l'angle de rotation n'est pas correctement calculé (cela a eu lieu 4 fois).
- Au cours des mouvements de la tête, les images capturées par la caméra de la scène peuvent ne pas être assez nettes : nous ne pouvons donc pas obtenir la position relative du regard, car les marqueurs ne sont pas détectables (cela a eu lieu 10 fois).
- Le seuil au niveau de la vitesse du mouvement défini dans les algorithmes ne convient pas à tous les participants, car ils effectuent des mouvements plus lents pendant la phase d'apprentissage (cela a eu lieu 32 fois, surtout pour les deux dodelinements de la tête).
- Les mouvements de la partie haute du corps (vers l'avant et vers l'arrière) avec le regard fixé sur le même point peuvent également entraîner une variation horizontale ou verticale sur les coordonnées du regard, et sont donc détectés comme l'un des mouvements de la tête à regard fixe que nous utilisons (cela a eu lieu pour 1 participant).
- Lorsque le participant est très proche de la partition, les micro-mouvements

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

et les mouvements oculaires rapides ayant provoqué des variations sur les coordonnées du regard ne peuvent pas être éliminés, car le changement au niveau des coordonnées relatives du regard est trop petit et reste dans la région de la même mesure (cela a eu lieu pour 2 participants).

Les 3 premiers problèmes sont principalement causés par le manque de netteté des images de la scène. Ceci peut être partiellement résolu en utilisant un capteur vidéo avec une fréquence d'image plus élevée qui est adapté pour capturer des traces lors des mouvements. Le quatrième problème peut être résolu en donnant à l'utilisateur plus de temps pour s'exercer : une fois que le participant se sent confiant pour effectuer des mouvements rapides de la tête avec le regard fixé, il sera plus facile pour lui d'interagir avec notre application. En outre, il est également possible de définir le seuil de durée comme une variable adaptative qui peut s'adapter à l'habitude de chaque utilisateur. Les 2 derniers problèmes sont liés à la distance entre l'utilisateur et la partition : ils peuvent être résolus en ajoutant des seuils liés avec cette distance, qui peut être approximativement calculée par la taille des marqueurs détectés.

4.5.6 Préférence de Mouvement

Il n'y a pas de différence significative au niveau du taux de succès de la détection des 6 mouvements de la tête à regard fixe différents, ni pour le pourcentage des essais réussis du premier coup. Au contraire, le nombre d'essais échoués pour chaque mouvement est différent. Comme montré par la Figure 4.9, le nombre d'essais ayant échoué pour M13 et M31 est supérieur à celui de M24 et M42, alors que M56 et M65 ont le moins d'essais échoués. L'analyse sur le nombre de participants qui ne peuvent pas effectuer des mouvements nous a donné un résultat similaire.

En ce qui concerne la fréquence d'utilisation au cours du processus de lecture, les mouvements verticaux de la tête à regard fixe (hochement de la tête) sont parfois utilisés pour battre le tempo. Cela peut nous apporter des faux positifs pour M24 ou pour M42, alors que les autres mouvements sont rarement utilisés lors de la lecture d'une partition.

Dans le questionnaire post-test, nous avons demandé aux participants de choisir

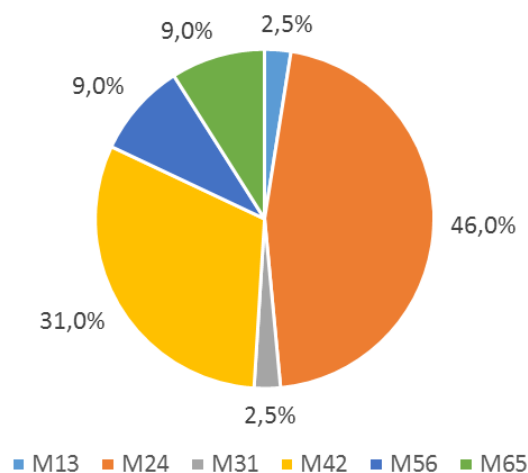


FIGURE 4.15 : Préférences du mouvement de la tête à regard fixe des participants.

leur mouvement préféré. Le résultat est représenté sur la Figure 4.15. Les mouvements les plus populaires sont les mouvements verticaux : M24 (sélectionnés par 46% des participants) et M42 (sélectionnés par 31% des participants), suivis par les dodelinements : M56 et M65 avec un taux de 9% pour chacun. Les mouvements horizontaux (M13 et M31) ont obtenu une préférence clairement plus faible, peut-être parce que les utilisateurs trouvent qu'il est difficile de fixer le regard pendant ces mouvements ou bien parce que parfois les gens peuvent confondre la gauche et la droite.

4.5.7 Résultat du SUS (*System Usability Scale*)

Nous avons demandé aux participants de remplir le formulaire standard du SUS après l'expérience. Le score moyen calculé pour les 39 participants qui avaient rempli ce formulaire est 77,2 avec un écart type de 11,5. Ce résultat est bon car il est supérieur au score moyen (70) de tous les produits testés. Toutefois, 9 participants ont noté notre application avec un score inférieur à 70, soit presque 25% des participants. L'histogramme est présenté à la Figure 4.16.

Nous avons analysé la relation entre le taux de réussite brut des participants et le score du SUS attribué à notre application : le résultat est présenté dans la Figure 4.17. Le rayon de chaque bulle correspond au nombre de participants dans chaque

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

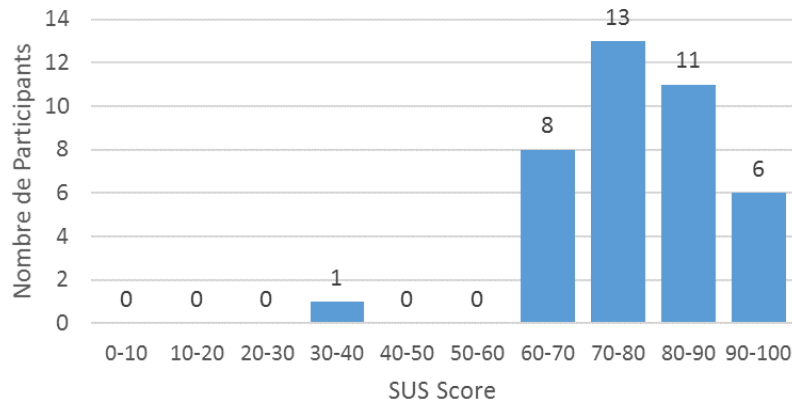


FIGURE 4.16 : Histogramme du score pour SUS.

catégorie : par exemple, la bulle bleue située dans le coin en haut à droite de la figure indique qu'il y a 2 participants dont le taux de réussite brut est compris entre 90% et 100% et qui ont noté notre application avec un score entre 90 et 100 dans la formule SUS. La couleur verte indique qu'il y a plus que 3 participants dans cette catégorie. Il est clair qu'il existe un lien fort entre le taux de réussite brut et le score du SUS : les participants ayant des taux de réussite brut plus élevés ont plus tendance à attribuer au dispositif un score élevé dans le SUS. De plus, les 6 participants qui n'ont pas réussi à faire des mouvements effectifs pour certains des mouvements de la tête ont attribué un score inférieur à 70 : cela est assez acceptable comme certaines commandes dans cette application ne sont pas utilisables par eux. Le taux de réussite qu'on applique dans cette analyse est le taux de réussite brut de la procédure d'apprentissage (présenté dans la Figure 4.7), parce que tous les participants ont participé à cette procédure de notre test, et qu'en plus, c'est le taux de réussite que les participants ressentent au cours de l'expérience.

Comme les déclarations positives et les déclarations négatives sont mélangées dans les 10 questions du SUS, on compte sur la contribution au score au lieu du score noté pour chaque question lors d'analyse des résultats expérimentaux. En comparant la contribution de chaque question au score du SUS dans le formulaire SUS, nous avons constaté que la première question ("Je pense que je vais utiliser ce service fréquemment") a obtenu la plus mauvaise contribution (2 en moyenne),

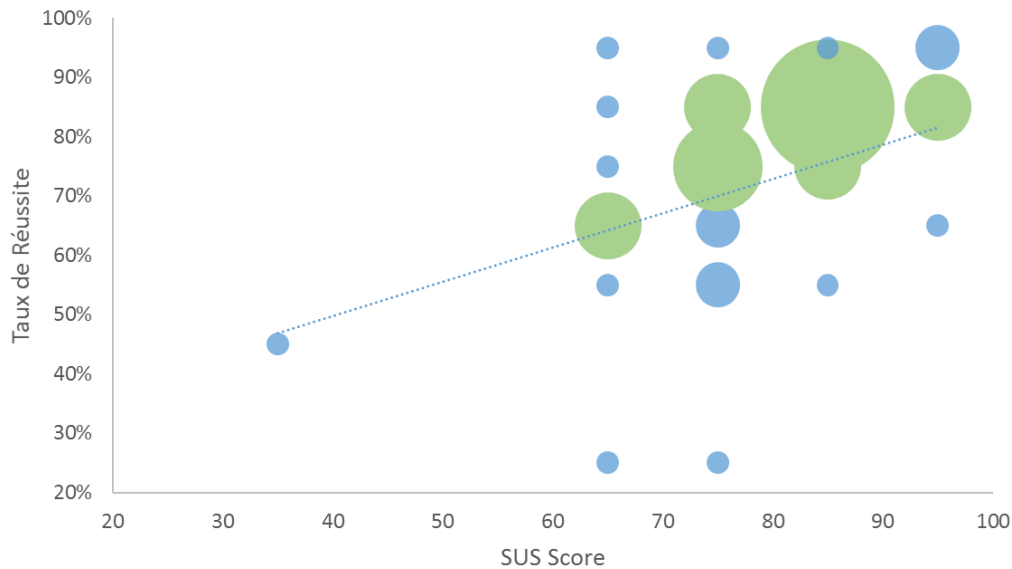


FIGURE 4.17 : Relation entre le score SUS et le taux de réussite brut.

alors que la contribution moyenne au score du SUS pour les autres questions est supérieure à 2,8. Cela est partiellement dû au fait que les participants que nous avons recrutés pour ce test ne sont pas tous des vrais musiciens novices : certains d'entre eux ont déjà une bonne connaissance de la notation musicale et n'ont donc pas besoin d'assistance pour apprendre une nouvelle mélodie, tandis que certains parmi les autres peuvent ne pas être intéressés du tout par l'apprentissage de la musique. En comparant la contribution notée par les participants avec différents niveaux de connaissance de la notation musicale, nous constatons que les 13 participants n'ayant aucune connaissance de la notation musicale nous ont fourni la meilleure contribution au score pour la première question. La contribution moyenne au score pour la première question de ces participants est 2,54, alors que 8 sur 12 des participants qui ont noté cette question avec une contribution supérieure à 3 appartiennent à cette catégorie. Par contre, pour les participants ayant au moins une connaissance élémentaire de la notation musicale, seulement 4 d'entre eux (sur 26) ont répondu à la première question avec une contribution au score supérieure à 3, et la contribution moyenne de cette question pour les participants dans ces 2 catégories fluctue entre 1,65 et 1,80. La Table 4.3 montre que bien que les participants

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

TABLE 4.3 : Taux de réussite brut et contribution au score du SUS de la première question pour les participants avec des niveaux différents de connaissance de la notation musicale.

Niveau de Connaissance de la Notation Musicale	Nombre de Participants	Taux de Réussite Brut	Contribution au Score pour la Première Question	Score du SUS
aucun	13	75,2%	2.54	80.2
élémentaire/ moyen	12	71,1%	1.67	75.4
bon/ excellent	14	76,3%	1.79	76.1
TOTAL	39	70,8%	2	77.2

avec une bonne/excellente connaissance de la notation musicale ont le meilleur taux de réussite brut pendant le procédure d'apprentissage de notre test d'utilisabilité, la contribution au score du SUS qu'ils ont noté pour la première question du SUS est même inférieure à la contribution moyenne. Au contraire, le groupe de participants avec aucune connaissance de la notation musicale ont un taux de réussite brut légèrement supérieur au taux de réussite moyen, mais la contribution au score qu'il nous ont attribué pour cette question est beaucoup plus haute que les autres groupes. En outre, le score SUS moyen pour les 13 participants n'ayant aucune connaissance de la notation musicale (groupe cible de l'application) est 80,2.

4.5.8 Utilité des Commandes

Dans le questionnaire post-test, nous avons demandé aux participants de noter l'utilité des 6 commandes que nous avons appliquées dans le prototype d'EyeMusic. Ils ont noté chaque commande par un nombre entier compris entre 1 et 5, et le résultat est présenté dans la Figure 4.18. Les notes pour 'baisser le volume' et 'augmenter le volume' sont 3,79 et 3,77, tandis que pour les autres commandes, la note est supérieure à 4,3. À leur avis, le changement de volume n'est pas aussi utile que les autres commandes car il ne doit être utilisé qu'une fois au début. L'utilité de 'arrêter' est légèrement supérieure à celle de 'reprendre', ce qui est tout à fait raisonnable, car le 'reprendre' ne peut être utilisé qu'après le 'arrêter'.

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

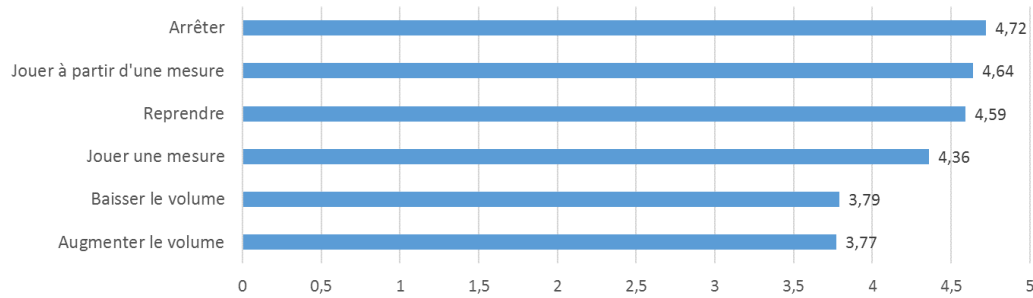


FIGURE 4.18 : Utilité de chaque commande notée par les participants.

Nous leur avons également demandé de proposer d'autres commandes qu'ils souhaiteraient ajouter dans notre application. Ils avaient le droit de proposer autant de commandes qu'ils voulaient et il y a 18 participants qui ont proposé au moins une nouvelle commande. Le résultat est présenté dans la Figure 4.19. La commande la plus populaire proposée par 8 participants consiste à pouvoir accélérer et ralentir la musique (changer le tempo de la musique).

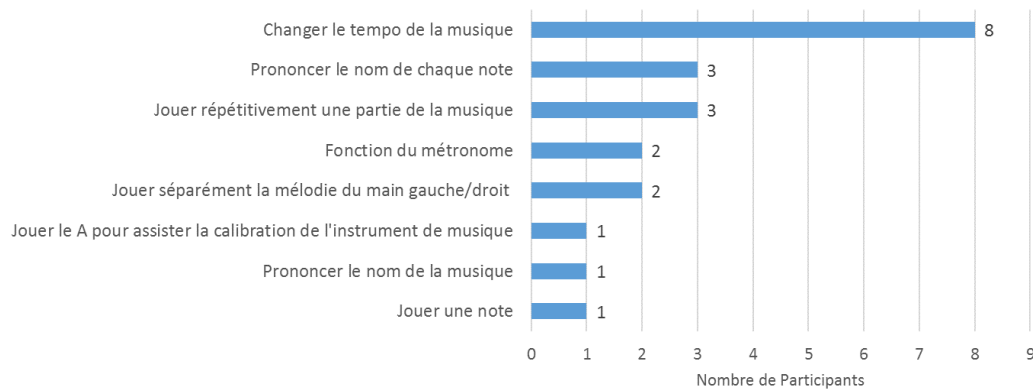


FIGURE 4.19 : Commandes supplémentaires proposées par les participants.

4.5.9 Supposition sur l'Utilité de EyeMusic

Dans le questionnaire post-test, nous avons demandé aux participants si cette application serait utile pour leur apprentissage de la musique. Selon le résultat, 79,5% des participants ont indiqué que cette application serait utile pour le processus d'apprentissage de la musique, 5,1% des participants ont indiqué qu'ils n'en auraient pas besoin car ils possèdent déjà une bonne connaissance de la notation

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

musicale, 10,3% des participants sont indifférents et 5,1% d'entre eux ne la trouvent pas utile, peut-être à cause du faible taux de réussite qu'ils ont obtenu lors de l'expérience. Bien que cette question soit similaire à la première question du SUS, les résultats sont différents. Pour la première question du SUS, la contribution au score moyenne n'est que de 2/4, mais dans ce questionnaire, l'utilité de notre application est approuvée par la majorité des participants. Peut-être cette différence est apportée par le mot 'fréquemment' dans la première question du SUS : Bien que cette application puisse être utile pour leur apprentissage de la musique, ils ne l'utiliseraient qu'occasionnellement, car ce n'est pas une application pour un usage quotidien.

La comparaison de l'utilité supposée pour les participants avec les différents niveaux de connaissance de la notation musicale est présentée dans le Tableau 4.4. Pour les 13 participants avec aucune connaissance de la notation musicale, 92,3% d'entre eux la trouvent utile pour le processus d'apprentissage de la musique, tandis que 7,7% d'entre eux sont indifférents. Le résultat pour les 2 autres catégories est similaire, à l'exception de 2 participants avec une bonne/excellente connaissance de la notation musicale qui ont déclaré qu'ils n'auraient pas besoin de cette application car ils possèdent déjà un bon niveau de connaissance de la notation musicale, et 2 participants possédant une connaissance élémentaire/moyenne de la notation musicale ne la trouvent pas utile, peut-être à cause du faible taux de succès qu'ils ont obtenu.

4.5.10 Implications sur la Conception du Prototype

Actuellement, la performance de notre approche est relativement faible en comparant avec les d'autres modalités d'interactions lorsqu'on utilise simultanément les 6 mouvements de la tête à regard fixe. Ce problème peut être résolu de plusieurs manières. D'une part, nous pouvons améliorer la performance de nos algorithmes de reconnaissance des mouvements de la tête à regard fixe selon les solutions proposées dans la Section 4.5.5. D'autre part, nous pouvons améliorer cette application en modifiant sa conception.

Par exemple, utiliser moins de mouvements de la tête serait une solution poten-

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

TABLE 4.4 : Supposition d'utilité de EyeMusic pour les participants avec des niveaux de connaissance de la notation musicale différents.

	Niveau de Connaissance de la Notation Musicale		
	aucun	élémentaire/ moyen	bon/ excellent
Utile	12	9	10
Indifférent	1	1	2
Inutile (sans raison)	0	2	0
Inutile (possède déjà une bonne connaissance de la notation musicale)	0	0	2
Nombre de Participants au Total	13	12	14

tielle pour rendre notre application plus fiable. Ceci est possible de deux manières. La première solution consiste à ne fournir que des commandes élémentaires, telles que 'jouer une mesure' et 'arrêter'. La deuxième méthode consiste à affecter les commandes aux mouvements de la tête de manière différente. Par exemple, la commande 'reprendre' est toujours utilisée après la commande 'arrêter' ; du coup, nous pouvons déclencher ces deux commandes avec un seul mouvement de la tête à regard fixe (s'il y a une mélodie en jouant, ce mouvement signifie de l'arrêter, tandis que s'il y a un morceau de musique en attente, ce mouvement signifie de le reprendre). De plus, nous pouvons utiliser les positions différentes de la fixation du regard pour différencier les commandes déclenchées par les mêmes mouvements de la tête. Comme par exemple, les mouvements verticaux de la tête permettent de jouer les notes quand le regard est fixé sur les mesures ; ou de modifier le volume lorsque le regard est fixé sur un bouton de volume imprimé dans le coin en haut à gauche de la partition ; ou encore d'arrêter et de reprendre la musique lorsque le regard est fixé sur le coin en haut à droite de la partition. Dans ce cas, 2 mouvements différents nous suffiraient déjà pour déclencher les 6 commandes, et nous pouvons même ajouter plusieurs commandes proposées par les participants dans notre application en utilisant moins de mouvements de la tête à regard fixe.

Les mouvements de la tête à regard fixe utilisés pour déclencher les commandes

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

peuvent également être choisis de plusieurs manières. La première façon consiste à combiner les mouvements qui suivent le même axe : dans ce cas, nous sommes capables d'éliminer la majorité des mauvaises détections, et nous pouvons encore détecter 3 mouvements différents. La deuxième façon est d'éliminer les 2 mouvements horizontaux de la tête et d'utiliser uniquement les hochements de la tête et/ou les dodelinements de la tête, car les mouvements horizontaux de la tête ne sont pas aussi populaires auprès des participants, et chacun d'entre eux n'est pas réalisable pour 10% des utilisateurs. En même temps, nous pouvons également éviter une grande partie des mauvaises détections des dodelinements de la tête en éliminant les mouvements horizontaux de la tête. De plus, nous pouvons permettre aux utilisateurs de personnaliser notre application en choisissant la paire de mouvements de la tête qu'ils souhaitent utiliser pour déclencher des commandes : les utilisateurs qui ont des difficultés avec les mouvements verticaux et horizontaux de la tête peuvent utiliser uniquement les dodelinements de la tête, tandis que les utilisateurs qui maîtrisent bien tous ces mouvements peuvent choisir les mouvements qu'ils préfèrent. En outre, les commandes peuvent également être personnalisées : les utilisateurs peuvent choisir les commandes les plus utiles pour leur phase d'apprentissage de la musique et utiliser leurs mouvements préférés pour les déclencher.

4.6 Conclusion du Chapitre

Dans ce chapitre, nous avons présenté notre application appelée EyeMusic, destinée à aider les débutants de la musique pour l'apprentissage d'un instrument. Nous avons décrit le choix du scénario, la procédure et les points techniques de la réalisation, et le premier prototype expérimental développé. Les mouvements volontaires de la tête à regard fixe présentés au chapitre 3 et les algorithmes de détection proposés sont mis en œuvre dans cette application afin de fluidifier les interactions. Nous avons également présenté le test d'utilisabilité que nous avons conduit sur ce prototype expérimental avec la participation de 41 volontaires afin d'évaluer la conception de l'application et la performance de nos algorithmes de détection des mouvements volontaires de la tête proposés. Les résultats expérimentaux obtenus

Chapitre 4. EyeMusic : Apprentissage Interactif de la Musique

ont été analysés et discutés à la fin de ce chapitre. En plus, nous avons proposé des solutions envisageables qui pourraient être menées pour améliorer la performance de nos algorithmes et des implications sur la conception du prototype. Dans le chapitre suivant, nous présentons deux autres scénarios d'application que nous avons conçus (EyeRecipe et EyePay). Ces scénarios reposent sur les mêmes principes de commande.

EyeRecipe et EyePay

À part l'application EyeMusic, nous avons conçu deux autres scénarios dans lesquelles l'eye-tracking et les mouvements volontaires de la tête à regard fixe sont utilisés afin de démontrer la potentialité de l'eye-tracking dans le domaine de l'IHM en tant que modalité d'entrée. Ces deux applications sont utiles pour une cible beaucoup plus large que les personnes potentiellement intéressées par EyeMusic. Elles peuvent être des applications à usage quotidien pour certains utilisateurs. Dans ce chapitre, nous allons décrire précisément ces deux scénarios appelés EyeRecipe et EyePay. En particulier nous allons présenter la réalisation d'EyeRecipe qui est une application interactive liée à la cuisine.

5.1 EyeRecipe

Le premier scénario conçu s'appelle EyeRecipe. C'est un scénario lié à la cuisine et, plus particulièrement, la consultation des recettes sur un site Internet via un écran. Cette application est destinée à aider les utilisateurs qui essaient de préparer un plat à l'aide d'une recette qu'ils ont trouvée sur Internet ou bien qu'ils ont envie de partager avec les autres.

5.1.1 Choix du Scénario

Faire la cuisine est une activité quotidienne pour toutes les familles, mais il n'est pas facile de mémoriser toutes les recettes par cœur. Dans ce cas, la solution la plus naturelle est de consulter la recette bien souvent disponible sur Internet pendant le processus de cuisine avec sa tablette ou son smart phone. À cause de ces besoins, il existe déjà beaucoup de sites Internet ou d'applications qui servent au partage des

recettes, comme par exemple Marmiton, 750g, cuisinez, etc.

Normalement, afin de bien suivre toutes les étapes d'une recette, l'utilisateur a besoin de consulter de temps en temps cette recette pendant le processus de préparation, mais à ce moment-là les deux mains de l'utilisateur sont souvent indisponibles (sales ou engagées dans une autre activité). En conséquence, la consultation de la recette n'est pas vraiment pratique et cela peut salir la tablette ou la souris.

Le système EyeRecipe est un bon moyen pour répondre à ce problème. Grâce à l'eye-tracking et aux mouvements volontaires de la tête à regard fixe, l'utilisateur est capable d'interagir avec la recette pratiquement sans besoin de libérer ou de se laver les mains. Il y a au total trois groupes de fonctions dans le système EyeRecipe :

- les fonctions pour la consultation d'une recette,
- les fonctions de prise de vidéo ou d'images afin de partager sa propre recette avec les autres, et
- les fonctions consistant à assister l'utilisateur en passant des commandes aux équipements intelligents (si l'utilisateur en possède).

5.1.1.1 Consultation d'une Recette

Au niveau de la consultation de la recette, les fonctions les plus utiles consistent au défilement de la page (pour les recettes longues), l'élargissement/le rétrécissement d'une image, et le 'jouer' et le 'arrêter' d'une vidéo.

Les deux dernières commandes peuvent être déclenchées avec un seul mouvement volontaire de la tête à regard fixe qu'on a présenté dans la Section 3.3 : lorsque le regard de l'utilisateur est fixé sur la région de la vidéo, le mouvement déclenche la commande 'jouer' ou la commande 'arrêter' selon l'état actuel de la vidéo ; le même mouvement signifie la commande 'élargir l'image' ou la commande 'rétrécir l'image' si le regard de l'utilisateur est fixé sur une image de taille originale ou élargie.

Le défilement de la page, par contre, n'est pas réalisé à l'aide des mouvements volontaires de la tête à regard fixe. Pour cette fonction, on envisage de la réaliser de manière implicite en s'inspirant du travail de Kumar et al. [62] car il est plus naturel et moins fatiguant que les mouvements de la tête. Comme il existe souvent des images et des vidéos dans une recette, on décide d'utiliser deux lignes de seuil



FIGURE 5.1 : Lignes de seuil pour le défilement de la page.

sur l'écran pour assister le défilement (voir la Figure 5.1) : lorsque le regard est fixé au-dessus du seuil du défilement en arrière, la page défile de haut en bas ; si le regard de l'utilisateur est fixé en-dessous du seuil du défilement en avant, la page défile de bas en haut ; le défilement s'arrête lorsque le regard de l'utilisateur se situe au milieu de ces deux lignes de seuil.

5.1.1.2 Prise des Images et de la Vidéo

Parfois, l'utilisateur a également envie de partager sa propre recette avec les autres. Afin de mieux illustrer cette recette, l'utilisateur a peut-être besoin de prendre des photos ou bien des vidéos pendant le processus de préparation de la recette. Mais comme ses mains sont souvent indisponibles pour prendre une caméra, la caméra de la scène incorporée à l'eye-tracker est une bonne alternative.

À l'aide du système EyeRecipe, les utilisateurs peuvent alors prendre des images et des vidéos par la caméra frontale en faisant des mouvements volontaires de la tête à regard fixe. Un des six mouvements est utilisé pour déclencher la prise d'une photo, un autre mouvement est appliqué pour le commencement et l'arrêt de la prise de vidéo. Toutes ces commandes sont accompagnées par un feedback audio afin que l'utilisateur puisse savoir si la commande s'est bien déroulée.

5.1.1.3 Contrôle des Équipements Intelligents

Si les équipements dans la cuisine de l'utilisateur sont intelligents (contrôlables par un terminal mobile), le système EyeRecipe peut également servir d'interface pour faciliter la cuisine de l'utilisateur en lui fournissant la possibilité de passer des commandes diverses à distance uniquement par le regard et des mouvements de la tête sans besoin de se déplacer ou de libérer ses mains. Comme par exemple, il est souvent requis de préchauffer le four à une température précise lorsqu'on est en train de battre des œufs ou de mélanger des ingrédients. Grâce au système EyeRecipe, cette commande peut être passée en effectuant un mouvement spécifique de la tête avec le regard fixé sur le mot clé de la recette sur l'écran, à l'aide de la technique d'OCR (*Optical Character Recognition*) [75]. De même, ce mouvement peut aussi servir à passer les autres commandes comme par exemple la mise en route d'un chronomètre. De plus, le contrôle des équipements intelligents peut également être réalisé en effectuant des mouvements spécifiques de la tête lorsque le regard de l'utilisateur est fixé sur le dispositif à contrôler. Ainsi les utilisateurs seront plus libres pour passer les commandes diverses (augmenter/diminuer la température du four par exemple). Le feedback de ces commandes est affiché directement sur l'objet contrôlé.

5.1.2 Réalisation

5.1.2.1 Choix de la Correspondance entre les Mouvements et les Commandes

Afin de réaliser tous les fonctions conçues pour notre application, il faut au moins six mouvements volontaires de la tête à regard fixe différents :

- un mouvement pour l'élargissement/le rétrécissement d'une image et le déclenchement/l'arrêt d'une vidéo car ces commandes ressemblent à l'effet de clique souris sur l'endroit regardé,
- un mouvement pour le contrôle des équipements intelligents qui nécessite une reconnaissance sur l'information regardée (ou une reconnaissance d'objet intelligent),

Chapitre 5. EyeRecipe et EyePay

- deux mouvements pour le raffinement des commandes à passer aux équipements intelligents (la température du four par exemple),
- un mouvement pour la prise des photos, et
- un mouvement pour la prise de vidéo.

La commande du défilement de la page est déclenchée de manière implicite, donc il ne demande pas de mouvement de la tête spécifique.

Selon les résultats expérimentaux du test d'utilisabilité d'EyeMusic et les études documentaires, les hochements (verticaux et horizontaux) et les dodelinements de la tête sont utilisables par la majorité de participants, et ils sont également les mouvements préférés des participants. Ainsi on peut décider d'appliquer ces six mouvements dans le système EyeRecipe.

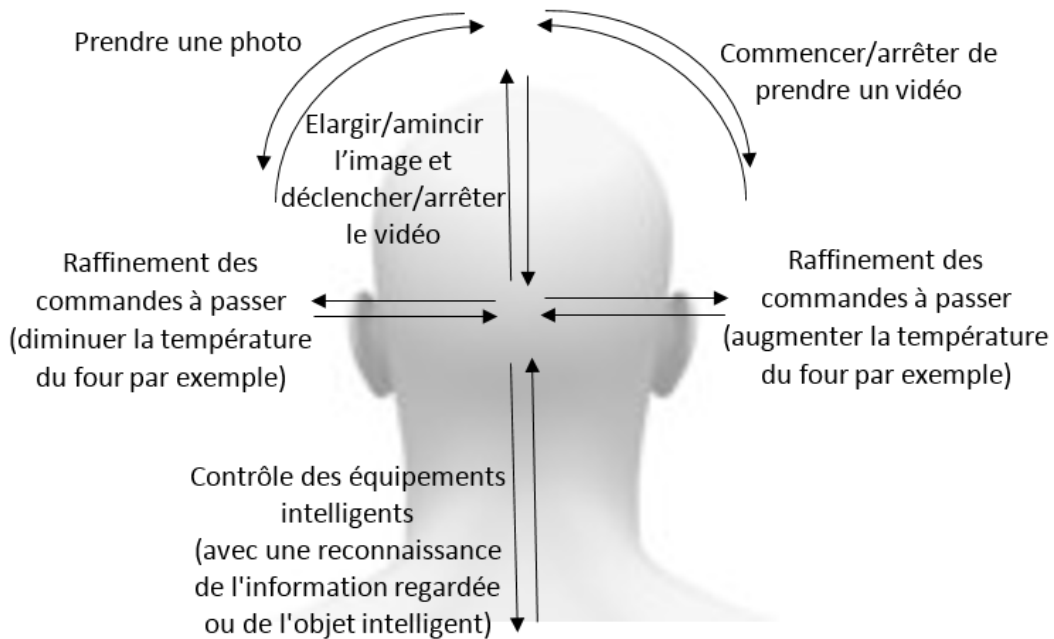


FIGURE 5.2 : Diagramme schématique de la correspondance entre les mouvements de la tête et les commandes de EyeRecipe.

La correspondance entre les mouvements de la tête et les commandes que l'on envisage dans ce système est illustrée dans la Figure 5.2. On peut supposer que les utilisateurs d'un tel dispositif s'en servent avant tout pour consulter des recettes et contrôler des équipements intelligents. On affectera donc les mouvements préférés (les hochements verticaux de la tête) à ces 2 commandes. Les dodelinements de la

tête sont appliqués pour les commandes les moins fréquentes car ils sont légèrement plus fatiguants que les hochements verticaux. En plus, comme les utilisateurs font beaucoup plus de mouvements verticaux naturels de la tête pendant la cuisine que pendant la lecture d'une recette sur un écran, il est raisonnable d'attribuer à la prise des images et des vidéos des mouvements qui ont lieu rarement naturellement (les dodelinements). Les hochements horizontaux de la tête sont alors affectés aux raffinements des commandes à passer aux équipements intelligents.

5.1.2.2 Calcul de la Position Relative du Regard par rapport à l'Écran

La position du regard que nous obtenons via l'API de l'eye-tracker est connue en coordonnées de l'image de la scène. Mais pour les commandes de consultation de recettes, c'est la position relative du regard par rapport à l'écran qui est utile (ou plus précisément la position relative du regard par rapport aux seuils prédéfinis pour le défilement de la page). Comme l'écran est souvent entouré par une bordure noire qui est facilement détectable à l'aide de fonctions des bibliothèques de traitement d'images, nous sommes en mesure de transformer les coordonnées du regard sur l'image de la scène en coordonnées relatives du regard par rapport à l'écran par la matrice d'homographie grâce aux coordonnées de la bordure noire détectée. On peut ensuite comparer la coordonnée relative y du regard avec les deux seuils afin de savoir si la commande du défilement doit être déclenchée.

5.1.2.3 Reconnaissance des Informations Regardées

Pour les fonctions au niveau du contrôle des équipements intelligents, un point essentiel est de reconnaître l'information regardée pendant le mouvement volontaire de la tête. Comme l'image de la scène capturée par la caméra frontale peut être floue surtout pendant les mouvements de la tête, il n'est pas toujours facile de reconnaître les mots clés en traitant l'image de la scène. En plus, comme l'image de la scène fournie par l'API de l'eye-tracker possède toujours un point vert superposé sur l'endroit du regard, il arrive souvent que le mot clé soit complètement caché par ce point. Du coup, afin de connaître les informations regardées, on ne traite la

Étape 2

Ajouter le beurre et les œufs, puis travailler avec les mains.

Étape 3

Couper le boule en 3.

Étape 4

Préchauffer le four à 200 °C.

Étape 5

Étaler la pâte avec un rouleau et découper des formes à l'emporte-pièce.

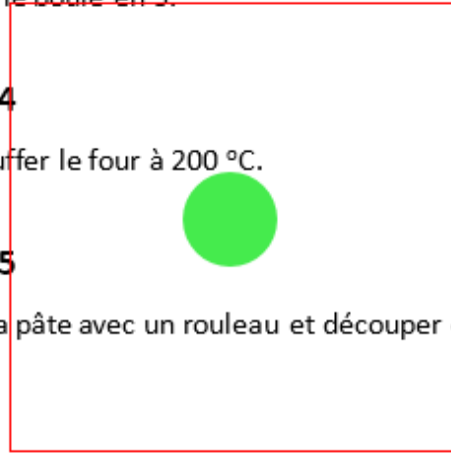


FIGURE 5.3 : Diagramme schématisé pour la capture d'images à traiter par EyeRecipe.

capture d'écran qu'une fois le mouvement spécifique détecté, et on ne traite que la région autour du point du regard détecté pour lequel les coordonnées relatives par rapport à l'écran sont déjà calculées.

Un exemple du processus est montré dans la Figure 5.3. Le point vert montre l'emplacement du regard calculé ; la capture d'écran est faite sur la région entourée par la bordure rouge. Ainsi, le traitement d'images est appliqué sur la Figure 5.4. Comme cela vient directement de l'écran, la netteté de l'image est garantie et il n'y a pas de point vert superposé. En plus, on ne s'intéresse qu'à la région à côté de la position du regard, par rapport au traitement de la capture de plein écran. Le temps de calcul est en conséquence largement diminué.

Sur l'image à traiter, on applique d'abord le processus d'OCR (*Optical Character Recognition*) [75] afin de détecter les mots clés comme par exemple '°C' et 'min'. Ce processus est facilement réalisable à l'aide des outils open source comme par exemple Tesseract [81]. S'il y a plusieurs mots clés détectés sur cette image, on ne

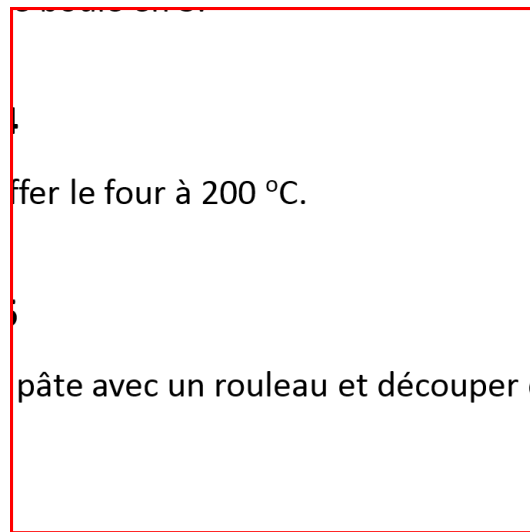


FIGURE 5.4 : Image réellement traitée.

prend que celui qui est le plus proche de la position du regard. Ensuite, on prend le chiffre qui se situe juste devant le mot clé pour compléter la commande. Par exemple, en appliquant l'OCR sur la Figure 5.4, on est capable de détecter le mot clé '°C', et puis on prend le chiffre '200' qui est devant le '°C' pour compléter la commande 'mettre le four à 200 degrés'. De même, s'il existe '30 min' dans l'image à traiter, on est en mesure de comprendre la commande 'démarrer un chronomètre de 30 minutes'. La distinction des équipements intelligents à contrôler est faite par les mots clés différents, le '°C' est lié au four, alors que le 'min' est lié au chronomètre.

5.1.2.4 Reconnaissance de l'Objet à Contrôler

Grâce aux méthodes de traitement d'images et de reconnaissance de formes, la reconnaissance de l'objet peut être réalisée en calculant la correspondance entre l'image de la scène en temps réel avec un dispositif pré-connu d'images enregistrées à l'avance dans une base de données, comme ce qui est proposé par Bonino et al. [8]. Puis, à l'aide des coordonnées de l'objet cible sur l'image de la scène, les coordonnées relatives du regard par rapport à l'objet détecté peuvent être calculées par la transformation des coordonnées.

5.1.2.5 Détection des Mouvements Verticaux et Horizontaux de la Tête à Regard Fixe

Globalement, l'algorithme pour la détection des mouvements verticaux et horizontaux de la tête à regard fixe d'EyeRecipe est similaire à l'algorithme qu'on a présenté dans la Section 3.4. On détecte d'abord la bordure noire (ou l'objet intelligent) sur l'image de la scène afin de calculer les coordonnées relatives du regard par rapport à l'écran (ou l'objet cible). On surveille la variation de la coordonnée y (ou x) du regard pour trouver les mouvements potentiels et on exclut les autres cas (les mouvements oculaires purs ou bien la tête et les yeux bougent dans le même sens) par les variations de la coordonnée relative y (ou x) différentes. La seule modification est que, dans le système EyeRecipe, on ne fixe plus un seuil sur le rapport entre la dérivée de la coordonnée relative y (ou x) par rapport à l'écran et la dérivée de la coordonnée y (ou x) sur l'image de la scène, car ce ratio est lié avec la distance entre l'utilisateur et l'écran et cette distance est plutôt variable lors de la préparation d'une recette de cuisine. Alternativement, dans le système EyeRecipe, le `THRESHOLD_RATIO` est une variable liée avec cette distance, c'est-à-dire la taille de la bordure (ou la taille de l'objet intelligent détecté) que l'on calcule pour chaque trame traitée.

5.1.2.6 Détection des Inclinaisons de la Tête à Regard Fixe

Dans le système EyeRecipe, les deux dodelinements de la tête sont utilisés pour la prise des photos et des vidéos. Il peut alors arriver que, de temps en temps, l'écran (la bordure noire) n'existe pas sur l'image de la scène pendant ces mouvements. On ne peut alors plus utiliser la matrice d'homographie calculée par les coordonnées de la bordure détectée pour obtenir l'angle de rotation en trois dimensions. Celui-ci est pourtant nécessaire dans l'algorithme décrit dans la Section 3.4. De plus, il n'y existe pas forcément un objet détectable qui peut servir en tant que référence, comme la bordure d'écran, car les images et les vidéos peuvent être prises n'importe où et sur n'importe quoi.

Dans ce cas-là, on décide d'utiliser l'algorithme de l'odométrie visuelle mono-

culaire qui est une partie essentielle pour la technologie SLAM (*Simultaneous Localization and Mapping*) [31, 100]. Pour chaque frame, on extrait un ensemble de caractéristiques saillantes de l'image (par exemple des points et des lignes), on les fait correspondre dans des images successives à l'aide de descripteurs invariants, et puis on récupère le mouvement de la caméra en utilisant la géométrie épipolaire. Toutes ces étapes peuvent être réalisées à l'aide de fonctions d'OpenCV, et il est alors possible d'obtenir le mouvement relatif de la caméra avec six degrés de liberté (3 ddls pour la translation et 3ddls pour la rotation). Grâce à cet algorithme, on est en mesure d'obtenir l'angle de rotation relatif selon l'axe Z nécessaire à notre algorithme dans la Section 3.4. On peut alors utiliser les mêmes contraintes pour détecter les dodelinements de la tête.

5.1.3 Algorithme Global d'EyeRecipe

L'algorithme global du système EyeRecipe est illustré dans la Figure 5.5. Pour chaque trame, on détecte d'abord la bordure noire de l'écran et les objets intelligents pré-enregistrés afin de connaître la position relative de l'écran (ou de l'objet cible) par rapport à l'utilisateur. À l'aide des coordonnées de la bordure noire détectée et les coordonnées du point du regard sur l'image de la scène, il est possible de calculer la position relative du regard par rapport à l'écran. Si la coordonnée relative y du regard a dépassé un des seuils pour le défilement de la page sur plusieurs trames successives, on déclenche la commande du défilement de la page dans le sens correspondant, sinon on applique l'odométrie visuelle monoculaire afin d'obtenir l'angle de rotation relative et ainsi détecter les dodelinements de la tête. Les deux dodelinements de la tête sont identifiés en avance, la variation des coordonnées du point du regard est examinée pour détecter les mouvements verticaux ou horizontaux de la tête uniquement (s'il n'y a pas de dodelinement de la tête en cours d'après la variation de l'angle de rotation z). Si l'un des six mouvements volontaires de la tête à regard fixe est détecté, on déclenche la commande correspondante, sinon on passe à la trame suivante. La commande correspond au mouvement M42 est illustrée dans la Figure 5.6, cette commande dépend de l'objet détecté sur l'image de la scène : la reconnaissance sur l'information regardée si la bordure de l'écran est détectée ; le

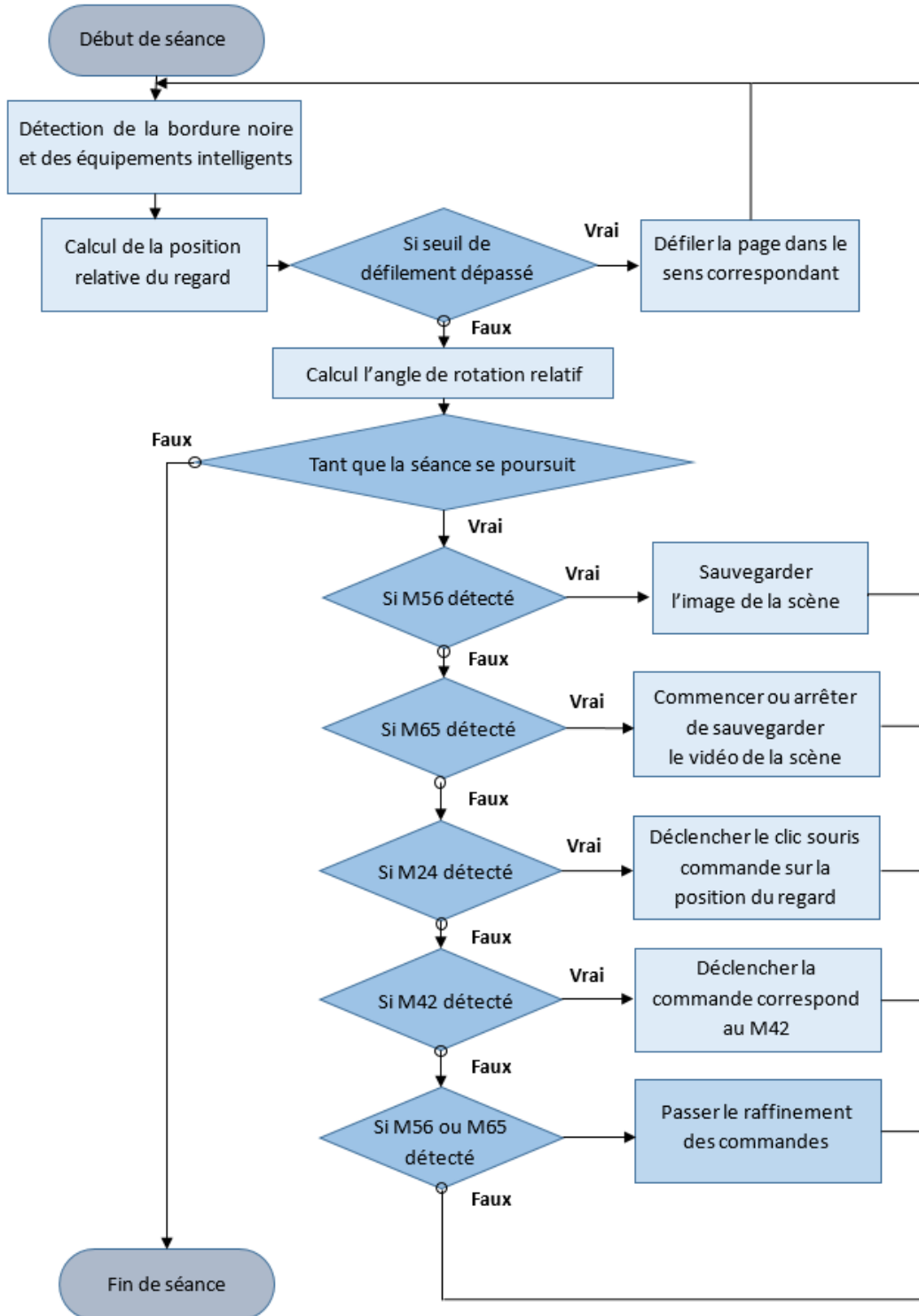


FIGURE 5.5 : Vue d'ensemble de l'algorithme principal d'EyeRecipe.

déclenchement de l'équipement si un objet intelligent est détecté.

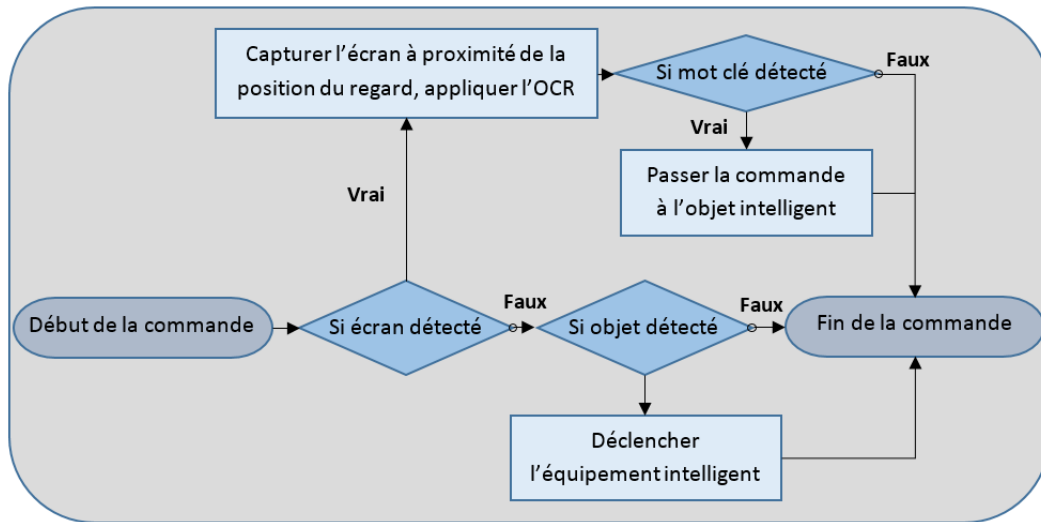


FIGURE 5.6 : La commande correspond au M42 d'EyeRecipe.

5.1.4 Conclusion de EyeRecipe

L'application EyeRecipe est destinée à aider les utilisateurs pendant le processus de la cuisine. Ceci est fait en leur fournissant la possibilité d'interagir avec l'écran sur lequel la recette est affichée et avec les objets intelligents dans l'environnement sans besoin de libérer leurs mains, grâce aux mouvements volontaires de la tête à regard fixe. Les commandes implicites et quelques améliorations de nos algorithmes de détection des mouvements volontaires de la tête à regard fixe sont prévues dans la réalisation de cette application. Des évaluations approfondies au niveau de la performance des algorithmes améliorés pourront être menés sur le prototype de cette application.

5.2 EyePay

Le deuxième scénario proposé est lié au paiement. Cette application est destinée à aider les utilisateurs lorsqu'ils ont besoin de payer pour acheter un objet ou un service.

Chapitre 5. EyeRecipe et EyePay

Afin d'effectuer un paiement, il nous faut soit signer un chèque soit sortir le portefeuille pour payer par carte ou bien en espèces. Dans tous ces cas, il nous faut une série de manipulations avec les deux mains et cela peut prendre quelques minutes afin d'accomplir ce processus. En plus, lorsqu'on a besoin de payer quelque chose, il arrive souvent que les deux mains sont occupées par ce qu'on achète ; ainsi la complexité de ce processus est à nouveau élevée. Donc la création d'une nouvelle façon de paiement 'mains libres' peut s'avérer intéressante.

5.2.1 Paiement Rapide sans Besoin de Libérer les Mains

Actuellement en Chine, on commence déjà à abandonner les modalités de paiement traditionnelles ; plusieurs applications de paiement sont créées en remplacement (WeChat Pay et Alipay, par exemple). Le paiement peut être effectué en scannant le code à deux dimensions du commerçant (ou du particulier), puis en entrant le montant à payer et puis, en confirmant le paiement. Cette méthode a beaucoup facilité la vie de tous en diminuant le nombre d'objets qu'on a besoin de prendre en sortant ; il suffit de mettre la téléphone portable dans la poche. Mais le problème lié à l'occupation des mains n'est toujours pas réglé.

Basé sur ces applications de paiement existantes, nous pouvons libérer les mains en utilisant le couplage entre l'eye-tracking et les mouvements volontaires de la tête à regard fixe. Lors d'un paiement, le commerçant affiche un code à deux dimensions qui contient 2 parties d'informations : l'identité du commerçant et le montant à payer. L'utilisateur pourra réaliser ce paiement en effectuant un mouvement volontaire de la tête spécifique avec le regard fixé sur ce code. Dans ce cas-là, le paiement devient beaucoup plus rapide et pratique.

5.2.2 Entrée de Code Privé par le regard

Actuellement, pour le paiement d'un montant élevé, l'utilisateur a souvent besoins d'entrer un code personnel afin de garantir la sécurité du paiement. Mais ce code privé peut être dérobé par les voleurs qu'il soit tapé sur le clavier de la machine de paiement ou bien sur l'écran tactile d'un téléphone portable. A l'aide

d'un eye-tracker, il est possible d'entrer ce code par le regard. Comme le regard est beaucoup plus discret que ne le sont les doigts sur un clavier, les voleurs ne peuvent pas détecter le code de l'utilisateur entré par le mouvement des yeux, la sécurité est ainsi bien mieux assurée.

5.2.3 Scanlib Version Eye-tracker

D'ailleurs, la fonction du système EyePay peut ne pas se limiter au paiement d'un montant fixe à un commerçant ou un particulier ; il pourrait remplacer complètement la fonction de caisse dans les supermarchés. Normalement les marchandises dans les supermarchés sont identifiées par un code-barre collé sur le paquet. En détectant ce code-barre on est en mesure de connaître le prix de ce produit. Actuellement, dans certains supermarchés (Carrefour par exemple), il existe un type de machine spécifique appelé Scanlib. En scannant le code-barre des produits qu'ils ont envie d'acheter, les utilisateurs peuvent payer directement en sortant du supermarché à une caisse spécifique. Ainsi, le client n'a plus besoin de sortir tous les produits et de les remettre dans le chariot afin de connaître la somme totale qu'il doit. Mais Scanlib est un dispositif qui occupe une main. Parfois il n'est pas pratique de scanner le code-barre si le produit est grand ou lourd. En plus l'utilisateur a toujours besoin de faire la queue pour payer en sortant (bien que ce ne soit pas la même queue que les gens qui n'utilisent pas cette machine). Ces deux problèmes peuvent être résolus grâce au système EyePay. À l'aide d'un eye-tracker porté sur la tête équipée d'une caméra frontale, l'utilisateur est en mesure de scanner le code-barre du produit qu'il a envie d'acheter en effectuant un mouvement volontaire de la tête avec le regard fixé sur celui-ci. Les deux mains sont toujours disponibles pour prendre la marchandise. En plus, le processus du paiement peut être fait directement lors du scan du code-barre ; l'utilisateur peut ainsi éviter de faire la queue en sortant.

5.2.4 Affectation des Commandes

Il y a principalement 3 catégories de fonctions pour le scénario d'application EyePay :

- scanner un code-barre dans les supermarchés (sans paiement),
- confirmer un paiement (à la fin de l'achat à l'aide d'un code à deux dimensions affiché ou lors de la prise d'une marchandise), et
- entrer le code privé.

La dernière commande peut être réalisée à l'aide d'un mouvement volontaire des yeux (prédéfini par l'utilisateur), donc 2 mouvements volontaires de la tête à regard fixe suffisent pour cette application. Comme les hochements verticaux de la tête sont moins fatiguants et qu'ils sont les mouvements préférés par les participants d'après l'expérience d'EyeMusic, nous décidons d'affecter ces deux mouvements à EyePay. De plus, pour les utilisateurs qui sont plus à l'aise avec les autres mouvements, ils pourraient personnaliser cette application en choisissant par eux-même la correspondance entre les mouvements et les commandes.

5.2.5 Conclusion de EyePay

Le système EyePay est destiné à permettre aux gens d'effectuer des paiements de manière rapide uniquement par le regard et les mouvements de la tête. En plus, la fonction 'entrée de code privé par le regard' peut assurer une meilleure sécurité de l'utilisateur lors d'un paiement. Cette application pourrait être utile pour le grand public.

5.3 Conclusion du Chapitre

Dans ce chapitre, nous avons présenté deux autres scénarios d'application conçus (EyeRecipe et EyePay) qui reposent sur les mêmes principes de commande que EyeMusic (les mouvements volontaires de la tête à regard fixe). Nous avons en particulier présenté la réalisation d'EyeRecipe qui est une application liée à la cuisine. Les commandes implicites et quelques améliorations de nos algorithmes de détection des mouvements volontaires de la tête à regard fixe sont intégrées dans cette

application afin de fluidifier l'interaction. Ces deux applications sont utiles pour une cible beaucoup plus large qu'EyeMusic et elles peuvent être des applications à usage quotidien pour certains utilisateurs. Le développement et les évaluations de ces deux applications pourront être menés dans des travaux ultérieurs afin d'approfondir la potentialité de l'eye-tracking dans le domaine de l'IHM en tant que modalité d'entrée. La conclusion et les perspectives de nos travaux seront présentées dans le chapitre qui suit.

Conclusion et perspectives

6.1 Travaux Réalisés

Dans cette thèse, nous avons mené une étude documentaire approfondie sur l'utilisation de l'eye-tracking pour l'interaction homme-machine en tant que modalité d'entrée. Nous avons étudié les applications proposées par les chercheurs, y compris les applications servant à faciliter la vie des personnes à mobilité réduite, les interactions avec un écran d'ordinateur utilisable pour tous, les applications mobiles pour contrôler des objets intelligents, et les interactions externes diverses, afin de nous en inspirer pour la conception de nos propres scénarios. Puis, nous avons examiné les problèmes et les difficultés rencontrés dans ce domaine. Nous avons constaté que l'eye-tracking ne disposait pas de méthodes pour le déclenchement des commandes, ce qui restreint grandement son intérêt surtout au niveau de la vitesse. Nous avons ensuite étudié les solutions envisageables proposées (mouvements volontaires des yeux, poursuite lisse, couplage avec un autre dispositif, etc).

En comparant les solutions proposées par les chercheurs sous plusieurs aspects, nous avons proposé de coupler l'eye-tracking avec les mouvements volontaires de la tête à regard fixe pour déclencher des commandes diverses, et pour fluidifier les interactions avec des objets dans l'environnement physique. Nous avons en particulier proposé un algorithme pour la détection de 6 mouvements volontaires de la tête à regard fixe différents (2 hochements verticaux, 2 hochement horizontaux et 2 dodelinements de la tête) en traitant uniquement les images de la scène avec le point du regard superposé capturées par un eye-tracker porté sur la tête et équipé d'une caméra de scène.

Pour tester la performance de notre algorithme de détection des mouvements vo-

lontaires de la tête à regard fixe, et l'acceptation du déclenchement des commandes par ces mouvements lorsque les deux mains de l'utilisateur sont occupées par une autre activité, nous avons effectué des expériences sur une application originale conçue pour montrer la potentialité de l'eye-tracking dans le domaine de l'IHM : EyeMusic. EyeMusic est une application expérimentale visant à aider les débutants lorsqu'ils apprennent à lire des partitions musicales. A l'aide des mouvements volontaires de la tête à regard fixe et de l'eye-tracking, les utilisateurs peuvent se concentrer sur la partition sans perturbation pour écouter les notes d'une mesure spécifique. Pour cela, ils n'ont pas besoin de relâcher leurs mains occupées par leur instrument de musique. De plus, aucun appareil supplémentaire autre qu'un eye-tracker porté sur la tête équipé d'une caméra de scène n'est nécessaire pour cette application.

Nous avons effectué des tests d'utilisabilité approfondis du prototype de notre application EyeMusic avec 41 personnes. Le taux de réussite moyen est de 70%, bien que ceci est partiellement influencé par la conception des mouvements et la performance de l'eye-tracker utilisé :

- certains participants avaient du mal à fixer leur regard sur la cible pour certains des mouvements volontaires de la tête que nous avons utilisés dans cette application,
- parfois la performance de l'eye-tracker utilisé n'était pas satisfaisante.

La performance de notre algorithme de détection des mouvements volontaires de la tête à regard fixe est de 85%, et il n'y avait pas de différence significative entre la performance de chaque mouvement volontaire de la tête testé. La note moyenne de SUS est 77,2, ce qui est supérieur à la moyenne pour tous les produits testés (70), malgré le fait que nous avons une mauvaise contribution au score global pour la première question ("Je pense que je vais utiliser ce service fréquemment"). En effet, certains des participants n'étaient pas de vrais débutants de la musique et ce n'est pas une application destinée à un usage quotidien. Le score SUS moyen pour les participants qui ne possèdent aucune connaissance de la notation musicale est 80,2. De plus, selon le questionnaire post-test, 79,5% des participants ont indiqué que cette application leur serait utile pour leur processus d'apprentissage de la

Chapitre 6. Conclusion et perspectives

musique.

Dans cette thèse, nous avons également conçu deux autres scénarios (le EyeRecipe et le EyePay) dans lesquels l’eye-tracking et les mouvements volontaires de la tête à regard fixe sont utilisés pour faciliter les interactions. L’application EyeRecipe est destinée à aider les utilisateurs qui essayent de préparer un plat cuisiné à l’aide d’une recette qu’ils ont trouvée sur Internet ou bien qu’ils ont envie de partager avec les autres. Ceci est fait en leur fournissant la possibilité d’interagir avec l’écran, les objets intelligents dans l’environnement et la caméra de scène située en face de l’eye-tracker porté sur la tête par le regard et les mouvements volontaires de la tête à regard fixe sans besoin de libérer les mains. Le système EyePay est destiné à permettre aux gens d’effectuer des paiements de manière rapide uniquement par le regard et les mouvements de la tête. Ces deux applications pourraient être utiles pour un public cible beaucoup plus large qu’EyeMusic, et elles peuvent devenir des applications à usage quotidien pour certains utilisateurs. Les commandes implicites et quelques améliorations de notre algorithme de détection des mouvements volontaires de la tête à regard fixe sont prévues dans la réalisation de ces applications.

6.2 Contributions

Nos contributions principales sont les suivantes :

- Nous avons comparé les avantages et les inconvénients des différentes solutions proposées par les chercheurs pour répondre au problème du déclenchement des commandes. Nous avons découvert que le couplage de l’eye-tracking et du mouvement de la tête à regard fixe (mouvement de la tête volontaire avec le regard fixé sur le même point) possède des avantages indéniables par rapport aux autres solutions. Cela mérite donc d’approfondir les recherches afin de faire progresser l’application de l’eye-tracking dans le domaine de l’IHM.
- Nous avons proposé un nouvel algorithme pour la détection des mouvements de la tête à regard fixe en utilisant uniquement des images de la scène avec le point du regard superposé fournies par l’eye-tracker équipé d’une caméra

- frontale. Nous montrons qu'il peut être plus efficace que les algorithmes existants et qu'il permet d'interagir avec des objets de l'environnement réel.
- Nous avons mis en évidence un cas d'usage dans lequel l'eye-tracking apporte une réelle plus-value par rapport à d'autres dispositifs d'interaction dans le contexte d'apprentissage de la lecture d'une partition de musique. Dans ce système, que nous avons appelé EyeMusic, nous avons appliqué le couplage de l'eye-tracking et la détection du mouvement de la tête à regard fixe afin de fluidifier l'interaction. En plus, nous avons développé un premier prototype de celui-ci en utilisant notre algorithme de détection des mouvements de la tête à regard fixe proposé.
 - Nous avons évalué la performance de notre algorithme et la faisabilité de cette approche à travers un test d'utilisabilité intensif sur le premier prototype d'EyeMusic avec 41 participants.
 - Le système proposé n'est pas spécifique à l'application développée, et nous avons décrit d'autres scénarios d'application de notre prototype.

6.3 Perspectives

Plusieurs travaux pourront être menés dans le futur.

Au niveau de l'application EyeMusic, malgré ces résultats encourageants, de nombreuses améliorations peuvent encore être apportées pour baisser le taux d'erreurs de notre algorithme de détection des mouvements volontaires de la tête à regard fixe, pour rendre le déclenchement des commandes par ces mouvements plus fiable et pour améliorer l'expérience des utilisateurs. Une idée est d'éviter la répétition du processus de calibrage de l'eye-tracker en enregistrant les paramètres du regard de chaque participant dans un fichier de configuration spécifique. Nous pouvons ajuster le seuil au niveau de la durée de chaque mouvement ou le définir comme une variable adaptative capable de s'adapter aux habitudes de chaque utilisateur. De plus, il serait intéressant de re-concevoir cette application :

- en ajoutant quelques nouvelles commandes proposées par les participants pendant le test pour enrichir les fonctionnalités de notre application,

Chapitre 6. Conclusion et perspectives

- en considérant moins de mouvements volontaires de la tête à regard fixe pour améliorer la performance de la détection,
- en modifiant la correspondance entre les mouvements et les commandes d'après la compatibilité de chaque mouvement.

Un deuxième test plus concentré sur l'utilité de cette application sur de vrais débutants de la musique pourrait être mené pour évaluer cette application de manière plus complète.

Pour le moment, cette application ne fonctionne que pour les partitions enregistrées dans une base de données. Cependant, avec un algorithme de la reconnaissance de la musique (OMR) suffisamment performant, cette application pourra fonctionner pour n'importe quelle partition. De plus, en analysant le regard de l'utilisateur lors de la lecture des partitions, il sera peut-être possible de prédire si l'utilisateur a rencontré des difficultés et a besoin d'assistance, ce qui permettrait un comportement pro-actif (les commandes seraient déclenchées implicitement pour assister l'apprentissage au bon moment).

Pour les deux autres scénarios d'application (EyeRecipe et EyePay), le développement et les évaluations pourront être menés dans des travaux ultérieurs afin d'approfondir la potentialité de l'eye-tracking dans le domaine de l'IHM en tant que modalité d'entrée.

Enfin, le domaine d'application de cet algorithme de détection des mouvements volontaires de la tête à regard fixe ne se limite pas à ces trois applications. En combinant la technique de la reconnaissance des formes ou en ajoutant des marqueurs à côté de chaque objet intelligent, cet algorithme peut nous permettre d'interagir avec tous les objets physiques connectés dans l'environnement à distance avec les deux mains libres par les mouvements volontaires de la tête à regard fixe. Par exemple, l'utilisateur pourrait allumer ou éteindre la lumière à distance en faisant un mouvement spécifique de la tête lorsque son regard est fixé sur la lampe, il pourrait modifier le canal ou le volume de la télévision par les mouvements de la tête lorsque ses deux mains sont prises par une autre activité, etc.

Annexe Relative à l'Evaluation de EyeMusic

A.1 Présentation Générale d'EyeMusic

Ce qu'on va faire aujourd'hui est un test d'utilisabilité du 1er prototype d'une application qui s'appelle EyeMusic.

EyeMusic est un système d'apprentissage de la musique capable de jouer les notes d'une mesure d'une partition musicale que l'utilisateur supposé débutant dans son apprentissage n'arrive pas à appréhender. Lorsque l'utilisateur rencontre une difficulté dans la lecture d'une partition, par un simple mouvement de la tête il pourra déclencher le feedback audio de la mesure qu'il fixe du regard.

L'équipement qu'on utilise est un eye-tracker sous la forme d'une paire de lunettes. Sur cet eye-tracker, deux caméras internes filment le mouvement des yeux afin de calculer la direction du regard, et une 3ème caméra, située devant les lunettes, capture l'image de la scène regardée.

A.2 Formulaire de Consentement

Informations sur le projet

Titre du projet : Suivi du regard pour l'interaction mobile dans un environnement réel augmenté

Chercheurs : Qinjie JU, Stéphane DERRODE and René CHALON

Institution : Ecole Centrale de Lyon

Contact : qinjie.ju@doctorant.ec-lyon.fr

Sujet de l'étude : Nous (Qinjie JU, Stéphane DERRODE and René CHALON) effectuons un test d'utilisabilité pour le 1er prototype d'EyeMusic. Veuillez lire attentivement ce formulaire de consentement et poser autant de questions que vous le souhaitez avant de décider si vous voulez participer à cette étude de recherche. Vous êtes libre de poser des questions à tout moment avant, pendant ou après votre participation à cette recherche.

1. BUT DE CETTE ETUDE

Dans cette étude, nous souhaitons mieux comprendre la facilité d'utilisation du 1er prototype de l'application EyeMusic. Plus précisément, nous nous intéressons au taux d'erreur d'utilisation du prototype ce qui comprend les erreurs de détection de mouvement et la difficulté à se souvenir de la correspondance entre mouvements de la tête et commandes.

2. CONFIDENTIALITE

o Votre identité dans cette étude sera traitée comme confidentielle. Les résultats de l'étude peuvent être publiés à des fins scientifiques, mais ne communiqueront pas votre nom ou n'incluront aucune référence identifiable.

o Tous les enregistrements ou données obtenus à la suite de votre participation à cette étude peuvent être partagés en ligne avec la communauté scientifique académique, gratuitement. Les données partagées seront anonymes et ne donneront pas votre nom ou n'incluront aucune référence permettant de vous identifier.

o Aucun enregistrement ou donnée obtenu à la suite de votre participation à cette étude ne sera vendu à une tierce partie. Tous les enregistrements ou données obtenus seront utilisés uniquement à des fins de recherche scientifique.

3. FIN DE L'ÉTUDE

Vous êtes libre de choisir de participer ou non à cette étude. Il n'y aura aucune pénalité si vous choisissez de ne pas participer. Vous recevrez de nouvelles informations significatives au cours de cette étude qui peuvent avoir un lien ou influencer votre volonté de continuer à participer. Si vous décidez d'interrompre votre participation à l'étude, veuillez aviser Qinjie JU de votre décision, de sorte que votre participation puisse être terminée correctement.

4. AUTORISATION

Je certifie sur l'honneur que j'ai lu et compris ce formulaire de consentement, que j'ai lu et compris la description de l'expérience et que je me porte volontaire pour participer à cette étude.

Nom : _____ Date : _____

Signature :

Consentement du parent ou du tuteur légal, si la personne participant à l'étude est mineure

Je consens et j'accepte, individuellement et en tant que parent ou tuteur légal du mineur nommé ci-dessus, aux termes et dispositions précédents.

Signature :

Lien de parenté :

A.3 Post-test Questionnaire

Genre : homme femme Âge : _____

Niveau de connaissance de la notation musicale :

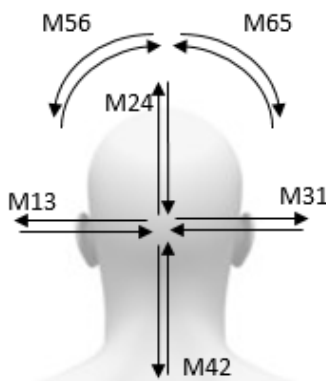
aucun élémentaire moyen bon excellent

Niveau d'étude:

école primaire collège lycée licence

master doctorat

1) Si notre application possède une seule commande (jouer une mesure), lequel mouvement de la tête préférez-vous pour déclencher cette commande ?



- M24 (mouvement haut- bas)
- M42 (mouvement bas-haut)
- M13 (mouvement gauche-droite)
- M31 (mouvement droite-gauche)
- M56 (inclinaison gauche-droite)
- M65 (inclinaison droite-gauche)

Annexe A. Annexe Relative à l'Evaluation de EyeMusic

2) D'après vous, quelle est l'utilité des 6 commandes que nous avons choisies pour notre prototype ?

	Pas du tout utile				Vraiment utile
Jouer une mesure:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
Jouer à partir d'une mesure :	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
Augmenter le volume:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
Baisser le volume:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
Arrêter:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5
Reprendre:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1	2	3	4	5

3) D'après vous, la correspondance mouvement-commande que nous utilisons dans ce test est appropriée ou pas ? Sinon, quel est la correspondance préférez-vous ?

4) Quelles sont les autres commandes que vous aimeriez ajouter dans notre application afin de mieux aider les élèves de la musique ? Quels sont les mouvements que vous aimeriez utiliser pour déclencher chaque commande ?

Annexe A. Annexe Relative à l'Evaluation de EyeMusic

5) D'après vous, sera cette application utile pour votre apprentissage de la musique ?

6) Est-ce cela vous intéresse de participer aux autres tests au niveau de notre application ?

A.4 SUS : System Usability Scale

	Pas du tout d'accord					Tout à fait d'accord
1. Je pense que je vais utiliser ce service fréquemment.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
2. Je trouve ce service inutilement complexe.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
3. Je pense que ce service est facile à utiliser.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
4. Je pense que j'aurai besoin de l'aide d'un technicien pour être capable d'utiliser ce service.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
5. J'ai trouvé que les différentes fonctions de ce service ont été bien intégrées.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
6. Je pense qu'il y a trop d'incohérence dans ce service.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
7. J'imagine que la plupart des gens serait capable d'apprendre à utiliser ce service très rapidement.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
8. J'ai trouvé ce service très lourd à utiliser.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
9. Je me sentais très en confiance en utilisant ce service.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
10. J'ai besoin d'apprendre beaucoup de choses avant de pouvoir utiliser ce service.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	

A.5 Fiche de Recueil pour l'Evaluation du Prototype EyeMusic

Date _____ Heure _____

Scénario 1 :

	N° Séquence	Etape 1	Etape 2	Etape 3
M13				
M31				
M24				
M42				
M56				
M65				

Scénario 2 :

(A)

Jouer la Partition (M42)	Mesure Choisie	Geste fait	Geste Détecté	Mesure Déclenchée	Consulter le Tableau ou Demander
	3				
	7				
Arrêt (M13)	Geste fait		Geste Détecté		Consulter le Tableau ou Demander
Reprise (M31)	Geste fait		Geste Détecté		Consulter le Tableau ou Demander

Annexe A. Annexe Relative à l'Evaluation de EyeMusic

(B)

Jouer la Partition (M42)	Mesure Choisie	Geste fait	Geste Détecté	Mesure Déclenchée	Consulter le Tableau ou Demander
	2				
	4				
Baisser le Volume (M56)	Geste fait		Geste Détecté		Consulter le Tableau ou Demander
Augmenter le Volume (M65)	Geste fait		Geste Détecté		Consulter le Tableau ou Demander

(C)

Jouer une mesure (M24)	Mesure Choisie	Geste fait	Geste Détecté	Mesure Déclenchée	Consulter le Tableau ou Demander
	1				
	5				
	6				

A.6 Participation à l'Evaluation du Prototype EyeMusic

N° du Sujet	Participer à la Phase d'apprentissage	Participer à la Phase Opérationnelle	Remplir le Questionnaire et le SUS
U01	Oui	Oui	Oui
U02	Oui	Oui	Oui
U03	Oui	Oui	Oui
U04	Oui	Oui	Oui
U05	Non	Non	Non
U06	Oui	Oui	Oui
U07	Non	Non	Non
U08	Oui	Oui	Oui
U09	Oui	Oui	Oui
U10	Non	Non	Non
U11	Oui	Non	Oui
U12	Non	Non	Non
U13	Oui	Non	Oui
U14	Oui	Oui	Oui
U15	Oui	Oui	Oui
U16	Oui	Oui	Oui
U17	Oui	Oui	Oui
U18	Oui	Oui	Oui
U19	Non	Non	Non
U20	Non	Non	Non
U21	Oui	Oui	Oui
U22	Oui	Oui	Oui
U23	Oui	Oui	Oui
U24	Oui	Oui	Oui
U25	Oui	Oui	Oui

Les sujets au fond gris sont les volontaires qui étaient incapables de participer à notre expérience.

Annexe A. Annexe Relative à l'Evaluation de EyeMusic

U26	Oui	Non	Oui
U27	Oui	Oui	Oui
U28	Oui	Oui	Oui
U29	Oui	Oui	Oui
U30	Oui	Oui	Oui
U31	Oui	Oui	Oui
U32	Oui	Oui	Oui
U33	Non	Non	Non
U34	Non	Non	Non
U35	Oui	Oui	Oui
U36	Oui	Oui	Oui
U37	Oui	Oui	Oui
U38	Oui	Oui	Oui
U39	Oui	Oui	Oui
U40	Oui	Oui	Oui
U41	Non	Non	Non
U42	Oui	Oui	Oui
U43	Non	Non	Non
U44	Oui	Non	Oui
U45	Oui	Oui	Oui
U46	Non	Non	Non
U47	Oui	Non	Oui
U48	Non	Non	Non
U49	Non	Non	Non
U50	Non	Non	Non
U51	Oui	Non	Non
U52	Oui	Non	Non
U53	Non	Non	Non
U54	Oui	Oui	Oui
U55	Oui	Oui	Oui
U56	Oui	Oui	Oui

Liste des tableaux

2.1	Classification des interactions du regard.	22
2.2	Propriétés des modalités de pointage d'après [20].	26
3.1	Comparaison des différentes modalités pour le déclenchement des commandes en combinaison avec un eye-tracker porté sur la tête. . .	52
4.1	Matrice de Confusion des 6 Mouvements.	82
4.2	Distribution du taux de réussite ajusté de la phase d'apprentissage pour les participants avec niveaux de connaissance de la notation musicale différents.	85
4.3	Taux de réussite brut et contribution au score du SUS de la pre- mière question pour les participants avec des niveaux différents de connaissance de la notation musicale.	93
4.4	Supposition d'utilité de EyeMusic pour les participants avec des ni- veaux de connaissance de la notation musicale différents.	96

Table des figures

1.1	Le schéma simplifié de l'œil.	1
1.2	Exemple de EOG : les électrodes sont fixées sur la peau autour des yeux.	6
1.3	Exemples d'eye-trackers : (a) eye-tracker stationnaire placé en-dessous d'un écran, (b) eye-tracker portable sous forme d'une paire de lunettes.	7
1.4	L'Eye Tracking Glasses 2 Wireless de SMI. Les deux caméras des yeux filment les mouvements des yeux tandis que la caméra de la scène capture l'image regardée.	8
2.1	Le système EagleEyes : (a) le premier écran des 2 niveaux , (b) l'écran après la sélection du groupe 'ABCDE'.	16
2.2	Application EyeDraw avec un dessin fait par l'un des auteurs.	17
2.3	Un système d'eye-tracking proposé par Tobii pour les personnes à mobilité réduite.	18
2.4	Une capture d'écran de GazeGalaxy.	19
2.5	EyeTuner composé d'un haut-parleur et d'un ECS digital.	20
2.6	Deux applications de SideWays : (a) navigateur de couverture d'albums, (b) quiz du regard.	23
2.7	EyeGuide assiste l'utilisateur en station de métro pour la recherche de la bonne route.	24
2.8	Système de traduction à l'aide d'un eye-tracker porté sur la tête avec une caméra frontale.	25
2.9	Exemple de EyeBook : lorsque l'utilisateur lit une annotation, l'événement correspondant se déclenche.	26
2.10	Exemple de séparation de la cible à sélectionner en 2 parties : le nom de la commande et la zone de sélection.	28
2.11	Sélection par deux fixations successives adoptée dans NeoVisus.	29

2.12	Système MDITIM : (a) emplacement des cibles, (b) exemples des interprétations 2D des caractères.	30
2.13	Système EyeWrite : (a) fenêtre dédiée à l'exécution et à l'interprétation des gestes, (b) interprétations des caractères.	31
2.14	Système Eye-S : (a) exemples des séquences des mouvements des yeux pour 'a' et 'b', (b) affichage explicite des cibles.	31
2.15	Exemples de séquences d'un mouvement d'œil non symbolique.	32
2.16	Un utilisateur est intéressé par un album, il suit le mouvement de cet album sur l'écran par ses yeux ; un extrait de la musique est alors joué automatiquement.	33
2.17	Les zooms et les pans lors de la sélection de la lettre 's'.	34
2.18	Exemple de l'augmentation du volume d'un lecteur de musique d'une montre intelligente.	35
2.19	Quatre prototypes d'AmbiGaze : (1) lecteur de la musique (virtuel+externe), (2) interface vidéo-sur-demande (virtuel+interne), (3) lampe (mécanique+interne), (4) ventilateur (mécanique+externe).	36
2.20	Représentation schématique des informations issues du geste piche- nette et du regard, la ligne (f) représente la direction du geste, la ligne (g) représente le regard de l'utilisateur.	39
2.21	Rotations du poignet selon les 3 axes.	40
2.22	Quatre types de rotation de cheville.	42
2.23	Lignes de seuil pour le défilement de la page.	45
2.24	La conception de l'application GeoGazemarks.	46
3.1	Les mouvements volontaires de la tête à regard fixe choisis.	54
3.2	Variation du point du regard lors des mouvements verticaux de la tête à regard fixe.	56
3.3	Variations des coordonnées du point du regard pendant un mouve- ment M24 à regard fixe : (a) variation de la coordonnée y, (b) dérivée de la coordonnée y, (c) variation de la coordonnée y relative.	57

Table des figures

3.4	Différences sur la coordonnée y relative entre les 3 cas qui peut entraîner une variation des coordonnées du regard souhaitée : (a) coordonnée y relative pendant un mouvement de la tête à regard fixe, (b) coordonnée y relative au cours d'un pur mouvement oculaire, (c) coordonnée y relative lorsque les yeux et la tête de l'utilisateur bougent en même temps.	58
3.5	Exemple de la variation du point du regard pendant le dodelinement de la tête.	61
4.1	Configuration typique d'EyeMusic pour apprendre à jouer du piano.	65
4.2	Exemple de la partition et de sa vue sur l'image de scène : (a) exemple de la partition avec 2 marqueurs ArUco, (b) exemple de l'image de la scène avec le point du regard superposé alors qu'une partie de la partition se situe en dehors du champs de vision de la caméra de la scène.	68
4.3	Vue d'ensemble de l'algorithme principal d'EyeMusic.	70
4.4	Diagramme schématique de la correspondance entre les mouvements et les commandes.	71
4.5	Configuration expérimentale.	72
4.6	Nombre d'essais réussis après 1, 2, 3, 4, 5 tentatives ou échoués pour chaque mouvement de la tête à regard fixe.	79
4.7	Histogramme du taux de réussite pour jouer la bonne mesure.	80
4.8	Histogramme du taux de réussite pour jouer la bonne mesure avec le bon mouvement détecté.	81
4.9	Nombre d'essais réussis après 1, 2, 3, 4, 5 tentatives ou échoués pour chaque mouvement de la tête à regard fixe.	82
4.10	Histogramme du taux de réussite des algorithmes de la reconnaissance des mouvements de la tête.	84
4.11	Nombre d'essais réussis après 1, 2, 3, 4, 5 mouvements effectifs ou échoués pour chaque mouvement de la tête à regard fixe.	85

4.12 Taux de réussite ajusté de la phase d'apprentissage pour les participants avec différents niveaux de connaissance de la notation musicale.	86
4.13 Histogramme du taux de réussite pour la phase opérationnelle.	87
4.14 Pourcentage d'essais réussis après 1, 2, 3, 4, 5 tentatives ou échoués pour chaque mouvement de la tête à regard fixe pour la phase opérationnelle.	87
4.15 Préférences du mouvement de la tête à regard fixe des participants.	90
4.16 Histogramme du score pour SUS.	91
4.17 Relation entre le score SUS et le taux de réussite brut.	92
4.18 Utilité de chaque commande notée par les participants.	94
4.19 Commandes supplémentaires proposées par les participants.	94
5.1 Lignes de seuil pour le défilement de la page.	101
5.2 Diagramme schématique de la correspondance entre les mouvements de la tête et les commandes de EyeRecipe.	103
5.3 Diagramme schématique pour la capture d'images à traiter par EyeRecipe.	105
5.4 Image réellement traitée.	106
5.5 Vue d'ensemble de l'algorithme principal d'EyeRecipe.	109
5.6 La commande correspond au M42 d'EyeRecipe.	110

Bibliographie

- [1] Jason Alexander, Teng Han, William Judd, Pourang Irani, and Sriram Subramanian. Putting your best foot forward : investigating real-world mappings for foot-based gestures. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 1229–1238. ACM, 2012. [42](#)
- [2] Vasileios Athanasios Anagnostopoulos and Peter Kiefer. Towards gaze-based interaction with urban outdoor spaces. In *Proc. of the 2016 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing : Adjunct*, pages 1706–1715, Heidelberg, Germany, September 2016. ACM. [21](#), [23](#)
- [3] Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Youmove : enhancing movement training with an augmented reality mirror. In *Proc. of the 26th annual ACM Symp. on User interface software and technology*, pages 311–320. ACM, 2013. [44](#)
- [4] A Terry Bahill, Michael R Clark, and Lawrence Stark. The main sequence, a tool for studying human eye movements. *Mathematical Biosciences*, 24(3-4) : 191–204, 1975. [2](#)
- [5] David Beymer and Daniel M Russell. Webgazeanalyzer : a system for capturing and analyzing web reading behavior using eye gaze. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1913–1916. ACM, 2005. [45](#)
- [6] Ralf Biedert, Georg Buscher, and Andreas Dengel. The eyebook—using eye tracking to enhance the reading experience. *Informatik-Spektrum*, 33(3) : 272–281, 2010. [25](#)
- [7] Pradipta Biswas and Patrick Langdon. Eye-gaze tracking based interaction in india. In *Proc. of the 6th Int. Conf. on Intelligent Human Computer Interaction*, pages 59–66, 2014. [44](#)
- [8] Dario Bonino, Emiliano Castellina, Fulvio Corno, A Gale, Alessandro Garbo, Kevin Purdy, and Fangmin Shi. A blueprint for integrated eye-controlled

- environments. *Universal Access in the Information Society*, 8(4) : 311, 2009. 20, 106
- [9] Stephen Brewster, Joanna Lumsden, Marek Bell, Malcolm Hall, and Stuart Tasker. Multimodal 'eyes-free' interaction techniques for wearable devices. In *Proc. of the SIGCHI Conf. on Human factors in computing systems*, pages 473–480. ACM, 2003. 43
- [10] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. Wearable EOG Goggles : Eye-based interaction in everyday environments. In *CHI Conf. on Human Factors in Computing Systems*, pages 3252–3264, Boston, USA, 2009. ACM. 32
- [11] Roger Carpenter. *Movements of the Eyes, 2nd Rev.* Pion Limited, 1988. 2
- [12] Roger Carpenter. The visual origins of ocular motility. *Vision and visual dysfunction*, 8 : 1–10, 1991. 3
- [13] Tom N Cornsweet and Hewitt D Crane. Accurate two-dimensional eye tracker using first and fourth purkinje images. *JOSA*, 63(8) : 921–928, 1973. 5
- [14] Andrew Crossan, Stephen Brewster, and Alexander Ng. Foot tapping for mobile interaction. In *Proc. of the 24th BCS Interaction Specialist Group Conf.*, pages 418–422. British Computer Society, 2010. 41
- [15] Andrew Crossan, Mark McGill, Stephen Brewster, and Roderick Murray-Smith. Head tilting for interaction in mobile contexts. In *Proc. of the 11th Int. Conf. on Human-Computer Interaction with Mobile Devices and Services*, page 6. ACM, 2009. 43
- [16] Andrew Crossan, John Williamson, Stephen Brewster, and Rod Murray-Smith. Wrist rotation for interaction in mobile contexts. In *Proc. of the 10th Int. Conf. on Human computer interaction with mobile devices and services*, pages 435–438. ACM, 2008. 40
- [17] Jakob De Lemos, Golam Reza Sadeghnia, Íris Ólafsdóttir, and Ole Jensen. Measuring emotions using eye tracking. In *Proc. of measuring behavior*, volume 226, pages 225–226, 2008. 10

Bibliographie

- [18] William Delamare. *Interaction à distance en environnement physique augmentée*. PhD thesis, Grenoble Alpes, 2015. [44](#)
- [19] Raymond Dodge and Thomas Sparks Cline. The angular velocity of eye movements. *Psychological Review*, 8 : 145–157, 1901. [4](#)
- [20] Heiko Drewes. *Eye gaze tracking for human computer interaction*. PhD thesis, Ludwig Maximilian University of Munich, 2010. [2](#), [4](#), [5](#), [9](#), [15](#), [18](#), [26](#), [133](#)
- [21] Heiko Drewes, Alexander De Luca, and Albrecht Schmidt. Eye-gaze interaction for mobile phones. In *Proc. of the 4th Int. Conf. on mobile technology, applications, and systems and the 1st Int. Symp. on Computer human interaction in mobile technology*, pages 364–371. ACM, 2007. [23](#), [32](#)
- [22] Heiko Drewes and Albrecht Schmidt. Interacting with the computer using gaze gestures. In *IFIP Conf. on Human-Computer Interaction*, pages 475–488, Rio de Janeiro, Brazil, 2007. Springer. [32](#)
- [23] Heiko Drewes and Albrecht Schmidt. The MAGIC touch : Combining magic-pointing with a touch-sensitive mouse. In *IFIP Conf. on Human-Computer Interaction*, pages 415–428, Uppsala, Sweden, 2009. Springer. [37](#), [38](#)
- [24] Andrew T Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, 34(4) : 455–470, 2002. [27](#)
- [25] Andrew T Duchowski. Eye tracking methodology. *Theory and practice*, 328 : 614, 2007. [2](#), [3](#), [29](#)
- [26] Marc Eaddy, Gabor Blasko, Jason Babcock, and Steven Feiner. My own private kiosk : Privacy-preserving public displays. In *Wearable Computers, 2004. ISWC 2004. Eighth Int. Symp. on*, volume 1, pages 132–135. IEEE, 2004. [23](#), [24](#)
- [27] Ralf Engbert and Konstantin Mergenthaler. Microsaccades are triggered by low retinal image slip. *Proc. of the National Academy of Sciences*, 103(18) : 7192–7197, 2006. [2](#)
- [28] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. Orbits : Gaze interaction for smart watches using smooth pursuit eye move-

- ments. In *Proc. of the 28th Annual ACM Symp. on User Interface Software & Technology*, pages 457–466, Charlotte, North Carolina, USA, 2015. ACM. 35
- [29] Assaf Feldman, Emmanuel Munguia Tapia, Sajid Sadi, Pattie Maes, and Chris Schmandt. Reachmedia : On-the-move interaction with everyday objects. In *Ninth IEEE Int. Symp. on Wearable Computers (ISWC'05)*, pages 52–59. IEEE, 2005. 40
- [30] David Fono and Roel Vertegaal. Eyewindows : evaluation of eye-controlled zooming windows for focus selection. In *Proc. of the SIGCHI Conf. on Human factors in computing systems*, pages 151–160. ACM, 2005. 18, 37
- [31] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo : Fast semi-direct monocular visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 15–22. IEEE, 2014. 108
- [32] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6) : 2280–2292, 2014. 67
- [33] Ioannis Giannopoulos, Peter Kiefer, and Martin Raubal. Geogazemarks : providing gaze history for the orientation on small display maps. In *Proc. of the 14th ACM Int. Conf. on Multimodal interaction*, pages 165–172. ACM, 2012. 46
- [34] Ioannis Giannopoulos, Peter Kiefer, and Martin Raubal. Gazenav : Gaze-based pedestrian navigation. In *Proc. of the 17th Int. Conf. on Human-Computer Interaction with Mobile Devices and Services*, pages 337–346, Copenhagen, Denmark, 2015. ACM. 23
- [35] James Gips and Peter Olivieri. Eagleeyes : An eye control system for persons with disabilities. In *The Eleventh Int. Conf. on Technology and Persons with Disabilities*, pages 1–15, 1996. 16
- [36] Etienne Grandjean and Karl HE Kroemer. *Fitting the task to the human : a textbook of occupational ergonomics*. CRC press, 1997. 40

Bibliographie

- [37] Jeremy Hales, David Rozado, and Diako Mardanbegi. Interacting with objects in the environment by gaze and hand gestures. In *Proc. of the 3rd Int. workshop on pervasive eye tracking and mobile eye-based interaction*, pages 1–9, 2013. 44
- [38] Teng Han, Jason Alexander, Abhijit Karnik, Pourang Irani, and Sriram Subramanian. Kick : investigating the use of kick gestures for mobile interactions. In *Proc. of the 13th Int. Conf. on Human Computer Interaction with Mobile Devices and Services*, pages 29–32. ACM, 2011. 42
- [39] Dan Witzner Hansen, Henrik HT Skovsgaard, John Paulin Hansen, and Emilie Mollenbach. Noise tolerant selection by gaze-controlled pan and zoom in 3D. In *Proc. of the 2008 Symp. on Eye tracking research & applications*, pages 205–212. ACM, 2008. 33
- [40] Hamilton Hartridge and LC Thomson. Methods of investigating eye movements. *The British journal of ophthalmology*, 32(9) : 581, 1948. 4
- [41] Franz Hatfield. System and method for controlling host system interface with point-of-interest data, June 5 2001. US Patent 6,243,076. 2, 33, 37
- [42] Anthony Hornof, Anna Cavender, and Rob Hoselton. Eyedraw : a system for drawing pictures with eye movements. In *ACM SIGACCESS Accessibility and Computing*, number 77-78, pages 86–93. ACM, 2004. 16, 27
- [43] Thomas E Hutchinson, K Preston White, Worthy N Martin, Kelly C Reichert, and Lisa A Frey. Human-computer interaction using eye-gaze input. *IEEE Transactions on systems, man, and cybernetics*, 19(6) : 1527–1534, 1989. 16
- [44] Poika Isokoski. Text input methods for eye trackers using off-screen targets. In *Proc. of the 2000 Symp. on Eye tracking research & applications*, pages 15–21. ACM, 2000. 30
- [45] Poika Isokoski and Roope Raisamo. Device independent text input : A rationale and an example. In *Proc. of the working Conf. on Advanced visual interfaces*, pages 76–83. ACM, 2000. 30
- [46] Howell Istance, Aulikki Hyrskykari, Lauri Immonen, Santtu Mansikkamaa, and Stephen Vickers. Designing gaze gestures for gaming : an investigation

- of performance. In *Proc. of the 2010 Symp. on Eye-Tracking Research & Applications*, pages 323–330. ACM, 2010. 29
- [47] Robert JK Jacob. What you look at is what you get : eye movement-based interaction techniques. In *Proc. of the SIGCHI Conf. on Human factors in computing systems*, pages 11–18. ACM, 1990. 27
- [48] Robert JK Jacob. Eye tracking in advanced interface design. *Virtual environments and advanced interface design*, pages 258–288, 1995. 3
- [49] Robert JK Jacob and Keith S Karn. Eye tracking in human-computer interaction and usability research : Ready to deliver the promises. In *The mind's eye*, pages 573–605. Elsevier, 2003. 4, 5
- [50] Qinjie Ju, René Chalon, and Stéphane Derrode. Fixed-gaze head movement detection for triggering commands. In *Workshop on Models and Analysis of Eye Movements*, Grenoble, France, 2018. 49
- [51] Qinjie Ju, René Chalon, and Stéphane Derrode. Assisted music score reading using fixed-gaze head movement : Empirical experiment and design implications. *Proc. of the ACM on Human-Computer Interaction*, 3(EICS) : 29p., 2019. 56, 65
- [52] Qinjie Ju, Stéphane Derrode, and René Chalon. Utilisation de l’eye-tracking pour l’interaction mobile dans un environnement réel augmenté. In *29ème conférence francophone sur l’Interaction Homme-Machine*, page 5p., Poitiers, France, 2017. AFIHM. Rencontres doctorales. 65
- [53] Charles H Judd, Cloyd N McAllister, and WM Steele. General introduction to a series of studies of eye movements by means of kinetoscopic photographs. *Psychological Review Monographs*, 7(1) : 1–16, 1905. 4
- [54] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Di Marca. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*, 10(5) : 285–299, 2006. 39
- [55] Peter Kiefer, Ioannis Giannopoulos, Dominik Kremer, Christoph Schlieder, and Martin Raubal. Starting to get bored : An outdoor eye tracking study

Bibliographie

- of tourists exploring a city panorama. In *Proc. of the Symp. on Eye Tracking Research and Applications*, pages 315–318. ACM, 2014. 47
- [56] Rick Kjeldsen. Head gestures for computer control. In *Proc. of IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 61–67. IEEE, 2001. 43
- [57] Takuya Kobayashi, Takumi Toyamaya, Faisal Shafait, Masakazu Iwamura, Koichi Kise, and Andreas Dengel. Recognizing words in scenes with a head-mounted eye-tracker. In *10th IAPR Int. Workshop on Document Analysis Systems (DAS)*, pages 333–338. IEEE, 2012. 25
- [58] David B Koons, Carlton J Sparrell, and Kristinn Rr Thorisson. Integrating simultaneous input from speech, gaze, and hand gestures. *MIT Press : Menlo Park, CA*, pages 257–276, 1993. 43
- [59] Richard J Krauzlis and Steve G Lisberger. Temporal properties of visual motion signals for the initiation of smooth pursuit eye movements in monkeys. *Journal of Neurophysiology*, 72(1) : 150–162, 1994. 3
- [60] Christine Kühnel, Tilo Westermann, Fabian Hemmert, Sven Kratz, Alexander Müller, and Sebastian Möller. I’m home : Defining and evaluating a gesture set for smart-home control. *Int. Journal of Human-Computer Studies*, 69(11) : 693–704, 2011. 44
- [61] Chandan Kumar, Raphael Menges, and Steffen Staab. Eye-controlled interfaces for multimedia interaction. *IEEE MultiMedia*, 23(4) : 6–13, 2016. 2, 45
- [62] Manu Kumar, Terry Winograd, Terry Winograd, and Andreas Paepcke. Gaze-enhanced scrolling techniques. In *CHI’07 Extended Abstracts on Human Factors in Computing Systems*, pages 2531–2536. ACM, 2007. 45, 100
- [63] Angela Kwan. 6 benefits of music lessons. <https://www.parents.com/kids/development/intellectual/6-benefits-of-music-lessons/>. Accessed March 21, 2018. 65

- [64] Hubert W Lilliefors. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association*, 62(318) : 399–402, 1967. 84
- [65] David G Lowe. Distinctive image features from scale-invariant keypoints. *Int. journal of computer vision*, 60(2) : 91–110, 2004. 20
- [66] Jane F Mackworth and NH Mackworth. Eye fixations recorded on changing visual scenes by the television eye-marker. *JOSA*, 48(7) : 439–445, 1958. 4, 5
- [67] Paul P Maglio, Teenie Matlock, Christopher S Campbell, Shumin Zhai, and Barton A Smith. Gaze and speech in attentive user interfaces. In *Advances in Multimodal Interfaces ICMI 2000*, pages 1–7. Springer, 2000. 19
- [68] Päivi Majaranta and Kari-Jouko Räihä. Twenty years of eye typing : systems and design issues. In *Proc. of the 2002 Symp. on Eye tracking research & applications*, pages 15–22. ACM, 2002. 15, 27
- [69] Diako Mardanbegi, Dan Witzner Hansen, and Thomas Pederson. Eye-based head gestures. In *Proc. of the Symp. on eye tracking research and applications*, pages 139–146. ACM, 2012. 50, 54, 55
- [70] Michael Mauderer, Florian Daiber, and Antonio Krüger. Combining touch and gaze for distant selection in a tabletop setting. In *Proc. of the Workshop on Gaze Interaction in the Post-WIMP World - ACM SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2013. 38
- [71] Apurva Mehta and Malay Bhatt. Practical issues in the field of optical music recognition. *IJARCSMS*, 2 : 513–518, 2014. 67
- [72] Slavko Milekic. The more you look the more you get : Intention-based interface using gaze-tracking. *Museums and the Web 2002 Selected Papers from an Int. Conf., Archives Museum Informatics*, 2003. 31
- [73] Emilie Mollenbach, John Paulin Hansen, and Martin Lillholm. Eye movements in gaze interaction. *Journal of Eye Movement Research*, 6(2), 2013. 3, 27, 29, 32, 36
- [74] Louis-Philippe Morency and Trevor Darrell. Head gesture recognition in intelligent interfaces : the role of context in improving recognition. In *Proc. of*

Bibliographie

- the 11th Int. Conf. on Intelligent user interfaces*, pages 32–38. ACM, 2006. 43
- [75] Shunji Mori, Hirobumi Nishida, and Hiromitsu Yamada. *Optical character recognition*. John Wiley & Sons, Inc., 1999. 25, 102, 105
- [76] Tomi Nukarinen, Jari Kangas, Oleg Špakov, Poika Isokoski, Deepak Akkil, Jussi Rantala, and Roope Raisamo. Evaluation of headturn : An interaction technique using the gaze and head turns. In *Proc. of the 9th Nordic Conf. on Human-Computer Interaction*, page 43. ACM, 2016. 50, 54
- [77] Ian Oakley and Sile O’Modhrain. Tilt to scroll : Evaluating a motion based vibrotactile mobile interface. In *Eurohaptics Conf. and Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 40–49. IEEE, 2005. 40
- [78] Ian Oakley and Jun-Seok Park. Designing eyes-free interaction. In *Int. Workshop on Haptic and Audio Interaction Design*, pages 121–132. Springer, 2007. 41
- [79] Alice Oh, Harold Fox, Max Van Kleek, Aaron Adler, Krzysztof Gajos, Louis-Philippe Morency, and Trevor Darrell. Evaluating look-to-talk : a gaze-aware interface in a collaborative environment. In *CHI’02 Extended Abstracts on Human Factors in Computing Systems*, pages 650–651. ACM, 2002. 19
- [80] Takehiko Ohno. Features of eye gaze interface for selection tasks. In *Proc. of the 3rd Asia Pacific Conf. on Computer Human Interaction*, pages 176–181. IEEE, 1998. 28
- [81] Chirag Patel, Atul Patel, and Dharmendra Patel. Optical character recognition by open source OCR tool tesseract : A case study. *Int. Journal of Computer Applications*, 55(10) : 50–56, 2012. 105
- [82] Marco Porta, Stefania Ricotti, and Calet Jimenez Perez. Emotional e-learning through eye tracking. In *Proc. of the 2012 IEEE Global Engineering Education Conf. (EDUCON)*, pages 1–6. IEEE, 2012. 10

- [83] Marco Porta and Matteo Turina. Eye-s : a full-screen input modality for pure eye-based communication. In *Proc. of the 2008 Symp. on Eye tracking research & applications*, pages 27–34. ACM, 2008. 31
- [84] Mahfuz Rahman, Sean Gustafson, Pourang Irani, and Sriram Subramanian. Tilt techniques : investigating the dexterity of wrist-based input. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 1943–1952. ACM, 2009. 41
- [85] C1 Rashbass. The relationship between saccadic and smooth tracking eye movements. *The Journal of Physiology*, 159(2) : 326–338, 1961. 3
- [86] Keith Rayner. Eye movements in reading and information processing : 20 years of research. *Psychological bulletin*, 124(3) : 372–421, 1998. 66
- [87] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre RS Marcal, Carlos Guedes, and Jaime S Cardoso. Optical music recognition : state-of-the-art and open issues. *Int. Journal of Multimedia Information Retrieval*, 1(3) : 173–190, 2012. 67
- [88] Adrian Reetz, Carl Gutwin, Tadeusz Stach, Miguel Nacenta, and Sriram Subramanian. Superflick : a natural and efficient technique for long-distance object placement on digital tables. In *Proc. of Graphics interface 2006*, pages 163–170. Canadian Information Processing Society, 2006. 38
- [89] Kerstin Ruhland, Sean Andrist, Jeremy Badler, Christopher Peters, Norman Badler, Michael Gleicher, Bilge Mutlu, and Rachel McDonnell. Look me in the eyes : A survey of eye and gaze animation for virtual agents and artificial systems. In *Eurographics state-of-the-art report*, pages 69–91, 2014. 3
- [90] Jacek Ruminski, Adam Bujnowski, Jerzy Wtorek, Aliaksei Andrushevich, Martin Biallas, and Rolf Kistler. Interactions with recognized objects. In *7th Int. Conf. on Human System Interactions (HSI), 2014*, pages 101–105. IEEE, 2014. 21
- [91] Johannes Schöning, Florian Daiber, Antonio Krüger, and Michael Rohs. Using hands and feet to navigate and manipulate spatial data. In *CHI'09 Extended*

Bibliographie

- Abstracts on Human Factors in Computing Systems*, pages 4663–4668. ACM, 2009. 42
- [92] Jeremy Scott, David Dearman, Koji Yatani, and Khai N Truong. Sensing foot gestures from the pocket. In *Proc. of the 23rd annual ACM Symp. on User interface software and technology*, pages 199–208. ACM, 2010. 42
- [93] Jeffrey S Shell, Ted Selker, and Roel Vertegaal. Interacting with groups of computers. *Communications of the ACM*, 46(3) : 40–46, 2003. 19
- [94] Fangmin Shi, Alastair G Gale, and Kevin Purdy. Direct gaze based environmental controls. In *The 2nd Conf. on Communication by Gaze Interaction*, pages 36–41. COGAIN, 2006. 37, 38
- [95] Fangmin Shi, Alastair G Gale, and Kevin Purdy. Eye-centric ICT control. *Contemporary Ergonomics : Proc. of the Ergonomics Society Annual Conf.*, pages 215–218, 2006. 20
- [96] Linda E Sibert and Robert JK Jacob. Evaluation of eye gaze interaction. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 281–288. ACM, 2000. 17
- [97] Oleg Špakov and Päivi Majaranta. Enhanced gaze interaction using simple head gestures. In *Proc. of the 2012 ACM Conf. on Ubiquitous Computing*, pages 705–710. ACM, 2012. 50, 54
- [98] Dave M Stampe and Eyal M Reingold. Selection by looking : A novel computer interface and its application to psychological research. In *Studies in visual information processing*, volume 6, pages 467–478. Elsevier, 1995. 17
- [99] Sophie Stellmach, Sebastian Stober, Andreas Nürnberger, and Raimund Dachsel. Designing gaze-supported multimodal interactions for the exploration of large image collections. In *Proc. of the 1st Conf. on novel gaze-controlled applications*. ACM, 2011. 18, 37
- [100] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual SLAM algorithms : A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1) : 16, 2017. 108

- [101] M Tall. Neovisus : Gaze driven interface components. In *Proc. of the 4rd Conf. on Communication by Gaze Interaction (COGAIN 2008)*, pages 47–51, 2008. 28
- [102] Desney Tan and Anton Nijholt. Brain-computer interfaces and human-computer interaction. In *Brain-Computer Interfaces*, pages 3–19. Springer, 2010. 37
- [103] JH Ten Kate, Edward EE Frietman, W Willems, BM Ter Haar Romeny, and E Tenkink. Eye-switch controlled communication aids. In *Proc. of the 12th Int. Conf. on Medical & Biological Engineering*, pages 19–20, 1979. 16
- [104] HM Tong and RA Fisher. Progress report on an eye-slaved area-of-interest visual display. Technical Report No. AFHRL-TR-84-36, Air Force Human Resources Laboratory, 1984. 45
- [105] Takumi Toyama, Thomas Kieninger, Faisal Shafait, and Andreas Dengel. Museum guide 2.0-an eye-tracking based personal assistant for museums and exhibits. In *Proc. of Int. Conf. on Re-Thinking Technology in Museums*, volume 1, 2011. 24
- [106] Koji Tsukada and Michiaki Yasumura. Ubi-finger : A simple gesture input device for mobile and ubiquitous environment. *Journal of Asian Information, Science and Life (AISL)*, 2(2) : 111–120, 2004. 41
- [107] Jayson Turner, Andreas Bulling, and Hans Gellersen. Combining gaze with manual interaction to extend physical reach. In *Proc. of the 1st Int. workshop on pervasive eye tracking & mobile eye-based interaction*, pages 33–36. ACM, 2011. 37, 38
- [108] Radu-Daniel Vatavu. A comparative study of user-defined handheld vs. free-hand gestures for home entertainment environments. *Journal of Ambient Intelligence and Smart Environments*, 5(2) : 187–211, 2013. 44
- [109] Eduardo Velloso, Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. An empirical investigation of gaze selection in mid-air gestural 3D manipulation. In *Human-Computer Interaction*, pages 315–330. Springer, 2015. 37, 43

Bibliographie

- [110] Eduardo Velloso, Markus Wirth, Christian Weichel, Augusto Esteves, and Hans Gellersen. Ambigaze : Direct control of ambient devices by gaze. In *Proc. of the 2016 ACM Conf. on Designing Interactive Systems*, pages 812–817. ACM, 2016. 35
- [111] Roel Vertegaal, Aadil Mamuji, Changuk Sohn, and Daniel Cheng. Media eye-pliances : using eye tracking for remote control focus selection of appliances. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pages 1861–1864. ACM, 2005. 19
- [112] Mélodie Vidal, Ken Pfeuffer, Andreas Bulling, and Hans W Gellersen. Pursuits : eye-based interaction with moving targets. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 3147–3150. ACM, 2013. 33
- [113] Colin Ware and Harutune H Mikaelian. An evaluation of an eye tracker as a device for computer input2. In *ACM SIGCHI bulletin*, volume 17, pages 183–188. ACM, 1987. 17
- [114] Jacob O Wobbrock, Brad A Myers, and John A Kembel. Edgewrite : a stylus-based text entry method designed for high accuracy and stability of motion. In *Proc. of the 16th annual ACM Symp. on User interface software and technology*, pages 61–70. ACM, 2003. 30
- [115] Jacob O Wobbrock, James Rubinstein, Michael W Sawyer, and Andrew T Duchowski. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proc. of the 2008 Symp. on Eye tracking research & applications*, pages 11–18. ACM, 2008. 15, 30
- [116] Tetsuya Yamamoto, Masahiko Tsukamoto, and Tomoki Yoshihisa. Foot-step input method for operating information devices while jogging. In *Int. Symp. on Applications and the Internet*, pages 173–176. IEEE, 2008. 41
- [117] Alfred L Yarbus. *Eye movements and vision*. Springer, 1967. 3
- [118] Laurence R Young and David Sheena. Survey of eye movement recording methods. *Behavior research methods & instrumentation*, 7(5) : 397–429, 1975.

- [119] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 246–253. ACM, 1999. 27, 37
- [120] Yanxia Zhang, Andreas Bulling, and Hans Gellersen. Sideways : a gaze interface for spontaneous interaction with situated displays. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 851–860. ACM, 2013. 23
- [121] Yun Zhou, Tao Xu, Bertrand David, and René Chalon. Innovative wearable interfaces : an exploratory analysis of paper-based interfaces with camera-glasses device unit. *Personal and ubiquitous computing*, 18(4) : 835–849, 2014.

Bibliographie
