

Dispositif de capture des mouvements oculaires pour un simulateur de vision

Professeur de diplôme : Thierry Pun, Université de Genève (Thierry.Pun@cui.unige.ch)
Chef du projet : Jacques François, Ecole d'Ingénieurs de Genève, (Jacques.Francois@eig.unige.ch)
Diplômant : Julien Kronegg (jkronegg@bigfoot.com)

Table des matières

CHAPITRE 1: PRÉAMBULE	5
1.1 INTRODUCTION	5
1.2 REMERCIEMENTS	6
CHAPITRE 2: INTRODUCTION ET BUT	7
2.1 INTRODUCTION	7
2.2 BUT DU TRAVAIL DE DIPLÔME.....	10
CHAPITRE 3: ŒIL, ÉLECTRODES ET IMPLANTATION	11
3.1 INTRODUCTION	11
3.2 DESCRIPTION EXTERNE.....	11
3.3 DESCRIPTION INTERNE.....	12
3.4 MOUVEMENTS DE L'ŒIL	15
3.4.1 Programmation de la saccade.....	16
3.4.2 Excitation du muscle.....	16
3.4.3 Rotation du globe oculaire	16
3.4.4 Fixation.....	17
3.5 DESCRIPTION DES ÉLECTRODES ET IMPLANTATION	18
CHAPITRE 4: EYE TRACKERS – STATE OF THE ART	21
4.1 INTRODUCTION	21
4.2 DOMAINES D'UTILISATION.....	21
4.3 TECHNIQUES D'EYE TRACKING	22
4.3.1 Calcul de la position de l'œil.....	22
4.3.2 Calcul du point de regard.....	24
4.4 FABRICANTS	25
CHAPITRE 5: TRAVAIL À EFFECTUER ET CHOIX PRÉLIMINAIRES	27
5.1 TRAVAIL À EFFECTUER	27
5.2 CHOIX D'UNE TECHNOLOGIE DE TRACKING	27
5.3 DESCRIPTION DÉTAILLÉE DU TRAVAIL À EFFECTUER	28
5.4 ETUDE DU SMI EYELINK.....	29
5.5 CAHIER DES CHARGES.....	31
CHAPITRE 6: CARACTÉRISTIQUES D'UNE IMAGE DE L'ŒIL.....	33
6.1 INTRODUCTION	33
6.2 IMAGE DE CAMÉRA TRADITIONNELLE	33
6.2.1 Ombres	34
6.2.2 Reflets.....	34
6.2.3 Pupille.....	35
6.2.4 Luminosité	36
6.2.5 Lunettes et lentilles de contact.....	36
6.3 IMAGE DE CAMÉRA INFRAROUGE (IR).....	36
CHAPITRE 7: TRAITEMENT D'IMAGE POUR LA DÉTECTION DU CENTRE DE LA PUPILLE.....	39
7.1 INTRODUCTION	39
7.2 PRINCIPE DE FONCTIONNEMENT	39
7.3 DESCRIPTION DE L'ALGORITHME.....	39
7.3.1 Seuillage.....	40
7.3.2 Calcul du barycentre	41
7.3.3 Elimination de l'erreur de type 1	44
7.3.4 Elimination de l'erreur de type 2	48
CHAPITRE 8: RÉSULTATS OBTENUS.....	51
8.1 IMPLANTATION DE L'ALGORITHME.....	51
8.2 PRÉCISION DE L'ALGORITHME	52
8.2.1 Erreur sur la position de l'œil.....	53

8.2.2	<i>Erreur sur le point de regard.....</i>	54
8.2.3	<i>Test sur des images infrarouges</i>	55
8.2.4	<i>Test sur des images traditionnelles.....</i>	59
8.2.5	<i>Analyse des résultats.....</i>	62
8.3	VITESSE DE L'ALGORITHME.....	63
8.3.1	<i>Performance avec Java sur Pentium II.....</i>	63
8.3.2	<i>Performances avec C sur C6711</i>	65
8.3.3	<i>Analyse des résultats.....</i>	68
CHAPITRE 9: CONCLUSION.....		71
9.1	TRAVAIL RÉALISÉ	71
9.2	AMÉLIORATIONS POTENTIELLES	71
9.2.1	<i>Identification de zones.....</i>	71
9.2.2	<i>Vitesse de l'algorithme.....</i>	72
9.3	CONSIDÉRATIONS FINALES.....	72
ANNEXE A : INTRODUCTION AU TRAITEMENT NUMÉRIQUE D'IMAGES		75
A.1	HISTOGRAMME ET SEGMENTATION	75
A.2	CONVOLUTION	77
A.3	EROSION ET DILATATION	77
A.4	FILTRE PASSE-BAS ET FILTRE MÉDIAN	78
ANNEXE B : RÉFÉRENCES		79

Chapitre 1: Préambule

1.1 Introduction

Ce travail de diplôme porte sur une expérience qui rendra possible la création d'électrodes pouvant être implantées dans un œil humain non fonctionnel, ce qui permettrait de rendre une partie du sens visuel à un sujet aveugle.

La structure de ce document reflète la méthode de travail qui a été utilisée. Pour faciliter la compréhension et donner au lecteur une vue d'ensemble, nous allons décrire sa structure.

Ce travail de diplôme s'inscrit dans un projet de grande envergure. Le **chapitre 2** permettra donc de cibler le projet et de bien comprendre quel est le but qui a été fixé.

Ce projet exploite certaines caractéristiques de l'œil humain. Il nous a semblé nécessaire de donner quelques informations sur sa structure, sur les électrodes ainsi que sur leur implantation. Ce sera l'objet du **chapitre 3**.

Comme nous l'aurons vu dans le premier chapitre, le but de ce travail de diplôme est de réaliser un *eye tracker*. Le **chapitre 4** permettra de voir quelles sont les techniques existantes et quel est l'état de l'art en la matière.

Le **chapitre 5** montrera quel est le travail à effectuer pour la réalisation d'un *eye tracker*. Nous y verrons également la méthode de *tracking* qui sera exploitée.

Cette méthode étant basée sur de l'imagerie, nous consacrerons le **chapitre 6** à la description d'une image de l'œil.

L'algorithme de détection de la position de l'œil basé sur la méthode de *tracking* choisie sera décrit dans le **chapitre 7**.

L'évaluation de l'algorithme décrit dans le chapitre précédent sera réalisée dans le **chapitre 8**. Cette évaluation portera sur le plan des performances en précision et en vitesse.

Le **chapitre 9** permettra de discuter le cahier des charges et de voir si l'algorithme proposé répond à ce cahier des charges. Il permettra également de conclure ce travail de diplôme.

Les chapitres suivants sont consacrés aux annexes (références, etc.).

Toutes les notes de bas de page faisant référence à des fichiers sont stockés sur le CD-ROM accompagnant ce document.

1.2 Remerciements

Comme dans tout travail de diplôme, il est coutume de laisser une petite place pour remercier les personnes qui ont soutenu l'auteur. Il est clair que je dérogerais pas à cette coutume et c'est même avec un certain plaisir que je m'y plie.

Je voudrais commencer par remercier les collaborateurs du Laboratoire de Systèmes Numériques de l'EIG dont les locaux ont été comme une seconde résidence durant près de 9 mois. Le frigo, allié de première nécessité, a été d'un grand secours durant les journées de canicules en stockant sandwiches, glaces et boissons fraîches. Merci encore à Buzz l'éclair et à sa plante verte qui m'ont surveillé de près pendant tout ce temps.

Je tient également à remercier toute l'équipe de la clinique d'ophtalmologie et en particulier Jürg Summerhalder pour sa grande disponibilité et sa gentillesse.

J'adresse également un immense merci à toutes les personnes qui ont pris la peine de relire mon travail et dont les remarques et commentaires ont été d'une grande aide (Jacques, Stéphane, Thierry et Vanessa).

Enfin, un grand merci à vous, lecteur, pour toute l'attention que vous porterez à ce travail. Il me reste à vous souhaiter beaucoup de plaisir pour votre lecture.

Julien Kronegg

Chapitre 2:

Introduction et but

2.1 Introduction

Une prothèse oculaire est un dispositif destiné à pallier un défaut de la vision (par exemple la cécité) de manière similaire aux prothèses auditives. Les recherches concernant ces prothèses sont actuellement en plein développement. Dans ce domaine, la Clinique d'Ophtalmologie de l'Hôpital Universitaire de Genève a engagé plusieurs études en collaboration avec différents centres de recherche européens.

En l'état actuel des travaux, la voie principale de recherche envisage l'insertion d'une surface d'électrodes (tapis) au niveau de la rétine, surface en connexion avec les fibres du nerf optique situé derrière la rétine. Les électrodes, fournissant au nerf optique les signaux appropriés, remplacent alors la fonction de transducteur "signal lumineux-signal électrique" de la rétine défaillante. Par cette méthode, il serait alors possible que les sujets aveugles ayant une dégénérescence des cellules photoréceptrices (cellules captant la lumière dans l'œil) retrouvent partiellement la vue.

Un implant oculaire doit permettre au patient de réaliser les activités de base de la vie courante :

1. la lecture;
2. les opérations sensori-motrices (saisir un objet, le manipuler, etc.);
3. les opérations de déplacement dans un environnement.

Actuellement, de nombreuses connaissances font encore défaut quant aux informations visuelles nécessaires à une personne pour ces activités de base. Or, ces connaissances sont indispensables pour déterminer les caractéristiques de l'implant :

- dimensionnement des électrodes;
- détermination des caractéristiques essentielles que les images transmises doivent comporter;
- détermination de la stratégie de codage des signaux de contrôle des électrodes.

Si quelques études ont été réalisées pour les conditions nécessaires à la lecture, les deux autres catégories d'activité n'ont fait l'objet d'aucune étude sérieuse.

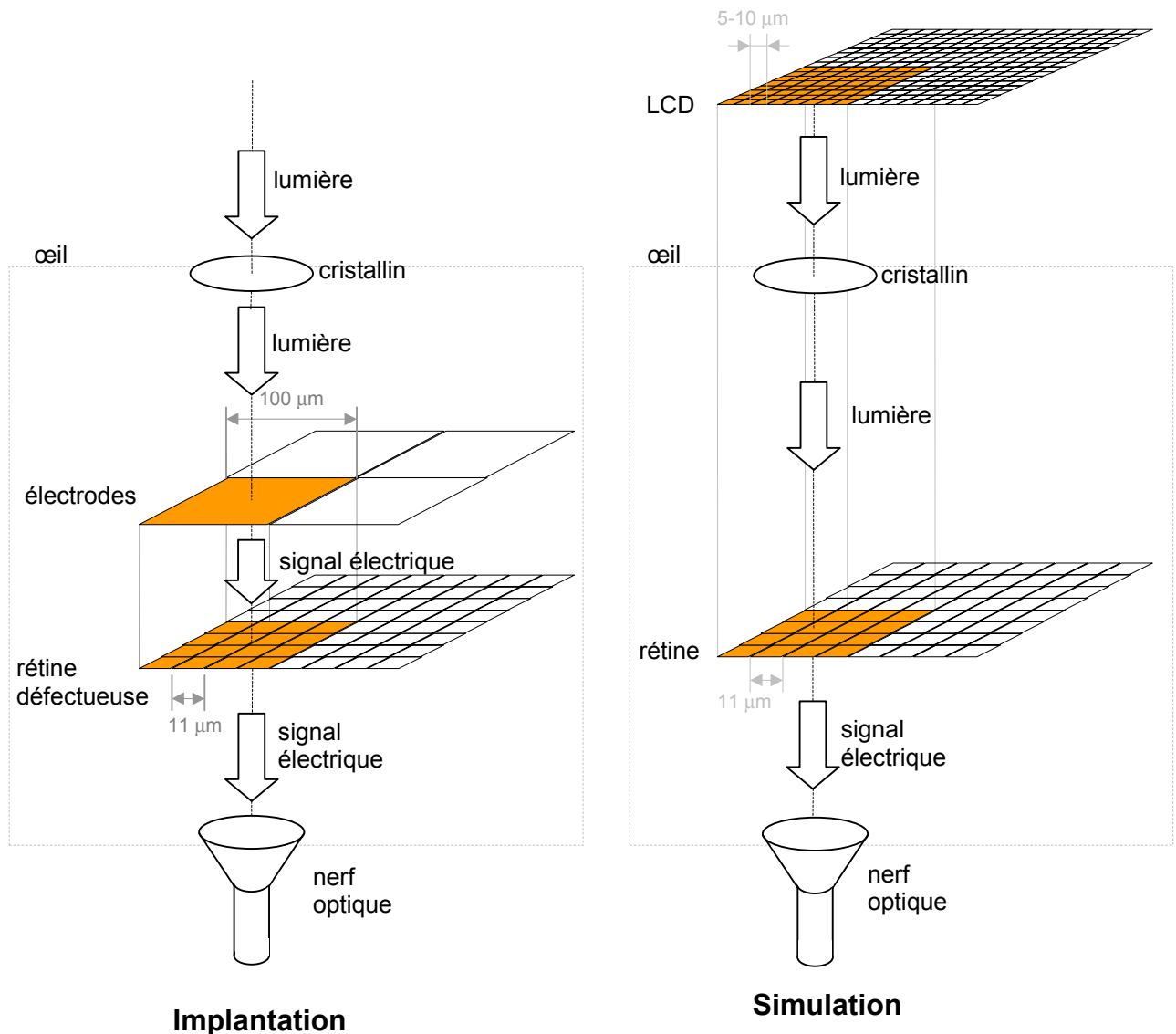
C'est la raison d'un projet du Fond National qui réunit les partenaires suivants :

- Clinique d'Ophtalmologie de l'Hôpital Universitaire de Genève;
- Clinique de Neurologie de l'Université de Genève;
- Faculté de Psychologie de l'Université de Genève;
- Laboratoire de Systèmes Numériques de l'Ecole d'Ingénieurs de Genève (EIG).

Parallèlement, un projet du Laboratoire de Systèmes Numériques de l'EIG, à travers un financement de la HES-SO, doit permettre de réaliser un système portable capable de simuler

le système d'implants oculaire. Grâce à ce simulateur, il devrait être déterminé de découvrir quelles sont les caractéristiques de l'implant en réalisant des tests sur un sujet dont le système de vision est fonctionnel.

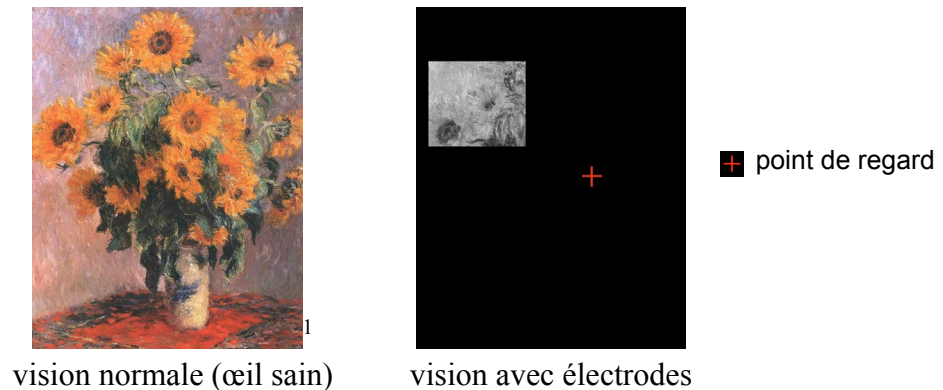
Illustrons le principe de fonctionnement de l'œil, tant pour la simulation que pour l'implantation des électrodes :



Dans l'état actuel de la technique, les électrodes seraient implantées sur la rétine, induisant ainsi un signal électrique palliant au défaut de fonctionnement des photorécepteurs de la rétine. Le signal électrique fourni par les électrodes serait créé en fonction de la lumière reçue à travers le cristallin (le détail de la pose des électrodes sera décrit plus en détails dans les chapitres qui suivent).

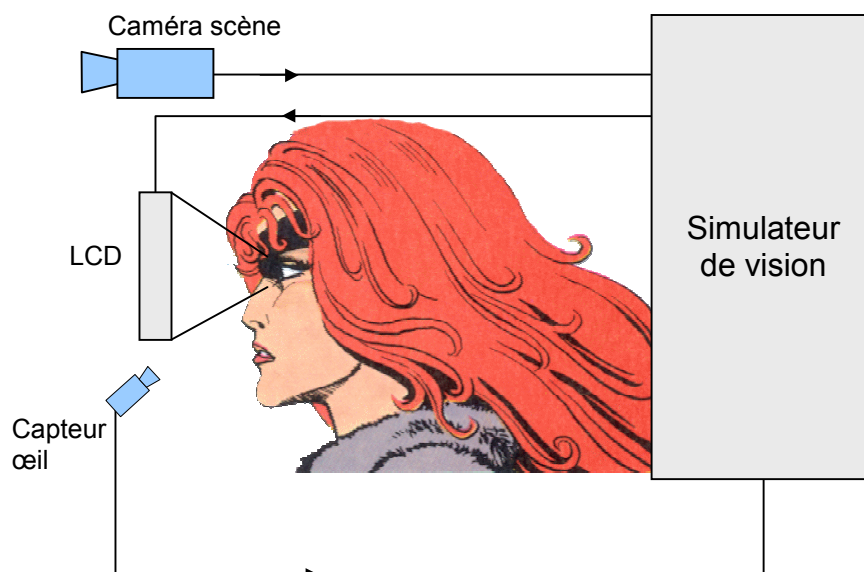
Un écran à cristaux liquides fournirait une simulation de la lumière naturelle qui apparaîtrait dans la phase d'implantation. La différence essentielle entre la phase de simulation et celle d'implantation est au niveau de la représentation du signal : signaux électriques pour l'implantation, signaux lumineux pour la simulation.

Les électrodes ne seraient pas implantées sur toute la surface de la rétine pour des raisons techniques (consommation électrique, difficultés chirurgicales, etc.) que nous verrons plus tard mais uniquement dans la zone de vision périphérique. La vision d'un sujet serait alors la suivante :



Avec des électrodes, une personne regarderait à un endroit (vision centrale) mais verrait à un autre endroit (vision périphérique). Notez bien la différence essentielle entre voir et regarder. Ce comportement est très important et doit être reproduit dans la phase de simulation.

Pour effectuer la phase de simulation, un simulateur de vision (appelé également simulateur d'électrodes) doit être créé. Un simulateur de vision est une version particulière d'un casque virtuel. Comme nous l'avons déjà dit, il doit permettre de dimensionner les électrodes.



Le simulateur d'électrodes est composé d'un écran à cristaux liquides projetant devant l'œil une image provenant d'une caméra filmant la scène (cf. illustration ci-dessus). Cette image peut être traitée (filtrage, modification de résolution, etc.). En variant le traitement, il devrait être possible de déterminer les caractéristiques que doivent avoir les électrodes pour qu'une vision suffisante aux activités de base soit offerte au patient.

¹ Source : Claude Monet, *Bouquet de soleils*, 1881.

² Visage tiré de la bande dessinée *Thorgal : La magicienne trahie*, éd. Lombard, p.9.

Par la simulation, un mécanisme de capture du mouvement de l'œil permet de déplacer l'image pour qu'elle corresponde au mouvement de l'œil, de manière à simuler le fait que les électrodes occupent seulement une partie périphérique de la rétine, comme on l'a vu avec la différence entre voir et regarder.

Au cours de ce document, nous parlerons quelques fois de "tromper le système". Ce terme se rapporte au cas suivant : le système de capture du mouvement oculaire assure un décalage entre le point de regard et le point de vision de manière à forcer le sujet à voir avec sa vision périphérique. Si le temps de réaction du système est trop long, le sujet peut effectuer un mouvement rapide de l'œil de manière à faire coïncider l'ancienne zone de vision avec le nouveau point de regard. Le sujet peut alors voir avec sa vision centrale, ce qui n'est pas désirable. Ce phénomène s'appelle "tromper le système".

Un certain nombre de laboratoires et de groupes de recherche sont impliqués dans ce projet. On peut notamment citer :

- Laboratoire de Systèmes Numériques de l'EIG (simulateur d'électrodes);
- Faculté de Médecine de l'Université de Genève (tests du simulateur);
- EPFL (conception des électrodes).

2.2 But du travail de diplôme

Le but de ce travail de diplôme est de réaliser un appareil capable de mesurer les mouvements oculaires. Ce genre d'appareil est communément appelé un *eye tracker*.

Les caractéristiques de cet *eye tracker* doivent être les suivantes :

- une précision de 1° (si l'œil bouge de 1°, l'appareil doit pouvoir le mesurer);
- une fréquence de fonctionnement de 250 Hz (nécessaire pour capter les mouvements rapides de l'œil);
- un temps de latence de l'ordre de 10 ms (pour éviter que le sujet puisse tromper l'appareil).

Ces exigences ont été demandées à partir de résultats de tests réalisés sur un *eye tracker*¹ de l'Hôpital Universitaire de Genève. Ces tests, effectués pour des expériences sur la lecture dans la zone de vision périphérique, ont montré que le sujet sous test ne pouvait pas tromper un appareil possédant les caractéristiques énoncées ci-dessus. Malgré tout, aucun essai n'a été réalisé avec des exigences moins élevées, ce qui permet de dire qu'il serait possible de construire un *eye tracker* avec de moins bonnes performances mais sans dégrader la qualité des résultats. Nous verrons cela en détails plus tard.

La réalisation d'un *eye tracker* a été décidée car les appareils disponibles sur le marché conviennent pas pour l'utilisation désirée pour des raisons de manque de portabilité) et de coût. Les modifications à apporter à un appareil du marché pour le rendre portable auraient probablement coûté davantage que de réaliser un *eye tracker* spécifique aux besoins de l'expérience.

Ce travail de diplôme est réalisé sous la direction de Thierry Pun, en collaboration avec le Laboratoire de Systèmes Numériques de l'Ecole d'Ingénieurs de Genève.

¹ SMI EyeLink *eye tracker* (<http://www.smi.de/>).

Chapitre 3:

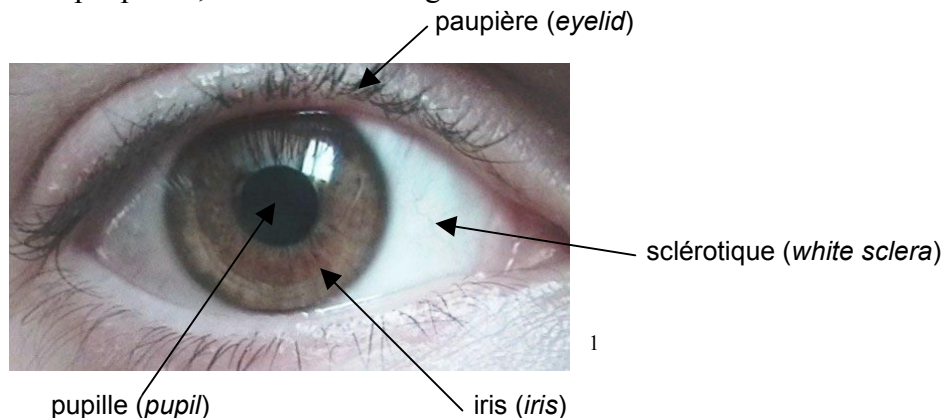
Œil, électrodes et implantation

3.1 Introduction

Cette partie décrit les différentes caractéristiques de l'œil. Les explications commencent par une description de l'extérieur de l'œil (cornée, pupille, etc.) suivie par une description de l'intérieur de l'œil (rétine, fovéa, etc.) et terminée par une description des mouvements de l'œil.

3.2 Description externe

Selon certaines définitions, l'œil humain est composé uniquement d'un globe oculaire mais pour ce projet, nous en parlerons plutôt comme de l'ensemble visuel comprenant le globe oculaire, l'orbite et les paupières, comme sur l'image suivante :



L'orbite de l'œil contient un globe oculaire de la forme proche d'une sphère. Ce globe oculaire contient toutes les cellules photosensibles permettant de capter la lumière ainsi que toutes les cellules nerveuses permettant la transmission de ces informations au cerveau.

La lumière traverse la pupille pour exciter les photorécepteurs de la rétine (cf. description interne). La quantité de lumière traversant cette pupille est réglée par l'iris, muscle qui a la capacité de se rétracter pour diminuer le diamètre de la pupille. La pupille n'est qu'une composante apparente puisqu'il s'agit en réalité d'une partie visible du cristallin. Celui-ci étant transparent, il laisse passer la lumière, raison pour laquelle la pupille est de couleur noire.

Le cristallin et l'iris sont protégés par une structure transparente appelée cornée. Les paupières apportent une protection supplémentaire en permettant de cacher l'œil au monde extérieur.

¹ Source : oeil_001.jpg.

Le globe oculaire possède trois degrés de liberté pour le déplacement :

- mouvements horizontaux;
- mouvements verticaux;
- mouvements de rotations.

Ces différents mouvements sont possibles grâce à six muscles qui sont fixés autour de l'œil. Les différents types de mouvements de l'œil seront détaillés à la fin de ce chapitre.

3.3 Description interne

L'image suivante montre l'iris et la cornée dont il a été question précédemment. Elle montre également le cristallin et la rétine, deux composantes essentielles du globe oculaire.

1

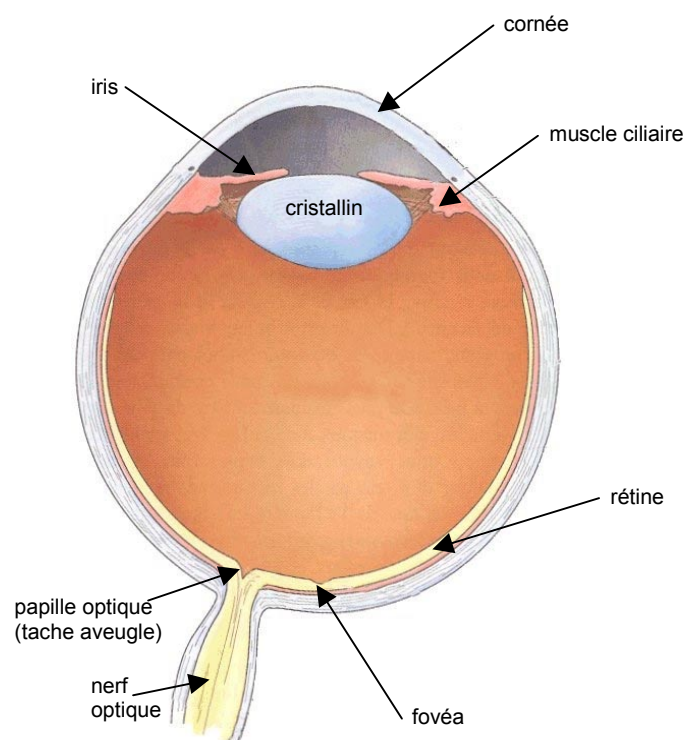
Le cristallin est un composant transparent et flexible permettant l'accommodation. Lorsque l'on regarde des objets éloignés, le cristallin a une forme fine alors qu'il a une forme épaisse lorsque l'on regarde des objets proches. La forme du cristallin est modifiée par l'action des muscles ciliaires l'entourant. Il permet de régler la netteté de l'image formée sur la rétine ainsi que la profondeur de champ (distance sur laquelle les objets ne sont pas flous).

Les rayons lumineux traversent le cristallin et forment une image sur la rétine. Les informations lumineuses correspondantes sont transférées au cerveau par l'intermédiaire du nerf optique.

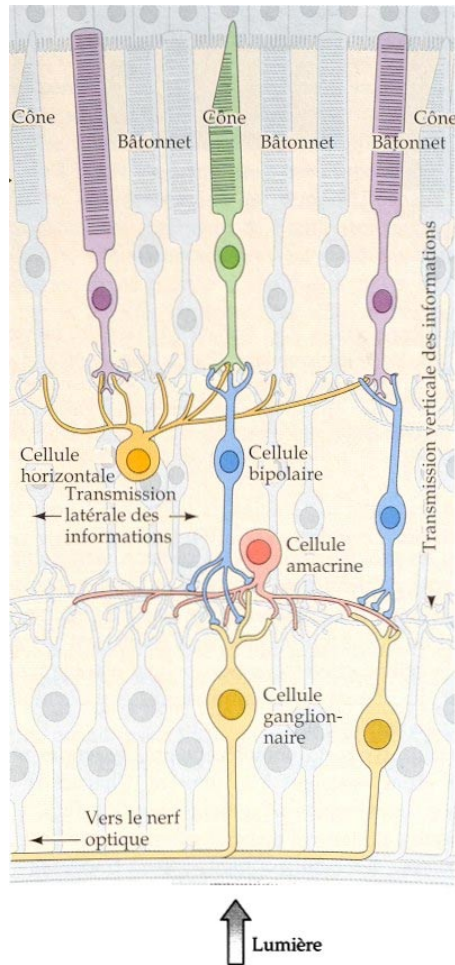
La rétine est composée de deux zones distinctes :

- la zone fovéale;
- la zone périphérique.

Ces deux zones seront décrites plus tard.



¹ Source : [24], p. 180.



La rétine (détail ci-contre) est composée de cellules de cinq types différents:

- les photorécepteurs (cônes et bâtonnets);
- les cellules bipolaires;
- les cellules amacrines;
- les cellules ganglionnaires;
- les cellules horizontales.

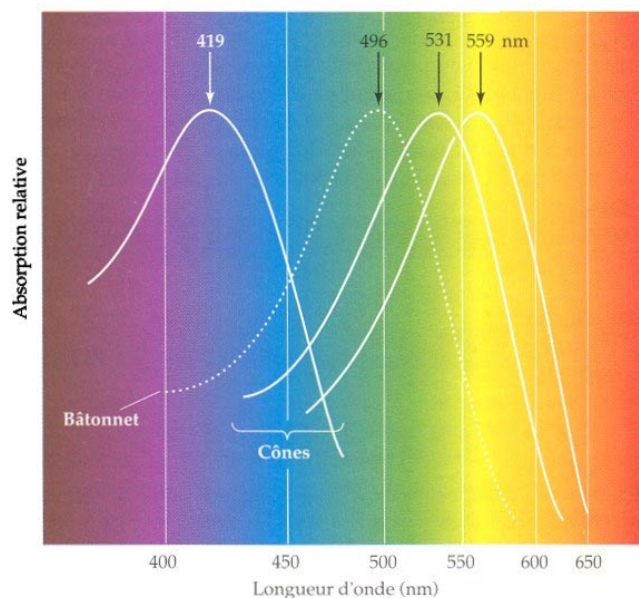
Toutes ces cellules sont des neurones dont le fonctionnement est similaire à ceux trouvés dans le cerveau. L'œil est d'ailleurs considéré par certains ouvrages comme une partie du cerveau.

Les photorécepteurs sont les seules cellules de la rétine sensibles à la lumière. Les autres types de cellules forment un réseau de neurones à deux couches que la lumière traverse avant d'arriver sur les photorécepteurs.

Les informations visuelles fournies par les photorécepteurs sont transportées en direction des fibres conduisant au nerf optique à travers le réseau de neurones à deux couches. Celui-ci assure un prétraitement de l'information (principalement un effet intégrateur). Cette information est diffusée verticalement (cellules bipolaires et ganglionnaires) et horizontalement (cellules amacrines et horizontales).

Le nerf optique est formé par les axones des cellules ganglionnaires.

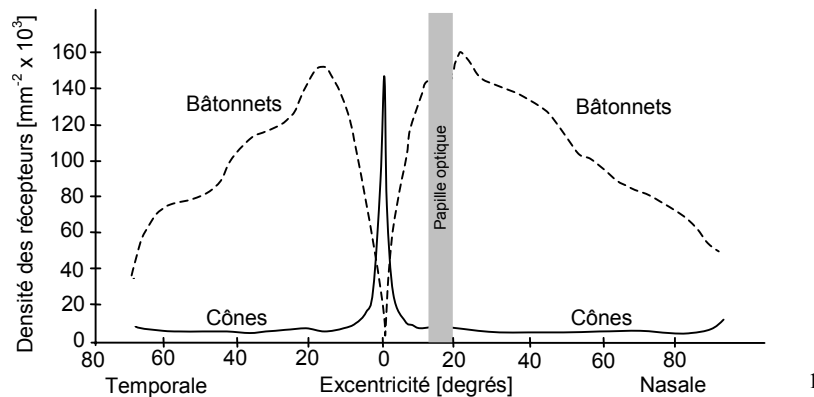
Il existe trois types de cônes absorbant chacun une longueur d'onde particulière (courbes en trait continu dans l'image de droite). Par contre, il n'existe qu'un seul type de bâtonnet qui absorbe lui aussi une longueur d'onde particulière (courbe en pointillés dans l'image de droite). Les cônes ont une résolution spatiale très élevée mais sont peu sensibles à la lumière. Inversement, les bâtonnets ont une résolution spatiale très faible mais ont une grande sensibilité à la lumière. Les cônes fonctionnent mieux avec la lumière diurne tandis que les bâtonnets fonctionnent mieux à la lumière nocturne.



¹ Source : [24], p.184.

² Source : [24], p.190.

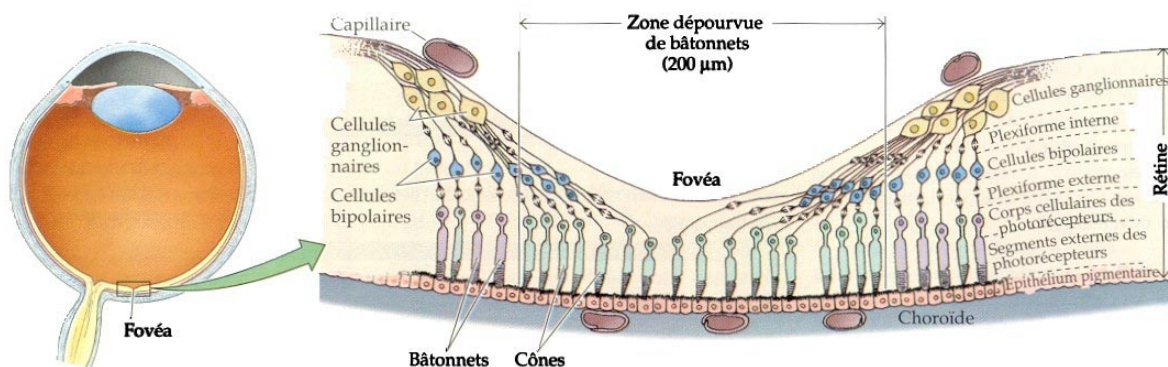
La disposition des photorécepteurs sur la rétine n'est pas uniforme. Elle dépend de l'angle entre le centre de l'œil (fovéa) et le rayon lumineux. Dans la zone fovéale, les cônes sont très nombreux alors que les bâtonnets sont très peu nombreux tandis que, dans la zone périphérique, c'est exactement l'inverse :



La grande densité de cônes en zone fovéale, leur capacité tri-chromatique ainsi que leur sensibilité diurne expliquent une bonne vision diurne des couleurs en zone fovéale. La grande densité des bâtonnets en zone périphérique, leur capacité mono-chromatique et leur bonne sensibilité nocturne expliquent une bonne vision nocturne en noir et blanc.

De manière générale, si l'on regarde les photorécepteurs dans leur ensemble, on peut dire que leur densité est maximale dans la zone fovéale et minimum lorsque l'on est très éloigné de celle-ci. Une exception réside dans la zone appelée papille optique ou tache aveugle qui est l'endroit de la rétine où se rejoignent toutes les fibres nerveuses pour former le nerf optique et qui, par conséquent, ne contient pas de photorécepteurs.

Si l'on s'intéresse de plus près à la zone fovéale, on peut voir qu'elle possède une structure qui diffère du reste de la rétine :



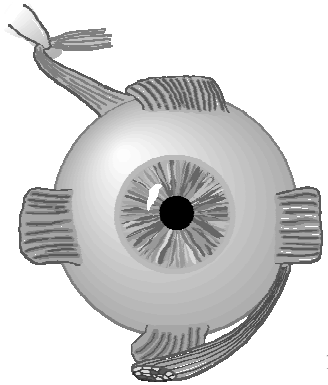
La partie centrale de la zone fovéale est dépourvue de bâtonnets, ce qui permet une plus grande densité de cônes et par conséquent une vision de haute précision. De plus, les deux couches de cellules assurant le prétraitement sont placées sur le côté de la zone fovéale (probablement à cause de la plus grande densité des cônes et afin que la lumière suive un chemin plus direct vers les photorécepteurs).

¹ Source : [24], p.191.

² Source : [24], p.191.

3.4 Mouvements de l'œil

Le mouvement de l'œil est provoqué par la contraction des six muscles entourant le globe oculaire :



Ces muscles permettent le déplacement avec trois degrés de liberté : vertical, horizontal et en torsion. Notons un petit abus de langage pour les termes "vertical" et "horizontal" : le globe oculaire étant considéré comme une sphère, il ne peut effectuer que des rotations. Par horizontal et vertical, on parle donc de la position du centre de la pupille projetée sur un plan faisant face à l'œil.

Les mouvements de l'œil peuvent être de deux types :

- mouvements volontaires;
- mouvements involontaires.

Les mouvements volontaires peuvent être déclenchés à volonté par la personne. Il s'agit des mouvements suivants :

- **convergence** : il s'agit d'un mouvement opposé des deux globes oculaires visant à se focaliser sur un objet;
- **saccades** : principal mouvement de l'œil pour examiner une scène. Il consiste à bouger rapidement l'œil d'une partie de la scène à une autre. Il s'agit du principal mouvement qui doit être reconnu par notre *eye tracker*.

Les mouvements involontaires sont effectués de manière naturelle. En général, on ne remarque même pas les mouvements de ce type. Les différents mouvements involontaires sont :

- **torsion** : rotation de l'œil selon un axe passant par le centre de la pupille. Ce mouvement apparaît par exemple lorsque l'on penche la tête de côté;
- **poursuite** : mouvement plus lent et plus fluide que les saccades. Il sert à suivre un objet qui se déplace et est initié involontairement lorsqu'un objet se déplaçant est dans le champ visuel;
- **nystagmus** : ce mouvement apparaît lorsque l'on tourne la tête en continuant à fixer un objet immobile. Également nommé réflexe vestibulo-oculaire (*vestibulo-ocular reflex, VOR*). Ce mouvement peut également être initié par un motif répétitif en mouvement (cf. le phénomène de la fenêtre dans un train en marche);
- **dérive et microsaccades** : apparaît durant les périodes de fixations (cf. description des saccades, ci-dessous). Consiste en une lente dérive suivie de petites saccades destinées probablement à corriger la dérive;

¹ Source : [1], p.13.

- **nystagmus physiologique** : oscillations hautes fréquences de l'œil destinées à bouger l'image affichée sur la rétine de manière continue (moins de 1°). Il apparaît en effet que si une image reste de manière fixe sur la rétine, elle disparaît. Le nystagmus physiologique permet de palier à cette disparition.

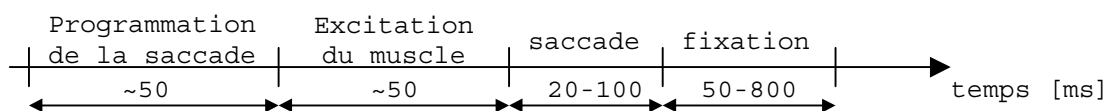
Pour notre travail, seules les saccades sont susceptibles d'intérêt, c'est pourquoi nous ne décrirons que celles-ci en laissant de côté les autres mouvements de l'œil.

La saccade est le mouvement le plus utilisé de l'œil. Il s'agit d'un mouvement d'amplitude variable ($0-135^\circ$) et de vitesse variable ($300-700$ voire $1000^\circ/s$). En général, il y a entre 3 et 5 saccades par seconde.

Une saccade peut être décomposée en plusieurs phases¹ :

- programmation de la saccade : décision du cerveau pour le prochain déplacement;
- excitation du muscle;
- rotation du globe oculaire : à vitesse variable;
- fixation : période pendant laquelle l'œil "regarde" la scène fixée.

Chaque phase est décrite en détails ci-dessous mais il est possible de résumer le déroulement temporel d'une saccade de la manière suivante :



Ces temps sont dépendant d'un certain nombre de paramètres. Il est d'ailleurs rare de voir plusieurs études arrivant aux mêmes conclusions. Les temps indiqués dans ce document sont ceux qui ont été le plus souvent rencontrés.

3.4.1 Programmation de la saccade

Durant cette phase, le cerveau choisit la prochaine position que l'œil devra atteindre et par conséquent le niveau d'excitation des muscles qu'il faudra produire. Le temps de programmation est difficile à mesurer avec un *eye tracker* car on ne peut pas le distinguer du temps de latence du muscle.

Le temps de décision est fonction de la tâche à exécuter. Un exemple simple est de suivre un spot lumineux se déplaçant sur un écran (beaucoup d'études utilisent cette tâche à tort pour mesurer le temps de programmation de la saccade).

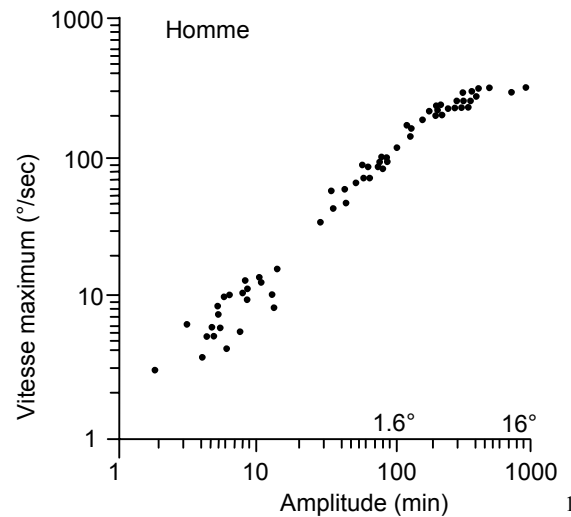
3.4.2 Excitation du muscle

Une fois que le cerveau a choisi la prochaine position du globe oculaire, il transmet l'ordre aux muscles entourant l'œil via les nerfs. Cela implique un petit délai de transmission.

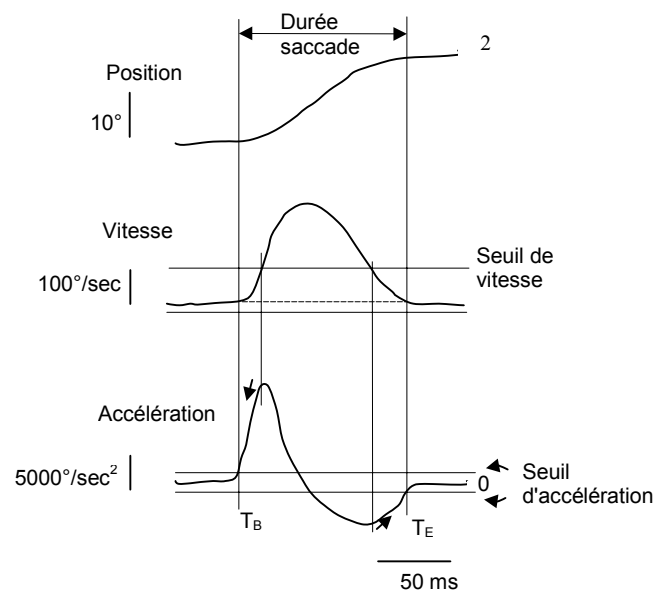
3.4.3 Rotation du globe oculaire

La vitesse angulaire de l'œil varie avec l'amplitude de la saccade. Cela permet à une grande saccade de ne pas prendre trop de temps. Selon certaines études, il semble que la vitesse angulaire sature vers $700^\circ/s$ (mais selon d'autres, ce serait plutôt le cas vers $1000^\circ/s$) :

¹ [2], p.89 donne une description légèrement différente.



La saccade est faite d'une accélération oculaire suivie d'une décélération oculaire pour finalement atteindre le but de la saccade. La vitesse maximale de la saccade est en général atteinte avant le milieu de la saccade :



Plusieurs chercheurs ont tenté de décrire des équations sur la durée d'une saccade. La plus simple équation qui semble raisonnable est due à Yarbus (1967) :

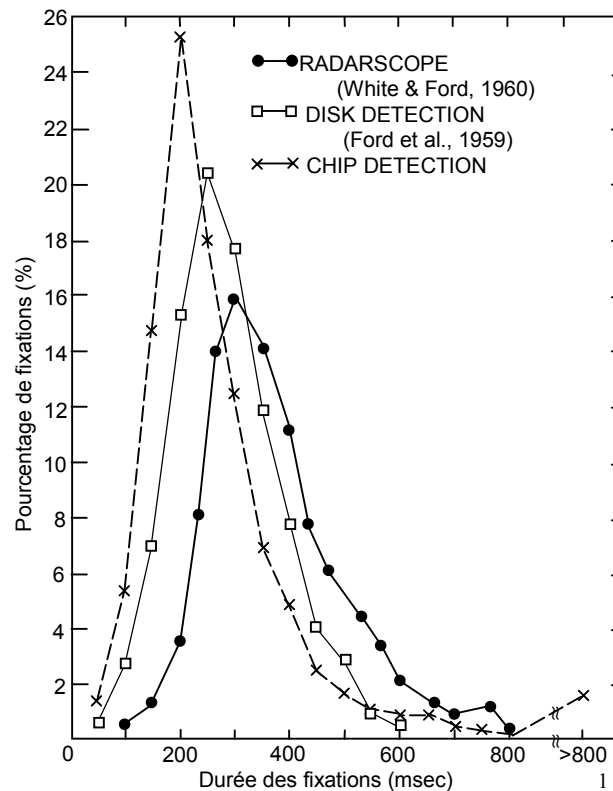
$$T = 21 \cdot a_0^{0.4} \quad a_0 = 5..45^\circ \text{ (amplitude de la saccade)}$$

3.4.4 Fixation

La période de fixation permet au cerveau de "lire" et de comprendre la scène fixée. Elle dure entre 50 et 800 ms. Cette durée est variable mais est en relation avec la tâche effectuée. Voici un exemple de distribution de temps de fixation pour différentes tâches :

¹ Source : [17], p.41.

² Source : [3], p.15.



La période de fixation dure le plus souvent environ 200 ms. Elle peut encore être décomposée en plusieurs phases :

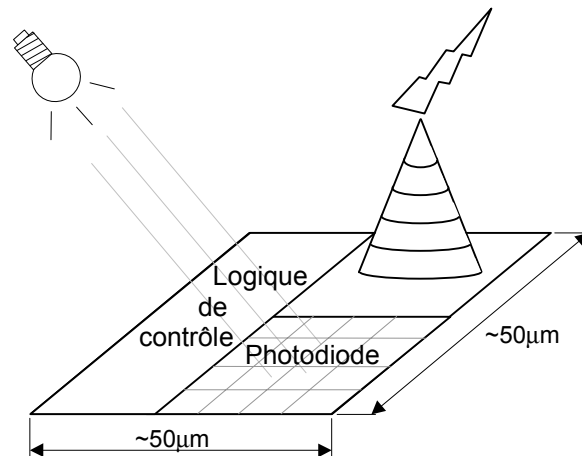
- acquisition de l'image par la rétine;
- transmission au cortex cérébral;
- traitement de l'information.

3.5 Description des électrodes et implantation

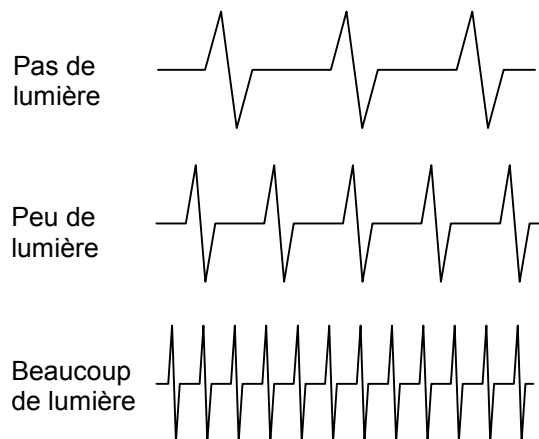
Les électrodes doivent permettre de remplacer des cellules photoréceptrices (cônes et bâtonnets) de l'œil qui sont défectueuses et qui ne peuvent donc plus assurer la conversion signal lumineux – signal électrique. Ces électrodes ont été développées au début de l'année 2001 à l'EPFL dans le cadre d'un travail de diplôme. Etant donné qu'il s'agit de travaux confidentiels, nous nous contenterons ici d'en énoncer les principes de base.

Une telle électrode est appelée *smart pixel*. Elle est composée d'une photodiode qui permet de capter la lumière, d'une pointe d'électrode et d'un circuit de contrôle assurant la conversion du signal :

¹ Source : [17], p.330.



Le fonctionnement est basé sur le principe relativement simple de l'accumulation-décharge. La pointe (probablement du titane) génère des impulsions dont la fréquence varie avec l'intensité lumineuse et produit donc un comportement similaire à celui d'un photorécepteur :

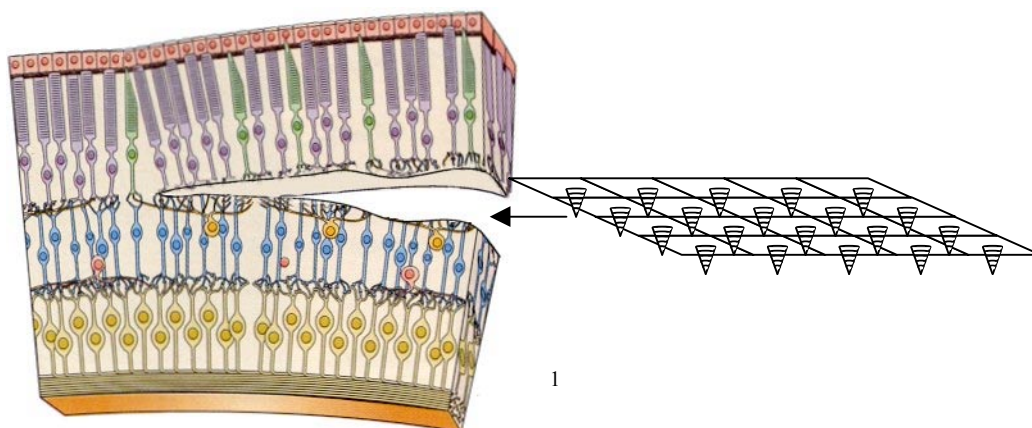


Cette pointe est pilotée par la logique de contrôle sur la base du niveau de lumière capté par la photodiode.

La taille d'un *smart pixel* est de l'ordre d'un carré de 50 à 100µm de côté, ce qui est relativement grand comparé aux 11µm de diamètre d'un photorécepteur. Cela veut donc dire qu'une électrode peut couvrir la surface de plusieurs photorécepteurs, suivant la densité de la rétine. L'énergie diffusée par la pointe doit donc être ajustée pour stimuler un certain nombre de cellules, ni trop ni trop peu. Il est probable que l'intensité de cette énergie diffusée soit d'une forme s'apparentant à une fonction gaussienne.

Les *smart pixels* sont organisés sous forme de tapis dont la forme reste à définir. Ces *smart pixels* doivent être implantés à une certaine distance les uns des autres de sorte que l'influence d'un *smart pixel* sur ses voisins soit minimale. Cet espacement, qui correspond à la densité du tapis d'électrodes, reste également à définir.

L'implantation de ce tapis d'électrode nécessite une opération chirurgicale de l'œil. Il est prévu d'implanter le tapis d'électrodes de manière à ce qu'il remplace les photorécepteurs. Pour cela, l'opération consiste à déchirer la rétine en deux parties entre les photorécepteurs et les cellules bipolaires :



Etant donné que les trois activités désirées (déplacement, lecture et manipulation) nécessitent une vision précise, le tapis d'électrodes devrait être idéalement implanté dans la zone fovéale, ce qui est malheureusement impossible. En effet, la zone fovéale possède une densité de fibres trop élevée et sa structure est trop complexe pour rendre possible une telle opération (il est impossible de retrouver la relation spatiale entre les électrodes et les nerfs). Le tapis d'électrodes serait donc implanté dans la zone périphérique de la rétine où la densité de fibres est moins importante. Compte tenu du fait que les électrodes ne seraient pas implantées en zone fovéale, un certain réapprentissage du cortex visuel sera nécessaire pour retrouver une vision suffisante aux trois activités de base.

Les électrodes destinées à être implantées dans l'œil n'ont pas besoin de gérer les couleurs car, étant donné que le tapis d'électrodes sera implanté en zone périphérique, il y aura fort peu de cônes pour capter des signaux tri-chromatiques (cf. illustration p.14). De plus, la gestion des couleurs nécessiterait une augmentation de la complexité du *smart pixel* ainsi qu'une augmentation de sa taille, ce qui poserait de grands problèmes lors de l'implantation.

Une telle prothèse nous donne l'espoir d'émuler plus ou moins fidèlement le comportement des photorécepteurs. Bien entendu, la résolution sera probablement bien inférieure (environ 100 x 100 *smart pixels*) compte tenu de la taille des électrodes et de leur dissipation de chaleur. En effet, on ne peut se permettre d'avoir une source de chaleur trop importante qui pourrait endommager l'œil.

Pour les premiers tests, il est prévu d'implanter ce tapis d'électrodes sur des poulets. En cas de succès, il est probable qu'il soit implanté sur un œil humain.

Les sujets humains concernés sont les aveugles de naissance et les aveugles par maladie (dégénérescence des photorécepteurs). Les aveugles d'accident ne sont pas concernés car il est rare qu'un accident détruise uniquement les photorécepteurs. Il est probable que les résultats obtenus avec les aveugles par maladie soient meilleurs que ceux obtenus par les aveugles de naissance. En effet, les aveugles par maladie ont déjà vu et par conséquent, leur cortex visuel a déjà été entraîné à la vision, ce qui n'est pas le cas des aveugles de naissance. Dans les deux cas, il est certain qu'un apprentissage sera nécessaire pour obtenir des résultats probants.

La structure particulière des électrodes (forme, méthode de diffusion, etc.) permet de voir quelles sont les fonctionnalités (par exemple : sous-échantillonnage, morphing, filtrage, etc.) que le simulateur doit implanter pour déterminer l'électrode idéale.

¹ Le tapis de forme rectangulaire est donné à titre d'illustration (la forme réelle du tapis est encore inconnue).

Chapitre 4:

Eye trackers – state of the art

4.1 Introduction

Dans ce chapitre, nous allons décrire le fonctionnement d'un appareil capable de capter les mouvements oculaires (*eye tracker*). Les principales techniques connues et les différents fabricants seront présentés.

4.2 Domaines d'utilisation

Un *eye tracker* est un appareil qui est destiné à capturer les mouvements de l'œil, ce qui permet de repérer le point de la scène vers lequel pointe l'œil de l'utilisateur. Ce point correspond en général à ce que le sujet regarde et ce à quoi il porte de l'intérêt. Ce genre d'appareil est utilisé dans une multitude de domaines aussi bien commerciaux ou industriels que militaires.

Parmi les applications typiques, on peut noter :

- commande d'appareils (l'œil sert de dispositif de pointage, comme une souris);
- diagnostic médical (p.ex. neurologie, ophtalmologie, etc.);
- évaluation de design (p.ex. sites web, musées, tableau de bord, etc.).

Il existe une multitude de techniques permettant de réaliser un *eye tracker*. Chacune a ses propres avantages et inconvénients, mais aucune méthode n'est significativement meilleure que l'autre. Des études¹ ont montré que les *eye trackers* pouvaient être classés selon une vingtaine de caractéristiques. Nous parlerons ici de celles qui sont importantes pour notre travail.

On distingue deux catégories d'*eye trackers* : ceux qui imposent un contact avec l'utilisateur (par exemple lunettes spéciales, casque, etc.) et ceux qui n'en imposent pas. En général, tous les appareils qui n'imposent pas le contact sont statiques (par exemple posés sur une table) et donc peu adaptés à la mobilité du sujet.

D'autres paramètres entrent en ligne de compte pour notre projet :

- la vitesse d'acquisition;
- la résolution;
- la précision.

¹ Hallet (1986), cité par [1], p.4.

4.3 Techniques d'eye tracking

Pour la réalisation d'un *eye tracker*, il faut bien distinguer deux phases très différentes : l'*eye tracker* doit être capable de repérer la **position de l'œil** (en fonction de l'état des muscles permettant son déplacement) et le **point où l'œil regarde** (point d'intérêt dans la scène regardée). Le point de regard dépend de plusieurs paramètres (position de l'œil, orientation de la tête, etc.) qui rendent le calcul direct difficile. Pour cette raison, les *eye trackers* décomposent souvent le problème en deux parties : position de l'œil puis point de regard.

4.3.1 Calcul de la position de l'œil

Pour la phase de calcul de la position de l'œil, il existe plusieurs familles de techniques¹ :

- basée sur le traitement numérique d'image. La position de l'œil est extraite à partir des informations apportées par le contenu d'une image de l'œil;
- basée sur le potentiel électrique de la peau. La position de l'œil est extraite à partir des variations de ce potentiel électrique;
- basée sur des lentilles de contact spéciales. La position de l'œil est extraite à partir des caractéristiques induites par la lentille.

Techniques basées sur l'imagerie

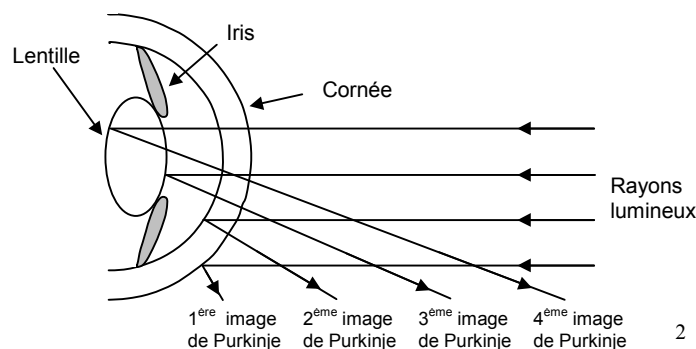
Les appareils exploitants cette technique se basent sur un principe simple. Une caméra capture des images de l'œil en séquence. Ces images sont ensuite traitées pour en extraire la position de l'œil. Pour cela, différentes caractéristiques des images sont utilisées. Il existe donc plusieurs méthodes pour le traitement.

On peut distinguer deux caractéristiques distinctes : les réflexions de lumière et les caractéristiques naturelles de l'œil.

Les **méthodes basées sur les réflexions de lumière** utilisent le reflet d'une source lumineuse comme information utile. Lorsqu'un rayon lumineux est dirigé contre l'œil, il est réfléchi. Etant la nature transparente de certaines parties de l'œil, la réflexion peut se faire sur quatre surfaces de l'œil :

- extérieur de la cornée;
- intérieur de la cornée;
- extérieur du cristallin;
- intérieur du cristallin.

Ces quatre reflets sont communément appelé *Image de Purkinje* :



¹ On peut trouver d'autres définitions (p.ex. [1]).

² Source : [1].

L'avantage de ce type de méthode est que le reflet est petit et brillant ce qui permet de le repérer facilement¹. Certaines de ces méthodes utilisent l'une ou l'autre de ces images de Purkinje. D'autres utilisent même deux images de Purkinje (DPI, *Dual Purkinje Images*), en général la 1^{ère} et la 4^{ème}. La position de l'œil est alors extraite de la différence entre les deux images de Purkinje.

Les **méthodes basées sur les caractéristiques naturelles de l'œil** utilisent des composants visibles sur l'image de l'œil comme :

- la pupille;
- l'iris.

Elles sont relativement plus simples que les précédentes car l'information à extraire de l'image est directement la position de l'œil et non une réflexion dont il faut tirer la position de l'œil. En général, l'utilisation de l'iris comme information de base implique une plage de fonctionnement (en degrés) plus réduite que lorsque la pupille est utilisée (de l'ordre de deux fois moins grande) car la surface de l'iris est le plus souvent cachée en partie par les paupières. Cependant, l'iris est plus facile à détecter car le contraste entre l'iris et la sclérotique est plus marqué qu'entre la pupille et l'iris.

Plusieurs méthodes de traitement ont été rencontrées (liste non exhaustive) :

- carte de gradient ([52]);
- transformée de Hough pour ellipses ([54] et [58]);
- seuillage de couleur ([52]);
- détection de contours ([51] et [52]);
- différence entre images (*dark & bright pupil*) ([53] et [60]);
- convolution avec un modèle ([52]);
- réseaux de neurones ([52] et [61]).

Toutes ces méthodes de traitement ont leurs avantages et leurs inconvénients et aucune n'est vraiment meilleure que l'autre.

Techniques basées sur le potentiel électrique

Lorsqu'un muscle se contracte, il génère un signal électrique appelé EMG (Electro-MyoGramme). Ce signal peut être capté par des électrodes spécifiques ([33]). A partir de ce signal EMG, il est possible de reconstituer le mouvement du muscle. En général, ces signaux peuvent être captés en plaçant des électrodes sur la peau et non directement sur le muscle. Cette méthode a donc l'avantage de ne pas être intrusive. Lorsqu'elle s'applique à un œil, les signaux générés sont appelés EOG (Electro-OculoGramme).

Les électrodes sont alors placées sur les muscles oculomoteurs qui permettent le déplacement du globe oculaire. Une fois que le mouvement de ces muscles a été reconstitué, il est possible de calculer la position de l'œil.

Cette méthode permet de travailler avec un œil qui n'est pas forcément visible. De ce fait, la plage de mesure est très importante (+/- 70° environ). Cependant, cette méthode n'est pas très pratique car la pose des électrodes proches de l'œil n'est pas toujours aisée (d'autant plus que la qualité des mesures dépend fortement du placement de l'électrode).

¹ Cf. Chapitre 6: pour des détails.

Cela permet de dire que l'usage quotidien d'une telle méthode serait difficile. Par contre, il est fort probable que les inconvénients cités ne soit pas trop contraignants lorsque celle-ci est utilisée de manière ponctuelle en laboratoire.

Techniques basées sur des lentilles de contact

L'idée maîtresse consiste à placer sur l'œil une lentille de contact possédant une propriété particulière. Celle-ci provoquera une modification d'une certaine caractéristique qui pourra être mesurée et dont on pourra tirer la position de l'œil. Deux des réalisations rencontrées utilisent ce principe.

La première consiste à placer l'œil au cœur d'un champ magnétique. La lentille de contact possède une petite inductance, ce qui a une influence sur le champ magnétique. Cette influence est mesurable car on peut capter les modifications du champ magnétique. A partir de cela, il est possible de déterminer la position de l'œil.

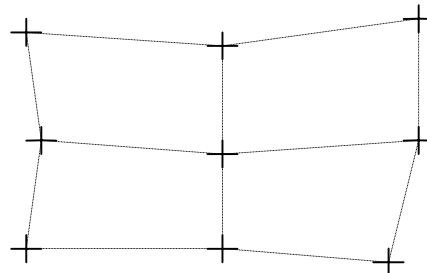
La seconde réalisation rencontrée propose une lentille contenant un petit miroir de forme particulière. Les réflexions d'une source lumineuse sur ce miroir provoquent un reflet possédant des propriétés qui permettent de ne pas le confondre avec un autre type de reflet. Ce reflet est capté à l'aide d'une caméra filmant l'œil. Cette technique est donc une combinaison entre les techniques d'imagerie et les techniques de lentilles de contact.

Il est fort probable que le port de lentilles de contact soit applicable aisément à l'usage quotidien. Notons toutefois que l'influence des champs magnétiques sur la santé n'est pas suffisamment connue actuellement.

4.3.2 Calcul du point de regard

A partir de la position de l'œil, le calcul du point de regard est calculé par interpolation après une phase de calibration.

Dans la phase de calibration, le sujet sous test doit regarder un certain nombre de points fixes dans la scène. La correspondance entre la position de l'œil et le point de regard est mémorisée et sert ensuite de base pour l'interpolation (par exemple : linéaire, cubique, spline). Exemple avec 9 points de calibration et une interpolation linéaire :



La notion de point de regard est très importante car elle fait intervenir une foule de paramètres :

- position de l'œil;
- position de la tête;
- géométrie de l'*eye tracker*;
- etc.

Ainsi, avec un *eye tracker* sans contact, il est possible de fixer le même point de regard avec plusieurs positions de l'œil différentes. Par contre, pour un *eye tracker* avec contact et pour autant qu'il soit solidaire avec la tête du patient, il n'y a pas de nécessité de faire de calibration car il n'est pas possible de fixer un point de regard avec plusieurs positions de l'œil différentes.

4.4 Fabricants

Les *eye trackers* sont développés par deux types de sociétés distincts :

- domaine éducatif (par exemple : écoles, universités);
- domaine commercial (par exemple : fabricants, instituts de recherche).

Dans certains cas, les appareils conçus dans le domaine éducatif sont par la suite utilisés dans le domaine commercial.

Voici la liste des principales¹ sociétés ayant développé des *eye trackers* :

	Avec contact	Sans contact
Domaine commercial	SMI ISCAN Arrington Research ASL Optomotor Fourward	SMI LC Technologies EyeTech DS ASL IBM
Domaine éducatif	University of Southern California University of Maryland	University of Karlsruhe California Institute of Technology HHI Lehigh University

Plus de 80% des sociétés commerciales utilisent un principe de fonctionnement basé sur l'imagerie et les sources infrarouges. Plusieurs raisons peuvent expliquer cette claire préférence pour la méthode d'imagerie par infrarouge :

- la plupart des appareils sont destinés à capter le regard pour les besoins d'une interface utilisateur qui n'a pas envie de s'encombrer d'un appareil complexe;
- la méthode d'imagerie par infrarouge est simple, performante et peu coûteuse.

La majorité des *eye trackers* rencontrés ([51], [52], [53] et [58]) ont une précision de l'ordre de 0.5 à 1° à une fréquence de fonctionnement de 25 images par secondes environ. Fort peu d'appareils atteignent des vitesses supérieures : le maximum constaté est 250 images par seconde.

¹ Pour une liste plus complète des sociétés commerciales : list_of_equipment_manufacturers.html.

Chapitre 5:

Travail à effectuer et choix préliminaires

5.1 Travail à effectuer

La réalisation d'un *eye tracker* consiste à :

- 1) choisir une méthode de *tracking*;
- 2) implanter la méthode;
- 3) intégrer l'*eye tracker* au simulateur de vision.

Bien entendu, il s'agit là d'un descriptif très vague qui doit être compris comme une description du travail préliminaire. En effet, il est difficile de décrire ce travail en détails sans avoir d'idée générale. Nous commencerons donc par choisir une technologie de *tracking* de la position de l'œil, ce qui nous permettra d'expliquer plus en détails le travail à effectuer.

5.2 Choix d'une technologie de tracking

Comme nous l'avons déjà vu dans le chapitre précédent, il est plus simple de décomposer le problème du *tracking* en deux parties. Nous traiterons donc la position de l'œil et le point de regard séparément.

La méthode de repérage de la position de l'œil qui a été retenue est **une méthode basée sur le traitement d'images** et ceci pour plusieurs raisons :

- un certain nombre d'*eye trackers* utilisent ce principe, en particulier celui mis à disposition par l'hôpital, ce qui a permis d'avoir une bonne base de départ;
- ces méthodes ont montré qu'il était possible d'obtenir des performances élevées tant au niveau de la vitesse que de la précision, ce qui n'est pas le cas de toutes les méthodes;
- la technologie relative à ce genre de méthodes est relativement simple et cela à prix très abordable (une caméra, un processeur, un *frame grabber* et de la mémoire).

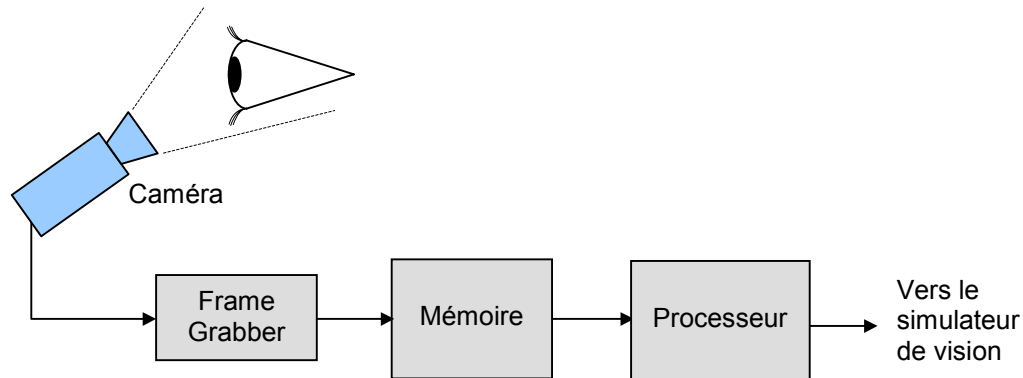
De plus, l'envie de maîtriser les techniques d'imagerie numériques a également joué un rôle dans le choix de cette technologie.

La méthode choisie est donc l'extraction de la **position de l'œil à partir des caractéristiques naturelles de l'œil, plus précisément de la pupille**. Cette méthode est très intéressante car elle fait intervenir de la reconnaissance de forme. De plus, le traitement temps-réel à haute vitesse est un défi à relever.

L'*eye tracker* réalisé étant un modèle "avec contact", il ne sera pas nécessaire d'effectuer une calibration à chaque utilisation de l'appareil. Par contre, le calcul du point de regard devra être adapté en fonction de la géométrie du simulateur de vision.

5.3 Description détaillée du travail à effectuer

Etant donné que la technologie choisie utilise de l'imagerie numérique, il est essentiel de disposer d'un certain nombre de composants (caméra, *frame grabber*, processeur, etc.). On peut dès lors schématiser l'*eye tracker* de la manière suivante :



A partir de cela, on peut maintenant déterminer avec précision le travail à effectuer. Sur le plan matériel, il faudra d'abord choisir les différents composants.

La **caméra** se charge de capturer l'image de l'œil. Elle devra être choisie en fonction de ses caractéristiques de vitesse et de résolution pour assurer que le sujet ne puisse pas "tricher" ainsi que pour ses faibles dimensions mécaniques pour laisser de la place pour le simulateur de vision.

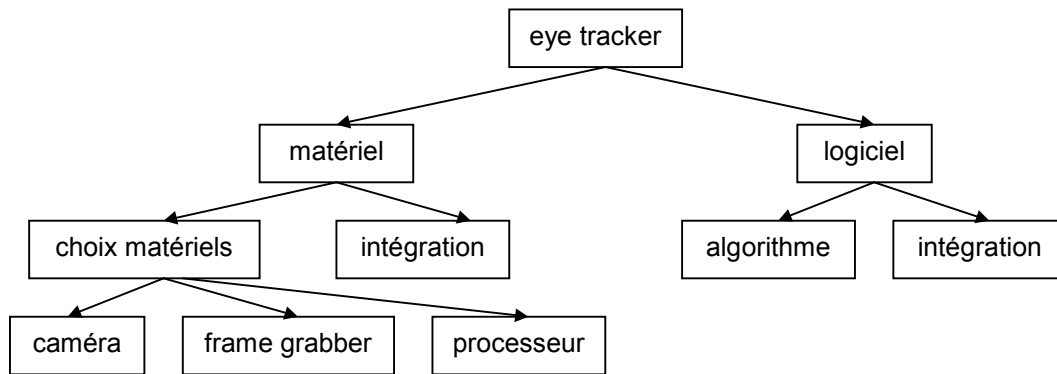
Le **frame grabber** se charge de convertir l'image et de la stocker dans un format numérique approprié au traitement par un processeur. Il devra être suffisamment performant pour suivre le rythme de la caméra.

La **mémoire** permet le stockage temporaire de l'image. Elle devra être de taille suffisante pour contenir une ou plusieurs images de l'œil.

Le **processeur** se charge de calculer la position de l'œil à partir de l'image en mémoire. Il devra être suffisamment puissant pour fournir la position de l'œil au rythme imposé par la caméra.

Sur le plan logiciel, il faudra créer un algorithme de recherche de la position de l'œil qui soit à la fois rapide et précis. Cet algorithme sera implanté dans le processeur.

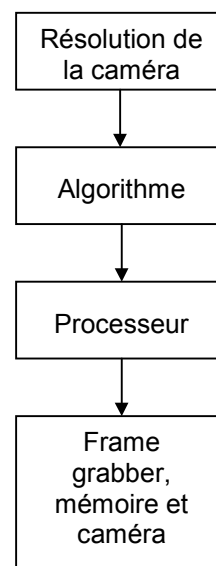
On peut résumer ces différentes étapes par le schéma suivant (les étapes "intégration" dénotent l'intégration avec le simulateur de vision qui est en cours de réalisation) :



Comme bien souvent dans un travail de recherche, le matériel et le logiciel sont très liés. Il est par conséquent déconseillé d'approfondir l'un ou l'autre sans avoir une vue d'ensemble.

La manière de procéder sera donc la suivante (voir illustration ci-contre) : la résolution de la caméra sera choisie en premier en se basant sur d'autres *eye trackers* existants. Cela permettra de dimensionner l'algorithme. Les performances d'un prototype de l'algorithme permettront de déterminer le processeur nécessaire pour traiter l'image. A partir de ce moment, le reste du matériel pourra être choisi.

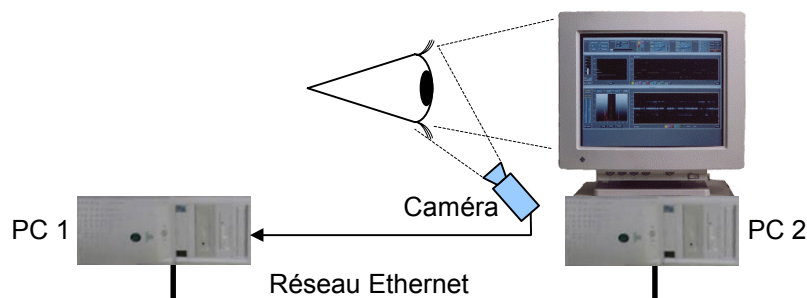
De prime abord, il est difficile de déterminer une résolution minimum pour la caméra. Nous avons choisi de prendre exemple sur un *eye tracker* existant pour se donner une idée de la résolution nécessaire pour obtenir une certaine précision d'algorithme. Pour cela, nous avons utilisé le EyeLink mis à disposition par la clinique d'ophtalmologie des Hôpitaux Universitaires de Genève.



Déroulement du travail (choix)

5.4 Etude du SMI EyeLink

Le SMI EyeLink¹ est un *eye tracker* basé sur une méthode d'imagerie numérique qui repère la position du centre de la pupille. Il fonctionne à une vitesse de 250 images par seconde avec une résolution de 0.01° (le temps de latence par image est de 10 ms). Il peut être schématisé de la manière suivante :

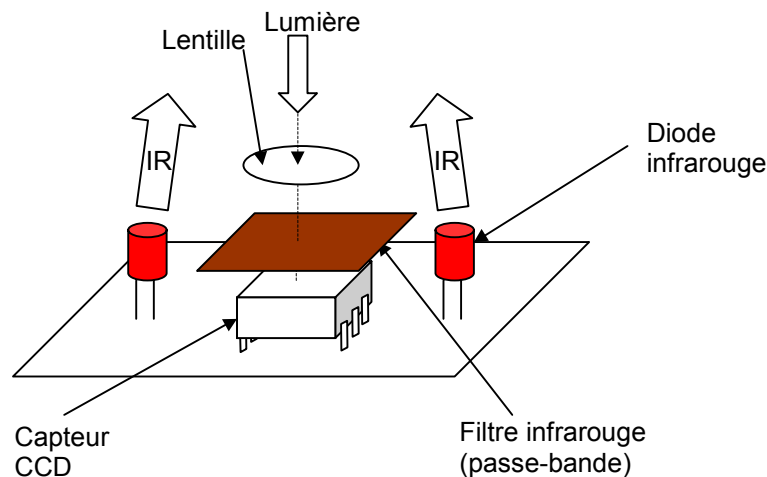


¹ www.smi.de, environ 70'000.-

L'œil du sujet regarde l'écran du PC 2 qui sert de station maître en générant des images à regarder à une fréquence d'affichage de 160 images/seconde. Le PC 1 se charge du traitement de la position de l'œil. La position de l'œil est transmise par un réseau Ethernet vers le PC 2 qui prend en compte cette donnée pour modifier le contenu de l'écran.

On comprendra aisément que ce système n'est pas portable et c'est la raison pour laquelle ce projet a été mis en place.

La caméra est relativement simple. Elle consiste en un capteur CCD recevant de la lumière infrarouge émise par deux diodes IR et réfléchi par l'œil. Une lentille permet la mise au point de l'image :

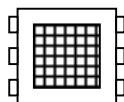


Les **diodes IR** permettent une bonne illumination de l'œil du sujet sans toutefois provoquer un éblouissement ou une gêne (tout au plus une très légère sensation de chaleur). Leur longueur d'onde est de 950 nm, ce qui les place dans le début de la zone infrarouge (rappel : l'infrarouge commence vers 700 nm). La puissance dissipée par les diodes est de 0.8 à 1.2 mW/cm².

Le **filtre passe-bande IR** permet de bloquer toutes les fréquences du spectre visible et ultraviolet. Ainsi, seule la lumière créée par les diodes IR et réfléchi par l'œil est perçue par la caméra (note : la lumière visible contient très peu d'infrarouge).

La **lentille** permet la mise au point de l'image en fonction de la distance de travail. Elle est de type DCX (double courbure) et son diamètre est de 4 mm. Elle est montée dans un cylindre à vis pour permettre son déplacement.

Le **capteur CCD** est un circuit 6 pattes mesurant 7.5mm de côté. Le nombre de pixels a été estimé¹ à 200 x 200 (±50) pixels. Il porte pour seule inscription "TI JAPAN" :



Il est fort probable que ce circuit soit un TC211² de Texas Instruments. La résolution du TC211 est de 192 x 165 pixels et sa fréquence d'acquisition maximum est de 315 images par

¹ Expérience réalisée avec un microscope 40x (Jurg Sommerhalder et Julien Kronegg, décembre 2000).

² Datasheet : socs008b.pdf.

seconde, ce qui semble bien correspondre aux constatations personnelles ainsi qu'aux données fournies par le constructeur de l'*eye tracker*.

Quelques mesures de vitesse ont également été effectuées pour vérifier si les performances de l'EyeLink n'étaient pas trop élevées pour l'utilisation souhaitée. Nous sommes arrivés à la conclusion que l'affichage pouvait être réalisé à une fréquence de 60 images par seconde sans que le sujet puisse "tricher"¹, la fréquence de la caméra restant à 250 images par seconde.

De la même manière, en gardant une fréquence d'affichage de 160 images par seconde, nous avons diminué la vitesse d'échantillonnage de la caméra jusqu'à 21 images par seconde sans qu'il soit possible de tricher.

Ces deux tests montrent que des performances aussi élevées en vitesse ne sont pas forcément nécessaire pour l'utilisation qui a été prévue. Néanmoins, pour ne pas limiter notre *eye tracker* à une utilisation unique, il est essentiel que la vitesse soit la plus élevée possible.

5.5 Cahier des charges

Pour garantir une bonne précision, il est essentiel que la **caméra** puisse fournir une image d'environ 200 x 200 pixels à une fréquence minimale de 21 images par seconde mais si possible supérieure. La technologie de la caméra (infrarouge ou "traditionnelle" est laissée libre).

L'**algorithme** doit fournir une bonne précision à partir de l'image fournie par la caméra. La fréquence de fonctionnement de l'algorithme doit être au moins équivalente à celle de la caméra et son temps de latence doit être inférieur à 10 ms (pour éviter la triche).

¹ A 50 images par seconde, le phénomène bien connu du scintillement était si désagréable qu'il n'a pas été possible de savoir s'il était possible de "tricher".

Chapitre 6:

Caractéristiques d'une image de l'œil

6.1 Introduction

Pour pouvoir dimensionner les traitements numériques qui seront appliqués aux images de l'œil, il est très important d'en connaître les caractéristiques. En effet, ces images contiennent une foule de détails qui influencent les traitements numériques nécessaires pour détecter la position de l'œil.

Ce document est séparé en deux parties décrivant deux types d'images distinctes :

- les images en provenance de caméras dites traditionnelles (spectre visible pour l'Homme);
- les images en provenance de caméras infrarouges (spectre IR).

Ces deux types d'images sont utilisées dans les *eye trackers* mais on constate une préférence pour les images infrarouges. La comparaison des deux types d'images ne permet pas de choisir la technologie de caméra qui sera utilisée et il est obligatoire de se livrer à des tests sur ces deux types d'image.

6.2 Image de caméra traditionnelle

Cette partie décrit les images de l'œil provenant de caméras dites "traditionnelles". Le terme "caméra traditionnelle" regroupe toutes les méthodes permettant d'obtenir une représentation numérique d'une image dont les couleurs sont visibles pour l'œil humain (spectre de lumière visible). Pour illustrer notre propos, prenons comme exemple l'image suivante :



¹ Source : Caméra numérique Panasonic NV-EX1, œil de l'auteur.

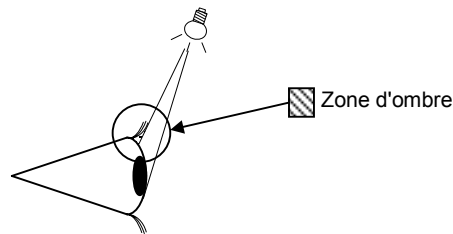
Cette image comporte toutes les caractéristiques que l'on trouve généralement sur des images d'œil :

- ombres;
- reflets;
- pupille;
- luminosité.

Ces caractéristiques sont détaillées dans les paragraphes qui suivent. Celles-ci sont particulières au type de caméra utilisé et aux conditions d'expérience. Ainsi, une image provenant d'une caméra infrarouge n'aura pas les mêmes caractéristiques.

6.2.1 Ombres

Les ombres sur l'œil sont en général provoquées par tout corps se trouvant entre l'œil et une source lumineuse. L'exemple typique est le reflet de la paupière lorsque la source lumineuse et la caméra forment un angle trop grand :



Dans l'image ci-dessus, la source lumineuse est située en dessus de l'axe de la pupille tandis que la caméra est située dans l'axe de la pupille. Cela provoque une ombre d'environ 3 mm de hauteur que l'on peut voir juste sous la paupière.

6.2.2 Reflets

Le reflet est un phénomène particulier dû à la structure de la cornée. Celle-ci étant lisse, elle a tendance à refléter la scène qui fait face à l'œil. L'œil peut donc s'apparenter à un miroir convexe (comme par exemple le dos d'une cuillère).

Du point de vue du traitement d'image, ce miroir peut être considéré comme la superposition de deux images de transparence différente. Plus le coefficient de transparence de l'image correspondant au reflet est faible (donc opaque), plus le reflet sera apparent.

Le coefficient de transparence de l'image correspondant au reflet est fonction de deux paramètres :

- l'intensité lumineuse dégagée par la scène;
- la distance entre l'œil et la scène.

L'intensité lumineuse dégagée par la scène provient des sources lumineuses qui composent la scène. Il s'agit par exemple des lampes ou de la lumière naturelle traversant une fenêtre. Plus une scène est lumineuse, plus elle est reflétée sur la surface de l'œil. C'est pour cette raison que les sources lumineuses sont les plus visibles dans l'image formant le reflet. Ce phénomène est expliqué par l'intensité lumineuse définie selon les réflexions ambiantes et spéculaires¹.

¹ Cf. [8], chapitre I.4.3.5.

La distance entre l'œil et la scène a une influence sur la visibilité du reflet : plus un objet est proche de l'œil, plus le reflet correspondant sera grand et donc davantage visible que si l'objet était éloigné (la taille du reflet est au maximum égale à la taille de l'objet lorsque celui-ci est contre la cornée)¹.

Comme on l'a vu, l'œil reflète la scène. Les éléments les plus visibles dans le reflet sont les suivants :

- cils;
- objectif de la caméra;
- réflexions spéculaires;
- réflexions ambiantes.

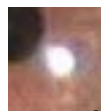
Tous ces éléments sont présents dans l'image présentée précédemment. On peut constater qu'il s'agit des objets les plus proches de l'œil et des sources lumineuses, ce qui vient appuyer les deux règles définies pour le coefficient de transparence.

Pour toutes les réflexions lumineuses (ambiantes et spéculaires), le reflet est de la couleur de la source lumineuse.

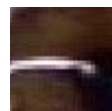
Il existe plusieurs sortes de sources lumineuses provoquant plusieurs types de reflets sur l'œil. Voici les trois formes qui ont été rencontrées (rectangle, ellipse, ligne) :



fenêtre²



lampe³



néon⁴

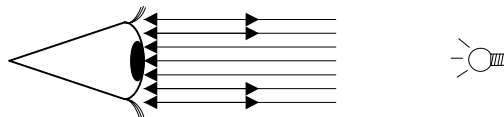
Plus le reflet de la source lumineuse est grand, plus une grande partie de l'image est cachée et donc perdue, ce qui rend plus difficile la détection du centre de l'œil. Il est donc judicieux de réduire les reflets au maximum.

Les reflets de sources lumineuses sont caractérisés par leur haute luminosité. Il s'agit en général des zones les plus claires de l'image. Le reflet de la source lumineuse étant de petite taille (environ une dizaine de pixels), il est difficilement observable.

Pour faciliter le traitement, il est nécessaire d'éviter le maximum de reflets sur l'œil. Pour cette raison, il est important que la scène soit le moins lumineuse possible. Une scène trop sombre n'est cependant pas idéale pour la détection. En réalité, ce n'est pas la luminosité qui est dérangeante mais plutôt la non uniformité de cette luminosité.

6.2.3 Pupille

Contrairement au reste de l'œil, la pupille ne reflète pas la lumière de par sa caractéristique translucide :



¹ Cf. [9], leçon 4 (miroirs convexes).

² Source : oeil_001.jpg.

³ Source : oeil_036.jpg.

⁴ Source : oeil_021.jpg.

Par conséquent, elle apparaît de couleur sombre, le plus souvent noire.

Cependant, si l'œil est éclairé par une source lumineuse dans le même axe que la caméra, la pupille apparaîtra avec une couleur claire (méthode *bright pupil* utilisée par certains *eye trackers*). Ce phénomène est bien connu des photographes sous le nom d'effet "yeux rouges" apparaissant sur les sujets photographiés avec le flash.

La taille de la pupille varie selon la luminosité (accommodation de la taille de l'iris) ainsi que de l'angle de vue de la caméra (plus l'angle de la caméra est faible, plus la pupille est grande). En fonction de sa taille, la pupille représente une surface variable de l'image. Les chiffres suivants ont été relevés (en % du nombre de pixels total de l'image) :

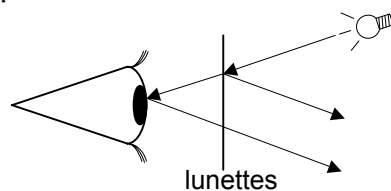
- 1 % : pupilles petites;
- 4 % : pupilles moyennes;
- 8 % : pupilles grandes.

6.2.4 Luminosité

Une bonne illumination de l'image est nécessaire pour permettre une détection correcte du centre de l'œil. En effet, lorsque l'image n'est pas bien illuminée, le contraste est mauvais et donc la pupille ne se détache pas bien du reste de l'image.

6.2.5 Lunettes et lentilles de contact

L'utilisation de lunettes ou de lentilles de contact accentue le phénomène de reflets, étant donné que les lunettes/lentilles modifient "virtuellement" la courbure de la cornée (permettant ainsi la correction visuelle) :

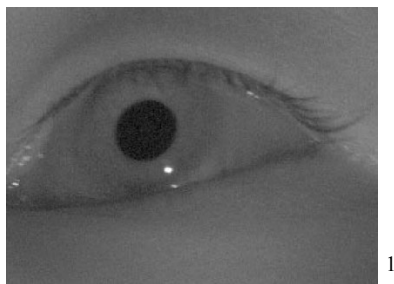


Des lunettes ou lentilles qui ne sont pas parfaitement propres introduisent un bruit de type impulsionnel (*salt & pepper*) sur l'image. Les lunettes ou lentilles teintées réduisent le contraste de l'image tandis que les lunettes fortement réfléchissantes (p.ex. les lunettes solaires de glacier) empêchent totalement de voir l'œil (coefficient de transparence nul). Pour conserver une image possédant les mêmes caractéristiques fréquentielles que l'image sans lunettes, il est donc impératif que les lunettes ou lentilles ne soient pas traitées.

6.3 Image de caméra Infrarouge (IR)

La caméra IR permet d'obtenir des images en une multitude de niveaux de gris (généralement 256). L'image IR est bien moins sujette aux reflets qu'une image prise avec une caméra traditionnelle. En effet, la majorité des reflets visibles sur une image "traditionnelle" est causée par la lumière naturelle. Or la lumière naturelle ne contient que très peu de composantes IR, ce qui explique que les reflets sont moins importants dans une image IR.

Bien entendu, comme la lumière IR est peu importante dans le spectre visible, la luminosité est trop faible pour donner des images de bonne qualité. Pour contrer cet effet, on éclaire l'œil avec une source IR. L'image ainsi obtenue a une luminosité suffisante. Voici un exemple d'image IR obtenue par ce principe :



On peut constater que le reflet de la source lumineuse IR est visible. Le changement majeur introduit par l'infrarouge est le changement de profondeur de champ (*depth of field*) ainsi que le changement de résolution. La profondeur de champ augmente linéairement avec la fréquence de la source IR tandis que la résolution diminue linéairement avec l'inverse de la fréquence de la source lumineuse. Le contraste d'une image infrarouge est ainsi supérieur à celui d'une image traditionnelle. Les images infrarouges possèdent encore l'avantage qu'il est possible d'éclairer l'œil du patient sans qu'il soit ébloui puisque l'œil humain ne perçoit pas l'infrarouge.

¹ Source : oeil_044.jpg.

Chapitre 7:

Traitement d'image pour la détection du centre de la pupille

7.1 Introduction

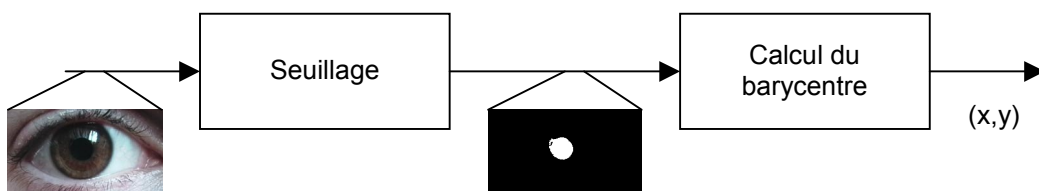
Ce chapitre propose un traitement d'image pour la détection du centre de la pupille. Il est écrit en partant du fait que le lecteur est familier avec les notions de traitement d'image (seuillage, histogramme, filtres, érosion, dilatation, etc.). Si ce n'est pas le cas, il est fortement conseillé de consulter l'Annexe A décrivant ces techniques de traitement d'image.

7.2 Principe de fonctionnement

Comme cela a été décrit dans un précédent chapitre, la pupille est de couleur sombre. Cette propriété est très intéressante car la pupille est en général la partie la plus foncée de l'image (celle qui renvoie le moins la lumière).

Grâce à la caractéristique foncée de la pupille, il est possible d'effectuer un seuillage sur l'image de manière à faire apparaître la pupille seule. Il est ensuite possible de calculer le centre de la pupille selon la méthode du barycentre.

Le fonctionnement de l'algorithme est alors le suivant :



7.3 Description de l'algorithme

L'algorithme de traitement imaginé comporte deux phases principales : le seuillage et le calcul du barycentre. Cependant, cette approche simple ne permet pas d'obtenir des résultats satisfaisants. Nous verrons par la suite quels sont les problèmes qui sont inhérents à cette approche et également comment procéder pour les éviter, ou du moins comment les minimiser.

7.3.1 Seuillage

Le but du seuillage est d'extraire de l'image de l'œil tous les pixels qui appartiennent à la pupille. La couleur de la pupille étant en général la plus sombre de l'image, la valeur du seuil choisie sera relativement faible. Voici par exemple l'image 1 avec un seuil de 26 :

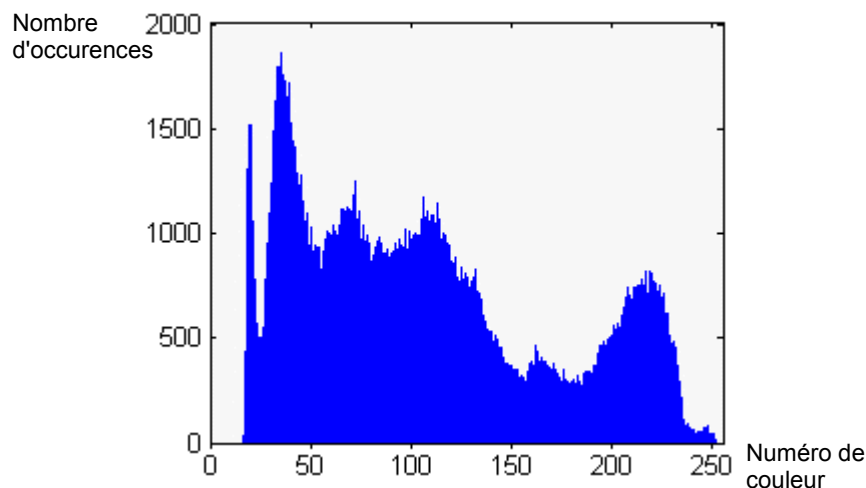


La principale difficulté dans le seuillage de l'image consiste à déterminer le seuil idéal. En effet, le seuil dépend des conditions d'expérience (quantité de lumière, orientation de la caméra, etc.). Certains *eye trackers* utilisent un seuil fixe qui est configuré par l'utilisateur lors de la phase de calibration de l'appareil. D'autres utilisent un seuil adaptatif calculé automatiquement.

L'utilisation d'un seuil fixe possède l'inconvénient de ne pas pouvoir fournir des résultats fiables lorsque les conditions d'expérience varient, ce qui peut entraîner le besoin d'une nouvelle calibration. L'utilisation d'un seuil adaptatif possède l'inconvénient d'être plus compliqué à réaliser que le seuil fixe. De plus, un certain temps de calcul est nécessaire pour la recherche du seuil. Cependant, aucune calibration n'est nécessaire.

L'*eye tracker* utilisé par la clinique d'ophtalmologie utilise un seuil fixe et les problèmes énoncés ci-dessus ont été constatés lors de plusieurs expériences. L'algorithme développé utilisera donc un calcul du seuil automatique.

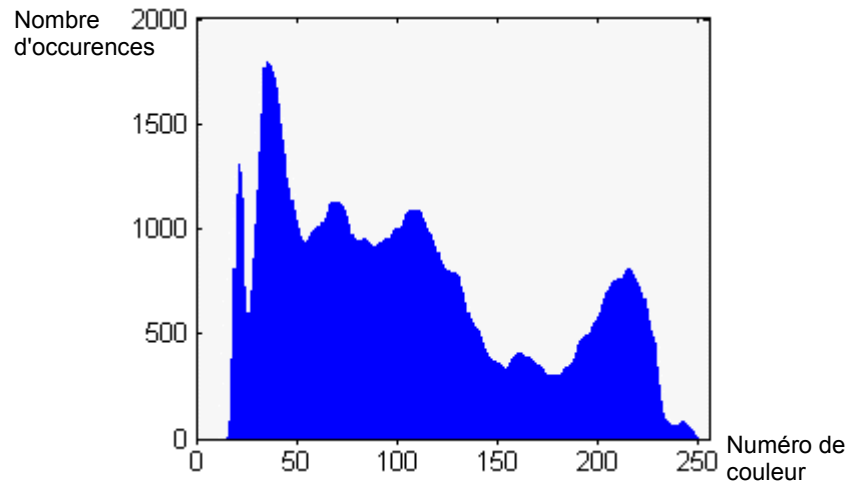
La méthode choisie consiste à utiliser l'histogramme de l'image pour déterminer le seuil optimal :



Le premier mode de l'histogramme contient les pixels les plus sombres de l'image de l'œil et par conséquent la pupille. En calculant les bornes de ce mode de l'histogramme, il est possible

¹ segmentation des couleurs 14 à 26 de `oeil_001.jpg`.

d'obtenir une bonne estimation du seuil optimal. Les bornes de ce premier mode sont détectées par le calcul des minimums locaux de l'histogramme (méthode du gradient). La forme non lisse de l'histogramme rend ce calcul difficile mais il devient très simple lorsque l'histogramme est filtré :



Deux filtres sont utilisés : un filtre médian d'ordre 5 qui a pour effet de supprimer le bruit impulsionnel et un filtre passe-bas d'ordre 3 qui a pour effet d'améliorer le lissage.

Dans certains cas cependant, il est possible que le premier mode de l'histogramme ne contienne pas uniquement la pupille mais également d'autres éléments comme les cils et les ombres. Dans ce cas, il est possible de diminuer la taille du premier mode de l'histogramme en faisant l'hypothèse que la pupille est la zone la plus sombre et qu'elle se situe donc dans le début du mode.

Comme nous l'avons vu dans le chapitre décrivant une image de l'œil, la pupille occupe une certaine surface de l'image qui varie entre 1 et 8%. Si l'on place une petite marge, on peut donc dire que tout premier mode contenant plus de 10% des pixels de l'image contient une pupille et d'autres éléments. Il est alors raisonnable de diminuer la borne supérieure du premier mode de l'histogramme jusqu'à ce qu'il contienne bien 10% de pixels de l'image au maximum. Pour réaliser cette adaptation de la taille du premier mode, une recherche dichotomique sur l'histogramme cumulé de l'image a été utilisée. Plusieurs limites de pourcentage ont été testées (5, 6, 7, 8, 9, 12, 15, 20%) mais il se trouve que 10% est un bon compromis, la diminution de la borne supérieure du premier mode entraînant une dégradation de la forme de la pupille et par conséquent une certaine erreur sur la position de la pupille.

Le temps de calcul du seuil adaptatif est de l'ordre d'un parcours de l'image, soit $O(n)$, n étant le nombre de pixels composant l'image. Bien que ce calcul puisse être fait idéalement à chaque image, on peut considérer que les conditions d'expériences varient de manière relativement lente ce qui pourrait permettre de ne calculer le seuil adaptatif que de temps en temps (par exemple une fois toutes les 1000 images).

7.3.2 Calcul du barycentre

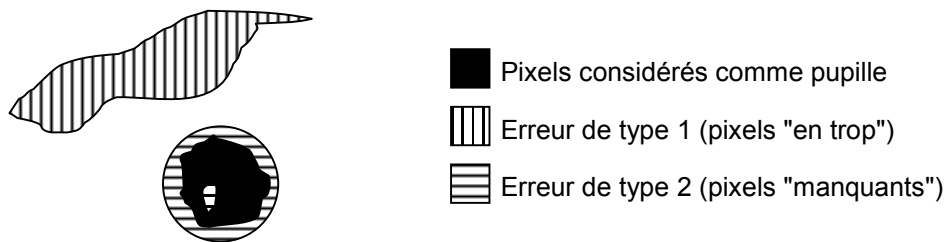
La méthode du barycentre consiste à calculer la position du centre d'un objet en utilisant toutes les particules composant l'objet. Dans notre cas, les objets sont des images de la pupille

et les particules sont des pixels. Chaque particule influence le barycentre calculé d'une manière équivalente (np est le nombre de particules intervenant dans le calcul du barycentre) :

$$\text{centre } x = \frac{\sum x_i}{np} \quad \text{centre } y = \frac{\sum y_i}{np}$$

La complexité du calcul du barycentre est de l'ordre d'un parcours de l'image, soit $O(n)$, n étant le nombre de pixels de l'image.

Lorsque l'on utilise la méthode du barycentre pour le calcul du centre de la pupille, deux type d'erreur peuvent se produire. Elles apparaissent une fois que l'image de l'œil a été seuillée. Ces erreurs sont nommées erreur de type 1 et erreur de type 2, par exemple sur le schéma suivant :



L'erreur de type 1 provient de toutes les composantes sombres de l'image (ombres, cils, etc.) qui ont été prises pour une pupille lors de l'opération de seuillage et qui correspondent à des pixels en trop. L'erreur de type 2 provient de tous les pixels manquants à la pupille et qui n'ont pas été pris en compte lors de l'opération de seuillage. Ces deux types d'erreur peuvent donc être considéré comme le résultat de l'incertitude sur le seuillage.

Dans le cas idéal, il faudrait que les pixels considérés comme pupille occupent toute la place de la pupille (erreur de type 2=0) et qu'aucun pixel à l'extérieur de la pupille ne soit considéré comme appartenant à la pupille (erreur de type 1=0). Comme ce n'est que très rarement le cas, il est nécessaire de calculer l'erreur sur la position du centre de la pupille qui est introduite par ces deux types d'erreur.

La partie qui suit montre un certain nombre d'équations permettant de connaître la précision de la méthode du barycentre. Ces équations utilisent la notation particulière suivante (notons un abus de notation : tous les symboles utilisés sont des vecteurs) :

$$\begin{aligned} \sum \blacksquare & \text{ somme des coordonnées des pixels se trouvant dans la zone } \blacksquare \\ \sum \boxminus & \text{ somme des coordonnées des pixels se trouvant dans la zone } \boxminus \\ \sum \boxplus & \text{ somme des coordonnées des pixels se trouvant dans la zone } \boxplus \\ \#\blacksquare & \text{ nombre de pixels se trouvant dans la zone } \blacksquare \\ \#\boxminus & \text{ nombre de pixels se trouvant dans la zone } \boxminus \\ \#\boxplus & \text{ nombre de pixels se trouvant dans la zone } \boxplus \end{aligned}$$

En tenant compte de l'erreur de type 2, on peut dire que le centre réel de la pupille est défini de la manière suivante :

$$\text{centre réel} = \frac{\sum \blacksquare + \sum \boxminus}{\#\blacksquare + \#\boxminus} \quad [pixels]$$

Le centre de la pupille effectivement calculé peut être considéré comme sous deux angles différent :

- **cas idéal** : erreur de type 1 nulle => centre calculé = $\frac{\sum \blacksquare}{\#\blacksquare}$
- **cas non idéal** : erreur de type 1 non nulle => centre calculé = $\frac{\sum \blacksquare + \sum \square}{\#\blacksquare + \#\square}$

A partir de ces trois équations, il est possible de déterminer quelle est l'imprécision sur la position du centre de la pupille introduite par les erreurs de types 1 et 2.

Commençons par la partie la plus simple où l'erreur de type 1 est nulle (notons que cette erreur n'est pas normalisée car celle-ci dépend de la structure du simulateur de vision) :

$$\begin{aligned} \text{imprécision} &= \text{centre réel} - \text{centre calculé} = \\ &= \frac{\sum \blacksquare + \sum \square}{\#\blacksquare + \#\square} - \frac{\sum \blacksquare}{\#\blacksquare} = \dots = \\ &= \frac{\#\blacksquare \cdot \sum \square - \#\square \cdot \sum \blacksquare}{\#\blacksquare(\#\blacksquare + \#\square)} \quad [pixels] \end{aligned}$$

De prime abord, cette équation ne nous apporte pas grand chose car plusieurs termes sont inconnus (notamment ceux faisant intervenir l'erreur de type 2). Cependant, il est possible de faire quelques hypothèses pour améliorer cela.

Supposons que le nombre de pixels de l'erreur de type 2 soit proportionnel au nombre de pixels considérés comme la pupille (ce qui revient à écrire : $\#\square = a \cdot \#\blacksquare$). Supposons également que les pixels considérés comme la pupille sont centrés de sorte que le barycentre de ces pixels soit environ celui de la pupille. Ce qui revient à écrire (ξ est l'erreur de barycentre en pixels) :

$$\frac{\sum \square}{\#\square} = \frac{\sum \blacksquare}{\#\blacksquare} + \xi = \frac{\sum \blacksquare + \xi_2}{\#\blacksquare}, \text{ avec } \xi_2 = \frac{\xi}{\#\blacksquare}$$

Ces deux hypothèses sont raisonnables car elles considèrent que les pixels qui forment la pupille (\square et \blacksquare) forment deux corps imbriqués de la forme d'un disque.

De là, nous pouvons écrire l'imprécision sous la forme suivante :

$$\begin{aligned} \text{imprécision} &= \frac{\#\blacksquare \cdot \frac{\sum \square}{\#\square} - \#\square \cdot \frac{\sum \blacksquare}{\#\blacksquare}}{\#\blacksquare(\#\blacksquare + \#\square)} = \dots = \\ &= \frac{a}{a+1} \frac{\xi_2}{\#\blacksquare} = \\ &= \frac{a}{a+1} \frac{\xi}{\#\blacksquare^2} \quad [pixels] \end{aligned}$$

Cette équation permet de dire que l'imprécision introduite par l'erreur de type 2 est très faible (inférieure à 1 pixel). En effet, les termes $\frac{a}{a+1}$ et $\frac{\xi}{\# \blacksquare^2}$ sont toujours inférieurs à 1 pixel.

Nous pouvons à présent nous concentrer sur l'imprécision introduite par l'erreur de type 1 (les deux hypothèses précédentes sont à nouveau utilisées).

$$\begin{aligned}
 \text{imprécision} &= \text{centre réel} - \text{centre calculé} = \\
 &= \frac{\sum \blacksquare + \sum \boxplus}{\# \blacksquare + \# \boxplus} - \frac{\sum \blacksquare + \sum \boxplus}{\# \blacksquare + \# \boxplus} = \dots = \\
 &= \underbrace{\frac{a}{a+1} \frac{\xi}{\# \blacksquare^2}}_{\text{terme dû à l'erreur de type 2}} + \underbrace{\frac{\# \boxplus \cdot \sum \blacksquare - \# \blacksquare \cdot \sum \boxplus}{\# \blacksquare (\# \blacksquare + \# \boxplus)}}_{\text{terme dû à l'erreur de type 1}} \quad [pixels]
 \end{aligned}$$

Contrairement à ce qui a été fait précédemment, il n'est pas possible de faire ici des hypothèses sur les termes $\# \boxplus$ et $\sum \boxplus$. Ceux-ci sont donc estimés en faisant des tests sur un certain nombre d'images traitées.

Ces tests¹ ont permis de déterminer plusieurs choses intéressantes :

- le coefficient a vaut entre 0.01 et 1.7;
- le $\# \boxplus$ moyen est de 1.5 fois le $\# \blacksquare$ (min=0.001, max=8.5).

Ces constatations ont permis de voir deux choses :

- le terme relatif à l'erreur de type 2 est inférieur à 0.05 pixels;
- le terme relatif à l'erreur de type 1 est compris entre -42 et 50 pixels (horizontal comme vertical).

Si l'erreur de type 2 n'influence guère la position de la pupille mesurée, on ne peut pas en dire autant pour l'erreur de type 1. Il faut cependant noter que l'erreur de type 1 est peu importante sur les images infrarouges du fait du contraste particulier inhérent à cette catégorie d'image.

Etant donné la résolution spécifiée dans le cahier des charges de l'*eye tracker*, il est nécessaire de concevoir un algorithme qui permette d'éliminer l'erreur de type 1. Par contre, l'erreur de type 2 a une bien moins grande importance et il est donc moins impératif de la compenser.

7.3.3 Elimination de l'erreur de type 1

Pour éliminer l'erreur de type 1, il faut faire disparaître les pixels qui ne font pas partie de la pupille. Plusieurs méthodes ont été envisagées (il en existe bien sûr d'autres) :

- masquage de zones de l'image;
- érosion;
- identification de zones.

¹ Cf. fichier MS-Excel : calcul d'erreur sur le barycentre.xls.

Le **masquage de zones de l'image** possède l'avantage d'être très simple à réaliser (il suffit d'effacer tous les pixels qui se trouvent dans une zone fixe de l'image définie à l'avance). Malheureusement, les zones à masquer (cils, ombres, paupières, etc.) sont mobiles et ne peuvent donc pas être masquées efficacement par une telle méthode.

L'**érosion** consiste à éliminer une zone de l'image qui ne satisfait pas à certains critères de morphologie mathématique. Pour éliminer les zones produisant l'erreur de type 1, il est nécessaire que l'élément structurant soit plus grand horizontalement ou verticalement que la zone à éliminer. Un inconvénient de cette méthode est donc que la pupille doit avoir une taille supérieure (horizontalement ou verticalement) à la zone à éliminer et à l'élément structurant. Un autre inconvénient de cette méthode est le temps de calcul nécessaire. En effet, les traitements à effectuer sont importants car l'image et l'élément structurant sont de grande taille.

L'**identification de zones** consiste à déterminer quelle est la zone de l'image qui correspond à la pupille et quelle sont les zones qui correspondent à l'erreur de type 1.

L'érosion et l'identification de zones permettent de bons résultats. Cependant, comme on le verra dans les paragraphes suivants, l'érosion est très coûteuse en terme de temps de calcul et c'est pour cela c'est la méthode de l'identification de zones qui sera utilisée.

Erosion

L'érosion permet de diminuer voir de supprimer l'erreur de type 1. Pour cela, il suffit d'éroder suffisamment pour que tous les pixels externes à la pupille disparaissent. L'inconvénient est que la taille de l'élément structurant est difficile à estimer. En effet, il faut éroder suffisamment pour enlever tous les pixels n'appartenant pas à la pupille mais pas trop pour qu'il reste des pixels appartenant à la pupille (c'est-à-dire qu'il faut que $\text{taille}(\text{pupille}) > \text{taille}(\text{reste})$).

Il y a donc deux possibilités : soit on donne une taille d'élément structurant fixe, soit elle est calculée automatiquement.

Si la **taille de l'élément structurant est fixe**, cela pose un problème parce que la taille de la pupille varie au cours du temps (elle est fonction de la quantité de lumière qui traverse l'iris). Il est donc possible que tous les pixels de la pupille soient éliminés si la pupille est trop petite. Par contre, dans le cas où la pupille est trop grosse, cela ne pose aucun problème particulier (sauf peut-être qu'on aurait pu éroder un peu plus).

Si la **taille de l'élément structurant est calculée automatiquement**, le problème de la variation de la taille de la pupille n'est plus dérangeant. L'inconvénient est par contre le temps de calcul nécessaire pour déterminer la taille de l'élément structurant.

L'algorithme de calcul automatique de la taille de l'élément structurant a été implanté et évalué. Cette méthode permet une bonne précision de l'érosion (on érode ni trop ni trop peu) et par conséquent sa robustesse au bruit est importante.

Pour déterminer la taille de l'élément structurant à utiliser, la méthode suivante a été utilisée : on commence avec un élément structurant de très grande taille (par exemple 128 x 128). On regarde ensuite s'il reste quelque chose sur l'image. Si l'image ne contient plus de pixels

blancs, cela veut dire que l'on a trop érodé et on diminue la taille de l'élément structurant. Si au contraire il reste quelque chose, cela veut dire que l'on a pas suffisamment érodé et on augmente la taille de l'élément structurant. En procédant ainsi avec une méthode dichotomique, la taille optimale de l'élément structurant est trouvée en un maximum de 7 érosions.

L'érosion peut être calculé de deux manières différentes : méthode spatiale et méthode fréquentielle.

La première consiste à balayer l'image avec l'élément structurant (**méthode spatiale**). La complexité est alors de $O(n \cdot m)$ avec n le nombre de pixels de l'image et m le nombre de pixels du masque. L'avantage de cette méthode réside dans le fait qu'elle est très rapide lorsque l'image et l'élément structurant sont de petite taille. Ce n'est malheureusement pas le cas ici, le masque ayant une taille moyenne de 60x60 pixels.

La seconde consiste à travailler dans le domaine fréquentiel¹ (**méthode fréquentielle**). Cette méthode est bien plus rapide que la méthode spatiale mais consomme davantage de mémoire.

Le temps de calcul de la taille de l'élément structurant est donc de $O(7 \cdot \text{érosion})$, ce qui est énorme compte tenu de la taille des éléments structurants utilisés. Si une telle méthode est utilisée, il est donc essentiel que cette taille ne soit pas calculée à chaque image.

Identification de zones

En constatant les problèmes de temps de calcul causés par la méthode d'érosion, une autre technique, basée sur un concept de zones, a été envisagée. Elle consiste en l'identification des zones de pixels blancs dans l'image seuillée, ce qui permet d'éliminer toutes les zones dont la structure n'est pas semblable à celle d'une pupille. La méthode est la suivante : tout d'abord, on sous-échantillonne l'image seuillée, de manière à ce que l'identification des zones se fasse sur une image de relativement petite taille et donc ne prenne pas trop de temps. Pour sous-échantillonner, on considère un bloc de pixels de l'image seuillée de taille fixée. Lorsqu'un certain pourcentage des pixels de ce bloc sont blancs, on considère que le pixel correspondant dans l'image sous-échantillonnée est aussi blanc. Il y a donc deux paramètres pour l'algorithme de sous-échantillonnage : la taille du bloc et le taux de remplissage du bloc. De manière heuristique, il a été déterminé que des blocs de 8 x 8 avec un taux de remplissage de 100% étaient des paramètres qui fournissaient de bons résultats :



image seuillée²



image seuillée sous-échantillonnée (8x8)

Une fois l'image sous-échantillonnée, on procède à l'identification des zones. Pour ce faire, on parcourt l'image séquentiellement à la recherche d'un pixel blanc qui indique le début d'une nouvelle zone. A partir de ce pixel, on procède à une recherche récursive de tous les pixels de la zone. Durant cette recherche, on mémorise les bornes X et Y de la zone ainsi que le nombre de pixels composant la zone (surface). Pour ne pas utiliser un pixel deux fois, on utilise une

¹ Selon un algorithme implanté dans Matlab 6.

² Source : oeil_001.jpg avec seuil de 26.

marque qui sera placée sur le pixel dès qu'il a été utilisé. L'algorithme de recherche est le suivant (algorithme de recherche récursif de zone par ligne avec connexité 4) :

```

Detecte_zone(x,y, marque, image_sous_ech, surface_zone, bornes_zone)
  Recherche de la borne gauche de la ligne
  Recherche de la borne droite de la ligne

  Surface_zone += longueur de la ligne
  Marquage des pixels de la ligne

  Mise à jour des bornes de la zone

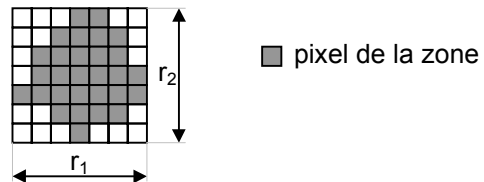
  For i=debut de la ligne jusqu'à fin de la ligne
    If (ligne_supérieure(i) in zone et pas marqué)
      Detecte_zone(i,y-1, marque, image_sous_ech, surface_zone, bornes_zone);
    End if
    If (ligne_inférieure(i) in zone et pas marqué)
      Detecte_zone(i,y+1, marque, image_sous_ech, surface_zone, bornes_zone);
    End if
  End for

```

Cet algorithme limite le nombre de niveaux de récursivité au nombre de pixels de l'image sous-échantillonnée lorsque la zone est concave et au nombre de lignes lorsque la zone est convexe. En admettant une image seuillée de taille 800x600 et un bloc de taille 8x8, cela correspond à un niveau de récursivité maximum de 70 (zone convexe) ou de 7000 (zone concave), ce qui reste acceptable en terme de quantité de mémoire comme en temps de calcul.

Certaines caractéristiques des zones, par exemple la densité, permettent de choisir la zone qui contient vraisemblablement la pupille. Ces caractéristiques sont définies par les formules suivantes :

$$\begin{aligned}
 dimension &= r_1 \cdot r_2 \\
 surface &= \# \text{ pixels} \\
 densité &= \frac{surface}{dimension}
 \end{aligned}$$



L'idée maîtresse est simple : plus la forme de la zone s'approche de celle d'une pupille (une ellipse), plus la densité de la zone s'approche de celle d'une ellipse :



$$\begin{aligned}
 r_1 &= 2 \\
 r_2 &= 2 \\
 dim &= 4 \\
 surf &= 4 \\
 d &= 1
 \end{aligned}$$



$$\begin{aligned}
 r_1 &= 3 \\
 r_2 &= 4 \\
 dim &= 12 \\
 surf &= 8 \\
 d &= 0.667
 \end{aligned}$$



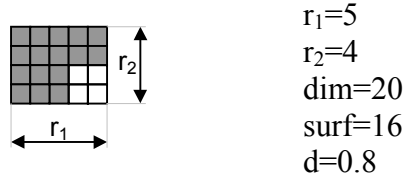
$$\begin{aligned}
 r_1 &=? \\
 r_2 &=? \\
 dim &= r_1 \cdot r_2 \\
 surf &= \pi/4 \cdot r_1 \cdot r_2 \\
 d &= \pi/4 = 0.785
 \end{aligned}$$

Une zone est finalement caractérisée par :

$$\Delta R = abs \left(\frac{surface}{dimension} - \frac{\pi}{4} \right)$$

On peut alors classer les zones en fonction de leur différence de rapport caractéristique (ΔR). La zone qui obtient le ΔR le plus faible est qui est le plus probablement la pupille.

Il est évident que cette méthode n'est pas absolue puisqu'il peut y avoir plusieurs zones candidates à être une pupille. En effet, le rapport ΔR tient compte uniquement de la densité de la zone et non de sa forme. Voici par exemple une zone qui a la densité d'une ellipse sans en satisfaire la forme :



Il est cependant possible de caractériser les zones plus finement lorsque l'on dispose d'information a priori, comme par exemple la taille de la pupille.

Une fois que la zone contenant la pupille a été identifiée, on peut calculer le barycentre uniquement sur cette zone, ce qui permet de limiter l'erreur de type 1.

Idéalement, il faudrait recomposer la zone correspondant à la pupille dans l'image seuillée, ce qui impliquerait une nouvelle recherche de zone. Cependant, comme on a fait l'hypothèse que la zone de la pupille a une forme proche d'un rectangle, on peut se contenter de calculer le barycentre dans une zone réduite de l'image seuillée (clipping). La zone de clipping est calculée à partir des bornes de la zone déterminée avec l'image sous-échantillonnée par un changement d'échelle correspondant à la taille du bloc. L'inconvénient de la technique de clipping est que la zone ainsi définie n'élimine pas forcément toute l'erreur de type 1 mais peut également augmenter l'erreur de type 2, ce qui pourrait rendre nécessaire le parcours de la zone "pupille" dans l'image seuillée.

La complexité de cet algorithme est de l'ordre d'un parcours d'image pour le sous-échantillonnage et d'un parcours de l'image sous-échantillonnée pour la classification des zones. Au total : $O(M+m)$ avec M la surface de l'image et m la surface échantillonnée. Cela équivaut à environ $1.1 \cdot M$ opérations, soit $O(M)$.

Les résultats¹ produits par cette méthode sont très intéressants car il permettent de réduire l'erreur maximum de 64 pixels horizontalement et 54 pixels verticalement (bloc de 8 x 8 avec un taux de remplissage de 100% et clipping). Cependant, dans environ 30% des cas, l'erreur de type 2 augmente, ce qui pourrait justifier un remplacement du clipping par un parcours de zone.

7.3.4 Elimination de l'erreur de type 2

Bien que l'erreur de type 2 soit bien moins importante que l'erreur de type 1, il peut aussi être intéressant de la corriger. Cela peut être le cas lorsque l'on traite des images infrarouges avec une erreur de type 1 et une erreur de type 2 du même ordre de grandeur. Cependant, pour que cette correction soit rentable, il est essentiel que les algorithmes de calcul soient très légers en terme de charge de calcul.

¹ Cf. fichier MS-Excel : calcul d'erreur sur le barycentre.xls.

Tout d'abord, caractérisons les éléments qui induisent l'erreur de type 2. On distingue deux types d'éléments :

- les éléments qui font partie de l'œil;
- les éléments externes à l'œil.

Parmi les éléments faisant partie de l'œil, on peut compter toutes les taches, défauts et imperfections de l'œil apparaissant lors de maladies. On peut également noter les éléments qui ne font pas directement partie de l'œil mais qui s'y apparentent comme par exemple les taches sur les lunettes ou les lentilles de contact. Ces taches sont de petite taille et peuvent donc être considérées comme du bruit impulsif, comme cela a déjà vu au chapitre précédent.

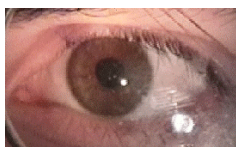
Parmi les éléments externes à l'œil, ce sont les reflets qui sont désignés comme principal coupable. Les ombres, elles, n'apparaissent en général pas sur la pupille.

Elimination des taches

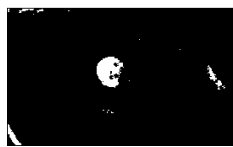
Admettons que l'appareil réalisé ici sera utilisé exclusivement sur des sujets sains et donc exempts de maladies oculaires. Nous pouvons alors dire que les seules taches qui peuvent apparaître sont des impuretés. Etant donné leur petite taille, il est possible de les éliminer par exemple avec ces deux méthodes :

- filtre médian;
- fermeture morphologique (dilatation suivie d'une érosion).

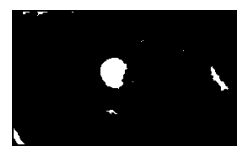
Voici un exemple d'élimination de taches avec la fermeture morphologique :



original¹



seuillage pupille



après fermeture morphologique 3x3

Cette méthode a été utilisée avec succès pour toutes les images présentant ce type de défaut (avec un élément structurant de taille 3 à 20, 3x3 donnant les meilleurs résultats).

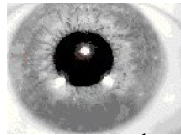
Malheureusement, l'amélioration de précision sur la position de la pupille ne justifie pas la charge de calcul supplémentaire et ce pour les deux méthodes indiquées précédemment. En effet, la quantité de bruit (nombre de pixel) étant très faible, il n'y a pas de changements significatifs lors du calcul du barycentre.

Elimination des reflets

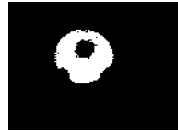
L'élimination des reflets pose le plus sérieux problème rencontré dans ce travail. En effet, comme cela a été indiqué précédemment, un reflet correspond à une perte d'information de l'image de l'œil. Il est alors très difficile de reconstituer les parties manquantes. Néanmoins, dans certains cas particuliers, il est possible d'annuler l'effet des reflets. C'est le cas pour le reflet de sources lumineuses de forte intensité comme les lampes. Pour cela un algorithme a été développé.

¹oeil_035.jpg.

Les sources lumineuses possèdent une caractéristique qui est à l'opposé des caractéristiques de la pupille. En effet, une source lumineuse est très claire (généralement la zone la plus claire de l'image) et sera donc représentée dans le dernier mode de l'histogramme de l'image. On peut donc reconnaître les sources lumineuses en déterminant les bornes du dernier mode de l'histogramme et en effectuant une segmentation de l'image.



original¹



seuillage pupille



seuillage sources



sources+pupille

Il est à présent possible d'effectuer une addition des images binaires de la pupille et des sources lumineuses (OR). Ainsi, si une source lumineuse produisait un reflet directement sur la pupille et produisait par conséquent un trou, ce trou sera bouché par l'image binaire des sources lumineuses, réduisant ainsi l'erreur de type 2. Si par contre une source lumineuse produisait un reflet à l'extérieur de la pupille, les pixels ajoutés par l'image binaire des sources lumineuses seraient éliminés lors de la phase d'érosion, ce qui n'augmenterait pas l'erreur de type 1.

Ce type d'algorithme fait intervenir plusieurs dilatations de taille différentes. La charge de calcul est donc relativement importante. Cette méthode est alors peu rentable, même si la quantité de pixels est plus importante pour les sources lumineuses que pour le bruit impulsif.

¹oeil_023.jpg.

Chapitre 8: Résultats obtenus

8.1 Implantation de l'algorithme

L'algorithme décrit au chapitre précédent a été implanté sous différentes formes de prototypes:

- Matlab : tests de précision;
- Java : tests de vitesse;
- C : tests de vitesse sur DSP.

Matlab a servi de base pour une première implantation de l'algorithme. Il s'agissait alors de voir si l'algorithme apportait les résultats escomptés. De par son grand nombre de fonctions prédéfinies, Matlab est un excellent outil pour du prototypage rapide.

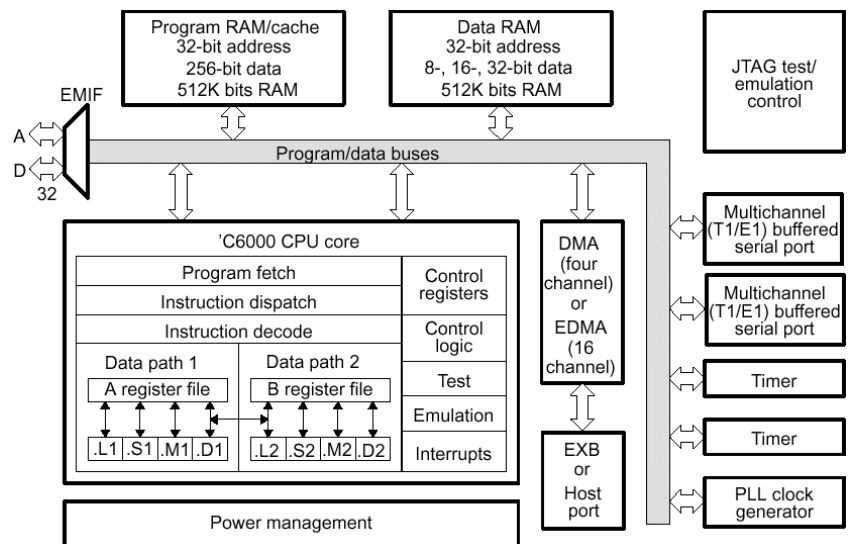
Java a ensuite servi de plate-forme intermédiaire pour valider l'algorithme en terme de vitesse. Le portage direct vers C n'a pas été choisi pour plusieurs raisons :

- tout d'abord, Matlab intègre Java ce qui permet de porter l'algorithme de Matlab en Java de manière progressive;
- ensuite, Java est très proche du C, langage de prédilection pour la programmation des processeurs de signaux, ce qui permet une adaptation rapide lorsque le processeur de signal sera choisi (ce qui n'était pas le cas lorsque le portage Matlab vers Java a débuté);
- bien que Java ne soit pas réputé pour sa très grande rapidité (facteur moyen de 3 à 5 fois plus lent que du C), des tests ont montré que la vitesse d'un programme Java était semblable à celle d'un programme en C pour autant que certaines fonctions de Java ne soient pas utilisées (héritage, graphisme, instanciation d'objets).

L'algorithme a ensuite été implanté en C sur un DSP de Texas Instruments (TMS320C6711). Ce choix a été fait pour plusieurs raisons :

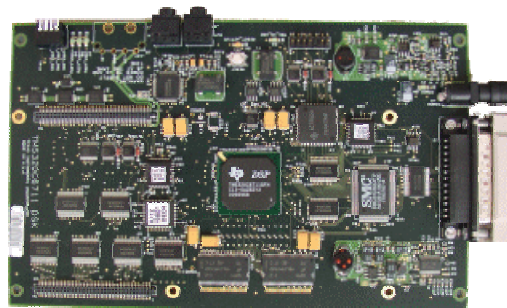
- tout d'abord, l'utilisateur du simulateur de vision doit pouvoir se déplacer. Pour cette raison, il est nécessaire de disposer d'un dispositif de calcul de taille réduite, ce qui n'est pas le cas d'un PC. Il faut néanmoins que la puissance de calcul soit suffisante pour que le cahier des charges soit respecté. Un DSP est donc un bon choix puisqu'il allie une puissance de calcul importante à un encombrement réduit.
- ensuite, Texas Instruments propose un kit de développement TMS320C6711 avec d'excellents outils de développement et ceci pour un prix très modique (environ 300\$).

Le TMS320C6711 est un processeur de Harvard (64 Kb instructions et 64 Kb données dans deux mémoires différentes) à virgule fixe ou flottante à technologie VLIW (*Very Long Instruction Word*). Son pipeline possède 5 étages. L'étage d'exécution est séparé en deux parties comprenant chacune 4 unités de traitement (multiplication, branchement, décalage, ALU) :



Avec une fréquence nominale de 150 MHz, il développe une puissance de traitement de 900 MIPS (puissance de pointe).

La carte de développement comprend bien entendu le processeur mais également une mémoire externe de 16 Mbytes et un codec audio :



Le coût (env. 300\$) ainsi que les excellentes performances (max. 900 MIPS) ont été des arguments indéniables pour le choix de cette carte de développement.

8.2 Précision de l'algorithme

Une base de donnée contenant une série d'images dont le centre et la taille de la pupille a été déterminée "à la main" avec un programme de dessin (Paint Shop Pro 6) sert de base pour le calcul de la précision de notre algorithme. Ces images sont tirées de différentes sources (caméra amateur, séquences vidéo, sites web, etc.). La répartition est la suivante² :

Nbr. d'images	Pourcentage	Source
30	59 %	Images "traditionnelles" (à partir de vidéos)
10	20 %	Images "traditionnelles" (à partir du web)
11	22 %	Images infrarouges (à partir de vidéos/web)

¹ Source : documentation Texas Instruments (spru197d.pdf).

² Pour une répartition détaillée, voir la base de donnée MS-Access : diplome.mdb.

Etant donné que plusieurs variations de l'algorithme existent (avec/sans érosion, avec/sans identification de zones, avec/sans suppression des sources lumineuses, avec suppression du bruit impulsif), il existe plusieurs cas d'erreur à traiter. Dans tous les tests d'erreur ci-dessous, on différenciera également les images infrarouges des images dites traditionnelles. Cela nous permettra de déterminer quel type de caméra est nécessaire pour notre *eye tracker*.

L'identification de zones est destinée à remplacer l'érosion pour des raisons de temps de calcul. Pour cette raison, l'érosion et l'identification de zones ne sont pas réalisées en même temps. Le tableau suivant montre quels sont les versions d'algorithmes testées (seules les versions principales sont présentées ici) :

Numéro de cas	Erosion	Identification de zones	Suppr. des sources lumineuses	Suppr. du bruit impulsif
1	-	-	-	-
2	X	-	-	-
3	-	X	-	-

X=avec, -=sans

Les résultats suivants ont été obtenus avec Matlab¹ et Java². Les résultats sont également disponibles sous forme de fichier Excel (*quantification_erreur.xls*).

Nous commencerons par définir les différentes erreurs mesurables puis nous verrons quelles sont ces erreurs sur différents ensembles d'images.

8.2.1 Erreur sur la position de l'œil

Le stockage des images dans une base de donnée permet une centralisation des données. De plus, il est possible de réaliser des tests sur certaines images de manière automatique, sans intervention humaine. On peut par exemple calculer l'erreur de centre et ceci de la manière suivante :

$$\Delta_{\text{centre}} = \text{centre}_{\text{réel}} - \text{centre}_{\text{calculé}} \quad [\text{pixels}]$$

Malheureusement, comme toutes les images n'ont pas la même taille, il est difficile de comparer l'erreur sur deux images différentes. Pour contrer cet effet, on définit une erreur relative (pourcentage) à partir des deux postulats suivants :

- 1) si $abs(\Delta_{\text{centre}}) \geq \frac{\text{taille}_{\text{pupille}}}{2}$, le centre calculé est en dehors de la pupille et par conséquent l'erreur est de 100 % (le centre ainsi calculé n'a aucune utilité);
- 2) si $\Delta_{\text{centre}} = 0$, il n'y a pas d'erreur sur le centre et par conséquent l'erreur est de 0 %.

A partir de cela, on peut dire que le pourcentage d'erreur est de :

¹ Programme : *get_pupille_db.m*, avec variation de la version d'algorithme dans *get_pupille.m*.

² Programme : *Algo_quantification_erreur.java*.

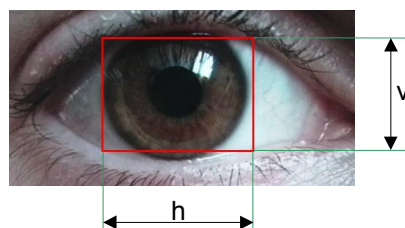
$$erreur \text{ [\%]} = \min \left(abs(\Delta_{centre}) \text{ [pixels]} \cdot \frac{2 \cdot 100 \text{ [\%]}}{taille_pupille \text{ [pixels]}}, 100 \right)$$

De cette manière, on tient compte de la taille de la pupille dans le calcul d'erreur. Il faut cependant noter qu'il s'agit uniquement de l'erreur sur la position de l'œil et non de l'erreur sur le point de regard.

L'erreur exprimée en fonction de la taille de la pupille n'a que peu d'intérêt puisqu'elle ne reflète pas la précision d'un degré demandée dans le cahier des charges. Cependant, certains constructeurs utilisent cette mesure pour caractériser leur *eye trackers*, ce qui permet la comparaison.

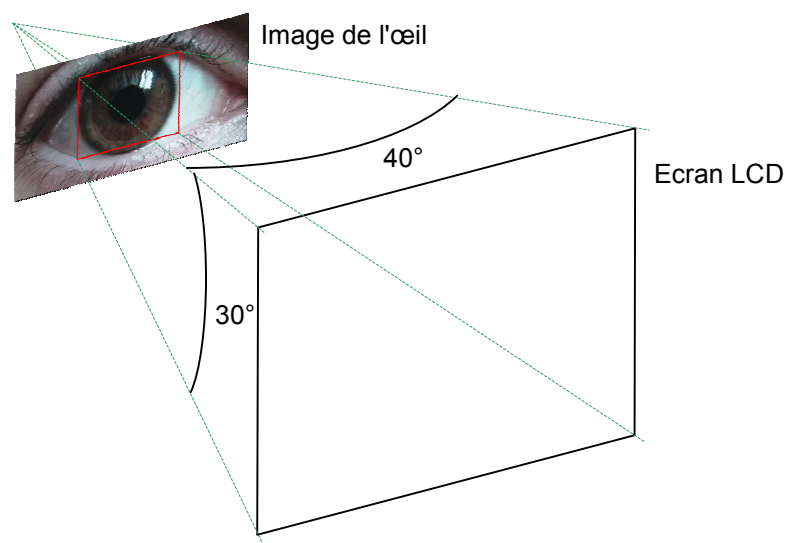
8.2.2 Erreur sur le point de regard

L'erreur sur le point de regard peut également être calculée. Le simulateur doit avoir un champ de vue rectangulaire de 40° horizontalement (θ_h) et 30° verticalement (θ_v), ce qui revient à dire que l'œil peut se déplacer dans un rectangle de cette taille. Numériquement, cela correspond à limiter la position du centre de la pupille à un rectangle de cette taille. Ce rectangle peut être représenté sur une image de l'œil (rectangle rouge, dimensions en pixels) :



Ce rectangle est inscrit dans la partie de l'œil où le déplacement est possible. Il a un rapport de 4:3 ce qui correspond à 40x30°.

Ce fameux rectangle nous permet de mieux comprendre comment se passe le calcul du point de regard et permet aussi de déterminer la résolution avec laquelle il est possible de calculer le point de regard :



Le plan déterminé par l'écran LCD est celui où se posera le point de regard. Le rectangle et l'écran LCD ont les mêmes dimensions à une transformation linéaire près ce qui permet de dire que la résolution du point de regard est fonction du nombre de pixels par ligne/colonne du rectangle rouge ainsi que de l'angle de vue :

$$résolution \left[\frac{^\circ}{pixel} \right] = \frac{\theta_h}{h}$$

On peut également constater qu'une portion importante de l'image n'est pas utilisée. Il est donc possible de réduire le champ de la caméra de l'œil pour n'effectuer des acquisitions que sur une surface correspondant à notre fameux rectangle.

Ces constations nous permettent de dire que plus le champ de la caméra est large (donc moins le rectangle rouge a de pixels), plus le déplacement de l'œil peut être important mais plus la résolution est faible. Dans le sens inverse, plus le champ de la caméra est restreint (donc plus le rectangle rouge a de pixels), plus le déplacement de l'œil est restreint mais plus la résolution est élevée. Pour une grande résolution, il est donc impératif de capter les mouvements oculaires avec un angle de vue raisonnable et de posséder une image de l'œil qui soit la mieux cadrée possible et qui contienne le plus possible de pixels.

La réduction du champ de la caméra au rectangle rouge permet également d'améliorer la précision de l'algorithme puisque certains éléments ne faisant pas partie de la pupille (par exemple les ombres et les paupières) sont en partie hors du champ de la caméra.

Tout ceci nous permet de dire que les indications de précision indiquées par certains fabricants sont des plus floues car la plupart des appareils permettent de régler le champ de la caméra et donc la résolution. La précision est dans ce cas une affaire de réglage puisqu'elle est fonction de la résolution :

$$précision [^\circ] = résolution \left[\frac{^\circ}{pixel} \right] \cdot erreur_position_œil [pixel]$$

8.2.3 Test sur des images infrarouges

Onze images infrarouges¹ ont été utilisées pour le test. Il est évident que ce nombre est trop peu élevé pour permettre une étude statistique des images infrarouges. Cependant, les images infrarouges étant très difficiles à trouver, nous nous contenterons d'une étude simplifiée. Pour les trois algorithmes testés, l'erreur sur la position de l'œil est de l'ordre de 20% de la taille de la pupille, ce qui est comparable à ce qui est donné pour d'autres appareils².

¹ Cf. `diplome.mdb`, images numéros : 22, 23, 24, 44, 45, 46, 47, 48, 49, 50 et 51.

² Par exemple : [14].

Cas 1

Dans cette partie du test, l'algorithme est réduit à sa plus simple expression :

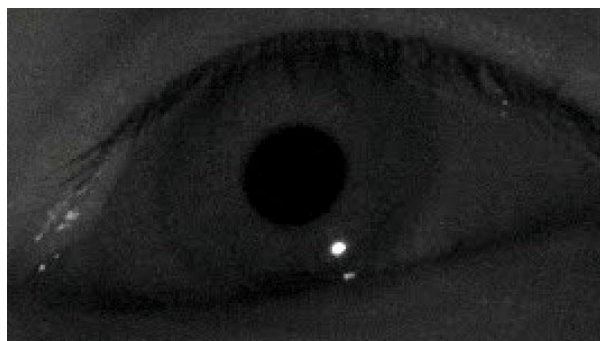
- calcul du seuil optimal sur une image infrarouge;
- barycentre sur l'image segmentée.

Les résultats sont les suivants (les erreurs sur le point de regard sont calculées à partir d'une estimation de la taille du rectangle rouge précité) :

Numéro d'image	abs(delta_centre)		erreur position de l'œil		erreur point de regard	
	X [pix]	Y [pix]	X [%]	Y [%]	X [°]	Y [°]
22	7,14	1,13	62,1	10,3	7,72	1,21
23	0,24	0,35	1,0	1,6	0,07	0,11
24	3,81	1,86	16,9	7,7	1,41	0,69
44	1,04	0,52	2,9	1,4	0,28	0,14
45	17,70	0,82	77,0	3,7	5,32	0,25
46	1,38	12,83	15,4	100,0	0,99	9,17
47	2,71	1,00	5,4	2,1	0,41	0,15
48	3,76	9,36	7,7	18,5	0,85	2,11
49	1,48	1,59	3,0	3,2	0,27	0,29
50	1,16	0,51	2,4	1,1	0,22	0,10
51	0,58	0,00	1,3	0,0	0,11	0,00

Comme le montre la grande dispersion de l'erreur sur le point de regard (0 à 15°), la précision de l'algorithme est très variable, ce qui implique qu'il n'est pas possible d'utiliser n'importe quelle série d'image infrarouge pour atteindre la précision demandée (1°). C'est néanmoins possible en choisissant une technologie de caméra adéquate. Par exemple, les images 47 à 51, dont l'erreur sur le point de regard est de l'ordre de 1°, sont extraites d'une même séquence vidéo, ce qui permet de faire l'hypothèse que n'importe quelle séquence vidéo acquise dans les mêmes conditions permettra d'atteindre la précision de 1° requise par le cahier des charges.

L'utilisation du barycentre seul sur l'image segmentée permet donc d'atteindre la précision requise pour autant que la caméra soit choisie et réglée correctement. On voit par exemple que l'image 51 permet d'obtenir le point de regard avec une précision de 0.11° x 0° (avec résolution de 0.2° x 0.2°) :



¹ oeil_051.jpg.

Cas 2

Cette partie du test est une version améliorée du cas précédent :

- calcul du seuil optimal sur une image infrarouge;
- réalisation d'une érosion optimale;
- calcul du barycentre.

Les résultats sont les suivants :

Numéro d'image	abs(delta_centre)		erreur position de l'œil		erreur point de regard	
	X [pix]	Y [pix]	X [%]	Y [%]	X [°]	Y [°]
22	3,56	0,83	31,0	7,6	3,85	0,89
23	1,00	10,00	4,3	45,5	0,31	3,13
24	5,34	2,32	23,7	9,7	1,98	0,86
44	5,00	16,00	14,1	44,4	1,35	4,32
45	10,50	5,00	45,7	22,7	3,16	1,50
46	1,87	1,49	20,8	11,9	1,34	1,07
47	0,50	0,00	1,0	0,0	0,08	0,00
48	6,03	10,22	12,3	20,2	1,36	2,31
49	2,00	10,00	4,1	20,4	0,37	1,85
50	19,62	4,62	40,0	9,7	3,72	0,88
51	1,00	4,00	2,2	9,1	0,19	0,75

Les résultats de cette méthode montrent que l'érosion n'est pas un algorithme très efficace dans le sens où il réduit l'erreur sur le point de regard sur certaines images l'augmente sur d'autres.

Ce cas s'explique simplement si l'on se souvient que les images infrarouges ont un excellent contraste et que par conséquent l'erreur de type 1 est relativement faible. La précision du barycentre est alors moins importante à cause de la réduction, par l'érosion, du nombre de pixels mis à contribution (augmentation de l'erreur de type 2).

L'utilisation d'une caméra fournissant des images de la qualité de l'image 47 permettrait d'atteindre la précision de 1° requise :



L'erreur sur cette image est de 0.1° x 0° (résolution estimée à 0.15° x 0.15°).

¹ oeil_047.jpg.

Cas 3

Cette partie du test est une variante du premier cas mais avec une étape d'identification de zones. La procédure est donc :

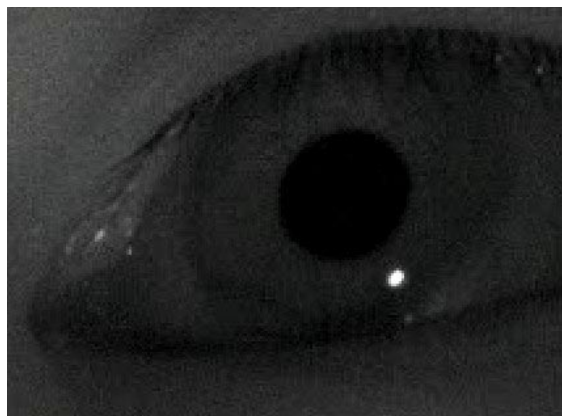
- calcul du seuil optimal;
- identification de zones;
- calcul du barycentre avec clipping.

Les résultats sont les suivants :

Numéro d'image	abs(delta_centre)		erreur position de l'œil		erreur point de regard	
	X [pix]	Y [pix]	X [%]	Y [%]	X [°]	Y [°]
22	2,64	0,71	22,9	6,4	2,85	0,76
23	4,70	3,51	20,0	16,0	1,45	1,10
24	0,91	1,77	4,0	7,4	0,34	0,65
44	0,89	0,65	2,5	1,8	0,24	0,18
45	5,82	39,35	25,3	100,0	1,75	11,80
46	0,46	20,88	5,1	100,0	0,33	14,91
47	0,47	0,27	0,9	0,6	0,07	0,04
48	1,35	8,00	2,7	15,8	0,30	1,81
49	0,44	0,39	0,9	0,8	0,08	0,07
50	0,00	0,53	0,0	1,1	0,00	0,10
51	0,86	0,36	1,9	0,8	0,16	0,07

Par rapport à l'algorithme de base impliquant uniquement le barycentre, cette version possède l'avantage de réduire l'erreur sur le point de regard de presque toutes les images du test. Cette réduction d'erreur est cependant relativement modeste (moins de 1°).

Ici, c'est l'image 50 qui fournit les meilleurs résultats : erreur sur le point de regard de 0° x 0.1° avec une résolution estimée à 0.2° x 0.2° :



L'utilisation d'une caméra fournissant des images de cette qualité permettrait d'atteindre la précision demandée par le cahier des charges.

¹ oeil_050.jpg.

8.2.4 Test sur des images traditionnelles

Quatorze images "traditionnelles"¹ ont été utilisées pour le test. Comme précédemment, il est évident que ce nombre est trop peu élevé pour permettre une étude statistique complète. Cependant, ce test permet de se donner une bonne idée de la qualité de l'algorithme pour des images "traditionnelles".

Cas 1

Dans cette partie du test, l'algorithme est réduit à sa plus simple expression :

- calcul du seuil optimal;
- calcul du barycentre sur l'image segmentée.

Les résultats sont les suivants :

Numéro d'image	abs(delta_centre)		erreur position de l'oeil		erreur point de regard	
	X [pix]	Y [pix]	X [%]	Y [%]	X [°]	Y [°]
1	17,86	27,52	44,6	67,1	3,12	4,80
2	0,90	2,31	2,4	6,1	0,20	0,51
8	169,98	2,92	100,0	6,4	29,56	0,51
19	6,23	1,99	29,7	12,9	1,90	0,61
20	2,02	2,23	10,1	12,1	0,64	0,71
21	3,22	0,04	14,0	0,2	0,89	0,01
26	1,91	3,18	42,4	70,7	2,39	3,98
28	15,70	44,63	100,0	100,0	5,61	15,94
30	0,97	2,22	6,9	17,0	0,31	0,58
31	37,62	49,22	86,5	100,0	5,08	6,65
32	2,13	3,22	19,4	26,8	2,08	3,11
33	1,17	0,56	5,3	2,5	0,36	0,17
40	20,45	0,81	100,0	4,4	5,80	0,23
43	35,09	15,85	100,0	88,1	12,53	5,66

Les erreurs sur le point de regard de ce test sont très disparates (entre 0 et 30° d'erreur) alors que cela n'était pas le cas avec le même algorithme appliqué aux images infrarouges. Ce phénomène est tout à fait normal puisque les images traditionnelles contiennent beaucoup plus d'erreur de type 1 que les images infrarouges.

Néanmoins, si l'on voulait utiliser cet algorithme, il faudrait utiliser une caméra capable de fournir des images comme l'image 33 (erreur sur le point de regard : 0.36° x 0.17°, résolution estimée à 0.3° x 0.3°) :



2

¹ Cf. `diplome.mdb`, images numéros : 1, 2, 8, 19, 20, 21, 26, 28, 30, 31, 32, 33, 40 et 43.

² `oeil_033.jpg` (il s'agit d'un dessin et non d'une photo).

Cas 2

Cette partie du test est une version améliorée du cas précédent :

- calcul du seuil optimal;
- réalisation d'une érosion optimale;
- calcul de barycentre.

Les résultats sont les suivants :

Numéro d'image	abs(delta_centre)		erreur position de l'oeil		erreur point de regard	
	X [pix]	Y [pix]	X [%]	Y [%]	X [°]	Y [°]
1	0,00	2,00	0,0	4,9	0,00	0,35
2	4,00	4,50	10,7	11,8	0,88	0,99
8	2,00	5,00	6,0	11,0	0,35	0,87
19	0,50	8,50	2,4	54,8	0,15	2,60
20	9,00	6,00	45,0	32,4	2,88	1,91
21	10,00	5,00	43,5	24,4	2,76	1,38
26	1,91	3,18	42,4	70,7	2,39	3,98
28	1,00	0,00	6,7	0,0	0,36	0,00
30	1,00	1,00	7,1	7,7	0,32	0,26
31	2,00	0,00	4,6	0,0	0,27	0,00
32	3,00	3,00	27,3	25,0	2,93	2,90
33	0,50	0,00	2,3	0,0	0,15	0,00
40	0,00	3,00	0,0	16,2	0,00	0,85
43	2,00	2,00	10,5	11,1	0,71	0,71

L'érosion sur des images traditionnelles permet de beaucoup réduire l'erreur de type 1 (plus de 2/3 des images ont un point de regard calculé avec une précision inférieure au degré).

C'est à nouveau l'image 33 qui produit le meilleur résultat du test et qui peut être utilisée pour obtenir la précision requise par le cahier des charges. L'erreur serait dans ce cas de $0.15^\circ \times 0^\circ$.

Cas 3

Cette partie du test est une variante du premier cas mais avec une étape d'identification de zones. La procédure est donc :

- calcul du seuil optimal;
- identification de zones;
- calcul du barycentre avec clipping.

Les résultats sont les suivants :

Numéro d'image	abs(delta_centre)		erreur position de l'oeil		erreur point de regard	
	X [pix]	Y [pix]	X [%]	Y [%]	X [°]	Y [°]
1	2,76	0,44	6,9	1,1	0,48	0,08
2	3,07	1,77	8,2	4,7	0,68	0,39
8	7,64	28,90	22,8	63,5	1,33	5,01
19	7,35	9,75	35,0	62,9	2,25	2,99
20	4,87	2,11	24,3	11,4	1,56	0,67
21	2,92	1,99	12,7	9,7	0,80	0,55
26	0,06	6,31	1,2	100,0	0,07	7,88
28	0,10	0,44	0,7	3,1	0,04	0,16
30	0,52	1,00	3,7	7,7	0,17	0,26
31	73,59	103,37	100,0	100,0	9,94	13,97
32	8,01	12,84	72,9	100,0	7,82	12,43
33	1,04	0,39	4,7	1,8	0,32	0,12
40	0,28	0,01	1,5	0,0	0,08	0,00
43	1,37	6,40	7,2	35,5	0,49	2,29

L'utilisation de l'identification de zones permet de réduire l'erreur sur le point de regard dans la majorité des cas. Cependant, l'amélioration n'est pas aussi bonne qu'avec l'érosion ou l'identification de zones sur les images infrarouges.

Une caméra fournissant des images de la qualité de l'image 40 permettrait de répondre aux exigences du cahier des charges pour ce qui est de la précision du point de regard :



L'erreur sur cette image est de $0.08^\circ \times 0^\circ$ avec une résolution estimée à $0.28^\circ \times 0.28^\circ$.

¹ oeil_040.jpg.

8.2.5 Analyse des résultats

L'erreur sur le point de regard est plus importante sur les images traditionnelles que sur les images infrarouges (environ cinq fois) lorsqu'on utilise un algorithme exploitant uniquement le barycentre. Cette grande différence s'explique facilement lorsque l'on compare quelques images infrarouges avec quelques images traditionnelles : le contraste est meilleur sur une image infrarouge, comme nous l'avions déjà vu dans le chapitre 6.3 (Caractéristiques d'une image de l'œil). Par conséquent, l'erreur de type 1 est plus réduite dans les images de type infrarouge.

On peut donc déjà dire, sans examiner les résultats des différents algorithmes, que la position de l'œil se calcule plus facilement avec une image infrarouge qu'avec une image traditionnelle, car il y a moins d'erreur à corriger.

Les différentes conclusions de ce chapitre ont été réalisées sur la base des mesures effectuées sur les ensembles d'images de test du chapitre précédent. Ces conclusions partent de l'hypothèse qu'il est possible d'utiliser une caméra produisant des images similaires à celles utilisées dans les tests et ceci pour une grande durée. Il sera nécessaire de vérifier cette hypothèse sur des séquences vidéo lorsque l'*eye tracker* sera fonctionnel.

Qualité des algorithmes

De manière étonnante, les différents algorithmes testés ne se comportent pas de la même façon suivant le type d'image utilisé (traditionnelle ou infrarouge) :

Algorithme	Efficacité	
	Image infrarouge	Image traditionnelle
Barycentre seul	oui	non
Erosion	non	oui
Identification de zones	oui	non

L'algorithme de type 1 (barycentre seul) donne de meilleurs résultats avec des images infrarouges. C'est principalement du au fait que l'erreur de type 1 est réduite sur ce type d'image.

L'algorithme de type 2 (érosion) donne de meilleurs résultats avec des images traditionnelles. C'est également l'erreur de type 1 qui est en cause. En effet, l'érosion a pour conséquence de diminuer l'erreur de type 1 et d'augmenter l'erreur de type 2. Dans ce cas, comme il y a beaucoup d'erreur de type 1 à corriger, on voit que l'érosion est très efficace. Car l'augmentation de l'erreur de type 2 n'est pas significative. Au contraire, sur des images infrarouges, l'erreur de type 1 est très faible et par conséquent la diminution n'est pas visible.

L'algorithme de type 3 (identification de zones) donne de meilleurs résultats avec les images infrarouges. En effet, l'identification de zones est plus efficace à éliminer l'erreur de type 1 lorsqu'elle est en quantités raisonnables. Malgré tout, les résultats obtenus par l'identification de zones sont très différents en fonction des variations de paramètres. De plus, l'algorithme reste très sensible à l'erreur de type 2 : si elle est trop importante, la pupille est mal voire pas du tout détectée (par exemple, si l'on élimine la moins bonne image IR du test, l'erreur en Y tombe à 1.1°).

Malgré les différences constatées, toutes les combinaisons de types d'images et d'algorithmes permettent d'atteindre une précision inférieure à 1°.

Les meilleures combinaisons d'algorithmes et de type d'images sont donc dans l'ordre décroissant¹ de précision :

Rang	Algorithme	Image	Résolution image	Précision
1	érosion	infrarouge	262 x 197	0.08° x 0.00°
1	identification de zones	traditionnelle	141 x 106	0.08° x 0.00°
3	identification de zones	infrarouge	211 x 158	0.00° x 0.10°
4	barycentre seul	infrarouge	215 x 161	0.11° x 0.00°
5	érosion	traditionnelle	132 x 99	0.15° x 0.00°
6	barycentre seul	traditionnelle	132 x 99	0.36° x 0.17°

Note : malgré les apparences, il ne semble pas y avoir de corrélation entre le type d'image et sa résolution.

Ces résultats permettent de choisir le type d'image qui sera utilisé pour notre *eye tracker* soit des image de type infrarouge. Ce choix est fait pour deux raisons essentielles :

- les images infrarouges sont plus simples à traiter (moins d'erreur de type 1) ce qui augmente la robustesse au bruit des algorithmes;
- l'erreur sur les images infrarouges est plus faibles que sur les images traditionnelles.

De plus, il est fort probable que le simulateur de vision se présente comme une variante de casque virtuel ce qui veut dire que l'œil du patient ne recevrait que la lumière provenant de l'écran LCD qui pourrait être trop faible pour obtenir une image de l'œil d'une qualité suffisante. Cela veut donc dire qu'il serait nécessaire d'éclairer l'œil du patient avec une source lumineuse. Si cette source produit de la lumière dans le spectre visible, il est probable qu'il sera ébloui. L'utilisation d'une source infrarouge permet alors d'éclairer l'œil sans éblouir le sujet, mais il est alors nécessaire d'utiliser une caméra infrarouge pour capter l'image de l'œil illuminé.

8.3 Vitesse de l'algorithme

Les essais en vitesse ont été réalisés avec Java (JDK1.3.1 de Sun) sur un Pentium II 400 MHz et en C sur la carte de développement TMS320 C6711 à 150 MHz.

Seules deux versions de l'algorithme ont été mesurées (barycentre seul et identification de zones). L'algorithme par érosion prenant trop de temps, nous n'avons pas jugé utile d'en tenir compte.

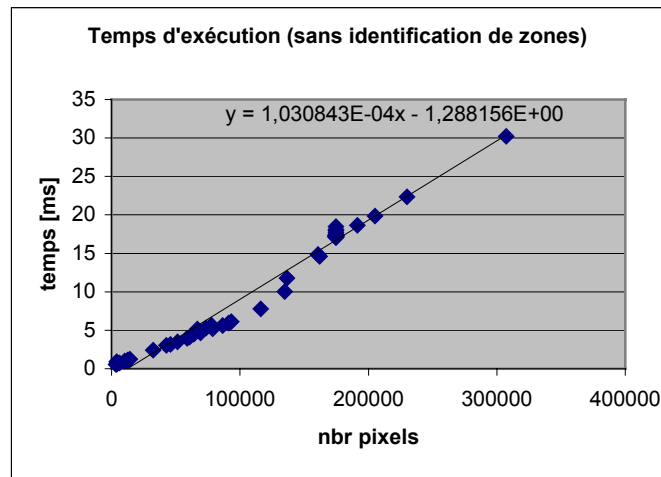
8.3.1 Performance avec Java sur Pentium II

Ce test de performance est mesuré sur le prototype Java. Par conséquent, il n'a pas vraiment d'intérêt, sauf peut-être à titre de comparaison avec le code C sur C6711. Cependant, la comparaison est difficile puisqu'il ne s'agit pas du tout des mêmes conditions d'expérience. En effet, le Pentium II est un processeur à usage général et comprenant un système d'exploitation alors que le C6711 est un processeur à usage particulier sans système d'exploitation.

¹ L'identification de zones est placée avant le barycentre seul pour sa capacité à réduire l'erreur en X. Cependant, elle est placée en seconde position pour sa mauvaise performance en Y.

Deux mesures ont été effectuées : une sans identification de zones et une avec. Chaque point du graphique est un temps moyen sur 300 échantillons (résolution : 3.3 μ s, équivalent à 1 ‰ minimum).

Mesure 1 : sans identification de zones

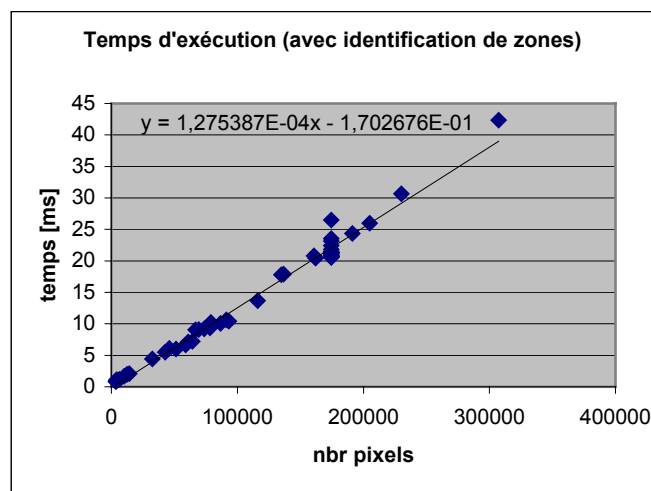


Ce graphique permet de voir le comportement de l'algorithme. L'aspect linéaire montre clairement que le temps de calcul est proportionnel à la taille de l'image (103.1 [ns/pixel]). L'ordonnée à l'origine comprend tous les calculs qui ne sont pas dépendants de la taille de l'image, par exemple tout ce qui touche à l'histogramme (environ 300 μ s). Cela correspond aux fréquences de fonctionnement suivantes (n est la taille du côté de l'image) :

n	t [ms]	f [Hz]
64	0,42	2368
128	1,69	592
256	6,76	148
512	27	37

La forme particulière du graphique (double pente linéaire) est probablement due à la mémoire cache du processeur.

Mesure 2 : avec identification de zones



Avec une identification de zones, la complexité de l'algorithme augmente de 30 % mais reste linéaire, ce qui se traduit par une pente de 127.5 [ns/pixel]. L'ordonnée à l'origine reste identique (aux erreurs de mesure près). Cela correspond aux fréquences de fonctionnement suivantes :

n	t [ms]	f [Hz]
64	0,52	1914
128	2,09	479
256	8,36	120
512	33,4	29,9

8.3.2 Performances avec C sur C6711

Le DSP C6711 dispose de deux blocs de mémoire distincts :

- Mémoire interne;
- Mémoire externe.

La **mémoire interne** est de capacité relativement réduite (64 Koctets) et ne permet par conséquent pas de traiter des images de grande capacité. Les algorithmes implantés ont été optimisés pour utiliser ce type de mémoire.

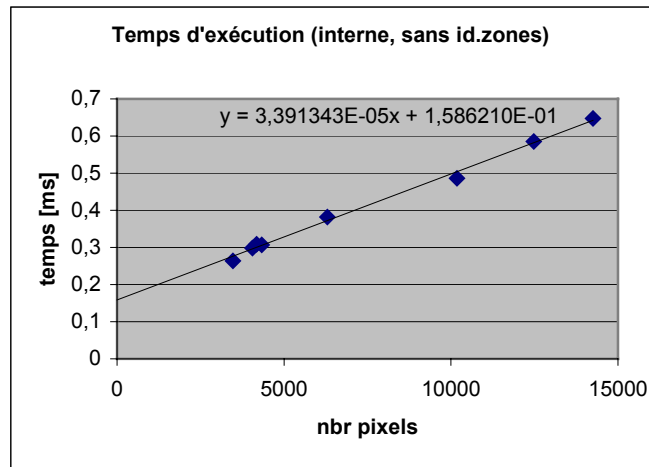
La **mémoire externe** a une capacité bien plus grande que la mémoire interne (16 Moctets) mais elle est plus lente. Les tests que nous verrons par la suite montrent que les algorithmes implantés sont environ 10 fois plus lents lorsqu'ils exploitent la mémoire externe au lieu de la mémoire interne. Il faut toutefois noter que ces tests ont été menés avec des optimisations spécifiques à la mémoire interne et non à la mémoire externe. S'il s'avérait que le traitement doive se faire en mémoire externe, il serait probablement possible d'optimiser un peu l'algorithme, par exemple par l'utilisation de mémoire cache de second niveau ou de DMA.

Un certain nombre de variables doivent cohabiter en mémoire interne et, par conséquent, il n'est pas possible d'utiliser les 64 Ko de mémoire interne pour stocker une image de l'œil. Les différents algorithmes implantés ont montré que la taille maximale d'une image en mémoire interne était d'environ 200 x 100 pixels (20'000 octets). Les études vues précédemment sur la précision de l'algorithme, ont montré qu'il s'agissait d'une résolution suffisante pour obtenir la précision de 1° demandée par le cahier des charges.

Les temps de calcul ont été mesurés pour les deux cas ci-dessus (mémoire interne et externe) et pour les deux versions d'algorithme (avec et sans identification de zones). Tous les temps mesurés en mémoire interne sont des moyennes sur un million d'échantillons (résolution : 1 µs, équivalent à 0.5 % minimum). Les temps mesurés en mémoire externe sont eux calculés sur des moyennes de 100'000 échantillons (résolution de 10 µs, équivalent à 1 % minimum).

Les mesures en interne sont réalisées sur très peu d'images à cause de la limite de taille et le peu d'images de faible résolution à disposition.

Mesure 1 : interne, sans identification de zones

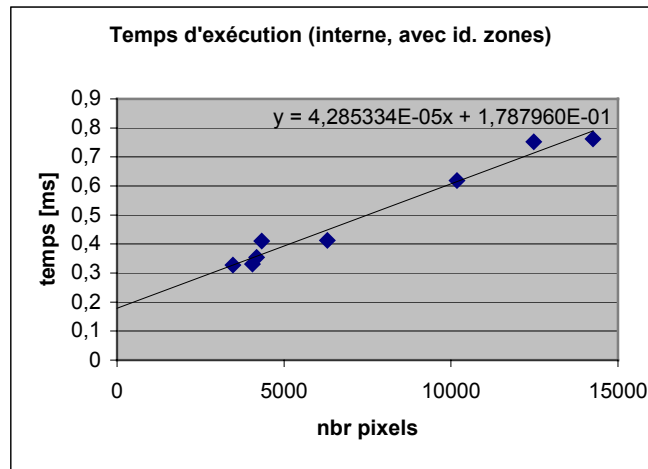


Contrairement à ce qui se passait pour les mesures de temps avec Java, on voit nettement l'ordonnée à l'origine (159 μ s). Comme on pouvait s'y attendre, l'allure du graphique est linéaire, avec une pente de 33.9 [ns/pixel]. Cela correspond aux vitesses de fonctionnement suivantes :

n	t [ms]	f [Hz]
64	0,3	3361
128	0,71	1400
256	2,38	420
512	9,05	111

On peut constater que ces vitesses sont environ 3 fois plus grandes que celles du prototype Java. A fréquence de fonctionnement égale, cela correspond à une amélioration d'un facteur 10 par rapport au code Java sur PII 400.

Mesure 2 : interne, avec identification de zones

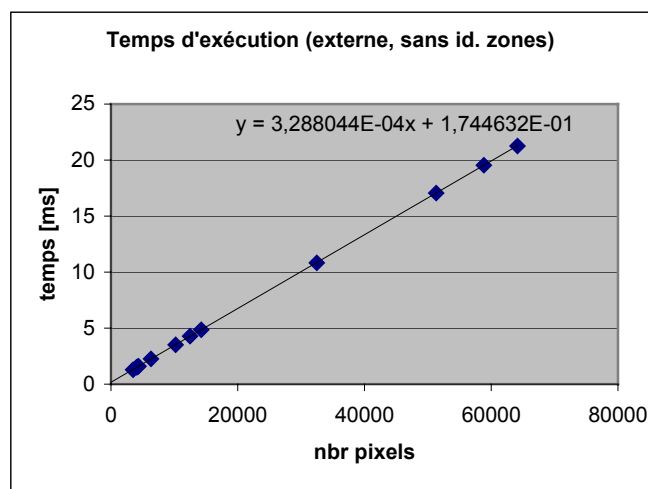


Comme précédemment, on voit bien l'ordonnée à l'origine qui est presque identique (178 μ s). La pente est de 42.9 [ns/pixel], soit environ 30 % de plus que la version sans identification de zones. On peut remarquer que ce surcoût dû à l'identification de zones est identique au surcoût pour le prototype Java. Cela correspond aux vitesses de fonctionnement suivantes :

n	t [ms]	f [Hz]
64	0,35	2822
128	0,88	1135
256	2,99	334,8
512	11,4	87,62

Mesure 3 : externe, sans identification de zones

Comme on l'a déjà dit, l'utilisation de la mémoire externe ralentit le traitement d'un facteur 10 environ :

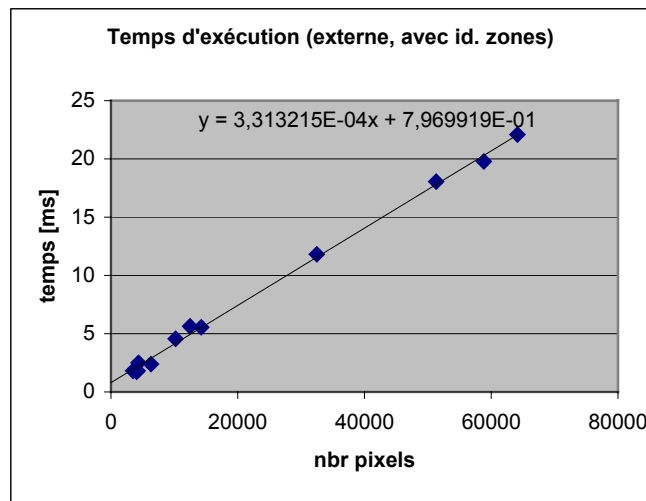


La pente est de 329 [ns/pixel], ce qui correspond bien à un facteur 10 par rapport au même algorithme en mémoire interne. L'ordonnée à l'origine (174 μ s) ne change pas d'un facteur 10

car certaines variables (par exemple l'histogramme) sont en mémoire interne. Cela correspond aux vitesses de fonctionnement suivantes :

n	t [ms]	f [Hz]
64	1,52	657
128	5,56	180
256	21,7	46
512	86,4	11,6

Mesure 4 : externe, avec identification de zones



La pente est de 331 [ns/pixel], ce qui correspond à un temps de calcul de 10% supérieur au même algorithme sans identification de zones. L'ordonnée à l'origine est de 797 μ s, soit un peu plus que précédemment à cause du plus grand nombre de variables indépendantes du nombre de pixels en mémoire externe.

Cela correspond aux vitesses de fonctionnement suivantes :

n	t [ms]	f [Hz]
64	2,15	464
128	6,23	161
256	22,5	44,4
512	87,7	11,4

8.3.3 Analyse des résultats

Les différents algorithmes implantés sur le DSP C6711 (barycentre seul et identification de zones) montrent des vitesses de traitement allant de 33.9 à 331 [ns/pixel]. Pour obtenir la précision de 1° demandée, une résolution de 132 x 99 semble suffire, comme cela été montré lors de l'analyse de la précision de l'algorithme.

Nous pouvons donc dire que la vitesse de traitement de l'image serait de :

Algorithme	Type de mémoire	Vitesse [Hz]
Barycentre seul	Interne	2'256
Barycentre seul	Externe	232
Identification de zones	Interne	1'785
Identification de zones	Externe	230

Ces chiffres permettent de choisir quel type d'algorithme sera utilisé pour notre *eye tracker*. L'image 132 x 99 étant suffisamment petite pour être stockée en mémoire interne, c'est ce type de mémoire qui sera utilisé. Pour l'algorithme proprement dit, c'est l'identification de zones qui sera utilisée. Les raisons sont les suivantes :

- l'identification de zones donne des résultats plus précis que le barycentre seul;
- la différence de vitesse de traitement entre le barycentre seul et l'identification de zones n'est pas significative;
- la vitesse de traitement de l'identification de zones est largement plus élevée que celle qui avait été demandée dans le cahier des charges.

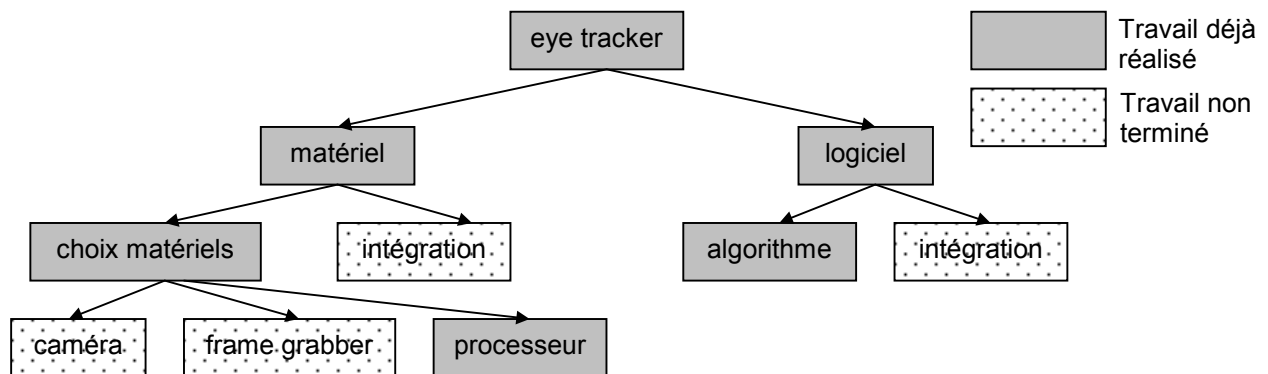
L'utilisation de l'érosion n'est pas envisageable pour des raisons de temps de calcul. En effet, l'élément structurant (masque) utilisé a une taille d'environ 60 x 60 pixels, ce qui augmenterait la complexité de l'algorithme d'un facteur 3600. Cela veut dire que la vitesse de traitement de l'algorithme descendrait à moins d'une image par seconde (identification de zone en mémoire interne) soit une vitesse trop faible par rapport aux besoins spécifiés dans le cahier des charges.

Chapitre 9: Conclusion

9.1 Travail réalisé

Des études sur les différents *eye trackers* a été réalisée, en parallèle avec une recherche sur les différents aspects biologiques de l'œil humain. Une méthode de *tracking* a ensuite été choisie et les images de l'œil ont été étudiées. Il s'en est suivi le développement de plusieurs versions d'algorithmes de sorte que les exigences en vitesse et en précision soient atteintes.

Le travail réalisé peut se schématiser de la manière suivante (reprise de l'arbre donné dans la description du travail à réaliser) :



Le travail qui n'a pas encore été terminé est :

- le choix de la caméra et du frame grabber;
- l'intégration matérielle et logicielle avec le simulateur de vision.

L'intégration n'est actuellement pas réalisable car la structure du simulateur de vision n'a pas encore été définie.

9.2 Améliorations potentielles

D'après ce qui a été vu dans les paragraphes précédents, les exigences du cahier des charges ont été remplies. Cependant, voici quelques idées qui permettraient d'améliorer l'efficacité de notre *eye tracker*.

9.2.1 Identification de zones

La première chose qui nous vient à l'esprit est de modifier l'algorithme d'identification de zones pour le rendre plus robuste et plus précis. Cette idée est envisageable car la fréquence de fonctionnement actuelle est beaucoup plus élevée que ce que ne l'exigeait le cahier des charges. Il est donc possible d'améliorer la précision et la robustesse de l'algorithme en utilisant du temps de calcul supplémentaire.

Les améliorations à apporter à cet algorithme sont multiples. Tout d'abord, il serait intéressant que notre classification de zones tienne compte de la forme de la zone et non uniquement de sa densité. L'idée est de prendre en compte plusieurs paramètres et que chaque zone se voie attribuer un score en fonction de la qualité des ces paramètres. La zone qui aurait le plus haut score serait alors considéré comme pupille. L'un de ces paramètres serait bien entendu la densité. Il faut alors trouver d'autres paramètres. On pourrait par exemple utiliser le barycentre de chaque zone : le barycentre idéal, celui d'une ellipse, est au centre de la zone. Cela permettrait de prendre en compte la forme de chaque zone.

Une deuxième amélioration serait, après l'identification de zones, de calculer le barycentre non avec une technique de clipping mais en faisant un nouveau parcours de zone dans l'image seuillée. De cette manière là, on réduirait encore l'erreur de type 1 tout en ne faisant pas augmenter l'erreur de type 2.

La difficulté de cette nouvelle approche est de trouver un point de la zone correspondant à la pupille dans l'image seuillée et ceci à partir d'une image sous-échantillonnée et donc de faible qualité. En effet, les pixels de l'image seuillée correspondants à un pixel sous-échantillonné peuvent appartenir à plusieurs zones et il faut alors déterminer quelle est la zone qui correspond effectivement à la pupille.

9.2.2 Vitesse de l'algorithme

D'après les résultats qui ont été obtenus, les différents algorithmes implantés en mémoire externe consomment dix fois plus de temps de calcul que s'ils sont implantés en mémoire interne. Cependant, ces algorithmes n'ont pas été optimisés spécifiquement pour l'utilisation de la mémoire externe.

L'idée est donc de procéder à ce genre d'optimisation, ce qui permettrait de diminuer la différence de temps de calcul entre mémoire interne et externe. De cette manière, il serait possible d'exploiter des images de dimensions supérieures à celles qui sont actuellement traitées sans trop sacrifier la vitesse de traitement. Il en résulterait une augmentation de la résolution et donc une augmentation de la précision.

9.3 Considérations finales

Les différents tests qui ont été menés ont permis de déterminer plusieurs paramètres important de notre *eye tracker* :

- résolution de la caméra : environ 132 x 99 pixels;
- vitesse de traitement : 1'785 images par seconde;
- algorithme : identification de zones;
- type d'image traitée : image infrarouge.

D'après ce qui a été vu, la caméra devrait avoir les caractéristiques suivantes :

- type : infrarouge;
- résolution minimale : 132 x 99;
- vitesse : < 1'785 Hz.

Avec de telles caractéristiques, notre *eye tracker* atteindrait une précision inférieure à 1°. Le temps de latence serait 1/1785 seconde soit 0.56 ms. Les résultats obtenus peuvent être résumés ainsi :

Caractéristique	Demandé ¹	Obtenu	Résultat
Précision	1°	0.1°	✓
Vitesse de fonctionnement	250 Hz	1'785 Hz	✓
Temps de latence	10 ms	0.56 ms	✓

Les résultats obtenus sont donc bien en delà de ce qui avait demandé dans le cahier des charges. Ce travail peut donc être considéré comme un succès même si la caméra n'a pas encore été choisie et si quelques améliorations pourraient être réalisées.

¹ Selon le cahier des charges.

Annexe A :

Introduction au traitement numérique d'images

Cette annexe est destinée aux personnes qui ne possèdent pas ou peu d'expérience dans le domaine de traitement numérique des images. Il est censé donner un petit aperçu des techniques utilisées dans ce travail.

Le terme traitement numérique d'image ou imagerie numérique consiste à appliquer un traitement à une image représentée sous forme numérique, c'est-à-dire sous la forme d'un tableau rectangulaire. Cette représentation correspond à l'échantillonnage d'une image analogique.

Chaque case du tableau représente un élément d'image ou pixel. Chaque pixel possède une couleur codée par un nombre. Le nombre de couleurs possible pour un pixel varie en fonction du codage choisi pour l'image. Cela correspond à la quantification. Les quantifications les plus répandues sont :

- 256 niveaux de gris (8 bits)
- 256 couleurs (8 bits)
- 65536 couleurs (16 bits)
- 16 Mio de couleurs (24 bits)

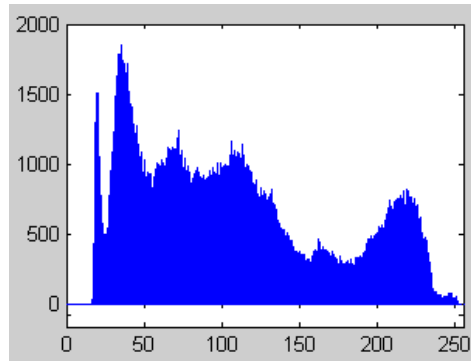
Tous les traitements décrits dans ce chapitre exploitent des images codées en 256 niveaux de gris. Il est cependant possible de les utiliser pour n'importe quel type d'image.

Lorsqu'une image est codée en 256 niveaux de gris, le nombre codant la couleur est compris entre 0 et 255. La valeur 0 signifie noir tandis que la valeur 255 signifie blanc. Toutes les couleurs intermédiaires sont des gris d'intensité proportionnelle à la valeur du pixel. Voici un exemple d'image en 256 niveaux de gris (565x309 pixels) :



A.1 Histogramme et segmentation

Le calcul de l'histogramme d'une image consiste à compter le nombre de pixels de chaque couleur et à stocker ce nombre dans un tableau qui contient autant de cases que le nombre maximal de couleurs de l'image (dans ce cas 256). Voici l'histogramme correspondant à l'image précédente :



L'histogramme permet de représenter l'information contenue dans l'image de manière compressée (on passe d'une image de 174'585 pixels (565x309) à un histogramme de 256 valeurs). L'information contenue dans l'histogramme permet dans certains cas d'extraire le contenu de l'image, typiquement d'extraire des zones de niveau de gris. En effet, les bosses (ou modes) de l'histogramme représentent une variation continue de couleur. Les pixels de l'image contenus dans ce mode sont dans la majorité des cas dans la même portion de l'image (pixels proches spatialement). L'extraction de contenu à partir de l'histogramme s'appelle **segmentation d'image par histogramme**. Elle consiste à créer une nouvelle image binaire (deux couleurs : noir et blanc) dont tous les pixels blancs sont ceux de l'image qui ont une couleur comprise entre deux valeurs fixées.

Par exemple, voici ce qui est obtenu par segmentation d'image obtenues à partir de l'histogramme précédent :



couleurs 12..24



couleurs 25..34



couleurs 35..53



couleurs 54..89



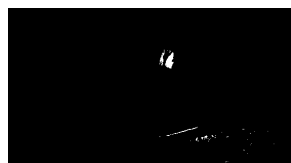
couleurs 90..153



couleurs 154..180



couleurs 181..242



couleurs 243..255

A.2 Convolution

La convolution est une opération qui consiste à faire passer un masque sur l'image et à appliquer un certain traitement :



Le pixel central du masque (X) se déplace sur l'image et l'opération de convolution implique chaque pixel de l'image qui est sous le masque. Chaque position du masque génère un pixel d'une nouvelle image et nécessite l'utilisation de tous les pixels du masque. Par conséquent, la convolution est une opération relativement complexe au niveau du temps de calcul : $O(M \cdot N)$ avec M le nombre de pixels du masque et N le nombre de pixels de l'image.

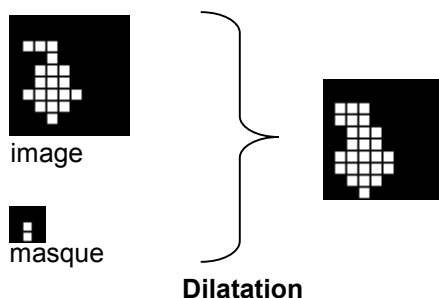
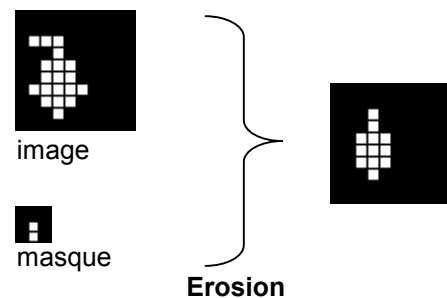
Une opération de convolution classique consiste à effectuer une somme de produits : pixel central du masque prend comme valeur la somme des produits de tous les pixels du masque et de ceux de l'image correspondants. Cela permet d'effectuer des filtres de fréquences.

Bien entendu, la convolution peut être utilisée à bien d'autres choses comme par exemple l'érosion ou la dilatation.

A.3 Erosion et dilatation

L'érosion et la dilatation sont deux opérations de convolutions complémentaires. Il s'agit d'opérations basées sur la morphologie mathématique. Ces deux opérations sont réalisées sur des images binaires (c'est-à-dire en 2 couleurs).

L'érosion consiste à "ronger" l'image comme si on la trempait dans l'acide. Numériquement, l'opération consiste à dire que si tous les pixels à '1' du masque sont également à '1' dans l'image, le nouveau pixel généré est mis à '1'. Dans le cas contraire, le nouveau pixel est mis à '0'. L'érosion est donc une opération qui diminue le nombre de pixels à '1' dans l'image.



La dilatation consiste à "faire croître" l'image comme si on la gonflait comme une chambre à air. Numériquement, l'opération consiste à dire que si le pixel de l'image sous le pixel central du masque est à '1', tous les pixels à '1' du masque sont copiés dans la nouvelle image. La dilatation est donc une opération qui augmente le nombre de pixels de l'image.

Il est possible d'utiliser l'érosion et la dilatation successivement pour obtenir un effet particulier. Cet effet s'appelle ouverture mathématique si l'érosion est appliquée avant la dilatation et fermeture mathématique dans le cas contraire. L'ouverture mathématique permet de lisser les contours des objets contenus dans l'image de départ. La fermeture mathématique permet de supprimer les trous (petites zones à '0' contenues dans des zones à '1'). Dans les deux cas, la quantité de pixels à '1' de l'image originale est quasiment conservée.

A.4 Filtre passe-bas et filtre médian

Le filtre passe-bas est une méthode qui permet de supprimer les composantes hautes fréquences d'une image, ce qui a pour effet de rendre les transitions de couleurs plus douces. Il s'agit d'une convolution avec somme de produits, chaque coefficient du masque valant l'inverse du nombre de pixels du masque.

Le filtre médian a pour effet de supprimer le bruit impulsionnel ou *salt and pepper*. Il s'agit également d'une convolution. Pour avoir une bonne idée de ce qu'est du bruit impulsionnel, il suffit de saupoudrer une photo avec un peu de sel. Pour réaliser un tel filtre, il faut trier toutes les valeurs des pixels qui sont sous le masque dans l'ordre croissant, puis choisir la valeur du milieu. Ainsi, les valeurs très grandes ou très petites sont éliminées.

Annexe B : Références

Les différentes références et ouvrages utilisés durant ce travail de diplôme ont été consignées dans une base de donnée gérée par le web.

L'application, qui a été conçue au début du travail de diplôme, consiste à offrir la possibilité à des utilisateurs enregistrés de stocker des références de documents (URL, éditeur, fichier, etc.) sur un serveur web centralisé. Il est ensuite possible de faire des recherches par mot clé sur les références indexées. De cette manière, il est possible d'ajouter et de consulter des références avec facilité et rapidité, ce qui rend bien plus agréable la création de bibliographie.

Les références de ce travail de diplôme sont donc consultable en ligne à l'adresse suivante :

`http://129.194.186.123/servlet_doc_db/index.html`

en donnant comme critères de recherche :

`domaine:vision num_utilisateur:1`

Note: les références du travail de diplôme ont un score de 2.

Voici la liste¹ de ces références (32 documents) dans un format plus approprié au papier :

- [1]. Arne John Glenstrup (panic@diku.dk), Theo Engell-Nielsen (beyond@diku.dk), *Eye Controlled Media: Present and Future State*, University of Copenhagen, 1995-06-01 (LSN_doc=23) Today, the human eye-gaze can be recorded by relatively unobtrusive techniques. This thesis argues that it is possible to use the eye-gaze of a computer user in the interface to aid the control of the application. Care must be taken, though, that eye-gaze tracking data is used in a sensible way, since the nature of human eye-movements is a combination of several voluntary and involuntary cognitive processes. The main reason for eye-gaze based user interfaces being attractive is that the direction of the eye-gaze can express the interests of the user-it is a potential porthole into the current cognitive processes-and communication through the direction of the eyes is faster than any other mode of human communication. It is argued that eye-gaze tracking data is best used in multimodal interfaces where the user interacts with the data instead of the interface, in so-called non-command user interfaces. Furthermore, five usability criteria for eye-gaze media are given. This thesis also suggests research into a new, interactive film medium: interest and emotion sensitive media (IES). IES exploits the viewer's eye-gaze and other affection measures to determine how script paths of an IES film are traversed. IES will be very difficult to implement today, as research is needed to investigate temporal problems in script construction, how multiple persons can use IES, how activation areas are constructed and how the producer of an IES film is best assisted in writing the script. This thesis also contains reviews of current technical possibilities, psychological aspects of eye-gaze tracking and current eye-gaze based systems.
- [2]. John W. Senders, Dennis F. Fisher, Richard A. Monty, *Eye Movements and psychological Functions*, Lawrence Erlbaum Associates, 1978-01-01 (LSN_doc=63) Description des mouvements de l'oeil, en particulier pour les saccades. Analyse fonctionnelle des saccades p.89. Suppression saccadique

¹ Générée avec le critère de recherche : `domaine:vision num_utilisateur:1 mode:ref`

- [3]. Rudolf Groner, Christine Menz, Dennis F. Fisher, Richard A. Monty, *Eye Movements and psychological Functions: International views*, Lawrence Erlbaum Associates, 159.95 EYE (biblio fapse), 1983-01-01
(LSN_doc=62) Description des mouvements de l'oeil, en particulier pour les saccades.
- [4]. Rainer Stiefelhaven (stiefel@ira.uka.de), Jie Yang (yang+@cs.cmu.edu), Alex Waibel (ahw@cs.cmu.edu), *Tracking Eyes and Monitoring Eye Gaze*, Interactive Systems Laboratories (University of Karlsruhe | Germany, Carnegie Mellon University, USA), Workshop on Perceptual User Interfaces, Banff, Canada, 1999-07-13
(LSN_doc=61) In this paper, we present a non-intrusive eye tracker which can detect and track a user's eyes in real time as soon as the face appears in the view of the camera without special lighting or any marks on the user's face. We also discuss the problem of gaze monitoring. We employ neural networks to estimate a user's eye gaze using the images of user's both eyes as input to the neural networks. We have collected 4 sets of data from 4 different users using the eye tracker and have trained and tested several neural networks. The eye gaze monitoring system has achieved accuracy of between 1.3 and 1.8 degrees with a user dependent setup and of 1.9 degrees for a multi-user setup.
- [5]. C.H. Morimoto, D. Koons, A. Amir, M. Flickner ({dkoons,arnon,flick}@almaden.ibm.com), *Frame-Rate Pupil Detector and Gaze Tracker*, Dept. Ciência da Computação IME/USP (São Paulo, Brazil) and IBM Almaden Research Center, course SIMG-503 Senior Research, 1999-07-13
(LSN_doc=60) We present a robust, frame-rate pupil detector technique, based on an active illumination scheme, used for gaze estimation. The pupil detector uses two light sources synchronized with the even and odd fields of the video signal (interlaced frames), to create bright and dark pupil images. The retro-reflectivity property of the eye is exploited by placing an infra-red (IR) light source close to the camera's optical axis resulting in an image with a bright pupil. A similar off axis IR source generates an image with dark pupils. Pupils are detected from the thresholded difference of the bright and dark pupil images. After a calibration procedure, the vector computed from the pupil center to the center of the corneal glints generated from light sources is used to estimate the gaze position. The frame-rate gaze estimator prototype is currently being demonstrated in a docked 300 MHz IBM Thinkpad with a PCI frame grabber, using interlaced frames of resolution 640x480x8 bits
- [6]. Jeffrey M. Cunningham, *Task Dependency of Eye Fixations & Development of a Portable Eye Tracker*, Center for Imaging Science at the Rochester Institute of Technology, course SIMG-503 Senior Research, 1998-01-01
(LSN_doc=59) This senior research project is easily divisible into two parts. The first part was modeled after work Alfred Yarbus completed over thirty years ago. Yarbus contributed much to our knowledge of eye movements, and is widely quoted. One of Yarbus' results was a dependency between eye movements and the task at hand. In this study his work was replicated using a modern video eye tracker, and task was shown to have a statistically significant effect on eye fixations. The second goal was the development of a comfortable, lightweight video eyetracker. The goal was to have an eyetracker that performs as well as the eyetracker currently in use (ASL's 501 model), have two hours battery time, and that was stable enough to be used on someone walking. New optics mounted on a baseball cap were used with the existing control box, and this system successfully tracked an eye. Battery power was sufficient for more than two hours of running time, but more work needs to be done.
- [7]. mUltimo3D Team (Heinrich-Hertz-Institut für Nachrichtentechnik), *Video Gaze Tracker*, Heinrich-Hertz-Institut für Nachrichtentechnik, 2000-10-17
(LSN_doc=58) We have developed a novel algorithm for determining the gaze line with active compensation of head movements. The measurement of the gaze line is derived from the measurements of both the gaze direction (the unit vector of the gaze line) and the eye location (determined by the head tracker). For the purpose of determining the gaze direction, we opted for the cornea reflex method [23] because of its high precision and stability and because it is non-intrusive (the measurements can be taken from a distance). The eye is illuminated with low-intensity infrared light. As a result, the pupil appears as a black elliptical region in the camera image. The center of the pupil and the reflection of the light from the cornea are determined by image processing. There is a monotonic relationship between the vector pointing from the center of the pupil to the light reflection (eye vector) and the user's gaze direction. After individual calibration, the gaze direction can be precisely derived from the eye vector. Hough transform

- [8]. Thierry Pun (thierry.pun@cui.unige.ch), *Cours d'imagerie numérique*, CUI, Université de Genève, 1996-01-01
(LSN_doc=57) Cours d'imagerie numérique de Thierry Pun (prof de diplôme) suivi en 1998-1999. Explique les notions de synthèse d'image et d'analyse d'image. Excellente description, très précis.
- [9]. Tom Henderson (THenderson@glenbrook.k12.il.us), *Reflection and the Ray model of light*, Glenbrook South High School, 1998-01-01
(LSN_doc=56) Description des rayons lumineux (réflexion) sur les miroirs planes, convexe et concave. Très didactique, avec des illustrations.
- [10]. Auteur inconnu, *Determination of Minimum Requirements for Useful Artificial Vision*, Projet Vision, 2000-01-01
(LSN_doc=55) Présentation du cahier des charges pour le projet vision. Ce document a été soumis au FNRS pour le financement. Décrit "qui fait quoi" et présente le travail à réaliser.
- [11]. McLaughlin, Robert A. , *Randomized Hough Transform: Improved Ellipse Detection with Comparison*, Centre for Intelligent Information Processing Systems, Department of Electrical and Electronic Engineering, The University of Western Australia, TR97-01, 1997-07-08
(LSN_doc=54) We describe an algorithm for the detection of ellipse shapes in images, using the Randomized Hough Transform. The algorithm is compared to a standard implementation of the Hough Transform, and the Probabilistic Hough Transform. Tests are performed using both noise-free and noisy images, and several real-world images. The algorithm was found to give improvements in accuracy, and a reduction in computation time, memory requirements and the number of false alarms detected. with source code.
- [12]. Blue-eyes Team (blue-eyes@notes.research.ibm.com), *Pupil finder*, IBM Almaden, Blue-eyes project/Pupil finder, 2000-01-01
(LSN_doc=53) we discovered a fast, robust, and low cost pupil detection technique that uses two infra red (IR) time multiplexed light sources, composed of two rings of 8 LED's each, synchronized with the camera frame rate. One light source is placed very close to the camera's optical axis, and the second source is placed off-axis. The pupil appears bright in the camera image during on-axis illumination (similar to the red eye effect from flash photography), and dark when illumination is off-axis. The off-axis light source is calibrated to provide roughly equivalent whole-scene illumination. Pupil detection follows from thresholding the difference of the dark from the bright pupil images. To reduce artifacts caused mostly by head motion, a larger temporal support is used. This method can be applied to detect and track several pupils (or several people). Experimental results from a real-time implementation of the system show that this technique is very robust, and able to detect pupils using wide field of view low cost cameras under different illumination conditions, even for people with glasses, and up to 5m from the camera.
- [13]. Alejandro Backer (alex@caltech.edu), Eva Peral (eva@cco.caltech.edu), *Remote video eye tracker*, California Institute of Technology, Technical report, CALTECH, CNS 248, 1997-06-09
(LSN_doc=52) An eye tracker is a device capable of determining the direction of gaze of a human observer. Current eye tracking technology is cumbersome to the user, prohibitively expensive for home or business use, requires extensive training and/or does not allow head movements. We present a system that only requires a video camera and computer software; unlike all other video-based systems we are aware of, it does not require infra-red or other light-emitting devices. Our algorithm first locates a few key features of the user's face (pupils and eye corners) using orientation correlation, and then feeds that location data to a 2-layer neural network with just a few (10) hidden units which, after training it with only 85 images, is able to locate the point on a screen that the user is looking at. A preliminary version of our system performs at present with under .15 units mean error, where units are 3 cm apart from each other. The user's head is free to move as long as both eyes remain in focus and in view of the camera. Hardware: CCD Black and white camera, no zoom, Frame grabber, Pentium Software: C, C++, Windows programming.
- [14]. Noel Annette Zabaronick, *EYE TRACKING USING FIBER OPTICS AND COMPUTER VISION TECHNIQUES*, Faculty of the Virginia Tech, Master thesis, 1997-04-04
(LSN_doc=51) We describe a system for tracking the motion and determining the velocity of the human eye using a combination of fiber optics and imaging optics integrated with a CCD camera, a framegrabber, and a PC. Systems such as the one we describe are needed for monitoring location of interest for better human- computer interfaces and virtual reality simulations, for monitoring pilot cognitive skills during flight simulations, and for detecting the early

warning signs of driver drowsiness. Where some eye-tracking devices are cumbersome because they require a person to limit head motion to remain in view of a camera or uncomfortable because they require the use of an elaborate headpiece, our proposed system is as simple to wear as a pair of eyeglasses, with the only added weight to the glasses being that of a fiber optic bundle. This fiber bundle guides the image of an 880 nm LED illuminated eye as it is reflected from a hot mirror built into the eyeglass assembly. The guided image is focused onto a CCD and the camera's output is linked to a PC framegrabber card. Software written in Microsoft Visual C++ locates the user's pupil in each video image and tracks the motion and velocity of the pupil over time. The prototype system tracks the center of the pupil to an accuracy of only 1/5 of the pupil size. This diminished accuracy results from the minimized computation scheme of pupil detection. The technique relies on edge detection and direction of image gradient over very few pixels so that the system is optimized for speed. Many other methods of pupil detection could be investigated in future research in order to optimize such a system for other parameters.

- [15]. Eileen Kowler, *Eye movement and their role in visual and cognitive processes*, Elsevier Science Publishers, 159.93 EYE, 1990-01-01
(LSN_doc=45) description des mouvements de l'oeil sous une vue mathématique (avec modèles math.)
- [16]. Hugh Davson, *The eye*, Accademic Press, 159.93 EYE, 1969-01-01
(LSN_doc=44) description générale de l'oeil, la plupart du temps en termes biologiques
- [17]. Richard A. Monty, John W. Sanders, *Eye movements and psychological processes*, Lawrence Erlbaum Associates Publishers, 159.93 EYE, 1976-01-01
(LSN_doc=43) description précise des mouvements de l'oeil. Suppression saccadique. Mouvements de l'oeil pendant la lecture. Graphiques très explicites et de qualité
- [18]. Ethel Martin, *Saccadic suppression : a review and an analysis*, Columbia University and the American Psychological Association, Psychological Bulletin, Vol. 81 No. 12, p.899-917, 1974-12-01
(LSN_doc=42) Résumé de tout ce qui a été écrit sur la suppression saccadique jusqu'en 1974.
- [19]. Fergus W. Campbell, Robert H. Wurtz, *Saccadic omission : why we do not see a grey-out during a saccadic eye movement*, Pergamon Press and the University of Cambridge, Vision Research, Vol. 18, p.1297-1303, 1978-01-04
(LSN_doc=41) Description temporelle de la suppression saccadique.
- [20]. T. Vilis, *Eye movement description*, University of Western Ontario, v1.8, 1998-01-01
(LSN_doc=40) Description des mouvements des yeux, avec animation Flash et document PDF. Pas formidablement bien détaillé mais c'est une bonne introduction.
- [21]. J. Kevin O'Regan, *SOLVING THE "REAL" MYSTERIES OF VISUAL PERCEPTION*, Université René Descartes, Canadin Journal of Psychology, 46:3, p.461-488, 1992-01-01
(LSN_doc=39) Visual science is currently a highly active domain, with much progress being made in fields such as color vision, stereo vision, perception of brightness and contrast, visual illusions, etc. But the "real" mystery of visual perception remains comparatively unfathomed, or at least relegated to philosophical status: why it is that we can see so well with what is apparently such a badly constructed visual apparatus? In this paper I will discuss several defects of vision and the classical theories of how they are overcome. I will criticize these theories and suggest an alternative approach, in which the outside world is considered as a kind of external memory store which can be accessed instantaneously by casting one's eyes (or one's attention) to some location. The feeling of the presence and extreme richness of the visual world is, under this view, a kind of illusion, created by the immediate availability of the information in this external store.
- [22]. Timothy K. Horiuchi & Christof Koch (timmer@klab.caltech.edu), *Analog VLSI Circuits for Visual Motion-Based Adaptation of Post-Saccadic Drift*, Computation and Neural Systems Program (California Institute of Technology, Pasadena, CA), 1996-01-22
(LSN_doc=35) Using the analog VLSI-based saccadic eye movement system previously developed in our laboratory, we are investigating the use of biologically realistic error signals to calibrate the system in a manner similar to the primate oculomotor system. In this paper we introduce two new circuit components which are used to perform this

task, a resettable-integrator model of the burst generator with a floating-gate structure to provide on-chip storage of analog parameters and a directionally-selective motion detector for detecting post-saccadic drift.

- [23]. Rajesh P. N. Rao (rao@cs.rochester.edu), *Eye movement ressources*, MIT Press, Advances in neural information processing systems (NIPS95), NIPS 95, 1996-01-01
(LSN_doc=33) Ressources, articles, ... sur les mouvements des yeux.
- [24]. Purves, Augustine, Fitzpatrick, Katz, Lamanta, McNamara, *Neurosciences*, DeBoeck University, Neurosciences (livre), p.179-198, 1999-01-01
(LSN_doc=31) Chapitre 10 de l'ouvrage "Neurosciences" décrivant l'oeil.
- [25]. William H. Dobelle, Donald O. Quest, Joao L. Antunes, Theodore S. Roberts, John P. Girvin, *Artificial vision for the blind by electrical stimulation of the visual cortex*, Congress of Neurological Surgeons, Neurosurgery, Vol. 5 No. 4, 1979-01-01
(LSN_doc=30) Electrodes greffées dans le cortex visuel par le docteur Dobelle. Description des impulsions utilisées pour les électrodes
- [26]. Carter Compton Collins, *Tactile image perception*, Smith-Kettlewell Institute and Departement of Visual Sciences, University of the Pacific, San Francisco, California, 1973-01-01
(LSN_doc=29) Electrodes posées sur la peau pour palier à la déficience de la vision. Avec dimensions des electrodes.
- [27]. Mike Kay, *The electric eye*, Popular Science, p.60-76, 1993-08-01
(LSN_doc=28) Electrodes posées sur le fond de la rétine. Décrit le déroulement de l'opération, combien de pixels sont nécessaires pour lire et pour voir.
- [28]. T. Gilmont (gilmont@dice.ucl.ac.be), X. Verians, J.-D. Legat, Cl. Veraart (veraart@fsio.ucl.ac.be), *Resolution reduction by growth of zones for visual prostheses*, Université Catholique de Louvain, Bruxelles, Belgique, International conference on image processing, Lausanne, Suisse, p.299-302, vol. II, 1996-09-16
(LSN_doc=27) algorithm for visual prostheses devices, where data bandwidth must be dramatically reduced in order to correspond to the biotechnological constraints. The resolution reduction is based on an image segmentation by growth of zones wich give good results.
- [29]. Robert W. Massof, Douglas L. Rickman, *Obstacles encountered in the development of the low vision enhancement system*, American Academy of Optometry, Optometry and vision Science, vol. 69, No. 1, pp. 32-41, 1992-01-01
(LSN_doc=26) High technology low vision aid for blind and visually impaired. Also know as LVES (Low Vision Enhancement System). Collaboration of NASA and John Hopkins Wilmer Eye Institute.
- [30]. Stéphane Marchand-Maillet, Yazid M. Sharaiha, *Binary Digital Image Processing*, Academic Press, 2000-01-01
(LSN_doc=74) Traitement des images discrètes, en particulier tout ce qui traite de la morphologie mathématique (érosion, dilatation, squelette, ...). Présente la méthode utilisée pour caractériser les zones utilisée dans ce travail (je l'ai vu après coup, mais étant donné que Stéphane est l'expert pour mon travail, je ne pouvais pas l'omettre ;)).
- [31]. Jacques François (jfrancois@eig.unige.ch), *Enoncé de diplôme - Simulateur de vision*, EIG-LSN, Genève, Suisse, 2000-05-15
(LSN_doc=25) Enoncé de diplôme de Julien Kronegg (CUI, UniGE).
- [32]. WM. H. Dobelle, *Artificial vision for the blind by connecting a television camera to the visual cortex*, American Society of Artificial Organs, ASAIO Journal 2000, 1999-12-01
(LSN_doc=24) Electrodes implantées dans le cortex visuel avec signaux provenant d'une caméra externe. Encore une oeuvre du docteur Dobelle.
- [33]. Jean-Benoît Page, *Système de contrôle pour une prothese mecanique*, EIG, LSN, 1997.