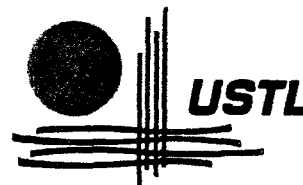


T
N° d'ordre : 1651



THESE
Présentée à
**L'UNIVERSITÉ DES SCIENCES
ET TECHNOLOGIES DE LILLE**

en vue de l'obtention du grade de

Docteur en Electronique

par

Pierre CHRISTIAEN

**Contribution aux recherches en Adéquation Algorithme
Architecture dans le cadre de la conception d'un suiveur
de regard « tête libre » par caméra**

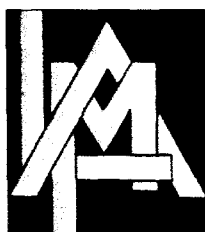
Soutenue le 22 Janvier 1996

devant la commission d'examen



Membres du Jury :

J.C. CARRU	Président	IEMN	UL Calais
J.L. FRANCESCHI	Rapporteur	LGET	Toulouse 3
M. CASTAGNE	Rapporteur	CEM	Montpellier 2
J.C. ANGUE	Examineur	LAMIH	Valenciennes
J.P. DUBUS	Examineur	LMA	USTL Lille 1
Y. SOREL	Examineur	INRIA	Rocquencourt
P. LOOSFELT	Directeur de Thèse	LMA	USTL Lille 1



LABORATOIRE DE MESURES AUTOMATIQUES

U.F.R d'INFORMATIQUE, ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE
UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE
Bâtiment P3 - 59655 Villeneuve d'Ascq CEDEX

Remerciements

Les travaux exposés dans ce mémoire sont le fruit de trois années de recherche au sein du Laboratoire de Mesures Automatiques (LMA) de l'Université des Sciences et Technologies de Lille, dirigé par Monsieur le Professeur Jean Paul DUBUS.

Je remercie chaleureusement Monsieur le Professeur Philippe LOOSFELT qui m'a accueilli dans son équipe et qui est l'initiateur de cette étude. Il a assuré la direction tout au long de ces années partagées, en conservant une disponibilité de chaque instant. Je lui suis particulièrement reconnaissant de ses nombreux conseils, la qualité de son suivi et de la confiance qu'il a eue dans le choix de mes axes de recherche. Je tiens à l'assurer ici de ma plus sincère reconnaissance.

Je remercie Monsieur Jean Claude CARRU, Professeur à l'Université du Littoral, de me faire l'honneur de présider le jury.

J'exprime ma gratitude à Messieurs les Professeurs Jean Luc FRANCESCHI, responsable d'une équipe de recherche au Laboratoire de Génie Électrique de Toulouse (LGET) et Michel CASTAGNE du Centre d'Électronique de Montpellier (CEM), pour l'honneur qu'ils m'ont accordé en acceptant d'être les rapporteurs de ce mémoire. Leurs remarques et critiques ont été précieuses. La disponibilité et la patience dont ils ont fait preuve m'ont touché. Qu'ils veuillent bien accepter l'expression de mes respectueux remerciements.

Je présente mes plus vifs remerciements à Monsieur le Professeur Jean Paul DUBUS qui a accepté d'examiner mon travail. Lors de mes études, j'avais apprécié la qualité de son enseignement. Au cours de mes années passées dans son laboratoire, j'ai eu la confirmation de son attachement à mener des recherches innovantes tout en conservant des règles de travaux rigoureuses. Son suivi et ses encouragements ont joué un rôle essentiel dans la bonne conduite de mon travail de recherche.

Monsieur Jean Claude ANGUE, directeur du Laboratoire d'Automatiques et de Mécaniques Industrielles et Humaines (LAMIH) de L'Université de Valenciennes a accepté d'examiner mes travaux. Sa compétence dans le domaine de l'oculométrie au service de systèmes homme-machine est incomparable en France. Il m'a permis d'enrichir mes connaissances dans ce domaine grâce aux différents travaux engagés à Valenciennes avec des partenaires industriels. Qu'il en soit vivement remercié.

La présence de Monsieur Yves SOREL, animateur du groupe de recherche du GDR 134 TDSI du CNRS, Ingénieur de recherche à l'Institut National de Recherche en Informatique et Automatique (INRIA) de Rocquencourt, comme membre du jury me touche particulièrement. L'approche flot de données, des langages synchrones et la recherche permanente dans le concept d'une meilleure adéquation entre l'algorithme et l'architecture, m'ont permis d'avancer dans ce travail. Je lui suis reconnaissant des remarques judicieuses et de sa disponibilité malgré un emploi du temps chargé dans ses fonctions d'organisateur de congrès nationaux et internationaux. Qu'il veuille bien trouver dans ce mémoire, l'expression de mes profonds remerciements.

Je voudrais associer à mes remerciements toutes les personnes qui ont permis la réussite de ce travail. J'ai une attention toute particulière pour l'équipe enseignante de l'IUP de Calais, sans laquelle il m'aurait été impossible de terminer la rédaction de ce mémoire à cette date.

Je remercie particulièrement Monsieur José LANNOO, directeur de l'établissement IUP filière des Systèmes Industriels de l'Université du Littoral pour ses précieux conseils et la confiance qu'il m'a accordée.

J'exprime à Messieurs Jean Bernard CHOQUEL, Franck DUFRENOIS, Régis LHERBIER, Gilbert MAGNIER, Joël MARICHEZ, Gilles ROUSSEL, Francis SALOME et Abdelmalik TALEB-AHMED, les responsables des études, mes sincères remerciements pour m'avoir donné une disponibilité à mes recherches et de leur compréhension.

La liste est trop longue pour que je cite toutes les personnes qui m'ont encouragé et aidé durant ces années. A Nacer, Serge, Ahcène, Sophie, Vincent, Francis, François, Patrick, Aziz, Brigitte, Ludovic, Moncef, Catherine, Isabelle et les autres que je n'oublie pas, qu'ils soient tous remercier de leur soutien amical.

Je remercie également Madame Régine CASTEGNIER et Madame Christine MARQUIS pour leur disponibilité et efficacité dans le travail de secrétariat.

Je tiens à remercier Monsieur Jean Pierre DEHORTER qui a assuré avec soin les travaux d'imprimerie et de reproduction de ce mémoire.

Merci à mes parents et à mon frère qui m'ont apporté leur soutien affectif et moral. Dans les moments de doutes, ils ont su me conseiller. Ils tiennent une part essentiel dans la concrétisation de mes travaux.

Table des matières

Introduction générale	1
Chapitre 1. L'oculométrie	3
1.1 Introduction	3
1.2 Définition	3
1.3 Données essentielles du système visuel	4
1.3.1 Mouvements oculaires	4
1.3.2 Structure de la rétine	5
1.3.3 Champ de vision	5
1.4 Les différentes techniques de mesure de déplacement oculaire	6
1.4.1 L'électro-oculographie	6
1.4.2 Anneau scléral	8
1.4.3 Méthodes optiques à cellules photo-électriques	10
1.4.3.1 Frontière iris sclérotique	10
1.4.3.2 Reflet cornéen	10
1.4.3.3 Méthode du miroir	11
1.4.4 Méthode différentielle de la double image de Purkinje	12
1.4.5 Méthode du reflet cornéen	13
1.4.5.1 Eye monitor	14
1.4.5.2 Méthode de registration des mouvements oculaires par recherche d'une forme pupillaire patron	15
1.4.5.3 Méthode portable NAC EYE MARK RECORDER	17
1.4.6 Méthode du vecteur cornéen ou pupille brillante	18
1.4.6.1 Oculomètre distant	20
1.4.6.2 Le suiveur de regard	21
1.4.6.3 Dispositif ASL4000 version portable	22
1.4.6.4 Eye gaze computer system ou ERICA	23
1.5 Conclusion	25
Chapitre 2. Extraction d'informations à partir d'une image vidéo	26
2.1 Introduction	26
2.2 Définition d'une image bidimensionnelle	26
2.2.1 Echantillonnage spatial	26
2.2.2 Echantillonnage dynamique	27
2.3 Les fondements du traitement d'images numériques	27
2.4 Les opérations de bas niveaux	28
2.4.1 Segmentation d'une image	28
2.4.1.1 La segmentation par classification de pixels	29
2.4.1.2 La segmentation par régions	29
2.4.1.3 La segmentation par contours	29
2.4.2 Profil des contours	30
2.4.3 La technique de détection par seuillage	31
2.4.4 La technique du filtrage	31
2.4.4.1 Filtre gradient	32
2.4.4.1.1 Opérateur gradient	32
2.4.4.1.2 Opérateurs de Roberts	33
2.4.4.1.3 Opérateurs de Prewitt et Sobel	33
2.4.4.1.4 Opérateur de Kirsh étendu à 3 et 5 niveaux	34

2.4.4.2 Filtre Laplacien	35
2.4.4.3 Filtre Gaussien	35
2.4.4.4 Filtre optimal	36
2.4.4.4.1 Filtre de Canny	36
2.4.4.4.2 Filtre de Deriche	38
2.4.4.4.3 Shen et Castan	39
2.4.5 Implantation récursive de l'opérateur optimal de Deriche	39
2.4.5.1 Décomposition sous forme d'opérateurs récursifs de niveaux 2	40
2.4.5.2 Conséquences de la récursivité sur le nombre d'opérations élémentaires	41
2.4.5.3 Extension au cas d'un traitement 2D	41
2.5 Codage en chaîne des contours d'une image binaire	45
2.5.1 Définitions générales du traitement binaire d'une image	45
2.5.1.1 Termes généraux	45
2.5.1.2 L'imbrication	46
2.5.2 Description d'une image décrite à partir d'une séquence de scrutation	46
2.5.2.1 Le voisinage de pixels en huit voisins	46
2.5.2.2 Le segment	47
2.5.2.3 La forme	47
2.5.2.4 Décomposition en type de segment	48
2.5.3 Représentation sous forme d'une liste de contours chaînés	49
2.5.3.1 Description du codage en chaîne de contours	49
2.5.4 Codage du contour d'une image binarisée par la technique du seuillage	50
2.5.4.1 L'accès séquentiel	50
2.5.4.2 L'opérateur de suivi de contour	51
2.5.4.2.1 Le suivi de contour	51
2.5.4.2.2 Le marquage	52
2.6 Codage d'image multi-niveaux de gris	53
2.6.1 Bref aperçu du chaînage de pixels sur une image multi-niveaux de gris	53
2.6.2 Chaînage des pixels sur des images binaires	54
2.7 Les résultats d'un traitement de segmentation de l'image	55
2.8 Conclusion	58

Chapitre 3. Recherche d'une implantation matérielle du filtre récursif de Canny-Deriche 59

3.1 Introduction	59
3.2 Mise en correspondance de plusieurs techniques	60
3.2.1 Mise en oeuvre d'une implantation matérielle	60
3.2.1.1 Solution proposée par le laboratoire GERE	60
3.2.1.1.1 Anticipation par ajout d'un îlot de pôle	62
3.2.1.1.2 Anticipation par ajout de pôles dispersés	62
3.2.1.2 Utilisation de l'arithmétique distribuée	64
3.2.1.2.1 Exemple d'un calcul	66
3.2.1.2.2 Structure de l'accumulateur	67
3.2.1.3 Extension de l'arithmétique distribuée au filtre récursif échantillonné	68
3.3 Disproportion traitements - problèmes	71
3.4 Conclusion	73

Chapitre 4. Technique originale de chaînage 74

4.1 Introduction	74
4.2 Principes du contourage	74
4.2.1 Surcodage du pixel par fenêtre 3*2	76
4.2.2 Le vecteur « Sémaphore » des fenêtres	76

4.2.2.1 Sémaphores liés au segment	77
4.2.2.2 Sémaphores liés au type du segment auquel appartient le pixel	77
4.2.2.3 Sémaphores liés aux noeuds	78
4.2.2.4 Sémaphores liés aux types de noeuds	79
4.2.3 Recherche du type d'un segment par balayage par le vecteur ETAT	79
4.2.4 Compression et Normalisation des informations	80
4.2.5 Chaînage des traits	82
4.3 Implantation algorithmique sous forme parallèle	83
4.3.1 Notion de temps réel	83
4.3.2 Recherche d'un traitement parallèle	84
4.3.2.1 Description fonctionnelle	84
4.3.2.2 Graphe de dépendance	85
4.3.2.3 La formulation flot de donnée	86
4.3.3 Le concept flot de donnée	87
4.3.3.1 Le langage SIGNAL	87
4.3.3.1.1 Syntaxe et sémantique	88
4.3.3.1.2 Fonctions	88
4.3.4 Application au codage d'une image binarisée	90
4.3.4.1 Expression de deux sommets ajoutés à la liste des sommets habituels	91
4.3.4.2 Explication de l'architecture du graphe	92
4.3.4.3 Résultat du traitement d'une image test	96
4.3.4.4 Conclusions sur l'efficacité d'une implantation massivement parallèle	97
4.4 Conclusion	97
 Chapitre 5. Dispositif de pointage sur un écran informatique	 98
5.1 Introduction	98
5.2 Le système d'acquisition d'une image	99
5.2.1 Calibrage de la caméra	100
5.2.2 Positionnement de la caméra	102
5.3 Utilisation d'un ensemble de sources ponctuelles délimitant le champ visuel	103
5.4 Caractérisation des données anatomiques de l'oeil	104
5.4.1 Image de l'oeil éclairé par une source I.R proche de l'axe optique de la caméra	104
5.5 Exploitation des images vidéo pour déterminer le centre du regard	106
5.5.1 Détection de la pupille noire	106
5.5.2 Détection des reflets	107
5.5.3 Procédure automatisée de la détection de la pupille et des reflets	108
5.6 Technique de la mesure absolue de la direction du regard	110
5.6.1 Procédure expérimentale de calibrage	111
5.6.2 Expérimentation sur un individu	112
5.7 Conclusion	114
 Conclusions et perspectives	 116
Annexes	118
Table des illustrations	127
Bibliographie	130

Introduction générale

L'informatique a subi les mutations des architectures matérielles des circuits spécifiques qui lui permettent d'être de nos jours un outil formidable de communication. Les bouleversements technologiques en intégration électronique, l'avancée des technologies numériques sont les paramètres qui ont permis que l'informatique s'ouvre à des échanges d'informations les plus diverses : son, images, textes. Aujourd'hui, on peut à moindre coût disposer d'une machine dont la puissance de calcul égale celle dont disposaient les mini-ordinateurs entrant dans les processus de commandes industrielles.

L'avènement du multimédia voici dix ans permet d'envisager encore bien des bouleversements dans les échanges et interactions homme-machine. On voit se dessiner les réseaux mondiaux, où les individus dialoguent avec des interlocuteurs distants de plusieurs milliers de kilomètres grâce à l'image et au son. Toutes ces évolutions font que l'ordinateur devient un élément central où se concentrent des informations en masse. L'homme se voit offrir des possibilités qui deviennent limitées si sa capacité d'action et si sa compréhension du flux d'informations ne sont plus en adéquation avec le rythme naturel des échanges dont il est capable de soutenir.

C'est dans cette perspective que nous avons été amenés à étudier un dispositif de communication qui soit plus en harmonie avec différents effecteurs humains. Le grand défi des constructeurs d'interfaces de communication des prochaines années sera d'offrir de nouvelles possibilités de dialogue permettant de combiner tous les sens humains.

Notre contribution à ces travaux sera de montrer qu'un dispositif de pointage absolu sur un écran d'ordinateur, par l'intermédiaire de l'analyse de la direction du regard, peut être envisageable grâce à des techniques robustes de traitement d'image sur la base d'algorithmes conçus pour un traitement cadencé au rythme de la vidéo.

Notre souci dans un premier temps était de concevoir un système sans contraintes majeures pour une utilisation la plus conviviale au possible, l'ergonomie étant de nos jours un atout principal de toute conception. Le domaine de l'informatique a vu apparaître des interfaces modelées pour un meilleur confort d'utilisation : A la façon du « Natural Keyboard » de Microsoft (clavier ergonomique). Il existe également des souris manuelles spécialement conçues pour les gauchers et même dernièrement un sous-main regroupant le clavier, l'écran et la souris en un seul élément d'informatique bureautique.

La commande visuelle offre une nouvelle possibilité de communication dans un domaine encore peu exploité. La communication orale est entrée dans une phase suffisamment rodée et fiable pour que l'on puisse commander son ordinateur au son de la voix. La reconnaissance de l'écriture a vu le jour avec des dispositifs portables à la manière du « Newton » d'Apple. D'une façon générale l'effecteur main reste encore le seul moyen de commande rapide, fiable et surtout précis. Les écrans tactiles sont modifiés pour apparaître sous forme d'un plan actif local où le bout du doigt vient faire glisser le curseur à l'écran. La souris classique se voit donc fortement concurrencée par des moyens plus maniables et permettant d'accroître encore plus la communication avec la machine.

Notre volonté est donc de participer à notre mesure, à une meilleure performance de dialogue avec un PC sous l'aspect d'un suiveur du regard.

Dans le premier chapitre, on illustre le domaine de l'oculométrie par des techniques expérimentales permettant d'exploiter l'activité oculaire. La démarche tente à montrer qu'il est possible par le biais de méthodes plus ou moins envahissantes ou contraignantes, de disposer de mesures signalant la direction du regard. Pour les différentes méthodes citées, il est évident que la mise en oeuvre technique est plus ou moins difficile et certaines se voient exclues si elles ne sont pas couplées à un dispositif de localisation de la tête.

Afin d'offrir la meilleure ergonomie, on s'est vu dans l'obligation d'opter pour un traitement vidéo de l'image de l'oeil, suivant une méthode similaire à la mesure du reflet cornéen, citée dans le premier chapitre. D'où le deuxième chapitre qui fait un état de l'art dans le domaine de l'extraction de contours et de la segmentation en régions afin de disposer de primitives qui enrichissent l'image numérique pour aboutir à une décision automatisée sur les éléments pertinents qui composent la scène observée. Le chapitre se consacre principalement aux méthodes robustes et ayant fait l'étude de traitement « temps réel » (masques de convolution, filtres récursifs). Les deux démarches offertes sont de travailler en image multi-niveaux de gris ou binaire. L'analyse de ces techniques montrent qu'il est très complexe de mettre en place une extraction de contour sous la forme d'un filtre optimal dans le cas d'une analyse d'une image bruitée. Après un bref aperçu des résultats offerts suivant les différentes méthodologies décrites, on montre que seuls les filtres récursifs conservent une détection de contours exploitable dans le cas de contours bruités.

Le troisième chapitre met en avant l'implantation réursive du filtre de Canny-Deriche. On compare suivant deux techniques, une possibilité d'implantation matérielle sous la forme d'un circuit spécialisé ASIC. Notre architecture exploite les propriétés de l'arithmétique distribuée et se positionne face à une démarche différente de restructuration de l'ordre du filtre, pour apporter un facteur d'accélération en adéquation avec les contraintes de temps de calculs des multiplieurs et additionneurs. On arrive à la conclusion raisonnable qu'une telle implantation exige une mise en oeuvre complexe et qu'il est peut-être judicieux de revenir sur des techniques plus rudimentaires avec un meilleur compromis complexité mesure. Ce qui n'empêche pas la faisabilité des deux techniques proposées.

De là, nous avons été amenés à concevoir une méthode originale de chaînage à partir d'un surcodage pixel suivant une technique séquentielle ligne, décrite dans le quatrième chapitre. Il est ainsi possible de modéliser parfaitement toutes formes contenues dans une image afin d'extraire les seules informations nécessaires pour un traitement donné. La possibilité offerte par ce traitement séquentiel a permis de se poser le problème d'une accélération du temps de traitement et d'envisager une implantation massivement parallèle. On montre qu'une réalisation de ce type est permise à partir d'une description sous la forme d'un graphe flot de données, pour un traitement opérant au rythme d'une ligne complète.

Nos outils de traitement d'image permettant d'exploiter une image binarisée à partir de logiciels spécifiques, nous montrons la technique de mesure retenue au cinquième chapitre pour réaliser un suiveur de regard tête libre. Cette étude permet d'envisager une conception ayant des performances suffisamment sensibles à une exploitation en tant qu'interface de pointage absolu sur un écran informatique. Son utilisation offrira des possibilités de communications pour des personnes handicapées moteurs sous forme d'outils de dialogue avec des actions distinctes validées par la seule action des mouvements oculaires.

Chapitre 1

1. L'oculométrie

1.1 Introduction

Ce premier chapitre fait état des connaissances sur le principe de mesurer l'activité oculaire. Quelques notions relatives, à l'anatomie de l'oeil et au mécanisme de perception de la vision, sont citées ne faisant appel qu'aux éléments les plus significatifs.

La majeure partie du chapitre s'intéresse à énoncer les différentes études faites sur les dispositifs mesurant l'activité oculaire. La description des différentes méthodologies permet de mettre en évidence les lacunes de certaines applications et donne un aperçu des difficultés de mise en oeuvre.

On cite sur chaque méthodologie les dispositifs qui tirent le meilleur partie des propriétés intrinsèques des mesures électriques, électromagnétiques ou optiques.

1.2 Définition

On désigne par le terme *oculométrie* le fait de mesurer une activité oculaire motrice. Ce sont les mouvements du globe oculaire dans l'orbite de l'oeil, référencés à la boîte crânienne, qui servent à commander un dispositif de contrôle visuel.

Les ordres oculaires, reconnus et évalués par un système de surveillance, permettent de mettre en jeu différents actionneurs : commande d'appareillages, suivi d'un pointeur, assistance à une activité contrôlée, analyse du système perceptif humain, évaluation de la bonne assimilation d'un slogan publicitaire; toutes ces applications existent actuellement.

C'est une étude américaine de la NASA, vers les années 1960, qui a initié ce projet. La NASA désirait améliorer l'efficacité des pilotes de chasses dans les combats aériens. Puis les applications médicales ont vu le jour pour des tests de dépistages, ou pour venir en aide aux handicapés moteurs par une assistance motrice, ou comme outil de communication. Dans le domaine industriel l'oculomètre offre des possibilités de réduction de charges de travail manuel, sur des postes hautement informatisés par le biais d'un deuxième canal de commande.

On distingue deux catégories d'oculomètres :

- Les oculomètres à désignation absolue du regard sur un objet référencé. Il faut généralement associer un deuxième dispositif qui a pour rôle de localiser la tête dans un repère fixe et ainsi ramener les mouvements oculaires relatifs à la tête dans ce même repère. Certaines techniques à base de traitement vidéo font une mesure absolue directement.
- Les oculomètres renseignant sur l'activité oculaire. Cette technique est plus simple à mettre en oeuvre, mais offre moins de champs d'application. On recherche les

mouvements relatifs des yeux afin de mouvoir un pointeur par des actions en tout ou rien (droite ou gauche et haut ou bas).

1.3 Données essentielles du système visuel

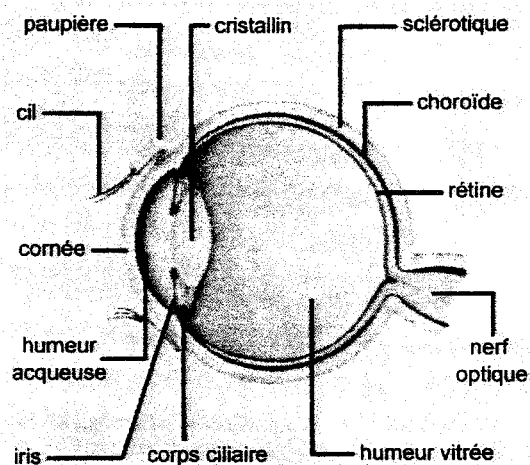


Figure 1-1 : Coupe de L'oeil.

L'oeil est limité par trois membranes (Figure 1-1):

- * La *sclérotique*, membrane protectrice, en avant elle est transparente, elle forme la *cornée*.

- * La *choroïde*, se prolongeant en avant par l'*iris*, ouvert par la *pupille*, à diamètre variable suivant l'intensité de la lumière entrante.

- * La *rétine*, membrane nerveuse sensible sur laquelle se forme les images fournies par les milieux antérieurs transparents de l'oeil (*cornée*, *humeur aqueuse*, *cristallin*, *humeur vitrée*).

1.3.1 Mouvements oculaires

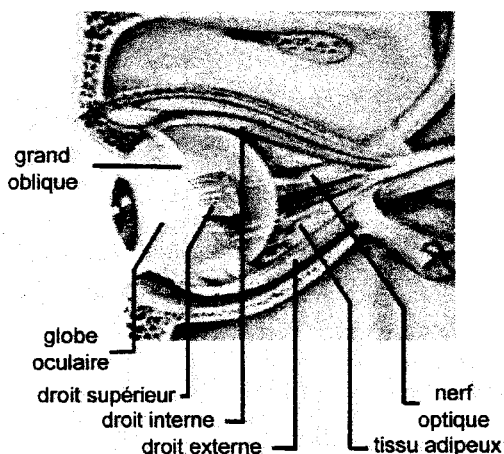


Figure 1-2 : Oeil et orbite.

Le globe oculaire est l'objet de mouvements rapides, saccadés mais de faible amplitude. Sa motricité est extrêmement rapide grâce à l'action de six muscles attachés au niveau de la sclérotique (Figure 1-2).

On distingue globalement, deux catégories de mouvements oculaires :

La première nous concerne principalement par le fait qu'elle est d'origine volontaire. Un double mécanisme combiné de saccades et de temps de fixation se met en place lorsqu'un sujet cherche à analyser une scène. L'oeil balaye rapidement par des changements brutaux de directions (450°/s pour une amplitude de 20°) toutes zones qui lui sont sensibles

(formes, mouvements d'objets, changement de luminosité). Les temps de pause (100 à 2000 ms) permettent de transmettre au cerveau les informations contenues dans une zone fixée.

La deuxième catégorie est de type réflexe donc non consciente. Il y a le mouvement visuo-vestibulaire qui correspond à un déplacement oculaire compensatoire d'un mouvement de la tête ou du corps ou de la poursuite d'un mobile à faible vitesse. Il existe également des déplacements très fins de l'oeil qui assurent la continuité du transfert d'information visuelle par une recomposition chimique de la *rhodopsine*¹.

¹ Rhodopsine: Pigment de la rétine indispensable à la vision.

La durée d'une fixation donne une idée sur le temps que le sujet met pour décoder, enregistrer, transmettre et mémoriser l'information visuelle. L'analyse du comportement visuel montre que plusieurs facteurs influent sur ce temps de fixation :

- * Le type de tâche.
- * Le processus de traitement.
- * L'aspect familier de l'environnement visuel.

L'intégration de stratégies de séquences visuelles avec une aptitude à les répéter sans un contrôle conscient permet de gagner un temps significatif sur les temps de fixation.

1.3.2 Structure de la rétine

La rétine est le capteur qui traduit la lumière en influx nerveux d'une manière spatiale et spectrale. L'information est véhiculée par le *nerf optique* vers le système nerveux qui la traite d'une manière sensitive. La conversion lumière-influx s'opère par deux types de cellules :

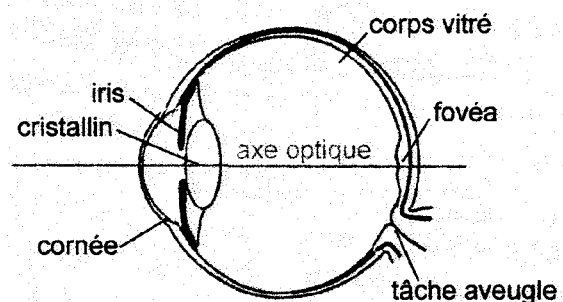


Figure 1-3 : Axe optique (ligne passant par la fovéa et le centre pupillaire).

- * Les *bâtonnets* sensibles aux formes ont un rôle sous faible éclairément (scotopique).
- * Les *cônes* interviennent dans la vision (photopique) et permettent d'évaluer les couleurs.

La concentration des cônes n'est pas la même sur toute la surface de la rétine. On situe une zone privilégiée, dite *fovéa* (Figure 1-3), située dans l'axe optique du cristallin. Les groupes d'éléments photosensibles en surnombres dans cette zone rétinienne permettent d'attribuer une limite à l'acuité visuelle. La distance entre cônes est d'environ de 2.2µm. Ce qui fixe l'acuité point à une minute d'arc environ.

Les fibres nerveuses qui constituent le nerf optique transmettent inégalement les informations des cônes. La répartition du nombre de cônes rattachés à une même fibre va en augmentant lorsque l'on s'éloigne de la fovéa.

1.3.3 Champ de vision

Les limites du champ de vision dépendent des limitations en concentration des cônes dans la zone périphérique de la rétine ; une description spatiale est donnée à la Figure 1-4. Pour des reconnaissances à des stimuli, le champ monoculaire approximatif d'un humain est 95° coté temporal, 60° coté nasal, 60° en supérieur et 70° en inférieur.

La structure sensorielle, cellules et fibres, est suffisamment variable sur la rétine pour que l'on puisse définir quatre zones concentriques pour lesquelles il intervient un mode de perception différent :

* La zone fovéale. L'acuité visuelle y est optimale. Pour une vision attentive de qualité, instinctivement, le sujet amène l'image à examiner dans cette zone.

* La zone de surveillance : vision à une distance d'observation de 2 à 3 fois la hauteur de l'image cadrée. L'attention sur la composition de l'image permet des changements rapides de l'oeil vers tels ou tels détails. On observe déjà une diminution de l'acuité visuelle. Les mouvements oculaires s'effectuent sans mouvements de tête.

* La zone d'impression induite : vision globale pour laquelle l'oeil est particulièrement sensible aux mouvements. L'observateur a une connaissance brute, position ou forme mais pas de distinction des détails, dans ce mode de perception nous avons des mouvements conjugués de la tête et du globe oculaire.

* La zone de perception latérale. L'oeil est sensible aux mouvements rapides des objets sans pouvoir les distinguer comme tel. Souvent l'objet est apprécié très rapidement et en découle des phénomènes de réponses motrices, contrôlées ou de défense.

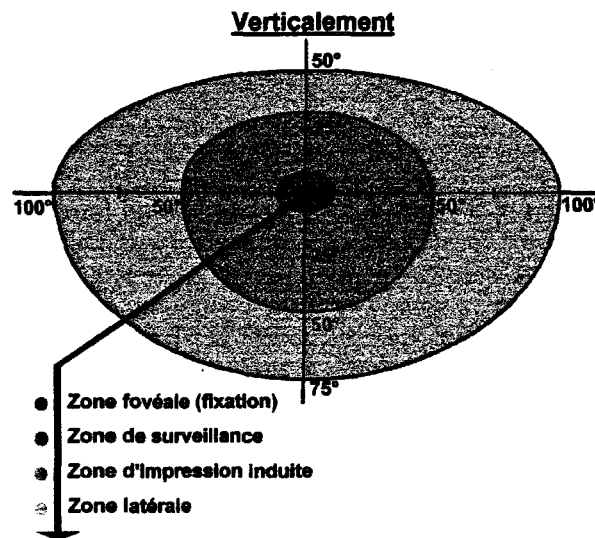


Figure 1-4 : Zones sensibles de la vision.

1.4 Les différentes techniques de mesure de déplacement oculaire

Les travaux dans le domaine de la détection des mouvements oculaires ont traversé les différents bouleversements technologiques. Certaines de ces techniques envahissantes et contraignantes par des appareillages spécifiques, peuvent apparaître dépassées vis à vis des méthodes optiques sans contact. Malheureusement, le choix de dispositifs à faibles contraintes n'offrent pas encore toutes les précisions escomptées dans certains types d'application. L'énumération des méthodes se fait dans l'ordre chronologique de leur apparition, depuis l'électro-oculographie jusqu'aux méthodes d'analyse vidéo actuelles.

1.4.1 L'électro-oculographie

La méthode Electro-OculoGraphique (E.O.G) consiste à la mesure des variations d'un potentiel produit par la rétine. L'origine de la différence de potentiel est l'existence d'un

courant ionique provenant de l'épithélium² pigmentaire. On recueille le potentiel par deux électrodes cutanées placées de part et d'autre de l'axe cornéo-rétinien. Les variations enregistrées par les électrodes sont référencées à un mouvement oculaire dans la cavité orbitaire et sont fonction approximativement du sinus de l'angle de rotation.

Cette méthode n'offre que peu de possibilités du fait que la mesure ne s'opère que pour un changement de rotation de l'oeil, sinon le potentiel est nul ; cette méthode permet d'effectuer des mesures relatives. C'est par un tracé continu du potentiel enregistré que l'on peut détecter réellement une activité oculomotrice. La mesure directement référencée à la tête ne permet la désignation d'une cible sinon que par immobilisation totale de la boîte crânienne.

La principale application de L'E.O.G était le domaine clinique, pour vérifier la sensibilité motrice du globe oculaire à des stimuli. Lacourse propose de mettre en pratique cette technique pour un outil de communication pour venir en aide aux personnes handicapées [LAC,90]². Il intitule le procédé DECS (Discrete Electro-oculographic Control System).

La Figure 1-5 montre la disposition de deux paires d'électrodes pour l'analyse des déplacements verticaux et horizontaux. Le capteur du DECS se présente sous la forme d'une paire de lunettes où les électrodes sont maintenues fermement entre la monture et la peau du visage. Un gel permet de réaliser le contact électrique. Le dispositif propose de sélectionner un port de commande par la désignation d'une des 8 directions, d'une cible visuelle ou un point central. Un retour d'information lumineuse de cette cible permet de connaître l'action validée.

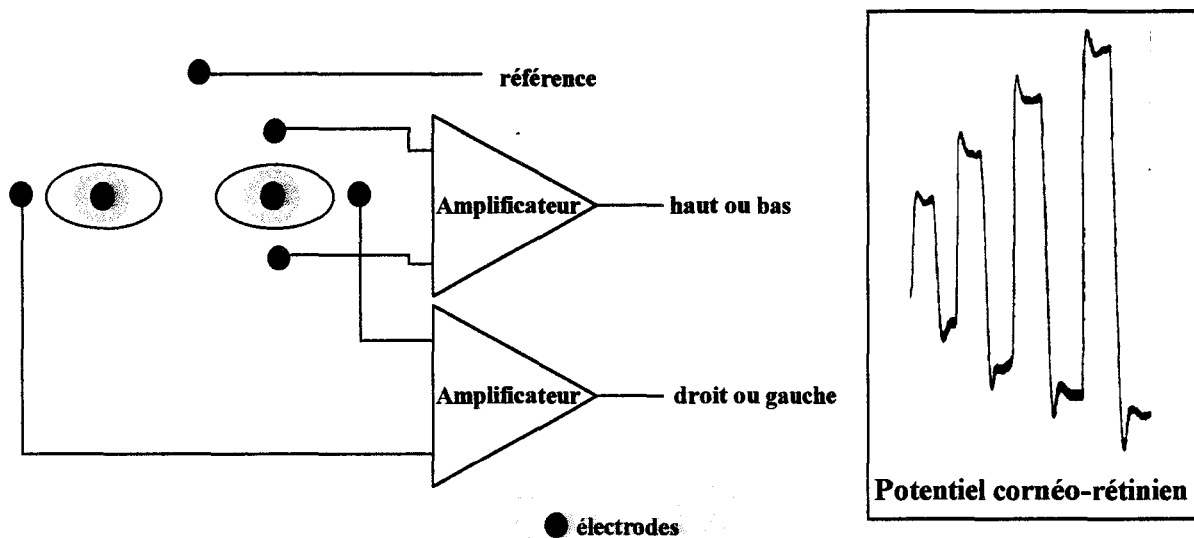


Figure 1-5 : Disposition des électrodes cutanées.

Cette méthode assez facile à mettre en oeuvre présente néanmoins des performances en évaluation des mouvements oculaires peu satisfaisantes.

Des phénomènes électriques au niveau du contact électrode-peau viennent perturber les relevés. Il faut attendre une vingtaine de minutes après la pose du capteur pour avoir une stabilité de la dérive du signal.

On observe également des variations des pics de potentiel lors du changement d'éclairage ambiant. La Figure 1-5 montre un potentiel cornéo-rétinien d'un déplacement du

² Epithélium: Tissu formé d'une ou plusieurs couches de cellules jointives et recouvrant le corps, les cavités internes, les organes.

globe oculaire de 20° entre le centre et une des deux directions horizontales sous quatre incidences lumineuses. Une recalibration est nécessaire pour éliminer la perturbation qui peut atteindre 40% en cas d'un fort contraste lumineux.

Les déplacements obliques sont très difficilement décelables par cette technique. En pratique Lacourse obtient un taux d'échecs de près de 25% dans les quatre directions obliques ce qui ramène le dispositif à 90% en accord avec les ordres oculaires émis. Le problème de la détection oblique est relatif à l'inhomogénéité de la conductivité électrique des tissus autour de l'oeil. Lors de mouvements palpébraux³ associés à un déplacement oblique ou vertical, on observe des glissements de contacts qui rendent la mesure imprécise.

L'avantage de cette méthode face aux autres techniques est qu'elle permet des mesures en excentricité plus importante. Le champ de mesure couvert est de 50° dans toutes les directions.

1.4.2 Anneau scléral

C'est une méthode électromagnétique développée au départ pour l'analyse des mouvements oculaires chez l'animal [ROB,63][○]. Une bobine placée dans une lentille de contact détecte le courant induit produit par deux champs perpendiculaires, alternatifs et déphasés. La résultante des deux champs est en relation avec la position de l'oeil. La contrainte de cette première étude réside dans la pose de la lentille rendue délicate par les fils de mesure qui limitent la durée de tolérance à 20 mn.

Afin d'éviter les fils, une nouvelle technique mettant à profit les variations de mutuelle inductance induit par une bobine fixe et une bobine en court circuit contenue dans le verre scléral, s'est révélée d'un meilleur confort d'utilisation. Reulen et Bakker [REU,82][○] décrivent une méthode similaire par double champ magnétique. Bour et al reprennent cette technique et apportent des corrections sur la non-linéarité inhérent au double champ [BOU,84][○].

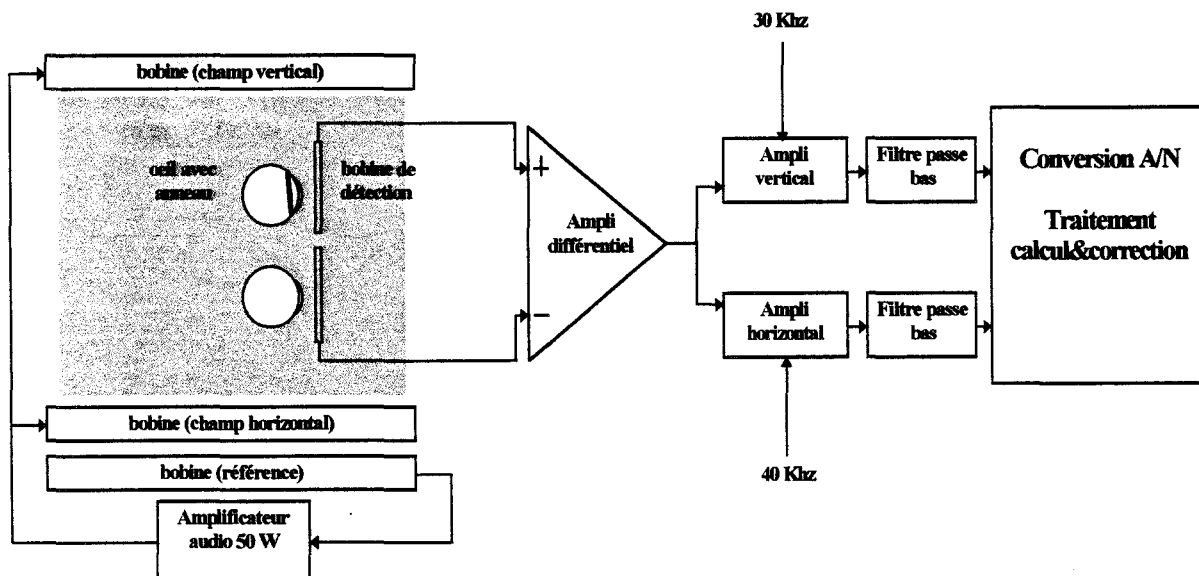


Figure 1-6 : Système expérimental DMI.

³ Palébral: Relatif à la paupière.

La méthode intitulée D.M.I (double magnetic induction) se présente sous le schéma fonctionnel représenté Figure 1-6.

La tête est plongée dans un champ magnétique homogène généré par deux bobines de 70 cm de 46 spires. Chaque bobine est référencée respectivement à une fréquence de 30 KHz et 40 KHz pour les mesures verticale et horizontale de la position de l'oeil. Une bobine d'une spire sert de référence et permet une contre réaction positive sur le contrôle de l'amplificateur. Le premier champ induit un courant alternatif dont l'intensité dépend de l'orientation de l'oeil équipé de l'anneau (Figure 1-7).

L'intensité est récoltée par une bobine détectrice placée au devant de l'oeil. Au devant du deuxième oeil une bobine détecte le champ primaire sans la deuxième composante induite par l'anneau. L'amplificateur différentiel permet de discriminer l'effet du champ des deux bobines génératrices pour ne récupérer que la composante induite dans le verre scléral. L'information verticale et horizontale du positionnement de l'oeil est recueillie par deux amplificateurs coupe bande, accordés sur les fréquences de référence de 30 et 40 KHz afin de séparer les deux composantes. Les tensions induites dépendent directement de l'orientation de l'anneau dans le cas où les bobines réceptrices et les bobines émettrices sont fixées dans un repéré fixe, avec la contrainte d'immobilisation complète de la tête.



Figure 1-7 : Anneau scléral.

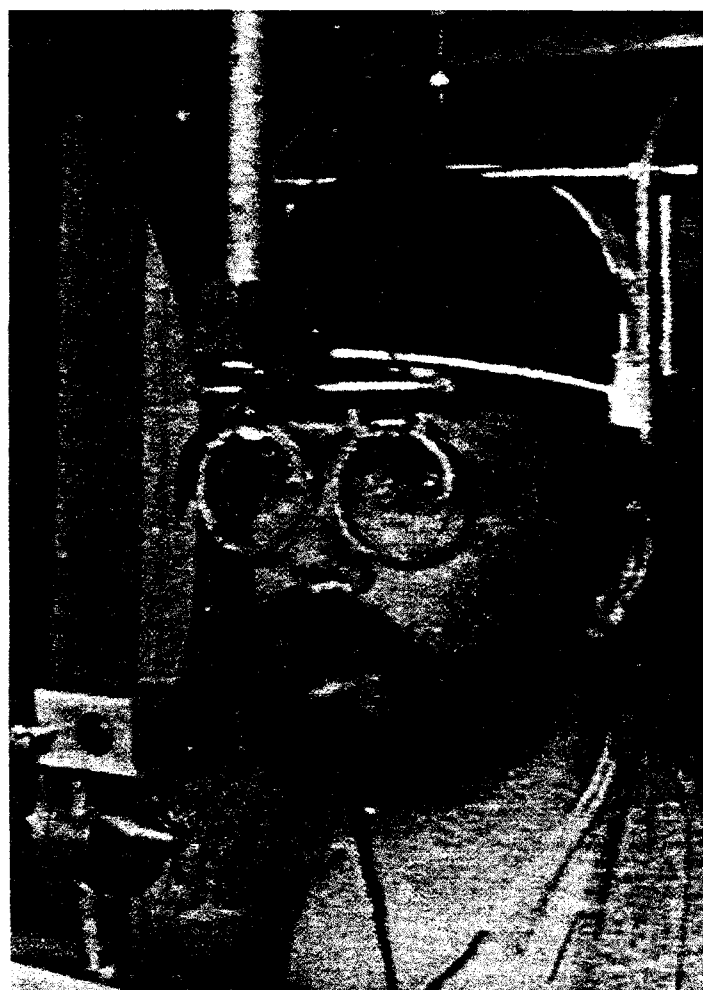


Figure 1-8 : Présentation du dispositif DMI pour l'homme.

La Figure 1-8 montre le dispositif très contraignant pour une utilisation courante. La contention de la tête très rigoureuse et une prise d'empreinte dentaire sur un plan de morsure, limite son utilisation à un dispositif expérimental.

La technique D.M.I présente plusieurs avantages. La surveillance des mouvements oculaires est très précise, 5 mn d'arc pour un champ de surveillance de -20° à 20° . La non linéarité du signal recueilli détériore légèrement les mesures pour les excentricités au-delà de 10° mais la correction reste performante pour des déplacements oculaires limités à $\pm 35^{\circ}$.

La possibilité de mesure paupières fermées ne pose aucun problème. Les fils supprimés offrent un confort et une tolérance d'utilisation de la lentille pour la majeure partie des patients. L'accoutumance au verre scléral s'opère à partir de quelques minutes

et n'apporte pas d'irritations pour une utilisation modérée d'une heure. Le champ de vision n'est pas perturbé contrairement à certaines autres techniques. Ce dispositif permet d'observer des saccades avec un maximum de vélocité de 700 °/s.

Les limitations de son utilisation restent un problème crucial. La pose du verre scléral demande une attention particulière; de plus l'appareillage exige une utilisation en milieu expérimenté avec un examen ophtalmologique. La surveillance est mono-oculaire.

1.4.3 Méthodes optiques à cellules photo-électriques

Une technique moins contraignante à base de photo-récepteurs, ce qui permet une mesure sans contact.

1.4.3.1 Frontière iris sclérotique

Cette méthode repose sur le principe de la variation de la réflectance entre l'iris et la sclérotique [TOR,51][°]. L'oeil est éclairé par une source lumineuse infra-rouge pointant vers l'iris et l'on absorbe la quantité de lumière réfléchie par une photo-cellule (Figure 1-9).

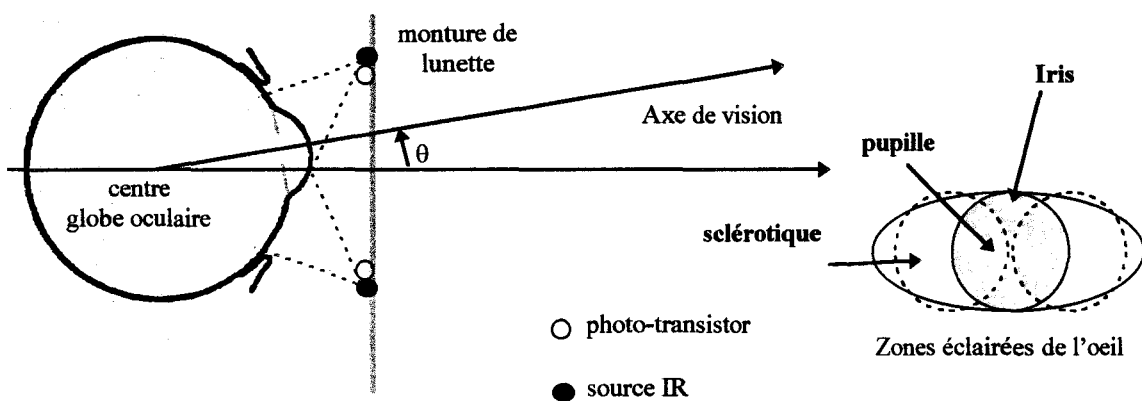


Figure 1-9 : Méthode de variation de réflectance iris-sclérotique.

La détermination de la frontière iris-sclérotique intervient pour les mouvements horizontaux et par l'intermédiaire de la frontière iris-pupille on désigne les mouvements verticaux. Le système se présente sous la forme d'une paire de lunettes, équipée de plusieurs photo-diodes, montée fixement afin de ne pas interférer sur les mesures d'un mouvement oculaire référencé à la tête. Ce procédé qui peut atteindre la résolution de 15 mn d'arc permet de détecter un déplacement angulaire de $\pm 10^\circ$. La détermination de la direction du regard sur cible fixe n'est possible que par l'adjonction d'un dispositif de mesure des mouvements de la tête. L'intérêt majeur d'un tel système est sa faible complexité de mise en oeuvre et la simplicité de l'appareillage : une paire de lunettes sur le patient. Les mouvements verticaux posent des problèmes délicats de distorsion, combinés au masquage partiel de la pupille par les paupières.

1.4.3.2 Reflet cornéen

Cette méthode se différencie de la frontière iris-sclérotique par le fait qu'elle capte la position d'un reflet infra-rouge se produisant sur la cornée. Le capteur de la commande visuelle de Rinard et Rugg [RIN,72][°] est logé sur la monture d'une paire de lunettes.

L'obtention d'un reflet cornéen est due à l'éclairement de l'oeil par une source infra-rouge (diode LED); voir Figure 1-10.

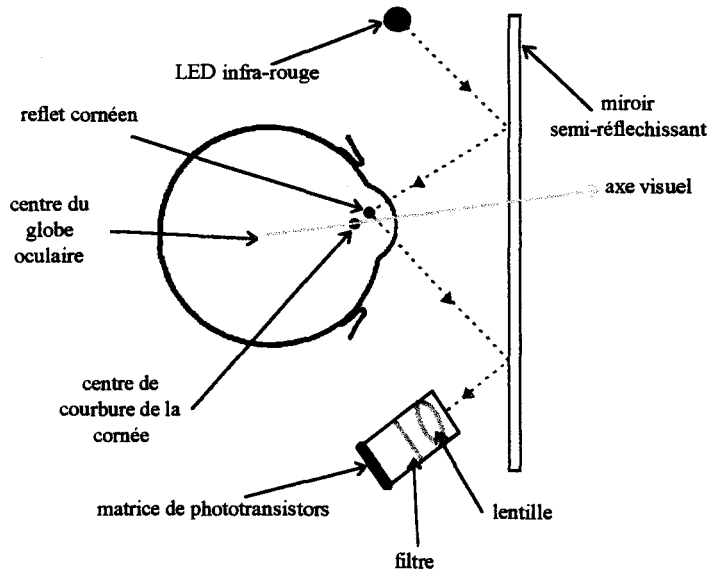


Figure 1-10 : Principe de récupération du reflet cornéen.

Le faisceau infra-rouge est envoyé sur l'oeil par réflexion sur un miroir semi-réfléchissant par un angle d'incidence de 45° . Le positionnement de ce reflet s'obtient par analyse spatiale de la distribution du reflet sur la cornée sur la matrice (32*32 éléments) de photo-récepteurs infra-rouges, après un passage dans une lentille de focalisation et un filtre. Le couple X,Y du phototransistor percevant le plus d'intensité lumineuse est directement en relation avec le mouvement oculaire référencé à la tête [GUI,88][⊗].

L'ensemble compact et très léger est bien adapté pour une utilisation intense sans trop d'incommodation. Le champ visuel couvert est de $\pm 25^\circ$ avec une fiabilité importante. L'utilisation recherchée étant un outil de désignation (commande visuelle d'un menu avec validation par clin d'oeil) pour des personnes atteintes d'une cécité musculaire du cou et de la tête, il convient suffisamment sans avoir à ajouter un dispositif de repérage de la tête.

1.4.3.3 Méthode du miroir

C'est un procédé qui fut mis en place pour réaliser une machine à écrire à commande visuelle [ORZ,78][⊗]. Le patient est face à un dispositif où il se positionne face à une fenêtre d'observation. La mesure s'opère tête fixe, la personne sélectionne des lettres sur une mire à retour de validation lumineuse. La direction du regard s'effectue par le suivi d'un faisceau lumineux réfléchi sur un petit miroir de 1 mm de diamètre, incorporé dans une lentille souple cornéenne. Le principe du dispositif est schématisé Figure 1-11.

Le faisceau émis par une diode laser infra-rouge est réfléchi par le miroir oculaire et il est renvoyé par un miroir sphérique de collection vers un miroir mobile. La commande du miroir tournant par une mécanique à 2 degrés de liberté permet de ramener le signal lumineux vers une photo-diode suivie d'une électronique de décision. La mise en correspondance des paramètres qui ont mis en mouvement le miroir et la direction du regard, se réalise grâce à un changement de repère.

L'avantage d'un tel système se situe principalement au niveau de l'émission du signal lumineux qui peut être de faible puissance grâce à un rapport signal/bruit excellent. La mire à une distance d'observation raisonnable permet de réduire l'amplitude des mouvements oculaires afin de réduire la fatigue oculaire. La précision atteinte est de 1° de résolution [DER,91].

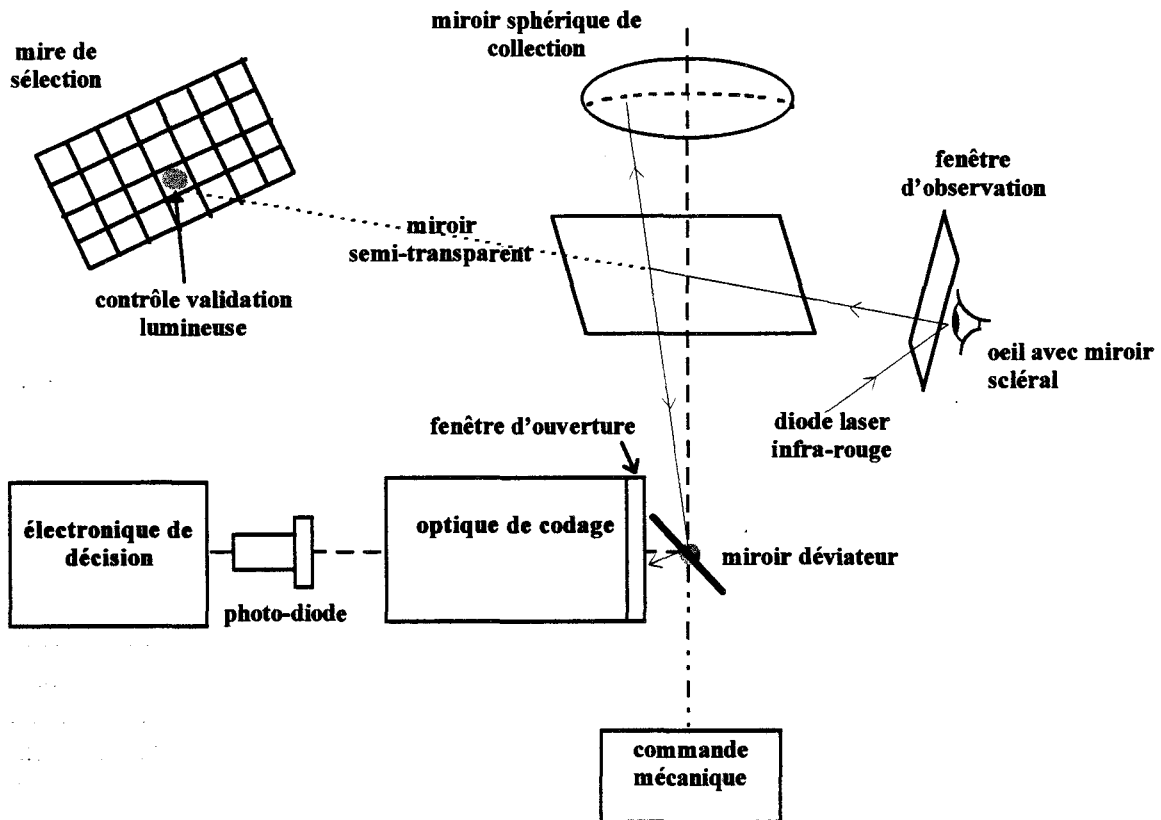


Figure 1-11 : Technique du miroir scléral (machine à écrire à commande visuelle).

1.4.4 Méthode différentielle de la double image de Purkinje

Un faisceau lumineux de forte intensité pénètre dans l'oeil qui va se réfléchir à chaque surface de séparation des milieux d'indice de réfraction différents. La première image de Purkinje est produite sur la face antérieure de la cornée, la quatrième est produite sur la face antérieure du cristallin. La Figure 1-12 illustre l'apparition de deux reflets de la 1^{ère} et 4^{ème} image de Purkinje.

Les deux images se forment sur un plan voisin et se déplacent en sens inverse de part et d'autre de l'axe visuel. La mesure de l'écart entre ces deux images donne la direction du regard vis à vis d'un appareillage extérieur fixe. Le niveau de brillance de chacun des deux reflets est très différent et pose quelques problèmes d'analyse pour la séparation des deux images. Le dispositif se compose d'un appareillage de laboratoire où le champ de vision est fortement perturbé [YOU,75]. Les mesures permettent des analyses de mouvements oculaires fins de 1° sur une plage de $\pm 12^\circ$. Son emploi de façon généralisé n'est pas possible. Cette technique différentielle apporte de nettes améliorations, en particulier sur le fait d'obtenir des mesures absolues (mesure indépendante des mouvements de la tête). Les techniques différentielles suivantes, dérivées de la technique énoncée, vont la supplanter grâce à une mise en oeuvre plus facile et un traitement plus aisé de l'image de l'oeil.

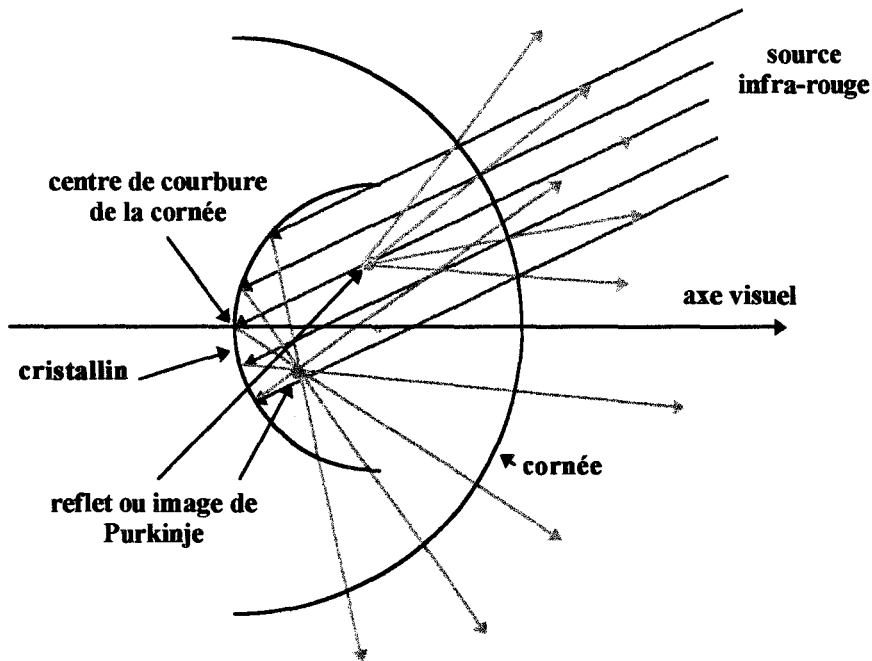


Figure 1-12 : Double image de Purkinje.

1.4.5 Méthode du reflet cornéen

La génération d'un reflet cornéen s'effectue par l'éclairage de l'oeil par une source ponctuelle lumineuse suivant le schéma suivant (Figure 1-13).

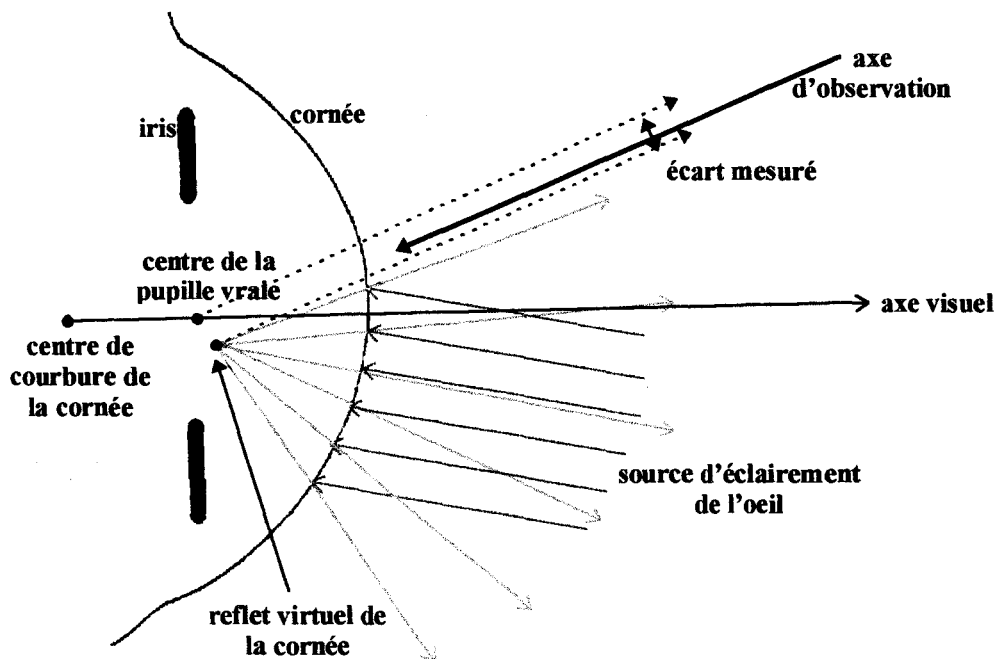


Figure 1-13 : Principe du reflet cornéen.

Le déplacement du reflet est proportionnel et opposé au sens de rotation du globe oculaire. L'écart X entre deux reflets obéit à la relation suivante :

$$X = d + OC \sin \theta$$

d : Déplacement latéral de l'oeil pour une rotation donnée du globe oculaire.

OC : Constante anatomique de l'oeil. Distance entre le centre du globe oculaire et le centre de courbure de la cornée. Sa valeur varie selon les individus; elle est proche de 5.5 mm.

Cette technique est la plus répandue au niveau des dispositifs de traitement vidéo des mouvements oculaires. Les contraintes d'utilisation sont faibles car l'acquisition des mesures se fait sans contact, sans limitation du champ visuel. Il faut préciser que ce système comporte des limitations d'utilisation. On observe une distorsion des mesures au-delà de $\pm 12^\circ$ du champ visuel du fait que le reflet se positionne dans ce cas en dehors de la partie sphérique de la cornée. Le déplacement latéral de l'oeil occasionne une source d'erreur dans l'obtention de la direction du regard : pour une saccade de 30° , d vaut jusque 1 mm du fait de la non-coïncidence entre l'axe visuel et l'axe optique de l'oeil. D'une manière générale on atteint une précision de 1° en corrigeant les non-linéarités et en prenant en compte les glissements de capteur sur la tête. Les paramètres physiologiques et anatomiques influent également sur la précision atteinte. Plusieurs travaux ont été menés dans cette voie; on citera leurs applications et une description matérielle qui les différencie.

1.4.5.1 Eye monitor

Le dispositif énoncé ici « eye monitor » équipe l'Université d'Iowa pour détecter des insuffisances de réponses pupillaires à des stimuli lumineux [MYE,91]⁹. L'originalité du système est qu'il n'y a pas de traitement numérique de l'image recueillie mais une poursuite du centre de l'image de la pupille par un calcul rudimentaire hardware du positionnement de la pupille. La Figure 1-14 décrit le calcul de l'aire et du centre pupillaire à base de compteurs.

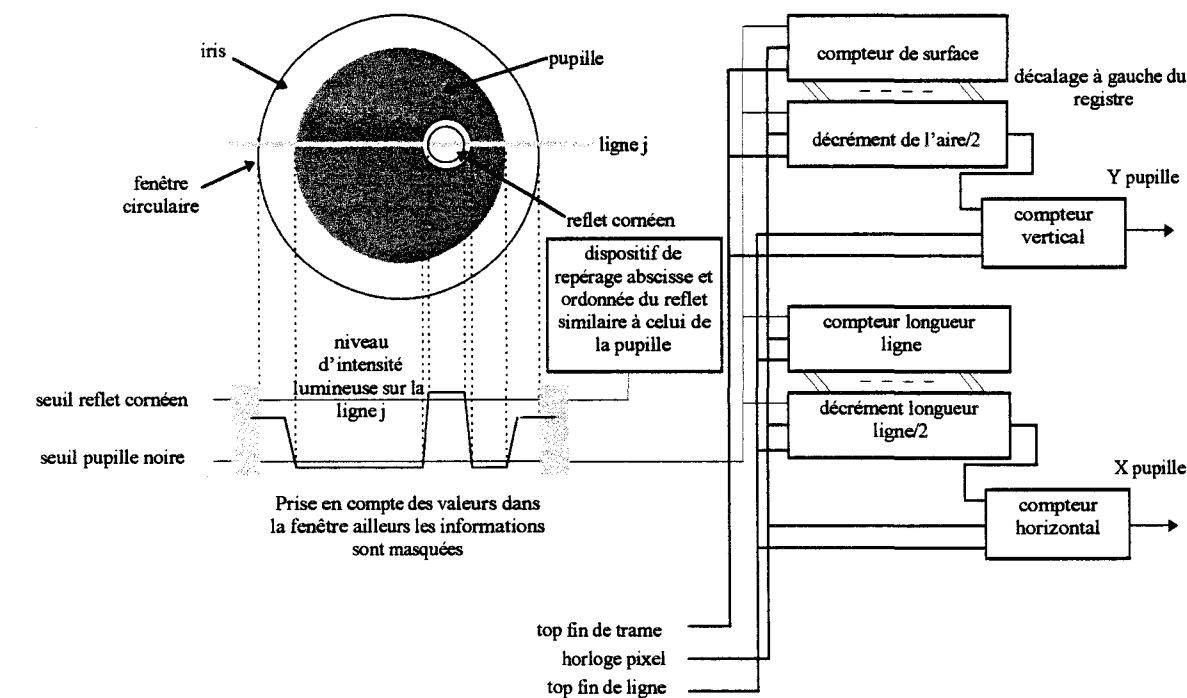


Figure 1-14 : Dispositif de la détermination de la position de la pupille au travers d'une fenêtre circulaire.

Le principe est d'opérer par une analyse semi-automatique de l'image infra-rouge de l'image de l'oeil, éclairé par une source ponctuelle I.R. Une caméra CCD envoie l'image sur le processeur vidéo au rythme de 60 images entrelacées par seconde. L'image est discrétisée par une horloge pixel qui permettra de totaliser les éléments appartenant aux formes respectives « pupille » et « reflet cornéen ». Pendant la première demie-trame, le compteur de surface totalise le nombre de pixels inférieurs au seuil référencé à la pupille noire. A la demie-trame suivante on mémorise les données du registre surface sur un décompteur qui absorbe la surface divisée par deux en opérant par un décalage à gauche du registre supérieur. On décrémente jusqu'à obtenir 0 pour commander la position verticale du centre de la pupille. Sur le même principe on opère sur un comptage des lignes traversant la pupille. Le positionnement n'est autorisé qu'au travers d'une fenêtre circulaire de surface adaptée à la surface pupillaire, ailleurs les données sont discriminées et n'influent d'aucune manière sur le résultat du positionnement. La tâche d'un opérateur humain est de repositionner le centre de la fenêtre à l'initialisation du dispositif et lors des décrochages dus aux variations rapides de la surface pupillaire, aux mouvements oculaires non contrôlés ou aux masquages intempestifs créés par les battements de paupières.

On observe qu'il n'y a pas de système de contrôle d'analyse de formes, ce qui oblige l'aide d'un opérateur humain. Ses interventions restent minimales du fait que l'analyse est dirigée par des stimuli dont on connaît le positionnement dans l'espace. On a une forte probabilité de percevoir un mouvement oculaire vers la zone de sélection (oeil esclave des ordres lumineux). Le dispositif autorise un mouvement de tête réduit à une amplitude de déplacement latéral de ± 3 cm. Le champ couvert est de $\pm 20^\circ$ pour une résolution de $\pm 0.2^\circ$. Pour des excentricités de $\pm 45^\circ$ on chute à 0.5° . Sachant que les mouvements de tête sont autorisés, qu'il n'y a pas de traitement complet des formes pupillaires et du reflet cornéen, que d'autres dispositifs plus contraignants ont des résultats modestes, on peut affirmer que ces résultats sont surestimés dans le cas d'une utilisation courante. Les résolutions indiquées sont établies pour des corrections et la mise en place de réétalonnage de fréquence élevée, ce qui le place dans un système clinique à forte contrainte d'utilisation nécessitant une salle d'expérimentation.

1.4.5.2 Méthode de registration des mouvements oculaires par recherche d'une forme pupillaire patron

Wagner propose de localiser une fenêtre rectangulaire qui délimite exactement la surface pupillaire pour diminuer la quantité d'information au minimum. Cette volonté s'explique par l'enregistrement des images de l'oeil face à un stimulus pour un traitement différé et conservation des images classifiées pour chaque patient [WAG,92]⁹. La tête est maintenue fixe avec une prise dentaire afin de faire abstraction des mouvements de tête perturbant le positionnement du reflet cornéen. La Figure 1-15 montre le dispositif optique par lequel le patient est analysé sur ses mouvements oculaires dirigés sur une cible fixe avec des stimuli lumineux couvrant un champ de vision de $\pm 30^\circ$ horizontalement et $\pm 20^\circ$ verticalement.

L'oeil est éclairé par une source infra-rouge par l'intermédiaire d'un miroir semi-transparent. Le champ de vision n'est pas perturbé par la caméra se

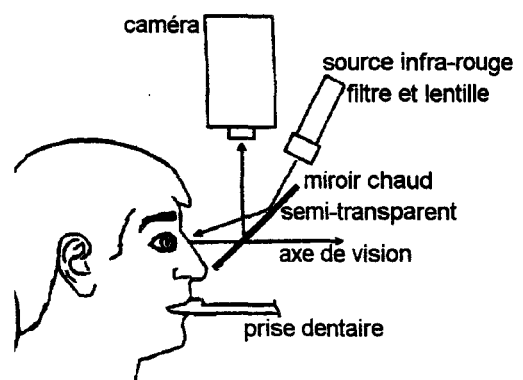


Figure 1-15 : Dispositif optique de l'analyse des mouvements oculaires par la méthode du reflet cornéen.

situant à la perpendiculaire et récoltant l'image infra-rouge par réflexion à 45° de l'axe central de la cible.

La démarche principale est de localiser la pupille par mise en correspondance d'une fenêtre mobile et de taille modulable, avec une image réduite type, de la forme pupillaire. Par un calcul de coefficient de corrélation entre l'image patron et la fenêtre sélectionnée on repère la pupille par un critère optimal de vraisemblance. L'image recueillie originale et l'image tampon patron sont illustrées Figure 1-16.

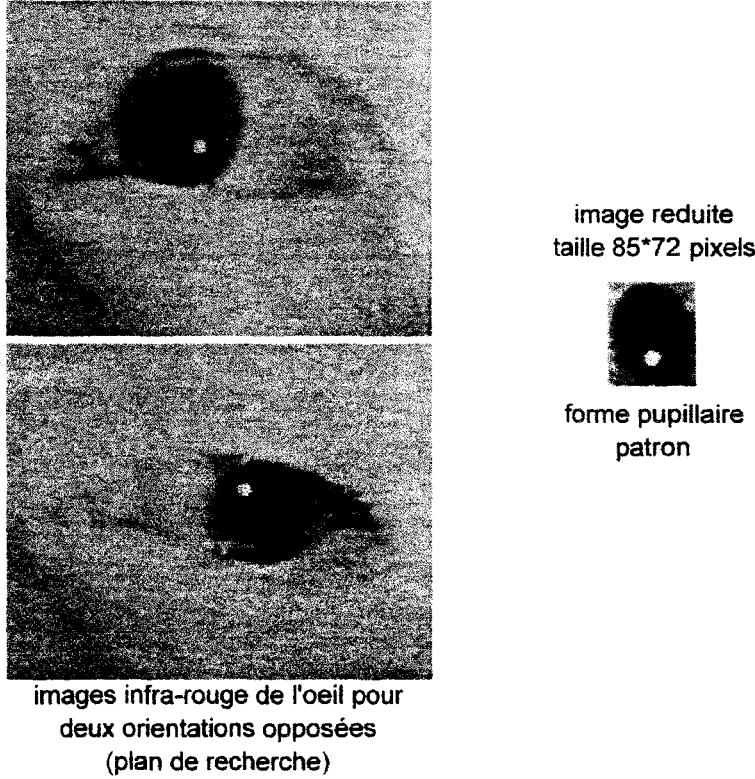


Figure 1-16 : Exemples d'images I.R et illustration de l'image patron recherchée.

Dans le cas du dispositif de suivi de pupille dynamique, Wagner montre que le calcul de coefficient de corrélation ne convient pas et propose d'identifier la pupille par le calcul de la somme des différences absolues entre les éléments de l'image originale et l'image de référence (SAVD). On obtient les relations suivantes:

Définition de la fenêtre positionnée en i,j et de taille K,L éléments :

$$S^{i,j}(k,l) \equiv S(i+k-1, j+l-1) \text{ où } 1 < k < K, 1 < l < L, \\ 1 < i < M-K+1 \text{ et } 1 < j < N-L+1$$

Définition du coefficient de corrélation entre l'image originale et le patron assorti :

$$\rho(i,j) \equiv \frac{\sum_{k=1}^K \sum_{l=1}^L [T(k,l) - \mu_T] [S^{i,j}(k,l) - \mu_S(i,j)]}{\sqrt{\sum_{k=1}^K \sum_{l=1}^L [T(k,l) - \mu_T]^2} \sqrt{\sum_{k=1}^K \sum_{l=1}^L [S^{i,j}(k,l) - \mu_S(i,j)]^2}}$$

avec $i : 1 \rightarrow M-K+1$ et $j : 1 \rightarrow N-L+1$

Les relations suivantes correspondent à la moyenne des niveaux de gris sur chacune des images. $T(k,l)$ est l'image discrétisée de l'image de référence et $S^{ij}(k,l)$, l'image originale au travers de la fenêtre.

$$\mu_T = \frac{\sum_{k=1}^K \sum_{l=1}^L T(k,l)}{K.L}, \mu_S = \frac{\sum_{k=1}^K \sum_{l=1}^L S^{ij}(k,l)}{K.L}$$

Cette technique répond au repérage de la pupille à 98% des cas d'un mouvement oculaire suivant le champ de vision établi par la cible de test. Les calculs de positionnement du reflet et du centre de la pupille sont traités par un algorithme de traitement d'image spécialement développé. Le problème majeur est l'occlusion partielle de la pupille pour des mouvements verticaux par la paupière. Dans ce cas précis, le traitement ne s'effectue que sur un demi-patron inférieur de référence. La déformation ellipsoïdale de la pupille pour des excentricités importantes reste maîtrisable et l'apparition du reflet à l'intérieur de la pupille n'intervient que très légèrement sur l'indice de vraisemblance de la forme pupillaire.

1.4.5.3 Méthode portable NAC EYE MARK RECORDER

Il s'agit d'une technique utilisant la réflexion sur la cornée par une source visible projetée sur l'oeil [CHA,87][○]. L'instrumentation repose sous la forme d'un masque qui permet une grande liberté de mouvement de la tête (Figure 1-17). Un mixage optique du reflet de la source de l'oeil et du champ visuel du sujet est réalisé par un dispositif optique. L'image combinée est transmise vers une caméra d'enregistrement par le biais du canal optique d'une fibre (Figure 1-18).

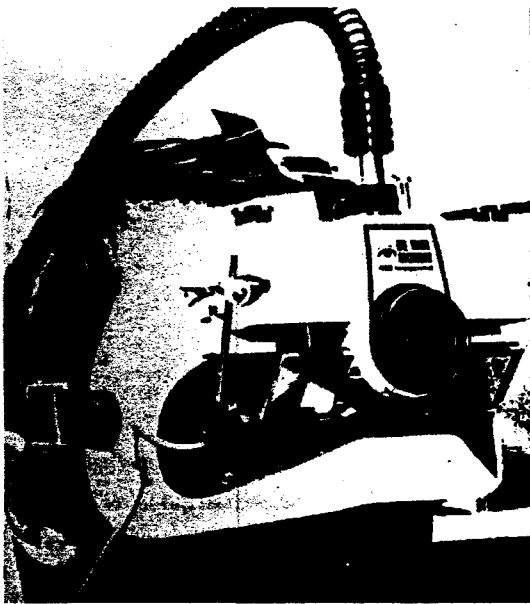


Figure 1-17 : Oculomètre NAC EYE MARK RECORDER 4.

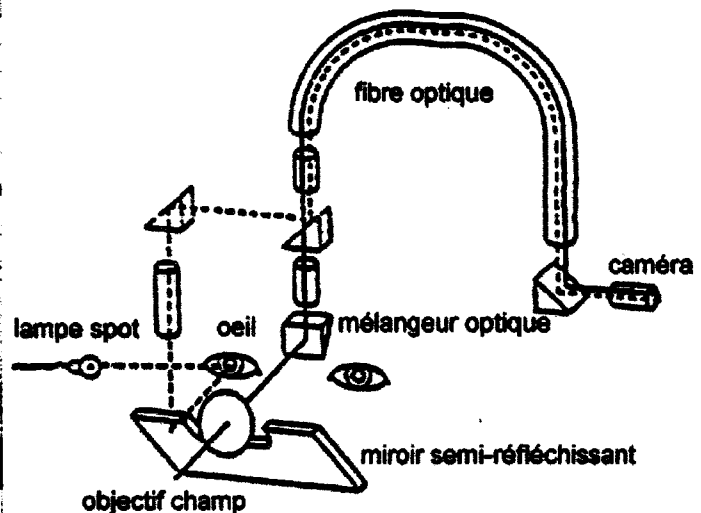


Figure 1-18 : Principe optique du capteur NAC 4.

L'identification du reflet superposé à l'image du champ est obtenue par sa forme en V, forme du filament de la source d'éclairage.

L'avantage de ce dispositif est la grande liberté de mouvement en vue d'une identification de la direction du regard par la matérialisation de marqueurs champ. L'inconvénient majeur de cet oculomètre est la pose du masque qui épouse le haut du visage maintenu fermement par des sangles qui ramènent le temps de l'expérience à 30 mn. Les

mesures sont sensibles aux glissements du capteur lors de mouvements brusques de la tête. Dans la configuration présentée, le traitement des mouvements oculaires ramenés dans l'axe visuel se réalise en différé par un dépouillement automatique suivant un marquage spatial de l'environnement visuel et un traitement vidéo spécifique [GER,87][Ⓢ].

Les déplacements oculaires limités à $\pm 12^\circ$ ne sont pas affectés directement au champ visuel puisqu'il y a tolérance de désignation de la direction du regard tant que les marqueurs sont repérés dans l'image recueillie par le capteur. La résolution atteint 1° .

1.4.6 Méthode du vecteur cornéen ou pupille brillante

La technique repose sur une méthode optique qui permet de réaliser une image brillante de la pupille par la réflexion sur la rétine du faisceau d'une source ponctuelle colinéaire à l'axe de vision d'un capteur vidéo (Figure 1-19).

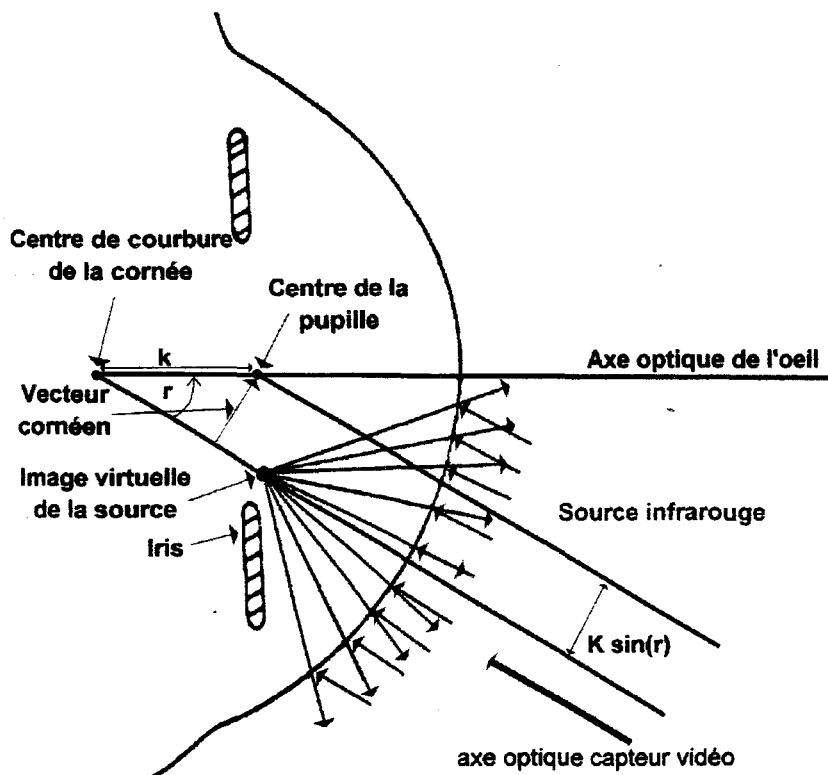


Figure 1-19 : Principe du vecteur cornéen.

Merchant et Morisette ont été les précurseurs dans le traitement vidéo de l'image de l'œil infra-rouge pour cette méthode différentielle [MER,74][Ⓢ]. L'angle entre le faisceau incident provenant de la source et le faisceau de retour capté par le capteur doit être faible, sinon la pupille est noire et la méthode différentielle n'est plus possible. Cette particularité exige un dispositif optique soigné représenté par la Figure 1-20. Une image typique de l'œil est illustrée Figure 1-21.

Le vecteur cornéen a pour origine le point de projection du centre de la pupille d'entrée sur la direction parallèle à l'axe optique vidéo passant par le centre de courbure de la cornée et le centre du reflet, et pour extrémité le centre de la pupille.

On a le module du vecteur cornéen qui vaut:

$$X = K \cdot \sin \theta$$

K : Constante anatomique de l'oeil. Distance entre le centre courbure de la cornée et le centre de la pupille d'entrée. Sa valeur est de l'ordre de 4.75 mm.

θ : Angle entre l'axe optique de l'oeil et la direction de la source lumineuse (identique à l'axe vidéo).

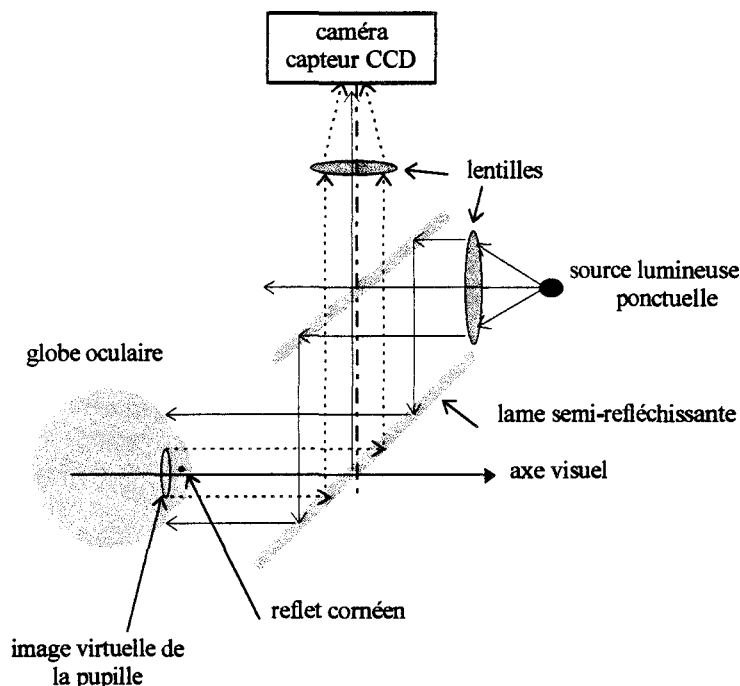


Figure 1-20 : Dispositif optique qui laisse le champ de vision libre.

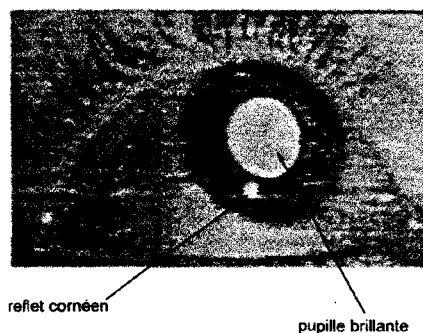


Figure 1-21 : Image du reflet pupillaire et du reflet cornéen, tache lumineuse réduite et d'intensité plus importante.

Dans le cas où la source est éloignée, on considère que la mesure est indépendante de la position absolue du centre de rotation de l'oeil. Nous observons donc la direction absolue du regard (axe visuel⁴). Le principe revient donc à localiser le plus précisément possible, le centre de la pupille et le reflet. La schématisation de la construction de la double image pupille-reflet

⁴ Axe visuel: Ligne joignant le point de fixation de l'oeil au centre de la fovéa. L'axe se déplace en fonction du mode de perception visuelle et d'une possible compensation des mouvements de la tête. L'axe optique de l'oeil est une notion différente par le fait qu'elle est définie par le centre du globe oculaire et le centre de la pupille.

est illustrée Figure 1-22 pour quatre types de mouvements combinés du système tête-oeil [MER,74].

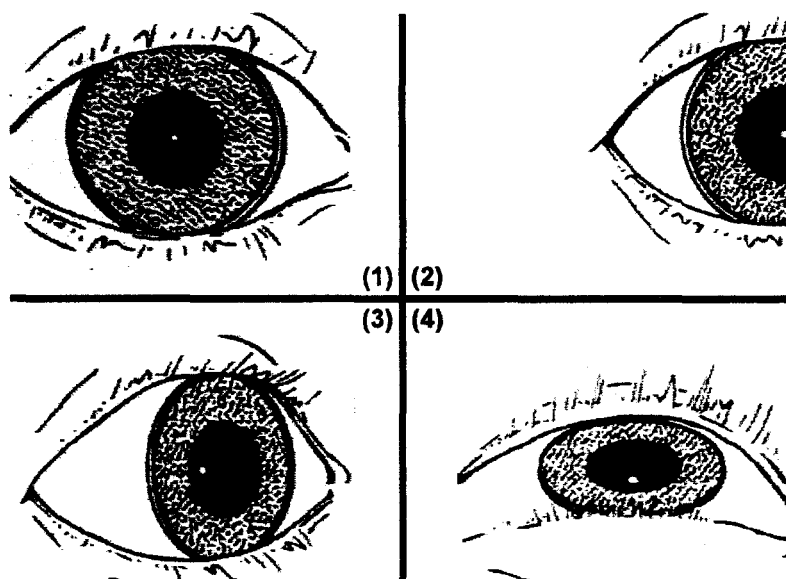


Figure 1-22 : Effets des mouvements de translation (tête) et rotation (oeil).

- (1) Regard droit devant. Coïncidence du centre de la pupille et du reflet.
- (2) Regard droit devant. La translation de la tête n'a pas affecté les positions relatives des deux centres.
- (3) Regard vers la droite. Pas de changement de la position du reflet dans l'image, par contre le centre de la pupille a suivi la translation sur la droite. La norme du vecteur cornéen n'est plus nulle on peut connaître l'angle de rotation de l'oeil.
- (4) Regard excentré vers le haut. Le vecteur cornéen est vertical. Déformation ellipsoïdale du contour de la pupille.

La technique du vecteur cornéen est insensible aux déplacements oculaires ce qui apporte une amélioration sensible sur la résolution d'un tel système comparé à la méthode du reflet cornéen. La résolution passe à quelques dizaines de minutes d'arc au prix d'une technique beaucoup plus élaborée, donc plus onéreuse.

Différents travaux sont cités par la suite en nommant les distinctions matérielles qui font leur part d'originalité.

1.4.6.1 Oculomètre distant

Ce sont les travaux initiaux de la méthode dite à « pupille brillante ». Le système expérimental est représenté Figure 1-23.

L'utilisation du système laisse une grande liberté de mouvement à l'utilisateur. Un espace toléré de 30 cm³ est possible grâce à un dispositif de suivi de l'oeil par un miroir asservi en rotation. Le patient peut avoir un retour visuel de l'image capturée par une caméra oeil.

Suivant les mouvements de la tête on observe des pertes de mesures oculaires. Pour un déplacement en profondeur le temps de récupération est de l'ordre de la seconde. Par contre pour des mouvements latéraux le temps de rétablissement est important (3.5 s en moyenne) du fait qu'il faille opérer à une recherche par le miroir, combiné d'un réglage du focus. Il y a

indétermination pendant une dizaine de secondes pour des déplacements brusques et occlusion de la pupille par les paupières fermées. Le dispositif présente une résolution de 1° d'angle de rotation de l'oeil, pour une recherche du capteur de son axe optique balayant de 0° à 20° d'élévation et $\pm 30^\circ$ d'azimut.

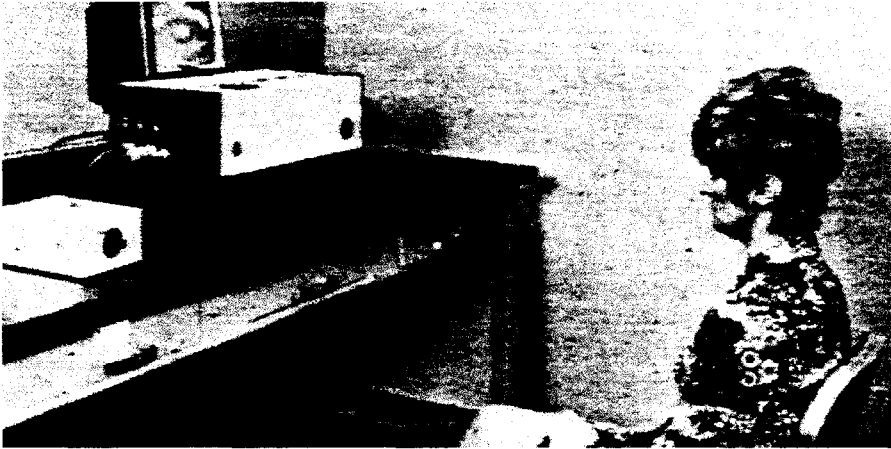


Figure 1-23 : Présentation du dispositif en milieu expérimental.

1.4.6.2 Le suiveur de regard



Figure 1-24 : Masquage partiel de la pupille par la paupière et les cils pour un déplacement oculaire vertical.

Les travaux de l'équipe U279 de l'ISERM de Lille ont repris la technique de la méthode différentielle pour des applications cliniques de recherche des zones de fixation du patient lors d'examen du champ visuel. Le problème du masquage partiel de la pupille par les paupières et les cils (Figure 1-24), les reflets parasites sur la sclérotique, le changement de taille pupillaire avec les modifications d'ambiance lumineuse les ont amenés à développer une méthodologie qui permet de s'affranchir des différents problèmes énumérés [CHA,82]⁹.



Figure 1-25 : Reflets infra-rouge du dispositif de discrimination des reflets parasites.

La Figure 1-25 présente l'image des différents reflets, du principe de discrimination d'éléments parasites. La technique permet le calcul de la direction du regard à partir de 5 reflets cornéens et de l'image de la pupille brillante.

Une version modifiée de l'appareillage clinique a vu le jour en Septembre 1986 sur l'initiative de la Cité des Sciences et de l'Industrie de la Villette à Paris intitulé « Suiveur de Regard » (Figure 1-26). Cette machine spécifique s'inscrit dans une démarche d'observation du comportement visuel individuel, de façon à montrer les mécanismes oculomoteurs qui nous

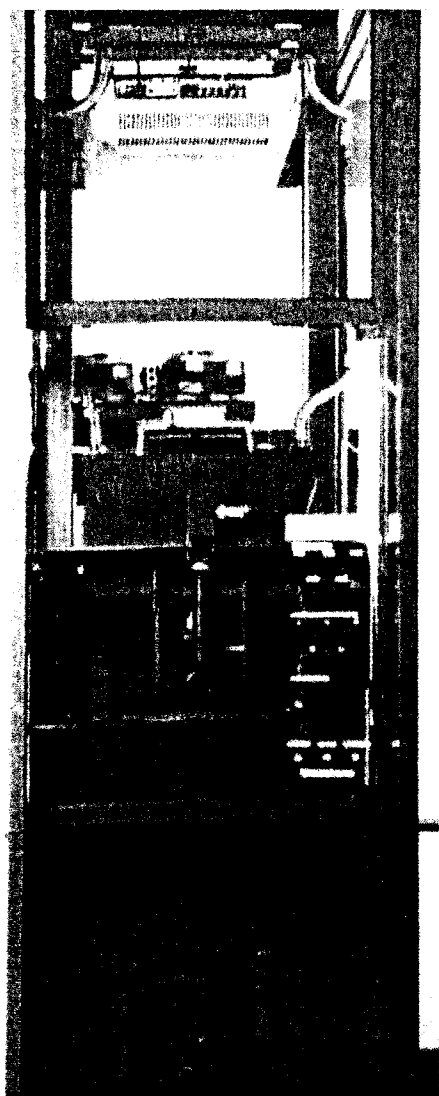


Figure 1-26 : Décomposition matérielle du Suiveur de regard.

échappent. La personne regarde une série d'images projetées sur un écran et ensuite elle observe les points de focalisation auxquels son oeil s'est attaché plus particulièrement [BUQ,88][○]. Ce système entièrement automatique et non restrictif a prouvé sa robustesse à près de 80% des cas.

Le dispositif clinique permet un enregistrement binoculaire, ce qui complique les mesures du fait du faible écart interpupillaire. Un dispositif de calibrage mécanique de prise de vue des deux caméras, la synchronisation des images, le dispositif optique permettant de prendre en compte les excentricités des mouvements verticaux font de ce dispositif un produit de grande précision [BUQ,93][○]. Le dispositif permet la mesure avec le port de verres correcteurs sans dégradations particulières grâce aux mouvements des caméras et un dispositif optique au plus près de l'oeil. La population testée qui a validé le dispositif appartient à une classe d'âge allant de 8 mois à 72 ans. Les mesures présentent une précision de 1° avec une sensibilité de 15 mn d'arc pour un champ visuel couvert de 50° dans toutes les directions. La fréquence des mesures de 60 Hz permet d'observer les phénomènes de mouvements rapides ; il faut toutefois que l'amplitude des saccades ne dépasse pas le seuil de 20°.

1.4.6.3 Dispositif ASL4000 version portable

L'oculomètre présenté ici présente l'avantage d'être une version portable. Il reprend la technique du vecteur cornéen mais l'acquisition de l'image de l'oeil s'opère par un dispositif optique monté sur un casque standard de pilotage. Une seconde caméra champ filme le repère visuel de la scène observé par l'utilisateur. La direction du regard est permise par l'extraction de points caractéristiques dans l'image champ afin de reconstruire un repère absolu. L'enregistrement des coordonnées oculaires s'effectue à l'aide d'une lame semi-réfléchissante placée devant les deux yeux, d'une caméra oeil miniature et d'un dispositif d'éclairage, le tout monté à la perpendiculaire de l'axe de vision. La caméra champ est sous la lame semi-réfléchissante, elle suit les mouvements de la tête sans gêner le champ de vision. La Figure 1-27 donne un aperçu du capteur ASL.



Figure 1-27 : Oculomètre ASL 4000.

L'encombrement réduit des capteurs CCD a permis cette position favorable qui n'introduit pas d'erreur de parallaxe. Le trajet des deux caméras en opposé est identique, l'une observe l'activité oculaire, l'autre recueille l'information visuelle du champ sur le même axe. Par contre une caméra scène montée sur le haut du casque n'aurait pas le même axe optique. L'erreur de parallaxe peut être un élément déterminant sur la qualité des mesures absolues de la direction du regard, même si la précision de l'oculomètre seul est importante [DER,91]. La sensibilité du dispositif au glissement de capteur implique des erreurs de mise en correspondance des deux acquisitions. Le traitement des mesures oculaires relatives au casque est effectué en temps réel; par contre le dépouillement des zones de fixations, s'opère en différé à partir de l'enregistrement des images caméra scène. La précision atteinte est de l'ordre du degré pour un champ couvert de $\pm 20^\circ$ horizontalement et $\pm 15^\circ$ verticalement.

Les oculomètres portables offrent une large possibilité de positionnement du regard avec un dispositif suiveur équipé sur la tête, ces outils n'ont malheureusement pas leur place dans des applications à usage intensif. On préférera donc une mobilité réduite dans les cas d'applications qui s'y prêtent, pour procurer un confort d'utilisation optimal d'une mesure sans contact, avec aucun appareillage, dans un volume restreint de mobilité de la tête. C'est le cas de l'application suivante, pour un appareillage de commande oculaire sur poste informatique où une caméra champ observe l'oeil à distance.

1.4.6.4 Eye gaze computer system ou ERICA

Le dispositif décrit ici, s'adresse à une utilisation ciblée sur micro-ordinateur pour des personnes handicapées. La caméra fixe qui recueille l'image de la pupille brillante et du reflet cornéen se situe en dessous d'un moniteur qui joue le rôle de cible visuelle sur lequel l'utilisateur sélectionne des options dans un menu [WHI,93][□]. Le système offre des possibilités multiples de commande d'appareil. La mobilité restreinte de la tête n'autorise les mesures que pour une position précise que l'utilisateur cherche à conserver par le biais d'un moniteur de contrôle de l'image champ oeil (Figure 1-29).

Le plan visuel est placé à 60 cm du visage, ce qui laisse un déplacement théorique latéral de 5 cm de la tête. La précision est indiquée par rapport à une désignation à 100% fiable pour une surface circulaire de 4 cm de rayon sur l'écran. Le dispositif optique n'a pas recours au miroir semi-transparent mais une prise directe de l'image de l'oeil avec au centre de l'objectif une LED infra-rouge permettant d'obtenir le phénomène de pupille brillante [HUT,89]⁶. La Figure 1-28 montre le dispositif optique retenu.

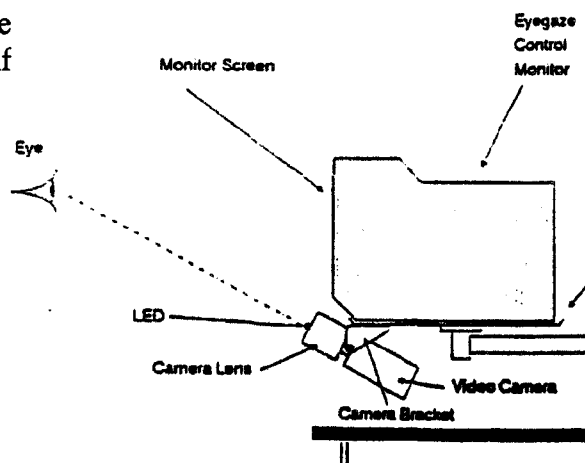


Figure 1-29 : Description matérielle de la commande oculaire ERICA.

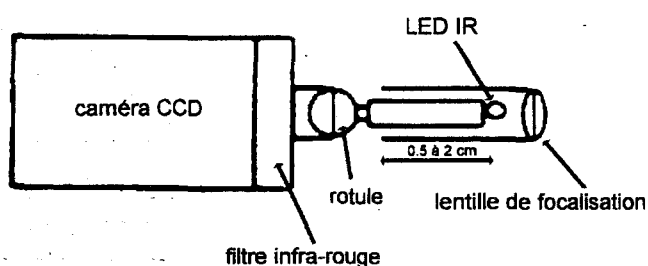


Figure 1-28 : Ensemble optique du dispositif ERICA.

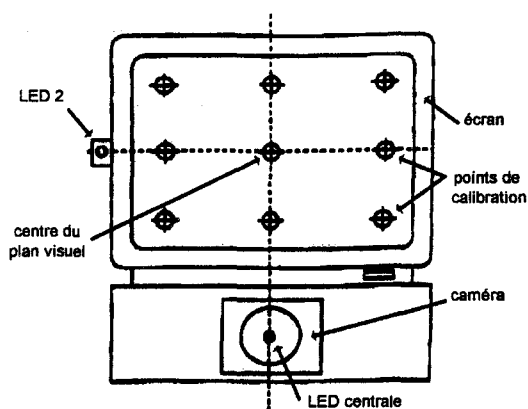


Figure 1-30 : Dispositif tenant compte de la position de la tête.

Le système ERICA est commercialisé depuis 1988 par une société américaine (LC Technologies) qui annonce une sensibilité de 0.8 cm sur l'écran disposé à 46 cm du visage mais à forte contrainte de mouvement de tête. La personne peut composer son texte par recherche de mots complets ou un mode épeler lettre par lettre [IBM,91]⁷. Les données relatives à l'équipe de recherche de L'Université de Virginie montrent une

précision moins importante. Elle propose de retenir une méthode de repérage de la tête dans le plan caméra afin de tenir compte de la non linéarité des mesures lors d'un grand déplacement de la tête excentré vis à vis du centre optique vidéo. La Figure 1-30 montre l'ajout d'une seconde LED de référence et une mire de test servant à déterminer les paramètres anatomiques de l'oeil à partir d'une phase d'initialisation. Le repérage de la tête permet par différenciation de rapporter le calcul du vecteur cornéen vis à vis d'une position spatiale de la tête.

Par l'utilisation de formules empiriques, par calibrage où il fait décrire 9 positions de références à différentes positions de la tête, il peut repositionner au mieux la direction du regard.

Sur la Figure 1-31 on voit qu'il est possible de positionner la tête à partir de la LED centrale et d'une deuxième LED de référence afin de créer un repère central du plan de

visualisation. La LED centrale intervient pour les deux mesures, celle du vecteur cornéen et celle du vecteur tête. Les améliorations apportées par le dispositif équipé de deux LEDs et la prise en compte de la non-linéarité des mesures dues à la courbure de la cornée permet de passer de 9 cases désignées à l'écran à 24 avec un taux de réussite de 85%.

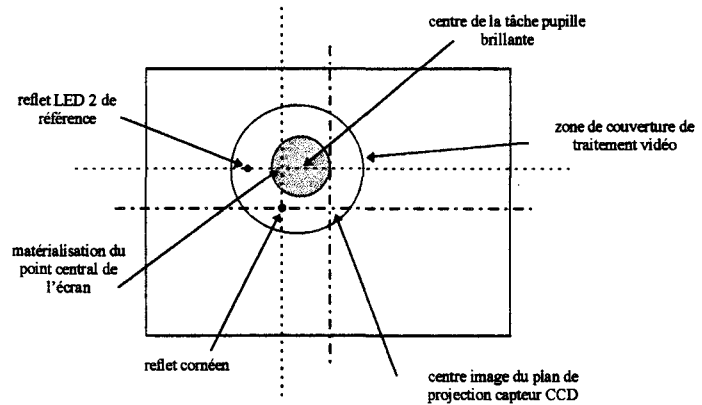


Figure 1-31 : Repérage du vecteur tête (origine centre image, extrémité intersection des axes verticaux et horizontaux passant par les deux reflets cornéens).

Un dispositif similaire à celui d'ERICA asservit la position de la caméra en rotation sur 5 zones où le traitement vidéo essaie de retrouver l'image de la pupille [EBI,91]²⁰. En moyenne il faut 2 s pour l'accrochage du dispositif qui laisse ainsi une mobilité de la tête pour un axe optique vidéo pouvant s'orienter d'un angle maximum de 15°. Le plan visuel est situé de 50 à 70 cm suivant la profondeur du déplacement de la tête par asservissement de la distance focale par un dispositif ultrasonore. La résolution atteinte est de 10 par 10 zones distinctes sur l'écran, le concepteur ne pouvant faire mieux du fait de la non-linéarité des mesures.

1.5 Conclusion

Les différentes méthodologies qui nous sont offertes se distinguent par la possibilité ou non d'une mesure absolue de la direction du regard, l'encombrement du dispositif, la tolérance du sujet à supporter les capteurs et la précision atteinte. Dans le domaine plus spécifique de l'analyse vidéo on repère deux familles d'oculomètres suivant que le capteur est monté ou non sur la tête donc autorisant ou non une mobilité conséquente de la tête.

Pour notre part, nous voulons laisser une large liberté des mouvements de la tête. La contrainte à tenir est de s'interdire toutes possibilités d'appareillages montés sur le sujet. On élimine d'office, la possibilité de l'emploi d'un capteur type ASL. On exclu par ailleurs, les techniques où les dispositifs sont basés sur des mesures référencées au crâne.

On sait qu'à partir des méthodes optiques différentielles, on peut réaliser aisément des mesures sans contact physique avec le sujet suivant l'exemple décrit au paragraphe 1.4.6.4, du suiveur de regard ERICA. Malheureusement cette application n'atteint pas la précision des oculomètres portables qui bénéficient de dispositifs optiques proches de l'oeil qui leur confèrent une précision accrue.

Suivant une disposition similaire au dispositif ERICA, la mobilité de la tête nous a conduit à retenir la méthode du reflet cornéen. Pour le procédé ERICA, il aurait fallu procéder à un asservissement mécanique de la position de la caméra de surveillance oeil, afin de conserver l'image dite de la « pupille brillante »; ce système procède à une surveillance éloignée par une caméra montée sous l'écran. Notre choix, et les expérimentations l'ont prouvé, consiste à une surveillance au plus proche de l'oeil en élargissant le champ de vision, ce qui laisse apparaître des formes nuisantes aux calculs avec les traitements d'images proposés dans les applications citées.

A nous de mettre en oeuvre les outils qui soient les plus adaptés à une mesure aisée de la direction du regard et si possible de façon complètement autonome.

Chapitre 2

2. Extraction d'informations à partir d'une image vidéo

2.1 Introduction

Ce chapitre consiste à faire état des différentes techniques de traitement vidéo, d'une image multi-niveaux de gris ou binarisée. On évalue les résultats qu'apportent les techniques de filtrage dans la détection de contours et on note les améliorations sensibles des filtres optimisés dans le cas du traitement d'images bruitées.

La liste des différents opérateurs énumérés ne se borne qu'à des traitements de convolution de l'image par des masques de dimension finie puisque la contrainte d'une détection au rythme de la cadence vidéo s'impose à nous dans la technique du reflet cornéen.

Le cas des images binarisées est étudié dans l'opération de suivi de contour par marquage ou par un traitement séquentiel d'étiquetage.

2.2 Définition d'une image bidimensionnelle

Une fonction image $I(x, y)$ est une fonction à valeurs réelles de deux variables réelles, définie à l'intérieur d'un domaine fini $D: 0 \leq I(x, y) < M, \forall x, y \in D$

La fonction image associe à chaque point une valeur appelée niveau de gris. Cette valeur dépend de la quantité de lumière réfléchie ou transmise par l'image physique en ce point pour une longueur d'onde donnée.

L'image numérique est le résultat discret de cette fonction image par un échantillonnage spatial et dynamique.

2.2.1 Echantillonnage spatial

Dans notre cas le type d'échantillonnage est un réseau rectangulaire, il est possible de rencontrer d'autres formes de réseau tel que le réseau hexagonal, triangulaire ou circulaire. On désigne l'image numérique par un tableau d'éléments *pixels* $a(i, j)$ de N lignes sur M colonnes. La valeur $a(i, j)$ représente la valeur moyenne de la fonction image sur $S(x, y)$. La

surface $S(x, y)$ est délimitée par :

$$\begin{cases} j-1 \leq y \leq j \\ \text{et} \\ i-1 \leq x \leq i \end{cases}$$

avec $a(i, j) \in \{0 \leq i < N, 0 \leq j < M\}$ (cf. Figure 2-1)

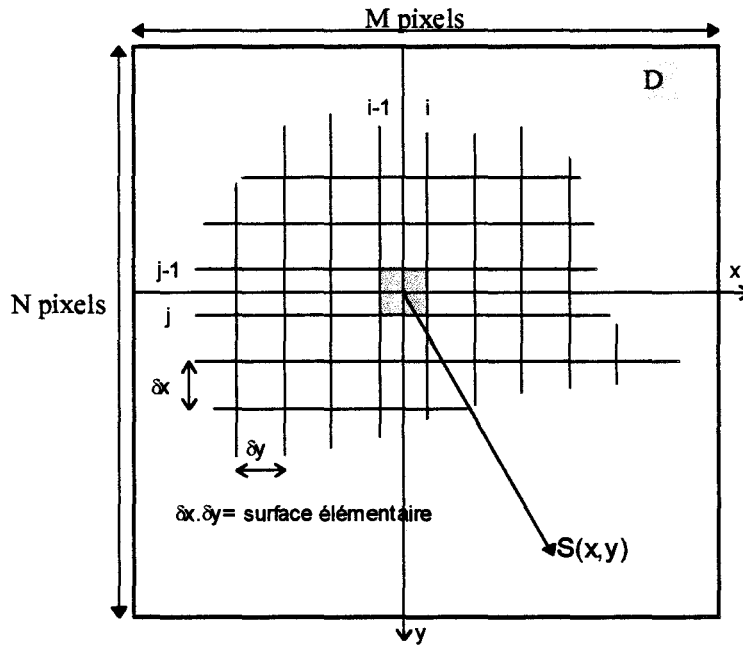


Figure 2-1 : Image découpée par le réseau rectangulaire.

2.2.2 Echantillonnage dynamique

La valeur quantifiée $a(i, j)$ représente le niveau de gris de l'image. On utilise un codage linéaire en intensité pour des niveaux de gris compris entre le blanc $a(i, j) = G - 1$ et le noir $a(i, j) = 0$. Le nombre de bits affectés au codage numérique permet d'étendre les valeurs de niveaux de gris à $2^{N_{bits}} = G$ gris différents.

2.3 Les fondements du traitement d'images numériques

La succession des traitements de l'image permet le passage d'une matrice de distribution d'intensité lumineuse vers des attributs qui identifient l'image afin de les exploiter de façon automatique pour contrôler, agir sur des commandes ou classifier.

Pour passer d'un domaine à l'autre, l'image subit des étapes de transformations suivant une chaîne de traitement .

- Echantillonnage et quantification (numériseur d'image).
- Codage en vue d'une compression de l'image pour réduire le débit de transmission ou la quantité d'informations de stockage.
- Amélioration et restauration, étapes qui vont permettre de dégager les informations à prélever (augmentation de la dynamique dans l'image).
- Segmentation et extraction de caractéristiques qui détermineront et représenteront la forme des contours. Nous serons à même de déduire la taille, la texture, le positionnement pour donner une interprétation modélisée de l'image.

Un niveau supérieur s'attache à donner une description suivant un modèle décrit sur la base de représentations symboliques, assimilable pour une cible donnée. Pour ce faire, ce traitement coordonne les informations décelées au niveau inférieur et se sert de sa base de connaissance pour évaluer complètement l'objet à identifier. Le passage d'un niveau à l'autre améliore la qualité des informations en réduisant leur volume. Les formes se révèlent à chacun des niveaux et guident vers les décisions. On peut représenter ce cheminement sous la forme d'une pyramide (Figure 2-2).

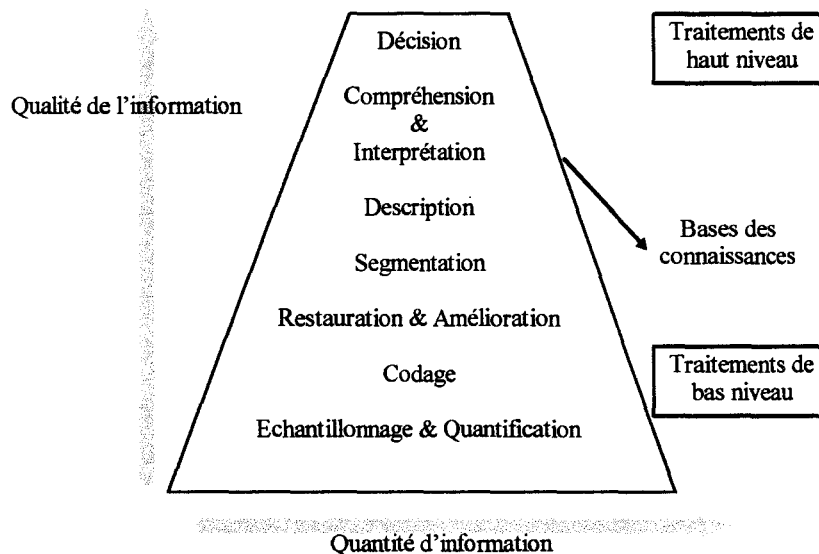


Figure 2-2 : Pyramide des traitements d'images numériques.

2.4 Les opérations de bas niveaux

On distingue dans cette classe de traitement, les opérations qui mettent en évidence les contours afin de segmenter l'image en régions distinctes.

2.4.1 Segmentation d'une image

L'objectif est de mettre au point une technique de segmentation qui extrait les contours d'une image en niveaux de gris. La génération des listes de codes modélisant les objets contenus dans l'image n'est possible que si l'on maîtrise parfaitement la détection.

La segmentation est l'étape qui crée le lien entre les traitements de bas niveau et ceux dits supérieurs. Elle permet l'extraction des différents attributs. L'image est découpée en régions distinctes suivant des caractéristiques reconnues pour engendrer une unité ciblant une forme ou un objet. Une description plus compacte que l'ensemble des pixels appartenant à cette région, facilite l'étape d'identification et de positionnement des différentes formes contenues dans l'image.

La technique de segmentation la plus simple est la binarisation. On réduit l'image à un plan de bits, selon que les pixels sont inférieurs ou supérieurs à un niveau de référence appelé seuil (voir 2.4.3). Elle est utilisée pour des images de bonne qualité, c'est à dire peu bruitées, avec un contraste important et des zones bien séparables.

Facile à mettre en oeuvre et rapide à l'exécution, la technique de binarisation ne donne pas satisfaction pour toutes les applications. De nombreuses autres méthodes, plus élaborées et

plus performantes, se caractérisent par le paramètre recherché pour isoler les régions et la méthode pour assembler les pixels qui l'englobe. Ces méthodes peuvent être distinguées selon les trois types de caractéristiques suivants :

- La classification de pixels.
- La détermination de régions.
- La détermination de contours.

2.4.1.1 La segmentation par classification de pixels

Contrairement aux deux autres techniques, la classification de pixels traite les pixels sans la notion de voisinage. Un contour est engendré par connexion de pixels adjacents. Si ce contour est fermé il représente une région.

Ce sont les caractéristiques spatiales de l'image qui classe le pixel vers tel ou tel ensemble. Cette technique fonctionne bien pour des zones homogènes de niveaux de gris suffisamment différents. La méthode est gourmande en temps de calcul du fait que chacun des pixels est comparé à l'ensemble de tous les autres pixels. Il est également nécessaire que les différentes régions de l'image soient bien attribuées vers différentes classes, gage d'une bonne séparabilité.

2.4.1.2 La segmentation par régions

Il y a prédominance de groupes de pixels adjacents ayant des caractéristiques communes. L'image est partitionnée en différents centres d'intérêts délimités par des régions. Les pixels sont traités individuellement par des relations les attachant au groupe de pixels voisins. Pour appartenir à une région le pixel doit obéir à certains critères.

Une des solutions envisageables est de diviser l'image en régions de taille définie, qui se subdivisent elles-mêmes jusqu'à parvenir à un critère d'uniformité.

Une variante, par un traitement ascendant, fusionne par succession d'étapes les régions qui sont similaires par des critères comme la recherche de distance inter-régions maximale ou similitude de niveaux de gris, jusqu'à satisfaction par un critère d'arrêt.

Une dernière famille de segmentation par régions s'appuie sur l'association d'une méthode descendante et ascendante. Le principe est d'utiliser les groupes de pixels révélés, comme noyaux pour une technique de croissance de régions. La solution est atteinte lorsque le découpage par division et fusion n'est plus possible.

2.4.1.3 La segmentation par contours

Le principe consiste à repérer les discontinuités locales dans l'image suivant des attributs offerts aux pixels afin d'engendrer les contours des régions. Plusieurs étapes sont nécessaires afin de parvenir au résultat escompté.

1. Lissage pour améliorer l'image pour réduire les contours parasites dus au bruit.
2. Révélation des contours de l'image.

3. Elimination des contours faibles par un seuillage adapté.
4. Réduction du contour à une épaisseur d'un pixel unique.
5. Chaînage des points ou segments détectés pour les attribuer à une liste en vue de former un contour unique fermé.

2.4.2 Profil des contours

Les changements ou discontinuités dans une image, telles que variation de luminance, texture sont les principales origines de distinction des objets.

Le contour d'une région est représenté par une frontière fermée un peu comme une limite topologique. Il faut donc que tous les pixels qui appartiennent à cette région soient contenus du même côté de cette ligne de démarcation. Les gabarits de ces contours sont représentés Figure 2-4.

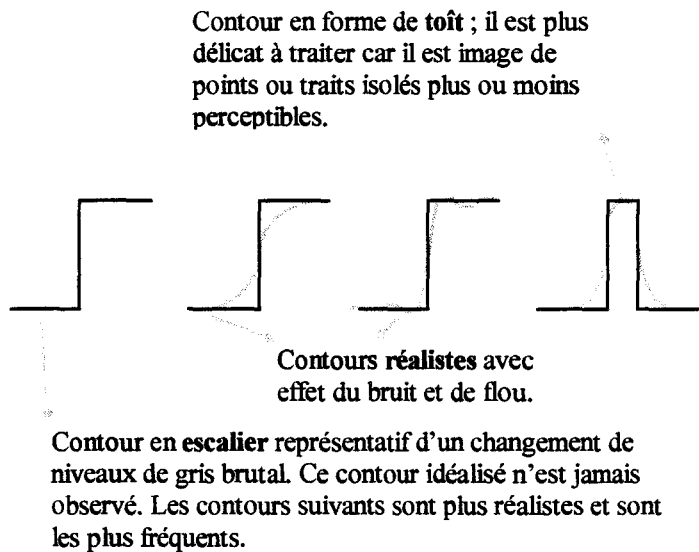


Figure 2-4 : Différents profils de contours.

Plusieurs paramètres entrent en compte dans la majeure partie des opérateurs pour lesquels le type de contour en escalier est retenu (Figure 2-3).

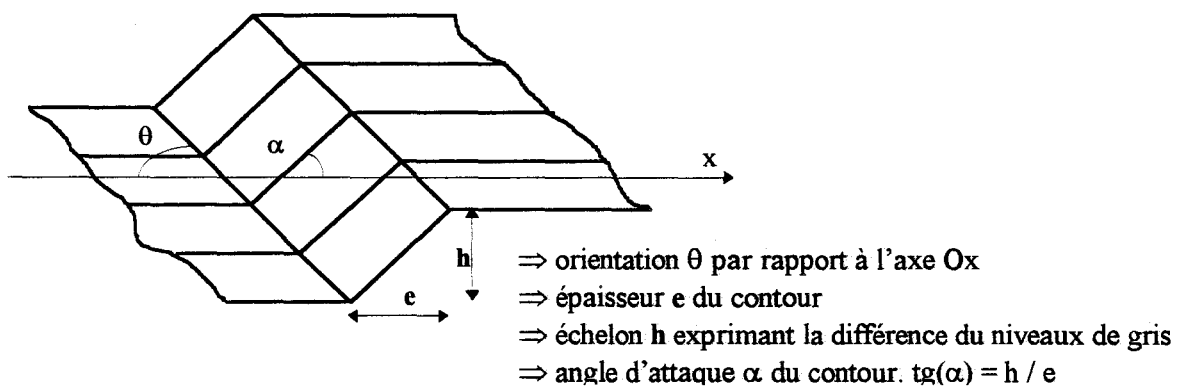


Figure 2-3 : Paramètres caractérisant un contour à deux dimensions.

D'une manière générale, on estime qu'un contour existe si le contraste est supérieur de 10% à la dynamique du signal sur plusieurs pixels et s'étendant dans la direction du contour.

2.4.3 La technique de détection par seuillage

Les contours sont des ensembles de pixels adjacents démarquant la limite tel que le niveau de gris d'un couple de pixels sur cette frontière obéit à la relation :

$$E(S) = \{[i, j], [k, l] \text{ tq } \min [h(i, j), h(k, l)] \leq S < \max [h(i, j), h(k, l)]\}$$

En d'autres termes, cela revient à repérer le couple pour lequel le niveau de gris de l'un d'eux est inférieur au seuil et que l'autre pixel soit supérieur. Afin que l'extraction soit faite au mieux, il faut placer le seuil entre le niveau des deux pixels de façon judicieuse pour que le contour généré soit le plus significatif possible. Cette technique ne peut répondre favorablement que dans le cas d'images fortement contrastées sur des zones homogènes en terme de niveau de gris.

Le résultat de cette opération donne une image intitulée *binarisée*, les éléments de sortie de la matrice image deviennent des booléens 1 ou 0 selon que le pixel appartient ou non à la région délimitée par le seuil.

2.4.4 La technique du filtrage

La mise en évidence des contours s'opère par une opération de transformation de l'image par convolution avec un filtre de type linéaire ou non linéaire.

Dans la plupart des cas, on procède par une approximation discrète de leur réponse impulsienne sous la forme d'un masque pondéré par des coefficients ajustés au mieux.

L'image étant la représentation d'un plan, certaines techniques utilisent plusieurs masques directionnels lorsque la séparation de ceux-ci est permise. Les opérateurs couramment utilisés ont été développés par Roberts, Prewitt, Kirsh, Sobel et plus récemment par Canny-Deriche.

Quand la fonction du filtre est dérivable, les coefficients des masques sont choisis pour représenter l'approximation discrète de sa dérivée au mieux.

Par contre, les dérivées partielles peuvent être approximées par différences finies pour le cas des fonctions non dérivables. L'opération de détection de contours est souvent précédée par une opération de lissage qui éliminera le bruit. Pour l'exemple d'un filtre dérivé d'un opérateur gaussien (SOBEL voir 2.4.4.1.3), on aura :

$$H(x) = L.\exp(-\alpha x^2) \text{ opérateur de lissage.}$$

$$H'(x) = -D.x.\exp(-\alpha x^2) \text{ opérateur de dérivation.}$$

Le choix d'un masque 3*3 nous impose les coefficients en $x = -1, 0$ et 1 :

$$H(-1) = H(1) = 1 \text{ et } H(0) = 2 \text{ et } H'(-1) = 1, H'(0) = 0 \text{ et } H'(1) = -1$$

Le masque qui approxime la dérivée directionnelle suivant l'axe Ox est en fait l'application d'une opération de lissage par l'opérateur $H(x)$, suivi d'une opération de

dérivation par l'opérateur $H'(y)$ dans la direction orthogonale. On retrouve bien l'expression du masque bien connu de Sobel :

$$H_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Les discontinuités d'ordre 0 sont révélées par les filtres de dérivation d'ordre premier intitulé *filtre gradient*. Ces filtres satisfont les contours de type escalier. L'approche du *filtre Laplacien* repose sur la notion de dérivée seconde.

Une description succincte des opérateurs les plus fréquemment employés est rappelée ci-dessous.

2.4.4.1 Filtre gradient

2.4.4.1.1 Opérateur gradient

La première approche possible pour détecter les variations locales de l'intensité f est une approche du type gradient. Un contour d'orientation θ au point (x,y) est détecté par un maximum de la dérivée directionnelle dans la direction θ . On a pour expression :

$$g(\theta) = \vec{\nabla} f(x,y) \cdot \vec{n}$$

où $\vec{n}$ représente le vecteur unitaire dans la direction du gradient :

$$\vec{n} = \cos(\theta)\vec{u} + \sin(\theta)\vec{v}$$

$$\text{On a donc : } g(\theta) = \cos(\theta) \frac{\partial f}{\partial x} + \sin(\theta) \frac{\partial f}{\partial y} :$$

Dans le cas d'une image numérique discrète, on approxime les dérivées directionnelles par de simples différences :

$$\frac{\partial f}{\partial x} \approx \Delta_x f(x,y) = f(x+1,y) - f(x,y)$$

$$\frac{\partial f}{\partial y} \approx \Delta_y f(x,y) = f(x,y+1) - f(x,y)$$

La norme du gradient s'exprime par les relations suivantes:

$$|\nabla f(x,y)| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

$$\text{avec un signal discret : } |\nabla f(x,y)| = \left[\Delta_x f(x,y)^2 + \Delta_y f(x,y)^2 \right]^{1/2}$$

Pour une recherche dans une direction précise, on cherche le maximum du gradient :

$$|\nabla f(x,y)| = \max(|\Delta_x f(x,y)|, |\Delta_y f(x,y)|)$$

Pour l'orientation du gradient nous avons :

$$\alpha(x, y) = \arctan\left(\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}}\right) + \frac{\pi}{2}$$

2.4.4.1.2 Opérateurs de Roberts

Cet opérateur désigne les dérivées directionnelles suivant les axes orientés à 45 degrés à l'aide de masques de convolution de dimension 2*2:

$$H0 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ et } H1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

L'image avec les contours accentués est l'image proche de la norme du gradient. Pour la construire, on opère par convolution de l'image originale par les masques directionnels :

$$f_x(x, y) = f(x, y) * H0$$

$$f_y(x, y) = f(x, y) * H1$$

L'image accentuée vaut en chaque point :

$$f_{acc}(x, y) = \left[f_x^2(x, y) + f_y^2(x, y) \right]^{1/2}$$

L'orientation du contour est donnée par :

$$\alpha(x, y) = \arctan\left(\frac{f_x(x, y)}{f_y(x, y)}\right) + \frac{\pi}{4}$$

2.4.4.1.3 Opérateurs de Prewitt et Sobel

Ici les masques de convolution approximent les dérivées directionnelles suivant x et y. Les masques sont de dimensions 3*3. Le paramètre qui les différencie est le coefficient c :

- c = 1 pour les masques de Prewitt.
- c = 2 pour l'opérateur de Sobel.

Les masques sont:

$$H0 = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \text{ et } H1 = \begin{bmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{bmatrix}$$

Ces filtres correspondent à la succession d'un lissage suivi d'une opération de dérivation. On peut décomposer les masques alors en 2 opérateurs ligne et colonne :

$$H0 = \begin{bmatrix} 1 \\ c \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \text{ et } H1 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & c & 1 \end{bmatrix}$$

L'image accentuée est le résultat de la norme du gradient approximé par convolutions des opérateurs directionnels sur l'image originale de la même façon que pour l'opérateur de Roberts :

$$f_{acc}(x,y) = [f_x^2 + f_y^2]^{1/2} \text{ et } \alpha(x,y) = \arctan\left(\frac{f_x(x,y)}{f_y(x,y)}\right)$$

2.4.4.1.4 Opérateur de Kirsh étendu à 3 et 5 niveaux

Contrairement aux opérateurs précédents où l'orientation n'est qu'une approximation, ici on détermine différents filtres directionnels adaptés à des orientations prédéfinies. Kirsh limite le nombre de masques à 8, chacun répondant à une direction préférentielle et par rotations successives de $\frac{\pi}{8}$ de l'opérateur de base $H0$.

$$H0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & 3 \\ -3 & -3 & -3 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 \\ h_7 & 0 & h_3 \\ h_6 & h_5 & h_4 \end{bmatrix}$$

Le gradient affecté au point est celui des huit masques qui aura la valeur maximale. De même l'orientation du contour est directement désignée par l'orientation privilégiée sur lequel le masque a répondu le plus favorablement.

Le masque avec ses coefficients indicés est donné par :

$$Hi = \begin{bmatrix} h_i & h_{i+1} & h_{i+2} \\ h_{i+7} & 0 & h_{i+3} \\ h_{i+6} & h_{i+5} & h_{i+4} \end{bmatrix}$$

La construction des autres masques s'obtient par permutation des membres de la matrice par la relation :

$$\{Hi_{0 \rightarrow 7} \Rightarrow h_{j_{0 \rightarrow 7}}\} = \{H0 \Rightarrow h_{(8-i+j) \text{ modulo } 8}\}$$

Les opérateurs de 3 et 5 niveaux fonctionnent sous le même principe. Ils se distinguent chacun par la valeur donnée à l'indice c :

- $c = 1$ approche dite de 3 niveaux.
- $c = 2$ extension à l'opérateur de 5 niveaux.

$$H0 = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 \\ h_7 & 0 & h_3 \\ h_6 & h_5 & h_4 \end{bmatrix}$$

et de même :

$$\{Hi_{0 \rightarrow 7} \Rightarrow h_{j_{0 \rightarrow 7}}\} = \{H0 \Rightarrow h_{(8-i+j) \bmod 8}\}$$

2.4.4.2 Filtre Laplacien

On décrit l'opérateur de type Laplacien suivant la relation :

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial n^2} + \frac{\partial^2 f(x,y)}{\partial n_{\perp}^2}$$

où n et $n_{\perp}$ définissent 2 directions orthogonales.

Si on prend les directions définies dans le système d'axes Ox et Oy , on retombe sur l'expression classique :

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

L'approximation discrète la plus simple du Laplacien est donnée par la fonction suivante :

$$L(x,y) = f(x,y) - \frac{1}{4} [f(x,y+1) + f(x,y-1) + f(x+1,y) + f(x-1,y)]$$

Son masque correspondant de taille 3*3 vaut:

$$H4 = \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

On cherche le passage par 0 de cette fonction afin d'obtenir les points qui distinguent le contour. Une opération de seuillage sur la norme du gradient est nécessaire afin d'éliminer les contours parasites dus au bruit dans l'image. L'inconvénient majeur de cet opérateur est sa grande sensibilité au bruit. Pour pallier au problème du bruit, on réalise un lissage de l'image en éliminant tous les détails de taille inférieure à un niveau critique.

2.4.4.3 Filtre Gaussien

A titre d'exemple on citera l'approche de Mahrr et Hildreth [MAR,80][©] parmi les nombreuses techniques de lissage (Filtre moyenne mobile, opérateur Nagdif, filtre médian et Filtre gradient inverse).

La méthode consiste en une transformation préalable de l'image par l'opérateur isotrope Gaussien $G(x,y)$ suivi d'une localisation des passages par 0 du Laplacien.

La distribution Gaussienne bidimensionnelle de moyenne nulle et d'écart type σ est :

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp \left(-\frac{x^2 + y^2}{2\sigma^2} \right)$$

et l'opérateur Laplacien d'une Gaussienne : $\nabla^2 G(x, y)$

D'où :

$$\nabla^2 G(r) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{r^2}{2\sigma^2} \right) \exp \left(-\frac{r^2}{2\sigma^2} \right)$$

$$\text{avec: } r^2 = x^2 + y^2$$

Ce filtre peut être mis sous la forme de filtres séparables. Chacun des filtres étant le produit d'une Gaussienne sur une dimension par sa dérivée seconde sur la deuxième dimension. Ce changement de disposition permet une mise en oeuvre de l'opérateur avec un nombre plus réduit d'opérations. L'opérateur Laplacien d'une Gaussienne devient alors :

$$\nabla^2 G(x, y) = g(x)g''(y) + g(y)g''(x)$$

Une troisième approche plus approfondie revient à définir différents critères d'optimisation sur la détection. Cette technique tend à se placer en bonne place dans l'extraction de contour au prix d'un temps de calcul souvent important et une complexité de mise en oeuvre accrue pour que ce traitement puisse s'opérer à la cadence du signal vidéo (contraintes des filtres récurrents).

2.4.4.4 Filtre optimal

Parmi les différentes approches par optimisation, celle développée par Canny semble tenir une place de référence incontestée; c'est pourquoi on y portera une attention toute particulière.

2.4.4.4.1 Filtre deCanny

Soit un signal monodimensionnel $I(x)$ d'échelon A situé à l'abscisse x_0 , noyé dans un bruit Gaussien $n(x)$ de moyenne nulle et de variance $\sigma^2 = E(n^2(x))$ (densité spectrale de puissance du bruit blanc).

Soit $R(x_0)$ la réponse au point x_0 de la convolution du signal d'entrée avec un opérateur de détection $D(x)$:

$$R(x_0) = \int_{-\infty}^{+\infty} I(-x) D(x_0 - x) dx$$

Le problème suivant Canny revient à trouver le détecteur tel que la réponse soit maximum sous trois contraintes précises [CAN,86][○]:

- Bonne détection (1).
- Bonne localisation (2).
- Non multiplicité des réponses (3).

(1) Il faut chercher le détecteur tel que le rapport signal sur bruit RSB soit maximum :

$$RSB = \frac{\left| \int_{-\infty}^{+\infty} I(-x) D(x) dx \right|}{\sqrt{\int_{-\infty}^{+\infty} n_0^2 D^2(x) dx}}$$

On cherche $\Sigma = \frac{A \int_{-\infty}^0 D(x) dx}{n_0 \sqrt{\int_{-\infty}^{+\infty} D^2(x) dx}}$ de façon à être maximal.

⇒ La réponse du filtre à l'abscisse $x = 0$ au signal échelon est :

$$R(0) = A \int_{-\infty}^0 D(x) dx + \underbrace{0 \int_0^{+\infty} D(x) dx}_{\text{terme nul pas de signal pour } x < 0}$$

⇒ Le bruit est supposé additif, blanc, gaussien et centré; on a donc pour valeur quadratique moyenne de la réponse du filtre au bruit :

$$\int_{-\infty}^{+\infty} D^2(x) n_0^2 dx = n_0^2 \int_{-\infty}^{+\infty} D^2(x) dx$$

D'où l'expression de Σ .

(2) Il faut minimiser la variance σ^2 de la position des passages à 0, donc maximiser la localisation :

$$\sigma_{x_0}^2 = E[x_0^2] = \frac{\int_{-\infty}^{+\infty} n_0^2 D'^2(x) dx}{\left| \int_{-\infty}^{+\infty} I'(-x) D'(x) dx \right|^2}$$

$$\text{D'où } \Lambda = \frac{A |D'(0)|}{n_0 \sqrt{\int_{-\infty}^{+\infty} D'^2(x) dx}} = \sqrt{\frac{1}{\sigma_{x_0}^2}}$$

⇒ La position d'un contour correspond à un maxima local de la réponse du détecteur ce qui revient à chercher le passage par 0 de la dérivée de la réponse. En absence de bruit la réponse se fait naturellement à l'abscisse $x = 0$. En présence de bruit, nous aurons un glissement pour un point proche de $x = 0$. On peut écrire par simplification :

$$D'(x_0)^{\text{contour seul}} + D'(x_0)^{\text{bruit seul}} = 0$$

La séparation des deux réponses nous amène à cette nouvelle relation par un développement local :

$$\underbrace{D'(0)^{\text{echelon}}}_{=0} + D''(0)^{\text{echelon}} \cdot x_0 + D'(x_0)^{\text{bruit}} = 0$$

⇒ L'expression de la distance devient :

$$x_0 = -\frac{D'(x_0)^{\text{bruit}}}{D''(0)^{\text{echelon}}} = \frac{H'_{\text{bruit}}(x_0)}{H''_{\text{echelon}}(0)}$$

$$\text{Donc : } E[x_0^2] = \frac{E[H'_{\text{bruit}}^2(x_0)]}{H''_{\text{echelon}}^2(0)}$$

On peut écrire l'expression de la valeur quadratique moyenne de la dérivée de la réponse au bruit (valeur gaussienne) par :

$$E[H'_{\text{bruit}}^2(x_0)] = n\sigma^2 \int_{-\infty}^{+\infty} D'^2(x) dx$$

Le résultat de la dérivée seconde du produit de convolution donne :

$$H''_{\text{echelon}}(0) = \int_{-\infty}^{+\infty} I'(-x) D'(x) dx = AD'(0) \text{ car convolution par un pulse de Dirac.}$$

On retrouve donc bien la relation du critère de localisation Λ .

(3) En présence de bruit, le signal détecté peut présenter plusieurs maxima et les réponses parasites doivent être éloignées de la réponse principale. Contraindre à la limitation du nombre de maxima-locaux détectés pour un même contour revient alors à maximiser la distance moyenne. Le critère de suppression des réponses multiples devient :

$$x_{\max} = 2\pi \sqrt{\frac{\int_{-\infty}^{+\infty} D'^2(x) dx}{\int_{-\infty}^{+\infty} D''^2(x) dx}}$$

Canny propose la maximisation du produit $\Sigma\Lambda$ sous la contrainte que $x_{\max}$ soit une constante arbitraire. Ce qui revient à trouver la solution de l'équation différentielle :

$$2D(x) - 2\lambda_1 D''(x) + 2\lambda_2 D''''(x) + \lambda_3 = 0$$

Nous avons comme solution générale :

$$D(x) = a_1 \exp(\alpha x) \sin(\omega x) + a_2 \exp(\alpha x) \cos(\omega x) + a_3 \exp(-\alpha x) \sin(\omega x) + a_4 \exp(-\alpha x) \cos(\omega x)$$

L'opérateur défini sur l'intervalle $[-W, +W]$ répond aux conditions aux limites suivantes :

$$D(0) = 0, D(W) = 0, D'(0) = S, D'(W) = 0$$

S est la pente à l'origine. Les termes α , ω sont positifs.

Il est à remarquer que la fonction est impaire et donc la solution est étendue aux x négatifs avec : $D(x) = -D(-x)$.

2.4.4.4.2 Filtre de Deriche

Contrairement à l'approche de Canny d'un détecteur sous forme d'un filtre à réponse impulsionnelle finie (RIF), Deriche a développé un opérateur optimal sous forme d'un filtre à réponse impulsionnelle infinie (RII) [DER,87]. L'opérateur devient alors :

$$D(x) = \frac{S}{\omega} \exp(-\alpha |x|) \sin(\omega x)$$

Il en découle les expressions suivantes du critère de détection et de localisation :

$$\Lambda = \sqrt{2\alpha}$$

$$\Sigma = \sqrt{\frac{2\alpha}{\alpha^2 + \omega^2}} = \sqrt{\frac{2m}{\omega} \left(\frac{1}{m^2 + 1} \right)}$$

avec $\alpha = m\omega$.

On obtient l'indice de performance : $\Sigma\Lambda = 2\sqrt{\frac{m^2}{m^2 + 1}}$.

Le cas limite pour $\omega \rightarrow 0$ correspondant à $m \gg 1$ offre le meilleur indice de performance $\Sigma\Lambda = 2$.

On a l'expression de l'opérateur optimal qui devient alors :

$$D_{\omega \rightarrow 0}(x) = S.x.\exp(-\alpha |x|)$$

2.4.4.4.3 Shen et Castan

Cette approche s'inspire du formalisme de Canny. Elle consiste en la recherche d'un filtre passe bas optimal pour la détection des contours. Le modèle de contour est celui d'un saut d'amplitude bruité et un critère à optimiser est alors dérivé [SHE,86][○].

Le filtre de lissage exponentiel se présente sous la forme:

$$S(x) = a \exp(-\alpha |x|)$$

Son implantation récursive, en deux temps, est donnée par les relations :

$$y_1(n) = x(n-1) + \exp(-\alpha) y_1(n-1) \quad n: 1 \rightarrow M$$

$$y(n) = y_1(n+1) + \exp(-\alpha) y(n+1) \quad n: M \rightarrow 1$$

La méthode consiste à filtrer l'image par l'opérateur de lissage exponentiel. Le résultat est soustrait de l'image d'origine pour aboutir à une approximation du Laplacien. On binarise l'image en mettant les pixels à 1 aux valeurs positives et 0 ailleurs. Les frontières des alternances positives et négatives forment les régions. Une discrimination des taches inférieures à un seuil permet d'éliminer les points isolés. Pour isoler les contours parasites dus au bruit, on calcule pour chaque passage par 0 la moyenne des niveaux de gris de l'image origine correspondant aux zones 1 et 0 de l'image binarisée.

2.4.5 Implantation récursive de l'opérateur optimal de Deriche

Le but recherché est de mettre l'opérateur de Deriche sous forme d'un filtre qui nous économise du temps de calcul en réduisant considérablement le nombre d'opérations élémentaires (multiplication et addition).

2.4.5.1 Décomposition sous forme d'opérateurs récursifs de niveaux 2

La mise en oeuvre récursive du filtre optimal est obtenue en appliquant les techniques de la transformée en Z. La relation qui permet ce passage dans le domaine des Z est :

$$F(Z) = \sum_{n=-\infty}^{+\infty} f(n)Z^{-n}$$

On cherche à rendre le filtre causal en décomposant le filtre en une somme de 2 séquences causales telles que :

$$f(n) = f_-(n) + f_+(n)$$

Les expressions de ces deux fonctions sont :

$$f_-(n) = \begin{cases} 0 & n \geq 0 \\ S.n.\exp(\alpha n) & n < 0 \end{cases}$$

$$f_+(n) = \begin{cases} S.n.\exp(-\alpha n) & n \geq 0 \\ 0 & n < 0 \end{cases}$$

D'après les propriétés de la transformée en Z on obtient : $F(Z) = F_-(Z) + F_+(Z^{-1})$

Par décomposition à l'aide de table de passage, on arrive aux expressions :

$$F_+(Z^{-1}) = \frac{aZ^{-1}}{1 + b_1Z^{-1} + b_2Z^{-2}}, \quad F_-(Z) = \frac{-aZ}{1 + b_1Z + b_2Z^2}$$

Les coefficients sont : $a = S \exp(-\alpha)$, $b_1 = -2 \exp(-\alpha)$, $b_2 = \exp(-2\alpha)$

Les fonctions de transfert de filtres récursifs opèrent chacun dans sa direction privilégiée. La réponse à l'opérateur de Deriche [DER,90] sous forme récursive est permis par les structures suivantes :

$$y_1(n) = x(n-1) + 2 \exp(-\alpha) y_1(n-1) - \exp(-2\alpha) y_1(n-2) \quad n: 1 \rightarrow M$$

$$y_2(n) = x(n+1) + 2 \exp(-\alpha) y_2(n+1) - \exp(-2\alpha) y_2(n+2) \quad n: M \rightarrow 1$$

$$y(n) = S \exp(-\alpha) [y_1(n) - y_2(n)] \quad n: 1 \rightarrow M$$

La constante S est obtenue par la contrainte de normalisation :

$$\sum_{n=0}^{+\infty} f(n) = - \sum_{n=-\infty}^0 f(n) = -1$$

On trouve ainsi :
$$S = \frac{(1 - \exp(-\alpha))^2}{\exp(-\alpha)}$$

En suivant la même démarche on peut obtenir les membres récursifs de la primitive $h(n)$, filtre de lissage de Deriche :

$$h(n) = k(\alpha |n| + 1) \exp(-\alpha |n|)$$

$$y_1(n) = k[x(n) + \exp(-\alpha)(\alpha - 1)x(n-1)] + 2\exp(-\alpha)y_1(n-1) - \exp(-2\alpha)y_1(n-2) \quad n:1 \rightarrow M$$

$$y_2(n) = k[\exp(-\alpha)(\alpha + 1)x(n+1) - \exp(-2\alpha)x(n+2)] + 2\exp(-\alpha)y_2(n+1) - \exp(-2\alpha)y_2(n+2) \quad n:M \rightarrow 1$$

$$y(n) = [y_1(n) + y_2(n)] \quad n:1 \rightarrow M$$

La constante k est obtenue par la contrainte de normalisation : $\sum_{n=-\infty}^{+\infty} h(n) = 1$

On obtient pour expression : $k = \frac{(1 - \exp(-\alpha))^2}{1 + 2\alpha \exp(-\alpha) - \exp(-2\alpha)}$

2.4.5.2 Conséquences de la récursivité sur le nombre d'opérations élémentaires

L'importance de ces structures récursives tient au fait que l'on réduit considérablement le nombre d'opérations. Contrairement au filtre RIF la structure récursive reste inchangée quel que soit la valeur du paramètre α . A titre d'exemple pour une implémentation sur 16 bits de l'opérateur de détection sous forme RIF pour α valant 0.25, il faut 105 opérations par point au lieu de 5 pour le filtre RII. Dans le cas de l'opérateur de lissage il faut 113 opérations par point au lieu de 8 dans le cas de l'implantation récursive.

Les filtres transversaux opérant de gauche à droite sont représentés Figure 2-5.

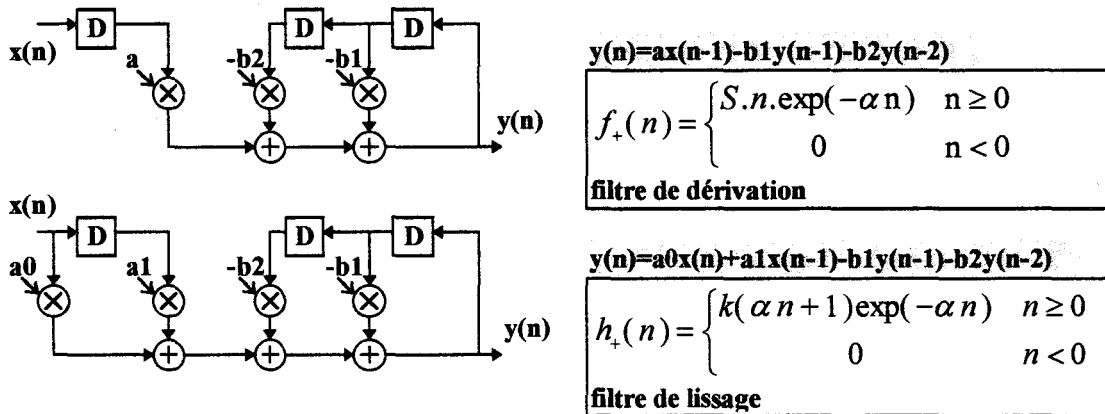


Figure 2-5: Opérateur RII filtre de dérivation et de lissage.

2.4.5.3 Extension au cas d'un traitement 2D

Deriche [DER,90] propose de généraliser son détecteur au cas du traitement d'une image sous les deux dimensions sous la forme d'une implantation récursive sous la base des opérateurs vus précédemment. On crée un masque 2D, produit du détecteur en x avec un opérateur de projection en y.

Pour détecter les contours d'une image bruitée, on opère par le calcul du gradient de l'image lissée $IL(x, y)$ suivant les deux directions orthogonales x et y suivant la relation :

$$\nabla(x, y) = \left(\frac{\partial IL(x, y)}{\partial x}, \frac{\partial IL(x, y)}{\partial y} \right) = (SS_x(x, y), SS_y(x, y))$$

L'opérateur de lissage 2D est mis sous la forme du produit de deux filtres :

$$SS(x, y) = h(x).h(y) = k_2(\alpha|x| + 1)\exp(-\alpha|x|)k_2(\alpha|y| + 1)\exp(-\alpha|y|)$$

D'où l'expression de l'image lissée sous les deux directions par convolution de l'opérateur lissage 2D avec l'image bruitée :

$$IL(x, y) = I(x, y) \otimes SS(x, y) = I(x, y) \otimes (h(x).h(y))$$

Lorsque l'on calcule les gradients directionnels en x et y, on remarque qu'en deux passages d'un masque de convolution 2D on obtient le résultat recherché. La décomposition suivante traite de l'opérateur horizontal, il en est de même pour l'opérateur vertical.

$$\begin{aligned} SS_x(x, y) &= \frac{\partial}{\partial x}(I(x, y) \otimes SS(x, y)) = I(x, y) \otimes \frac{\partial}{\partial x} SS(x, y) \\ &= I(x, y) \otimes \frac{\partial}{\partial x}(h(x)h(y)) = I(x, y) \otimes f(x) \otimes h(y) \end{aligned}$$

Les expressions des gradients directionnels sont :

$$SS_x(x, y) = k_1 x \exp(-\alpha|x|).k_2(\alpha|y| + 1)\exp(-\alpha|y|)$$

$$SS_y(x, y) = k_2(\alpha|x| + 1)\exp(-\alpha|x|).k_1 y \exp(-\alpha|y|)$$

La norme du gradient vaut pour tout point de l'image :

$$|\nabla(x, y)| = \sqrt{SS_x^2(x, y) + SS_y^2(x, y)}$$

La direction du gradient est obtenue par le quotient du gradient directionnel horizontal sur le gradient directionnel vertical :

$$\alpha(x, y) = \arctan\left(\frac{SS_x(x, y)}{SS_y(x, y)}\right)$$

La mise oeuvre de la détection de contours dans une image bruitée s'exécute par les étapes suivantes (Figure 2-6) :

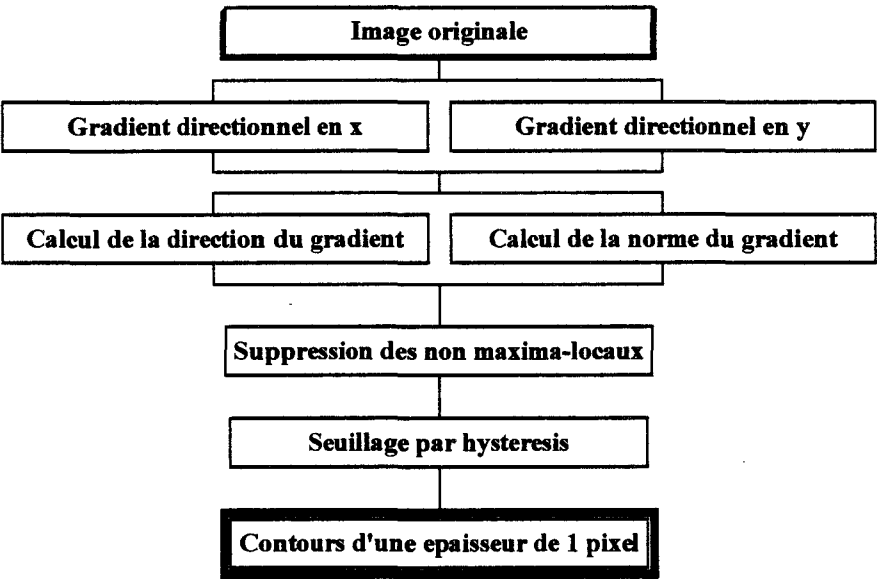


Figure 2-6 : Synoptique de la détection de contours par les opérateurs de Deriche.

La suppression des non maxima-locaux s'effectue sur un voisinage 3*3 du plan de la norme des gradients directionnels. La norme du gradient dans la direction déduite du rapport des gradients directionnels est interpolée (Figure 2-17) avec les éléments situés de part et d'autre du point central. Si la norme du gradient interpolé est inférieure à celle du point central, alors on retient le gradient dans le plan gradient sinon on met cet élément à 0.

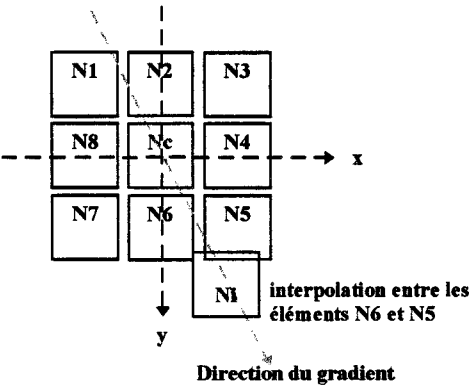


Figure 2-7 : Interpolation du gradient.

Configuration du filtrage récursif appliqué à chaque ligne de l'image :

$$y_1(m,n) = a_1x(m,n) + a_2x(m,n-1) + b_1y_1(m,n-1) + b_2y_1(m,n-2)$$
$$n:1 \rightarrow N, m:1 \rightarrow M$$

$$y_2(m,n) = a_3x(m,n+1) + a_4x(m,n+2) + b_1y_2(m,n+1) + b_2y_2(m,n+2)$$
$$n:N \rightarrow 1, m:1 \rightarrow M$$

$$r(m,n) = c_1(y_1(m,n) + y_2(m,n)) \quad n:1 \rightarrow N, m:1 \rightarrow M$$

Puis sous la même configuration avec des coefficients différents, on opère pour chaque colonne de l'image filtrée par la première passe.

$$y_1(m,n) = a_5r(m,n) + a_6r(m-1,n) + b_1y_1(m-1,n) + b_2y_1(m-2,n)$$
$$m:1 \rightarrow M, n:1 \rightarrow N$$

$$y_2(m,n) = a_7r(m+1,n) + a_8r(m+2,n) + b_1y_2(m+1,n) + b_2y_2(m+2,n)$$
$$m:M \rightarrow 1, n:1 \rightarrow N$$

$$y(m,n) = c_2(y_1(m,n) + y_2(m,n)) \quad m:1 \rightarrow M, n:1 \rightarrow N$$

Le tableau ci-dessous permet d'adapter à chaque opération souhaitée la valeur des coefficients.

Lissage 2D	Dérivée directionnelle en x	Dérivée directionnelle en y
$a_1 = a_5 = k$ $a_2 = a_6 = k(\alpha - 1)\exp(-\alpha)$ $a_3 = a_7 = k(\alpha + 1)\exp(-\alpha)$ $a_4 = a_8 = -k\exp(-2\alpha)$ $c_1 = c_2 = 1$	$a_1 = a_4 = 0$ $a_2 = -a_3 = 1$ $a_5 = k$ $a_6 = k(\alpha - 1)\exp(-\alpha)$ $a_7 = k(\alpha + 1)\exp(-\alpha)$ $a_8 = -k\exp(-2\alpha)$ $c_1 = -(1 - \exp(-\alpha))^2$ $c_2 = 1$	$a_1 = k$ $a_2 = k(\alpha - 1)\exp(-\alpha)$ $a_3 = k(\alpha + 1)\exp(-\alpha)$ $a_4 = -k\exp(-2\alpha)$ $a_5 = a_8 = 0$ $a_6 = -a_7 = 1$ $c_1 = 1$ $c_2 = -(1 - \exp(-\alpha))^2$

2.5 Codage en chaîne des contours d'une image binaire

Le principe est de chaîner les éléments qui composent une forme décrite suivant des primitives (exemple: ensemble connexe de pixels voisins deux à deux).

2.5.1 Définitions générales du traitement binaire d'une image

2.5.1.1 Termes généraux

- **Pixel** : Élément booléen non divisible constituant de base de l'image. Il appartient à l'ensemble $\{0,1\}$. Dans une terminologie de région, le pixel est *objet*, c'est à dire qu'il vaut 1 (*true*). Dans le cas contraire le pixel est *fond*, il vaut 0 (*false*).
- **Région** : Groupe de pixels voisins connectés par une ou plusieurs propriétés communes.
- **Courbe frontière** : Courbe fermée délimitant deux régions, l'un objet et l'autre fond.
- **Courbe frontière objet** : Courbe fermée particulière qui, parcourue dans le sens horaire, voit la région objet à sa droite.
- **Courbe frontière évidemment** : Même propriété que la définition précédente mais duale. Le parcours est déterminé dans le sens trigonométrique.
- **Fond** : Groupe de pixels *fond* interne à un objet.
- **Fond primaire** : Région particulière composée de pixels *fond*, délimitée dans le cadre de l'image.
- **Objet** : Lorsque la courbe frontière *objet* est commune au fond primaire alors tout son intérieur représente l'objet.
- **Objet interne** : Distinction supérieure à la notion précédente. C'est l'intérieur de la courbe frontière *objet* qui touche un fond.
- **Evidement** : Intérieur d'une courbe frontière d'évidement.
- **Forme** : Désignation globale indifféremment pour un objet, un objet interne ou un évidement.

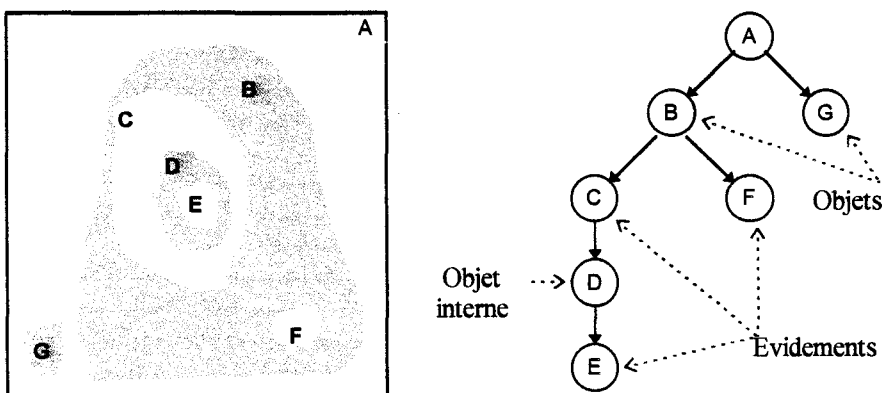


Figure 2-8 : Exemple de formes imbriquées. L'arbre permet de connaître le niveau d'imbrication pour chacune des formes.

On peut ordonner les différentes formes d'un objet au travers d'un arbre qui exprime les relations d'imbrications. Par exemple, un objet interne est toujours contenu dans un évidement, la Figure 2-8 montre cette notion illustrée par son arbre.

2.5.1.2 L'imbrication

Cette notion permet de définir l'existence d'une forme par une autre qui lui est parent. On a imbrication lorsque le passage d'une forme $T_i(n)$ d'un objet d'indice n au fond primaire nécessite le passage inévitable par au moins une seconde forme $T_k(n)$. La relation d'imbrication met en avant le degré de filiation d'une forme dans un objet par le nombre de passages de frontières parents nécessaires pour atteindre le fond.

2.5.2 Description d'une image décrite à partir d'une séquence de scrutation

On décrit l'image suivant un balayage de gauche à droite et de ligne en ligne sur le domaine complet de l'image numérisée.

2.5.2.1 Le voisinage de pixels en huit voisins

Cette notion fait appel à une connexion de deux pixels par un voisinage immédiat. Suivant les cas et les opérateurs de description utilisés, on opère par un voisinage uniquement horizontal ou vertical (voisinage en 4 voisins) ou l'on autorise également la connexion oblique (voisinage en 8 voisins) voir Figure 2-9. Le pixel positionné en abscisse et ordonnée d'indice ligne n et colonne m $P(m,n)$ a pour voisin $p(i,j)$ dans le cas restrictif où :

$dPp = \max(|i-m|, |j-n|) = 1$ et que $P(m,n) = p(i,j)$ soit : deux pixels objets ou deux pixels fonds, adjacents deux à deux.

on note dans ce cas: $Voi_{xy}[P(m,n), p(i,j)] = 1$

De même $Voi_x[P(m,n), p(i,n)] = 1$ implique une adjacence sur la même ligne.

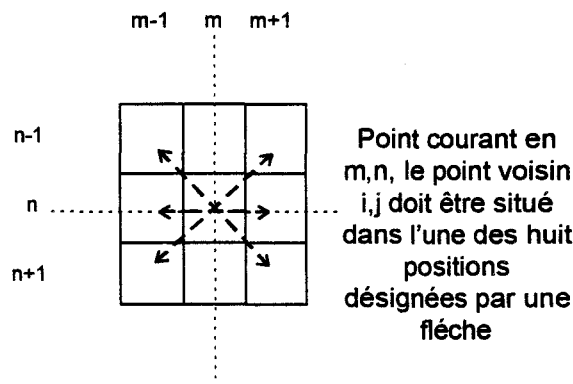


Figure 2-9 : Espace d'un voisinage de huit voisins.

2.5.2.2 Le segment

Le segment est une description « ligne » d'une séquence ininterrompue de pixels objets. On a :

$S_l(o) = \left\{ \xrightarrow{c:0 \rightarrow L-1} \text{Voi}_x[P(m+c,n), p(m+c+1,n)] = 1 \right\}$ avec $P(m,n)$, le premier élément du segment.

Le segment $S_l(o)$ de l'objet d'indice o est l'ensemble des points voisins deux à deux sur la ligne n sur une largeur L de pixels objets. Cela revient à écrire :

$$S_l(o) = \{ P(m,n) / \exists p(i,n) \in S_l(o) \text{ et } |i-m|=1 \}$$

Le segment est délimité à ses deux extrémités par un pixel départ $d_l(o)$ et un pixel fin $f_l(o)$.

Pixel départ: $d_l(o) = \text{Voi}_x[P(m,n), p(m-1,n)] = 0 \text{ et } P(m,n) \in S_l(o)$

ou : $d_l(o) = \{ P(m,n) / \nexists p(m-1,n) \in S_l(o) \}$

Pixel fin: $f_l(o) = \text{Voi}_x[P(m,n), p(m+1,n)] = 0 \text{ et } P(m,n) \in S_l(o)$

ou : $f_l(o) = \{ P(m,n) / \nexists p(m+1,n) \in S_l(o) \}$

2.5.2.3 La forme

C'est l'ensemble des segments adjacents deux à deux, donc appartenant à un même objet sur la hauteur de son contour projeté sur l'axe des ordonnées.

La relation d'adjacence est vérifiée par les relations suivantes :

$$\begin{cases} A_x[d_v(o)] \leq A_x[f_w(o)] + 1 \\ \text{ou} & \text{et } |O_y[S_w(o)] - O_y[S_v(o)]| = 1 \\ A_x[d_w(o)] \leq A_x[f_v(o)] + 1 \end{cases}$$

avec $A_x[P(m,n)] = m$ et $O_y[P(m,n)] = n$

On exprime par les relations $A_x[]$ et $O_y[]$ respectivement l'abscisse et l'ordonnée d'un élément du segment.

Si la relation d'adjacence est réelle, on a: $\text{Adj}[S_v(o), S_w(o)] = 1$

La forme est l'ensemble des segments adjacents deux à deux. On aura dans ce cas l'expression suivante :

$$T_l(o) = \{ S_v(o) / \exists S_w(o) \in T_l(o) \text{ et } \text{Adj}[S_v(o), S_w(o)] = 1 \}$$

Ces termes permettront de décrire un objet à partir d'un traitement séquentiel où l'objet se révèle par morceaux de formes qui s'imbriquent au rythme d'un balayage de gauche à droite et de haut en bas dans l'image. Les segments rattachés aux objets peuvent être également

représentatifs d'un évidement lorsque ceux-ci sont composés d'éléments pixels *fond*. On définit donc les mêmes structures pour le voisinage de pixels *fonds*. Ce qui nous amène à indiquer l'apparition des segments, segments *évidements* ou segments *objets*. Un exemple de décomposition (Figure 2-10) permet d'illustrer ces propos.

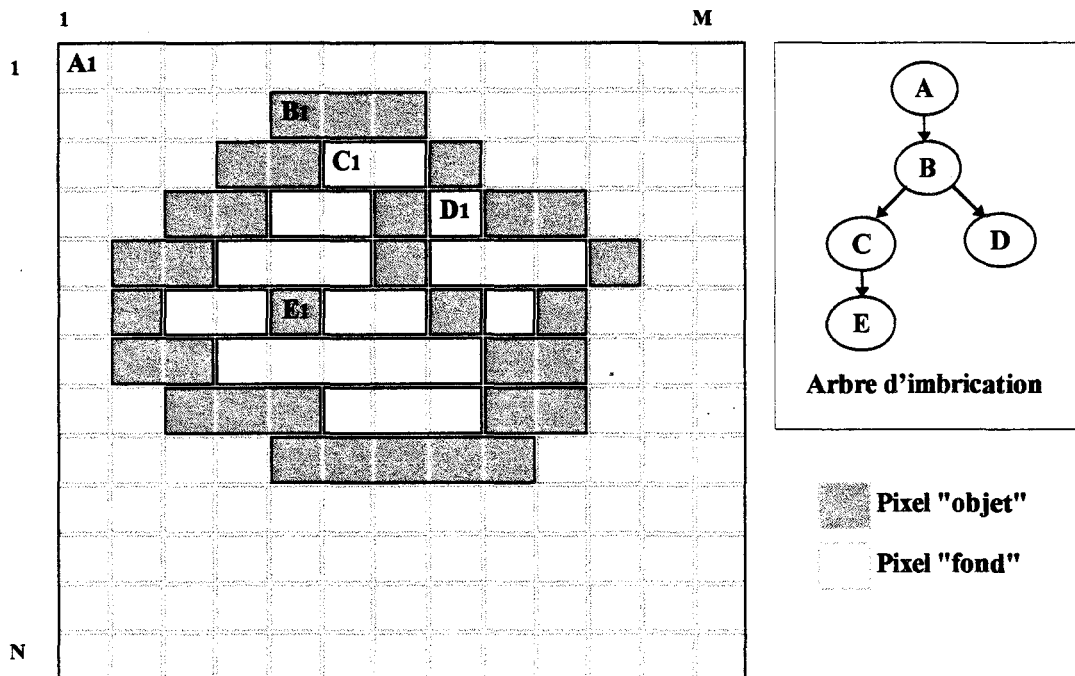


Figure 2-10 : Exemple permettant d'exprimer les différentes relations précédentes.

2.5.2.4 Décomposition en type de segment

Le point A1 est le point de départ de toute image. Il désigne le fond primaire.

La séparation des segments s'opère dans un premier temps par le repérage du premier segment objet tel que :

$$S_o(o) = \{S_v(o) \in T_i(o) / \forall v < w, \exists S_w(o) \text{ tq } Adj[S_v(o), S_w(o)]\}$$

Dans l'exemple, on a le premier objet dont le premier segment débute en B1 d'où $S_o(1)$ avec $d_o(1) = (B1)$.

Ensuite dans l'ordre d'imbrication, on a soit un nouvel objet donc même démarche ou un évidement à l'intérieur du premier objet. Dans le cas de l'évidement, on exprime par $S_e^0(o)$ le premier segment évidement de l'objet d'indice o . Dans le cas général on a l'expression : $S_e^n(o)$ où n est l'indice d'apparition d'un évidement sur cet objet o .

En suivant l'exemple on a : $S_e^0(1)$ avec $d_e^0(1) = (C1)$. Par contre l'évidement D est le deuxième évidement de l'objet B et son premier segment est référencé par :

$$S_e^1(1) \text{ et } d_e^1(1) = (D1).$$

De la même façon il faut ensuite pouvoir localiser le premier segment de l'objet interne par la désignation $S_{im}^0(o)$.

Dans le cas général d'une imbrication de profondeur quelconque on a : $S_{im}^n(o)$ où n est le $n^{ième}$ objet interne de l'évidement de niveau m dans l'objet référencé o . Pour exemple on a l'objet interne E qui débute en E1 par son premier segment $S_{io}^0(1)$ avec $d_{io}^0(1) = (E1)$. Le tableau suivant permet de récapituler le nombre d'évidements et objets internes à chaque niveau d'imbrication de la Figure 2-10.

Forme	$d_o(O)$	$d_e^n(o)$	$d_{im}^n(o)$	Contenu(s) pour chaque forme
A (image)	{B1}	{C1,D1}	{E1}	1 objet
B	{B1}	{C1,D1}	{E1}	2 évidements et 1 objet interne
C	x	{C1}	{E1}	1 objet interne
D	x	{D1}	x	$\emptyset$
E	x	x	{E1}	$\emptyset$

2.5.3 Représentation sous forme d'une liste de contours chaînés

Tous les éléments permettant de modéliser l'objet binaire sont maintenant prêts ; nous allons mettre en oeuvre le suivi de contour vers une structure chaînée. Cette étape réduit considérablement la masse d'informations brutes contenues dans la disposition matricielle de l'image numérisée.

2.5.3.1 Description du codage en chaîne de contours

L'objectif visé est de modéliser l'image par un codage supérieur afin de ne conserver que les formes de contours. Le système doit être réversible dans le cas d'une image traitée de type binaire. Ce codage est la retranscription des caractéristiques géométriques et topologiques des objets contenus dans l'image.

La première approche est celle de Freeman [FRE,61]⁶. Il opte pour le codage d'entités représentant les liens entre les pixels voisins. Le contour est rapporté à une séquence de segment de droite ayant une direction et une longueur caractéristique.

Le principe du codage est de découper l'espace des directions des pixels voisins proches, en un codage absolu de 8 vecteurs dans le sens horaire. Le contour est affecté à une séquence de codes qui débute par les coordonnées du point de départ puis une suite de codes décrivant les changements de directions pour se cheminer correctement sur le contour afin de retrouver le point de départ.

La démarche citée ci-dessus n'est pas la forme la plus condensée pour décrire un contour sous forme d'une liste de codes caractéristiques des pixels appartenant au contour. Si le contour est contenu par M éléments la chaîne vaut $M+1$ codes. Lorsque nous avons un contour peu bruité, de périmètre important avec des segments de droites de longueur supérieure au pixel, on peut apporter une variante au codage précédent. Il suffit d'indiquer par un compteur le nombre de séquences de directions identiques pour un segment de droite continu. A chaque cassure de direction, on a le nombre de fois que l'on suit continûment la

direction et le code de Freeman du vecteur direction. Un exemple illustre les deux méthodes de codage (Figure 2-11).

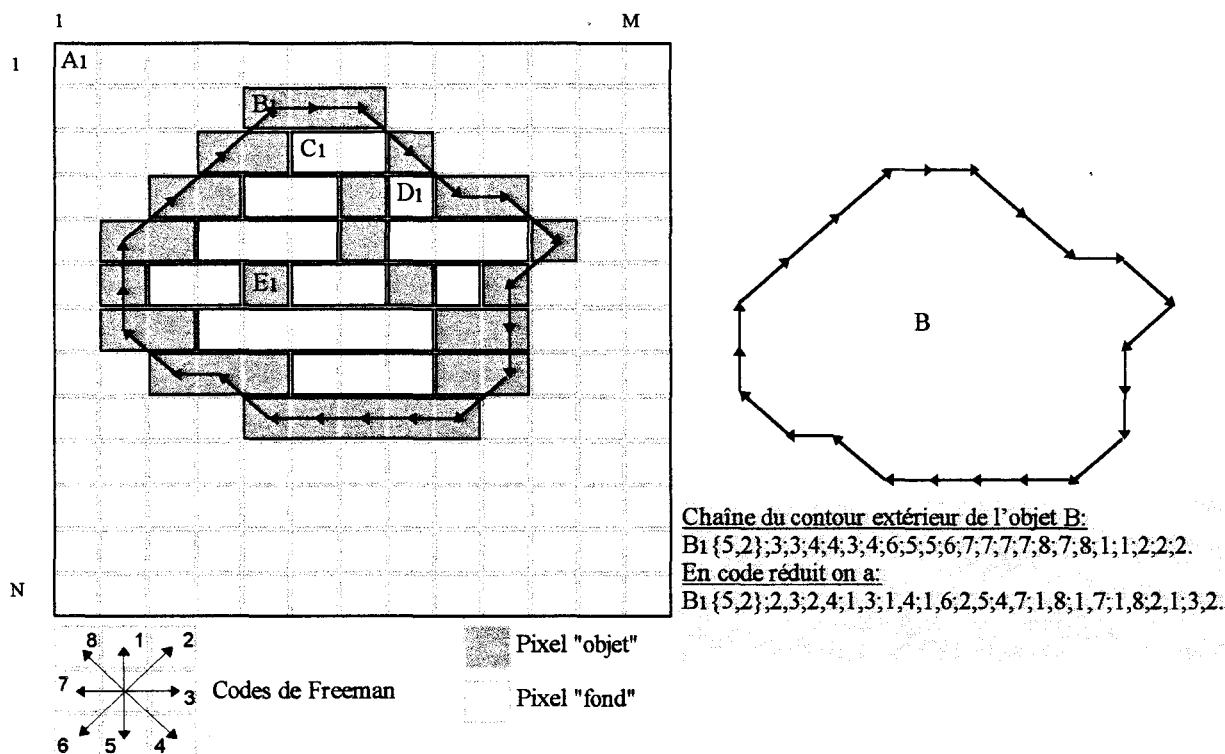


Figure 2-11 : Illustration du codage de Freeman.

2.5.4 Codage du contour d'une image binarisée par la technique du seuillage

Lorsque la qualité de l'image le permet, et sur des régions suffisamment contrastées, la technique du seuillage offre des avantages d'un traitement direct. Dans ce cas nous ne sommes pas face à des contours mais à des formes sur lesquelles on suit les courbes frontières. Deux techniques sont généralement exploitées : l'accès séquentiel et le suivi de contour.

2.5.4.1 L'accès séquentiel

On décrit un balayage systématique de l'image de gauche à droite et de bas en haut. Une première opération consiste à l'obtention de deux lignes consécutives de l'image [PAV,78]. De là on procède au repérage des segments pixels objets par la mémorisation des extrémités sur la ligne qui les composent. Ensuite on procède à l'étiquetage de segments suivant les relations décrites au paragraphe 2.5.1.1.

Pavlidis propose de représenter les liaisons inter-segments par cinq distinctions précises. La Figure 2-12 montre un exemple qui fait appel à ces différentes notions.

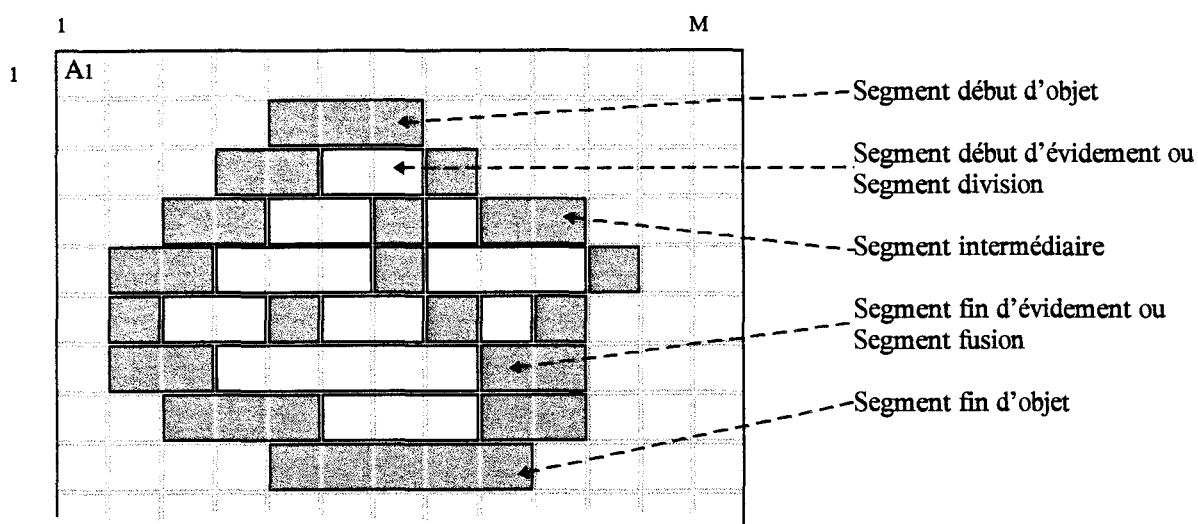


Figure 2-12 : Les cinq caractérisations des liaisons inter-segments.

2.5.4.2 L'opérateur de suivi de contour

Cette technique est surtout intéressante lorsqu'on veut procéder à une génération automatique de contours chaînés. Le principe reste toujours le même suivant les différentes variantes proposées par de nombreux travaux [HEG,85][⊗].

On recherche un point de départ sur la courbe frontière objet, puis on décrit par un opérateur de suivi de contour la frontière de l'ensemble des pixels constituant la forme à décrire.

La méthode de suivi de contour est intéressante dans le cas d'une application semi-automatique et sélective. L'exemple du germe placé à l'intérieur de l'objet et où le germe se déplace à abscisse constante pour atteindre en ordonnée le premier point contour offre une facilité de mise en oeuvre avec une efficacité optimale [DEC,92][⊗].

2.5.4.2.1 Le suivi de contour

L'opération s'effectue par la génération de la chaîne de codes en parcourant successivement tous les pixels de la courbe frontière (contour) de la forme. Le codage de Freeman permet le passage d'un pixel contour à l'autre, le code suivant est obtenu par une règle de poursuite.

Pour un parcours dans le sens horaire on utilise la règle du « Left Most Looking » [GON,87][⊗]. A partir du point de départ, on recherche le point suivant en décrivant dans le sens horaire les huit directions du code de Freeman jusqu'à rencontrer un nouveau point objet (Figure 2-13). A partir du point suivant, on estime une direction située à la perpendiculaire par rapport à la direction précédente. On a la relation de la direction testée par rapport à la direction précédente qui est : $\theta(test) = \left(\theta(prec) - \frac{\pi}{2} \right) \bmod(2\pi)$.

Ou encore avec le code de Freeman $D(test) = (D(prec) - 2) \bmod 8$.

Le suivi de contour nécessite une opération de marquage qui permet de ne pas surgénérer des contours et d'éviter les croisements.

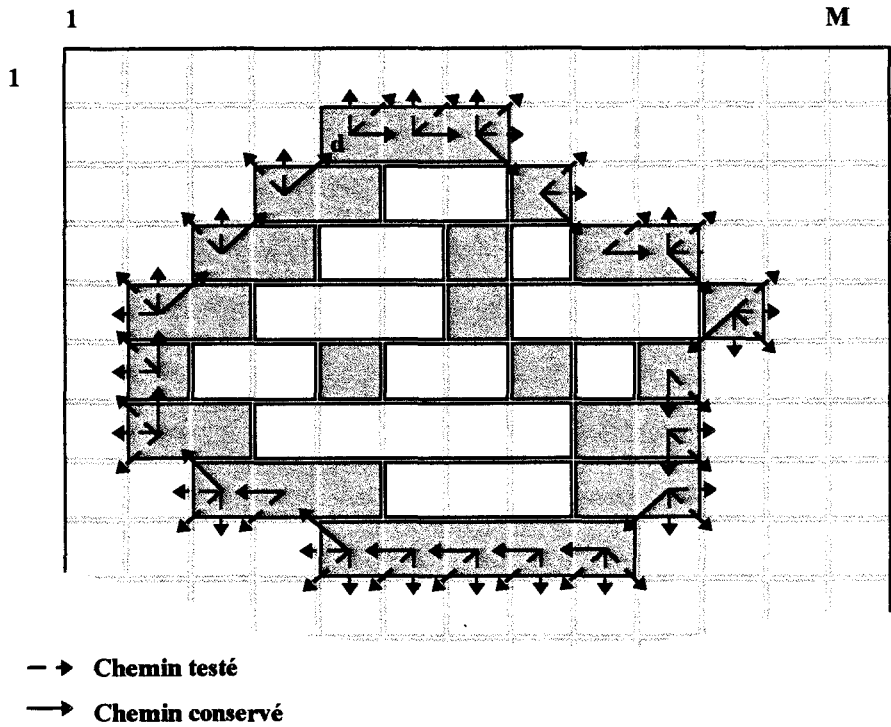


Figure 2-13 : Exemple de suivi de contour par la règle du Left Most Looking.

2.5.4.2.2 Le marquage

Il faut répéter l'action de recherche des points de départ en dehors de ceux qui ont déjà été traités pour permettre un suivi de contour sans redondance. La technique la plus performante est celle du double marquage [BOM,90][©]. On opère par un marquage en ligne avec un code d'entrée-sortie approprié lors de l'étape de suivi de contour, et un marquage final opérant sur la forme entièrement traitée ne comportant plus de points de départ de type objet ou évidement.

Le point de départ est décrit comme le premier pixel (objet ou fond) d'une forme (objet et objet interne ou évidement) contenu dans un objet d'indice *o* lors d'une description ligne à ligne de l'image. On a une procédure itérative qui répète l'action de recherche de point de départ puis de suivi jusqu'au niveau d'imbrication le plus élevé d'un objet donné. L'algorithme suivant (Figure 2-14) illustre la méthode sous forme récursive décrite par Bombardier[BOM,91][©].

L'algorithme proposé par suivi de contour et double marquage est robuste quel que soit le degré de complexité de l'image. Il offre la possibilité de calculer directement les caractéristiques géométriques par les relations du code Freeman. Le niveau d'imbrication des formes est conservé. L'inconvénient majeur est l'étape récursive, qui engendre un temps de calcul élevé pour des images avec des objets de niveaux d'imbrication élevés.

autres chaînes ne répondant pas ce critère sont détruites. L'optimisation de la méthode consiste à privilégier le voisinage à scruter en fonction de la dernière direction suivie [CHE,87][⊗] ou par un test logique suivant un ensemble de combinaisons envisagées [ROB,76][⊗].

Malheureusement ces techniques révèlent souvent des contours non fermés lorsque le critère choisi ne satisfait plus à une poursuite du contour jusqu'à l'élément qui a fait prendre naissance à la chaîne. D'où la mise en place d'algorithmes qui ont pour objet de réaliser la concaténation des chaînes primaires vers des chaînes les plus longues possibles.

2.6.2 Chaînage des pixels sur des images binaires

L'image binaire a le mérite d'avoir déjà les contours apparents. On a toujours à réaliser les jonctions de chaînes de contours afin de créer un ensemble cohérent ou également des éliminations de chaînes parasites. La méthode de recherche systématique opte pour la détection d'un voisinage proche au travers d'un masque 3*3 afin de définir le type de point rencontré. Dans cette optique nous pouvons détecter des embranchements de contours. Quatre configurations se proposent à nous:

- C'est un point de départ,
- C'est un point final,
- C'est un point intermédiaire,
- C'est un point de jonction.

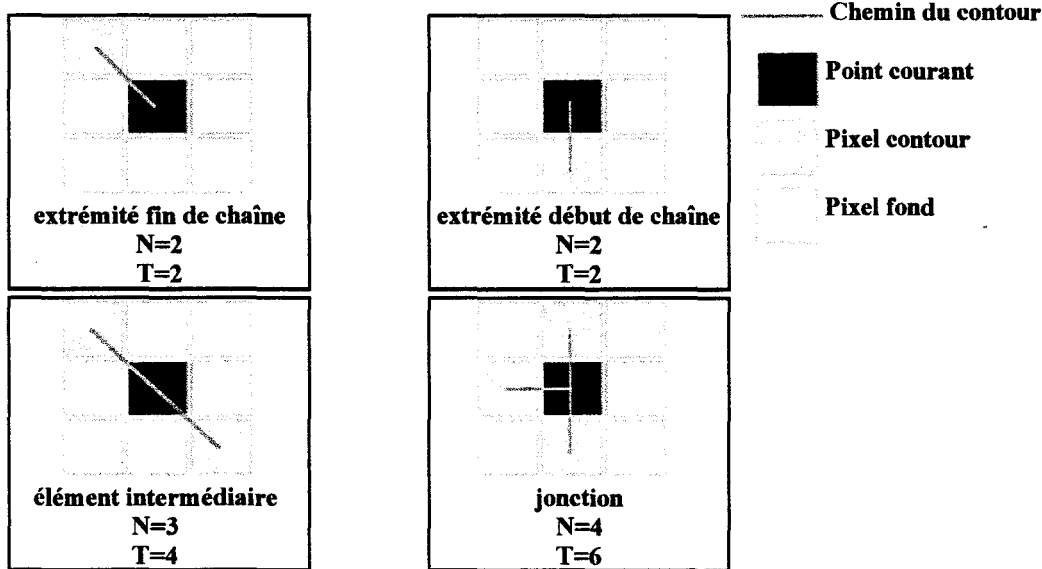


Figure 2-15 : Exemples des 4 configurations possibles.

Par une description simple au travers d'un masque 3*3 pixels on dénombre le nombre de points de contours (N) contenus dans le voisinage et le nombre de transitions pixel contour/fond (T) [ALI,88][⊗]. La Figure 2-15 récapitule les quatre configurations possibles:

- Si $N = 0$, le point est isolé,
- Si $N \leq 2, T = 2$ le point est une extrémité de chaîne,
- Si $N > 2, T > 4$ le point est une jonction.

La jonction des chaînes se fait par analyse du voisinage des extrémités de chacune d'entre elles. Pour réaliser le tri des chaînes à conserver ou non, on prend en considération la longueur du contour, et on élimine celle qui est inférieure à une longueur référencée.

2.7 Les résultats d'un traitement de segmentation de l'image

Quelques images traitées viennent éclaircir les différentes démarches de l'opération d'extraction des contours. On montre simplement les distinctions qualitatives des méthodes de segmentation sur une image test (Figure 2-16) infra-rouge représentant l'image d'un oeil.

Le but recherché est d'extraire le contour de la pupille noire et des reflets infra-rouge de quatre sources ponctuelles se réfléchissant sur la cornée. Le reste n'ayant pas d'incidence sur les mesures, on cherche à éliminer par un traitement simple les contours des cils, paupières et reflets parasites.

La Figure 2-17 représente l'image gradient par la fusion des gradients directionnels horizontaux et verticaux. Pour obtenir la norme du gradient il faut procéder en tout en trois

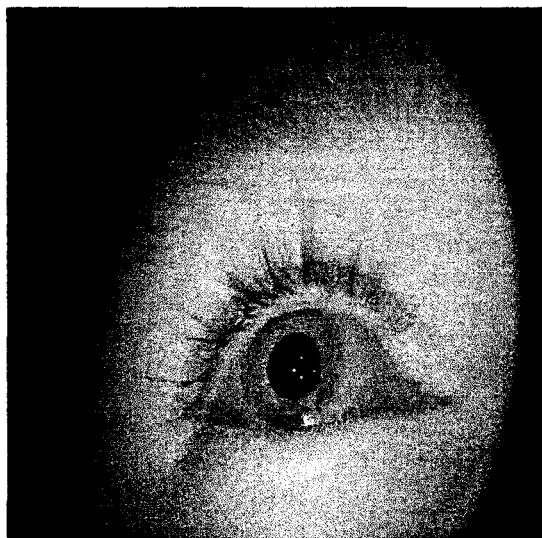


Figure 2-16 : Image originale.

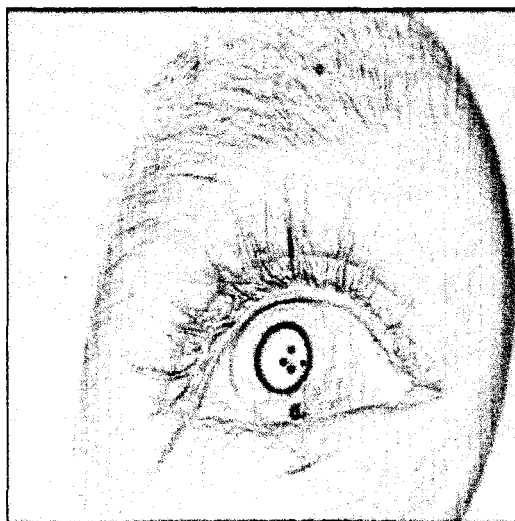


Figure 2-17 : Image norme du gradient sur l'image originale.

étapes avec pour chacune d'entre elles mémorisation momentanée du nouveau plan image (méthode de Canny-Deriche).

La première étape traite le gradient directionnel horizontal pour lequel on dérive (détecteur) dans la direction horizontale, puis on intègre (lissage) dans la direction verticale. A l'opposé pour le gradient vertical (deuxième étape) les deux opérations se font dans une séquence inverse sur le plan image pivotée de 90 degrés. Ces deux traitements sont indépendants et peuvent se réaliser de manière tout à fait parallèle. Les deux nouveaux plans images sont fusionnés par le calcul de la norme en chaque point.

La Figure 2-18 est la représentation de l'image binaire après opération des non maxima-locaux dans la direction exacte du gradient. Pour faire cette opération on procède par interpolation linéaire sur la norme.

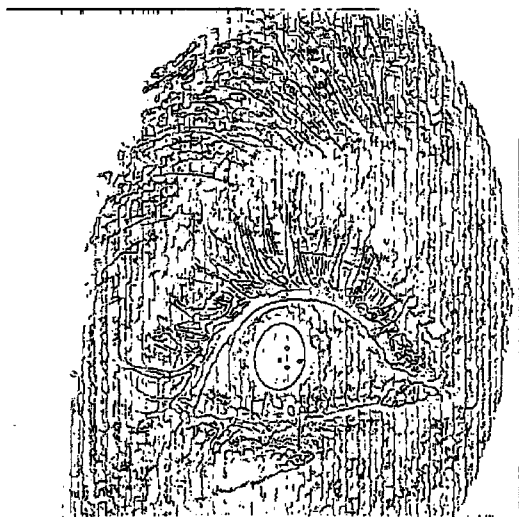


Figure 2-18 : Contours après suppression des non maxima-locaux (image originale).

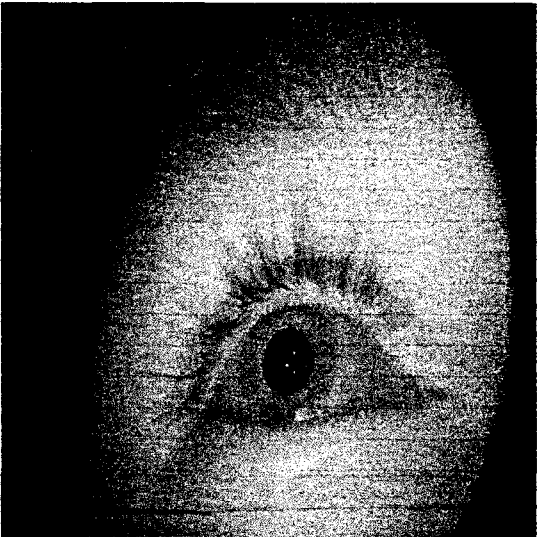


Figure 2-19 : Image noyée dans un bruit gaussien.

Afin d'estimer la bonne tenue du filtre de Deriche on procède à un ajout d'un bruit gaussien sur l'image test voir Figure 2-19.

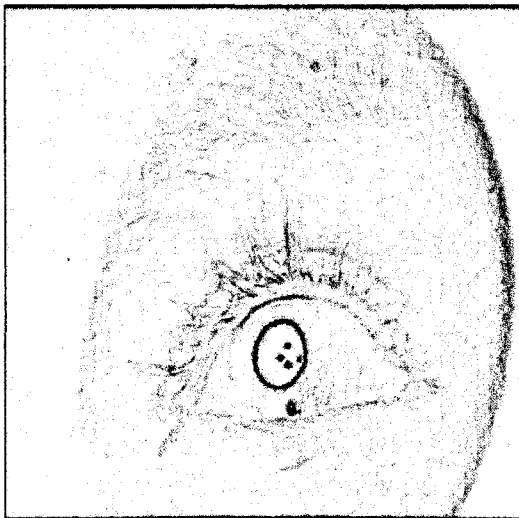


Figure 2-20 : Image norme du gradient (image bruitée).

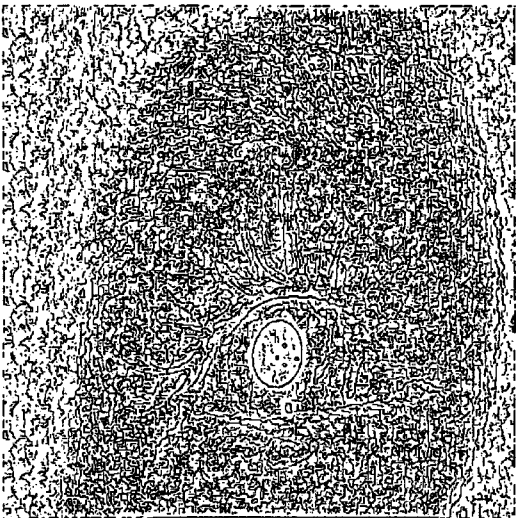


Figure 2-21 : Contours après suppression des non maxima locaux (image bruitée).

L'image du gradient Figure 2-20 est légèrement dégradée. On le remarque plus nettement sur l'image des contours par suppression des non maxima-locaux (Figure 2-21).

De nombreux contours non significatifs encombrant l'image. Dans ce cas on réduit assez fortement les informations non pertinentes par un filtrage par hystérésis (Figure 2-22). Pour obtenir ce résultat il suffit de binariser l'image par un seuil bas qui a le mérite d'éliminer simplement les contours non significatifs dus au bruit principalement. Ensuite on chaîne les informations restantes mais on ne retient que celles qui possèdent au moins un pixel sur son contour qui soit supérieur ou égal au seuil haut.

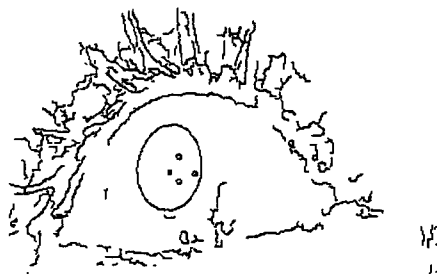


Figure 2-22 : Contours après seuillage par hystérésis. Seuil haut=30, seuil bas=15. Opération sur image bruitée.

La discrimination des contours non recherchés est permise par une adaptation du seuil haut du seuillage par hystérésis. La connaissance plus approfondie de l'image traitée permet d'obtenir le seuil qui soit optimal afin de révéler les contours qui sont nécessaires à la compréhension de l'image. Les Figure 2-23 et Figure 2-24, montrent qu'il est tout à fait possible d'améliorer considérablement l'image des contours, en vue d'une classification et d'une reconnaissance des formes recherchées pour une application donnée.



Figure 2-23 : Contours après seuillage par hystérésis. Seuil haut=67, seuil bas= 15. Opération sur image bruitée.



Figure 2-24 : Contours après seuillage par hystérésis. Seuil haut=67, seuil bas =15. Opération sur image originale.

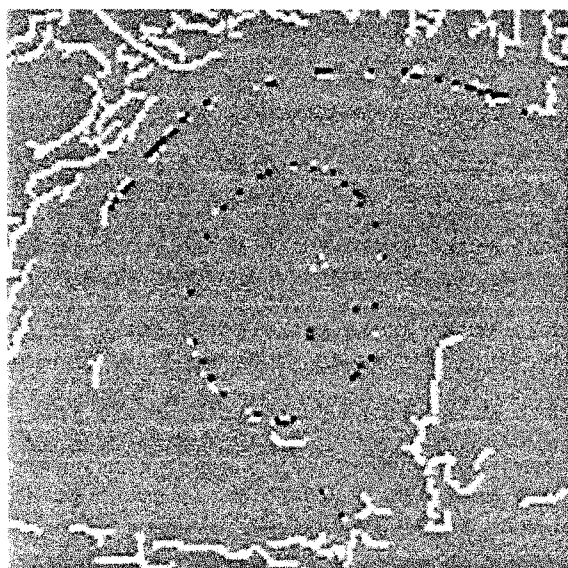


Figure 2-25 : Qualité de la localisation malgré l'ajout de bruit dans l'image.

La comparaison qualitative de la Figure 2-25 montre la bonne localisation du contour de la pupille en soustrayant la Figure 2-24 et Figure 2-23 entre la détection sur l'image brute et l'image noyée dans le bruit. Les points en noir et blanc sont ceux qui n'ont pas reçu de correspondants entre les deux images. On peut remarquer que l'épaisseur de ces points ne dépasse jamais au plus un pixel. En moyennant sur le contour complet l'erreur de positionnement a une valeur nulle. L'erreur se partageant de chaque côté du contour.

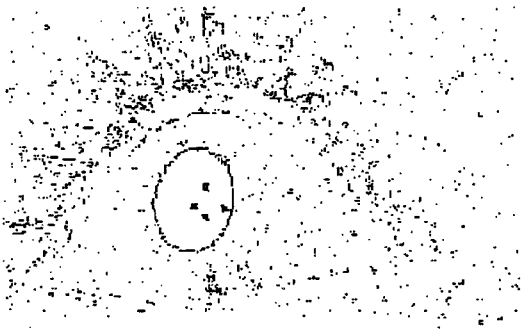


Figure 2-26 : Méthode de Prewitt. Image binaire par suppression des maxi-locaux.

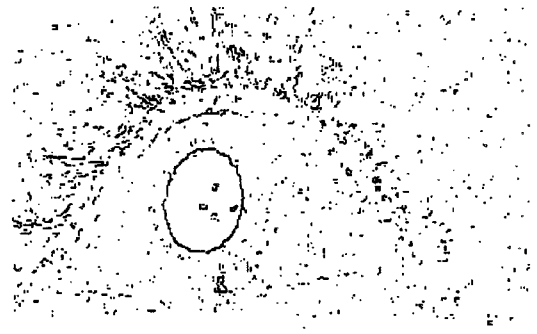


Figure 2-27 : Méthode de Kirsh.

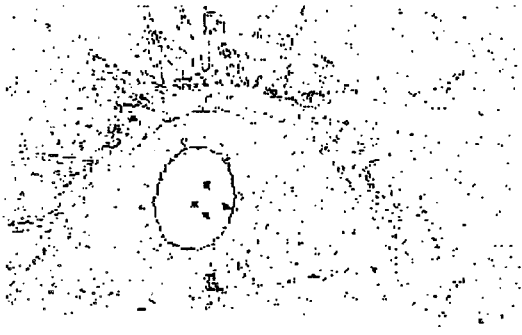


Figure 2-28 : Méthode de Sobel.

Les trois figures suivantes sont la preuve de la mauvaise tenue des filtres de détection de contour, s'ils n'ont pas fait l'objet d'une optimisation sur le positionnement du contour, lors du traitement d'une image bruitée. L'exploitation de ces trois résultats ne permet pas d'avoir suffisamment de précision pour que le contour soit chaîné de façon réaliste. De plus, de nombreux artefacts envahissent les contours, ce qui demande un traitement supérieur délicat de lissage de contours et fermeture de ceux-ci.

2.8 Conclusion

Au regard des différentes techniques de segmentation, on remarque une nette amélioration des performances de localisation pour les filtres optimaux type « Canny-Deriche ». Cette technique est composée d'une chaîne de traitements complexe à mettre en oeuvre.

Tout d'abord, la recherche des contours pertinents, nous impose de traiter l'image par un filtre détecteur de discontinuités. Ensuite par un procédé de suivi de la frontière objet, on maintient un opérateur qui suit fidèlement le maximum du gradient dans la direction du contour. Ce n'est qu'après un dernier traitement de chaînage et de fermeture des contours, que l'on peut analyser, les formes contenues dans l'image.

Le détecteur de Deriche décrit sous forme récursive, est délicat à implanter matériellement. On se propose d'analyser l'impact d'un tel filtre sur l'architecture qu'il engendre afin de tenir un traitement temps réel. C'est l'objet du chapitre suivant, qui met en avant la restructuration nécessaire du filtre pour que sa nouvelle architecture soit en adéquation avec les contraintes temporelles incontournables.

La technique de segmentation en région par l'opérateur de seuillage binaire sera plus largement appréciée pour les applications où la qualité de l'image le permet, afin de minimiser les calculs. Le traitement de segmentation s'opère alors directement sur l'image originale sans devoir réaliser l'étape de détection.

Chapitre 3

3. Recherche d'une implantation matérielle du filtre récursif de Canny-Deriche

3.1 Introduction

Après avoir étudié les opérateurs d'extraction de contour sous leur forme algorithmique, on se propose d'envisager leur implantation sous forme d'un circuit spécialisé.

L'accroissement de la puissance de calcul offerte par les circuits spécifiques de l'électronique numérique, nous amène à développer des traitements de complexité accrue.

Le dilemme de l'implantation de telles applications se situe dans la restructuration des algorithmes lorsque ceux-ci ne sont pas étudiés en amont pour une architecture cible.

Lorsque les contraintes matérielles et temporelles ne permettent plus la bonne tenue du traitement sur une machine dédiée, il faut parvenir à introduire du parallélisme qui intègre et pallie aux difficultés d'une réponse temps réel.

L'intégration des notions de temps de réponse, période d'échantillonnage, encombrement, prix va permettre d'optimiser et de définir les choix retenus pour accéder à la meilleure solution.

L'efficacité d'une implantation reste complexe à réaliser et fait largement appel à la notion d'adaptation d'une architecture à un algorithme ou inversement mutation de l'algorithme par des concessions afin de satisfaire à sa description sur une architecture cible.

Les notions d'adéquation en algorithme et architecture sont clairement synthétisées dans le document [AAA,92] qui rapporte les travaux en la matière d'universitaires et d'industriels confrontés à ce délicat problème.

La première étude s'appuie sur une technique de restructuration, permettant de laisser les opérations trop longues pour un traitement à la cadence vidéo se propager à l'intérieur d'une structure pipeline.

La deuxième étude fait partie d'un projet personnel d'implantation d'une structure récursive utilisant les propriétés de l'arithmétique distribuée.

Ces deux conceptions architecturales permettent de mettre en évidence la complexité et le cas individuel de chaque application en terme d'adéquation architecture et algorithme, le but à atteindre étant toujours un temps de calcul optimal pour une surface de silicium donnée.

3.2 Mise en correspondance de plusieurs techniques

La démarche d'extraction de contour selon le principe de Deriché permet d'extraire les contours avec une précision optimale sur sa localisation. On obtient ce résultat au détriment d'un temps de calcul trop important pour un traitement au rythme d'une trame vidéo. On peut citer les travaux du laboratoire GERE [BOU,94][⊗] sur une implantation matérielle du filtre de Canny-Deriché optimisé [DER,93].

3.2.1 Mise en oeuvre d'une implantation matérielle

La philosophie de Bourennane est de généraliser la méthodologie de Deriché à des contours à fronts de montée non idéaux suivant l'allure suivante (Figure 3-1):

3.2.1.1 Solution proposée par le laboratoire GERE

La fonction contour est la suivante:

$$C(x) = 1 - \frac{\exp(-Sx)}{2} \text{ pour } x \geq 0 \text{ et } C(x) = \frac{\exp(Sx)}{2} \text{ pour } x < 0$$

L'indice S est le paramètre du contour. Il peut être assimilé comme l'élément qui détermine le flou causé par la caméra. Il n'est donc pas nécessaire de déterminer pour chaque contour séparément l'indice optimal pour une prise de vue d'une scène avec un même capteur.

La réponse impulsionnelle du filtre optimisé sur les contours flous est:

$$f_-(x) = A(k \exp(\alpha x) \sin(\omega x) + \exp(\alpha x) \cos(\omega x) - \exp(Sx)) \text{ pour } x \leq 0$$

$$f_+(x) = A(k \exp(-\alpha x) \sin(\omega x) + \exp(-\alpha x) \cos(\omega x) - \exp(-Sx)) \text{ pour } x > 0$$

De même que pour Deriché on implante le filtre sous forme récursive:

$$y_+(n) = a_1 x(n-1) + a_2 x(n-2) + a_3 y_+(n-1) - a_4 y_+(n-2) + a_5 y_+(n-3) \quad n:1 \rightarrow N$$

$$y_-(n) = -a_1 x(n+1) - a_2 x(n+2) + a_3 y_-(n+1) - a_4 y_-(n+2) + a_5 y_-(n+3) \quad n:N \rightarrow 1$$

$$y(n) = y_+(n) + y_-(n) \quad n:1 \rightarrow N$$

La démarche de l'extension au traitement d'une image en deux dimensions est la même que pour Deriché, elle est largement détaillée dans [BOU,93][⊗].

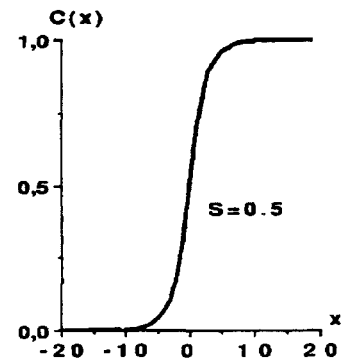


Figure 3-1 : Contour non idéal sous forme exponentielle.

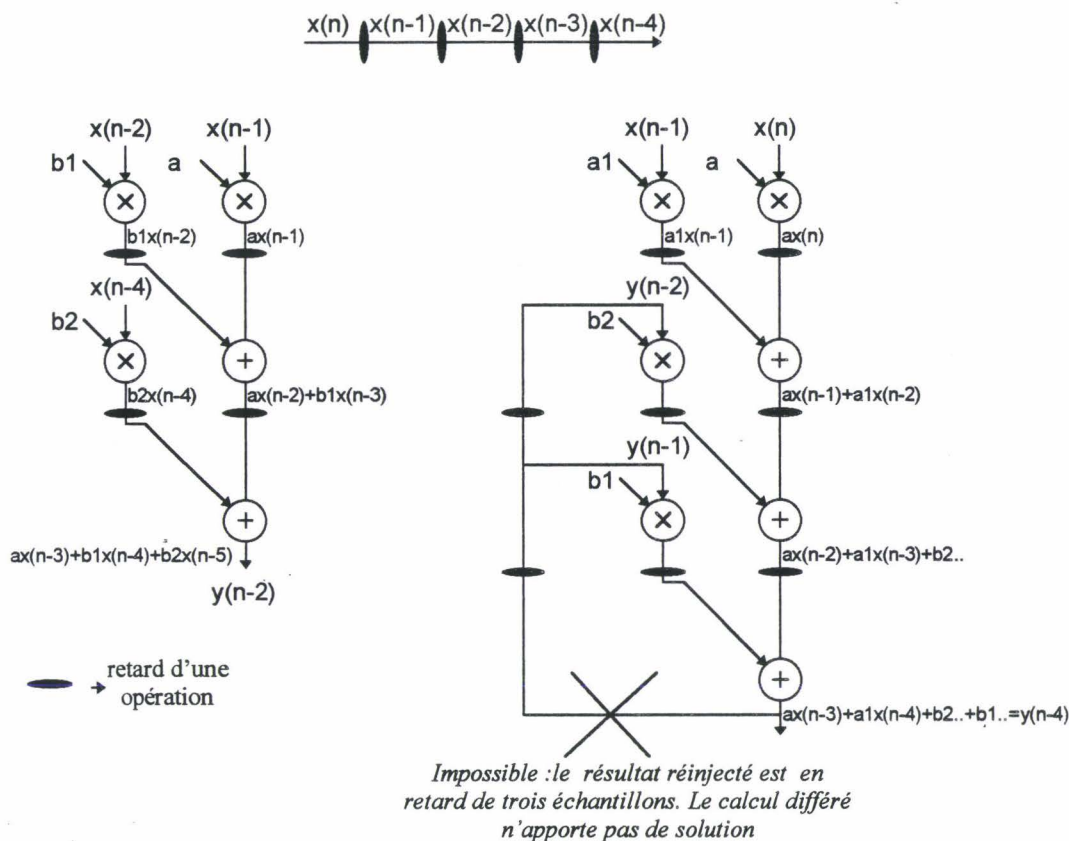


Figure 3-2 : Structures pipelines.

Le temps de traitement d'une image par ce filtre, avec un algorithme implanté sur PC même équipé d'un processeur Pentium et pour une image de taille 256*256 pixels quantifiée sur 256 niveaux de gris, déborde démesurément les 40 ms durée entre deux images successives à 25 images/s.

La technique consiste donc à étudier une architecture matérielle permettant d'accélérer le processus de calcul. Les techniques généralement employées sont l'implantation sur DSP, processeurs parallèles ou technologie d'intégration VLSI sur un circuit ASIC. Ici le choix de Bourennane s'est porté sur un circuit spécialisé ASIC en s'employant à paralléliser au maximum les calculs. L'architecture pipeline permet cette accélération par une séquence de tâches sur un parcours où chacune de ces tâches opère un calcul utilisant un élément qui le précède. Cet élément est le résultat de l'étage d'un temps différé d'une séquence. Des registres intermédiaires sous la forme de latches séparent les différentes tâches. Pour cela il faut pouvoir décomposer le traitement ce qui n'est pas toujours le cas. La décomposition pour un filtre RIF se prête bien à ce genre de technique; par contre dans le cas du filtre récursif il se pose le problème de la boucle de retour. Le schéma (Figure 3-2) ci-dessus expose le problème.

Sur le filtre RIF on sait que tous les échantillons d'entrée existent au moment du calcul sur le premier étage et ainsi de suite sur toute la profondeur du pipeline (Figure 3-3).

Sur le second filtre on doit injecter à partir du deuxième étage le résultat de la sortie de deux calculs précédents. Si ce résultat n'est pas disponible dans le temps imparti par la période d'échantillonnage du filtre, le résultat ne sera pas correct et on accumulera les erreurs à l'infini.

Dans le cas du filtre RIF, lorsque le temps de calcul est supérieur à la période d'échantillonnage, on accroît artificiellement le nombre de retards pour laisser le temps d'une

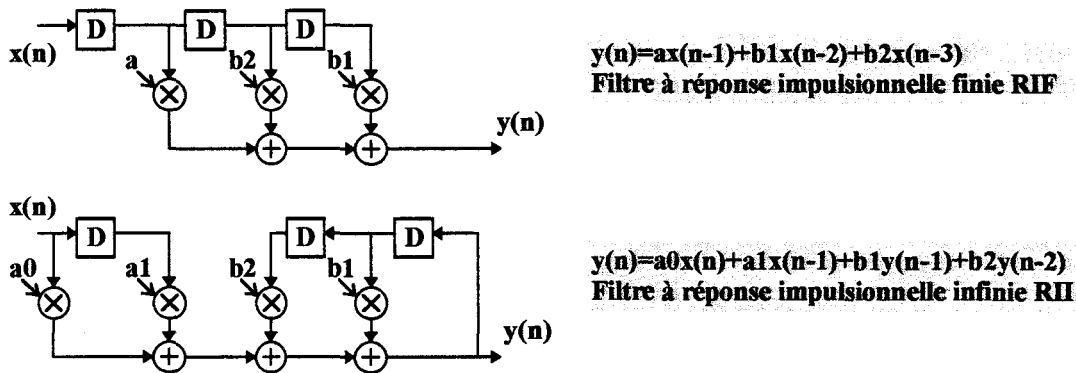


Figure 3-3 : Exemple de deux filtres sous forme RII et RIF.

opération à s'exécuter normalement (Figure 3-2). Le résultat est différé d'autant, que l'on a inséré de registres pour préserver les différentes couches d'opérateurs.

La technique employée par l'équipe du laboratoire GERE est l'utilisation de l'anticipation de calculs par ajout d'un flot de pôles supplémentaires et par dispersion de pôles.

3.2.1.1.1 Anticipation par ajout d'un flot de pôle

Pour mieux cerner la technique, nous reprenons la fonction de transfert d'un filtre quelconque :

$$H(Z) = \frac{\sum_{i=0}^N b_i Z^{-i}}{1 - \sum_{i=1}^N a_i Z^{-i}}$$

L'équation aux différences est :

$$y(n) = \underbrace{\sum_{i=1}^N a_i y(n-i)}_{\text{partie recursive}} + \underbrace{\sum_{i=0}^N b_i x(n-i)}_{f(n) \text{ non recursif}}$$

Comme nous l'avons remarqué dans le cas d'une structure pipeline on ajoute artificiellement des retards Z^{-D} qui ont pour objet de laisser les calculs se terminer même si la période d'échantillonnage est inférieure au temps d'une opération. On a remarqué également que cette technique n'était valable que lorsque le dénominateur est nul (RIF). L'anticipation par flot de pôles permet de transformer la fonction de transfert de façon que tous les coefficients deviennent: $Z^{-i} \rightarrow Z^{-(M+i)}$.

La boucle de retour de la partie récursive possède ainsi M étages pipelines et nous obtenons le résultat $y(n)$ en fonction des termes $\sum_{i=0}^{N-1} y(n-M-i)$.

Cette technique apporte souvent une instabilité sur la réponse du filtre restructuré. C'est pourquoi on étend cette méthode à l'ajout de pôles dispersés.

3.2.1.1.2 Anticipation par ajout de pôles dispersés

Le but recherché est de transformer la fonction de transfert de telle façon que le dénominateur ne contient que les termes : $\sum_{i=1}^N Z^{-iM}$.

Pour obtenir ce résultat, il faut procéder par introduction dans chacun de ces pôles, de (M-1) pôles et zéros qui s'éliminent, et conserver un module égal à celui du pôle original. Grâce à cette restructuration on conserve la stabilité du filtre original.

La partie récursive nécessite N multiplications et la partie directe (NM+1) multiplications. On obtient une complexité du filtre proportionnelle au carré de M, d'où une limitation d'implantation.

Une méthode étendue permet de réduire cette complexité proportionnellement au carré de M par une progression logarithmique. Soit la partie récursive d'un filtre avec k latches de pipeline dans sa boucle de retour initial :

$$H(Z) = \frac{1}{1 - \sum_{i=1}^N q_i(k) Z^{-ik}}$$

On obtient un filtre à 2k étages en pipeline en multipliant à la fois le numérateur et dénominateur par l'expression:

$$\left(1 - \sum_{i=1}^N (-1)^i q_i(k) Z^{-ik} \right)$$

L'extension d'un pipeline de profondeur 2^k s'opère par $\log_2(2^k) = k \log_2(2) = k$ transformations du filtre.

Bourennane a opté pour un facteur de multiplication d'ordre 4 ce qui l'a amené à transformer son filtre récursif d'ordre 3 deux fois suivant la méthode décrite ci-dessus. Il se retrouve alors avec un filtre restructuré d'ordre 12 suivant l'expression suivante :

$$FZ_+ = \frac{Y_+(Z)}{X(Z)} = H_1(Z)H_2(Z)H_3(Z)H_4(Z)$$

$$H_1(Z) = a_1 Z^{-1} + a_2 Z^{-2}$$

$$H_2(Z) = 1 + a_3 Z^{-1} + a_4 Z^{-2} + a_5 Z^{-3}$$

$$H_3(Z) = 1 + (a_3^2 - 2a_4)Z^{-2} - (-a_4^2 + 2a_3a_5)Z^{-4} + a_5^2 Z^{-6}$$

$$H_4(Z) = \frac{1}{1 - \{r_1 Z^{-4} + r_2 Z^{-8} + r_3 Z^{-12}\}}$$

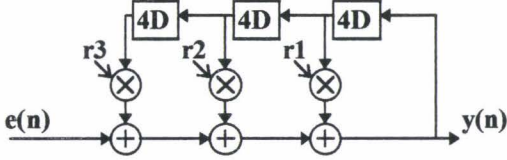
Les coefficients réduits sont :

$$r_1 = (a_3^2 - 2a_4)^2 + (-a_4^2 + 2a_3a_5)$$

$$r_2 = -(-a_4^2 + 2a_3a_5)^2 + 2a_5^2(a_3^2 - 2a_4)$$

$$r_3 = a_5^4$$

La nouvelle partie récursive du filtre se présente sous la forme de la Figure 3-4.



Pour la nouvelle décomposition nous avons un différé du résultat de sortie de deux temps d'échantillonnage, ce qui ne compromet en rien sur le résultat réinjecté.

En effet en décomposant les calculs on retrouve :

$$y_1(n) = e(n) + r_3 y(n-12)$$

$$y_2(n) = y_1(n-1) + r_2 y(n-9)$$

$$y(n-2) = y_2(n-1) + r_1 y(n-6)$$

Pour retrouver l'expression de la sortie on associe chacun des éléments dans leur temps de calcul associé :

$$y(n) = y_2(n+1) + r_1 y(n-4)$$

$$y_2(n+1) = y_1(n) + r_2 y(n-8)$$

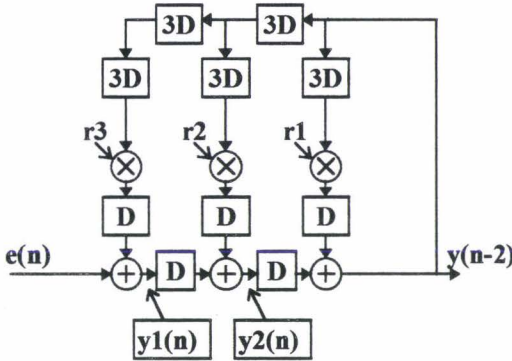


Figure 3-4 : Redistribution des retards dans la boucle de retour.

D'où l'expression retrouvée de $y(n) = r_3 y(n-12) + r_2 y(n-8) + r_1 y(n-4)$.

On voit qu'il est possible de réaliser des circuits spécialisés qui permettent de calculer une extraction de contour par un filtrage récursif au prix d'un accroissement considérable d'opérateurs, d'où des occupations en surface de silicium important. Le circuit décrit ici répond aux exigences d'un traitement temps réel jusqu'à une fréquence de 20Mhz. Sa surface de silicium est de 60 mm² pour une technologie de 1,2 μm pour une arithmétique interne sur 12 bits. Le filtre dans son intégralité intègre 11 multiplieurs de type pipeline (Baugh-Wooley [BAU,73][°]). Le traitement complet d'une image bidimensionnelle nécessite 4 filtres de ce type pour calculer le gradient, d'où une architecture complexe à mettre en oeuvre.

Le filtre d'origine d'ordre 3 impose cinq multiplications et quatre additions. La restructuration du filtre qui permet un traitement à 20 Mhz se traduit par un facteur d'accroissement des opérateurs de trois pour les additions et de deux pour les multiplications. Ce que l'on gagne en terme de rapidité se paye en augmentation sensible du nombre d'opérateurs donc en surface de silicium.

Une deuxième technique de reconstruction d'un filtre récursif peut être envisagée pour des performances tout aussi comparables. Cette étude personnelle a fait partie d'un essai d'implantation sur ASIC. Il n'y a pas eu concrétisation d'un circuit du fait que la technique d'extraction du gradient ne permet pas une mise en oeuvre aisée de l'accès aux différents plans mémoires à cause de l'action simultanée de plusieurs opérateurs. Néanmoins, la partie récursive du filtre isolé montre des avantages intéressants de temps de calcul par la méthode que nous décrivons par la suite.

3.2.1.2 Utilisation de l'arithmétique distribuée

La méthode permet une diminution considérable des calculs par l'utilisation de table de coefficients. Il suffit de décomposer le format des échantillons sous forme binaire. Le produit suivant peut se décomposer en :

$$a.I(n) = a \sum_{i=0}^p \delta_i(I) 2^i = a\delta_0(I) + 2a\delta_1(I) + \dots + 128a\delta_7(I)$$

avec $\delta_i(I) = 0$ ou 1 .

Le format des données est un nombre entier codé sur 8 bits.

Dans le cas généralisé au filtre récursif de Deriche on obtient :

$$\begin{aligned} y_+(n) &= aI(n-1) + b_1y_+(n-1) + b_2y_+(n-2) \\ y_+(n) &= a\delta_0(I(n-1)) + b_1\delta_0(y_+(n-1)) + b_2\delta_0(y_+(n-2)) \\ &\quad + 2(a\delta_1(I(n-1)) + b_1\delta_1(y_+(n-1)) + b_2\delta_1(y_+(n-2))) + \dots \\ &\quad \dots + 128(a\delta_7(I(n-1)) + b_1\delta_7(y_+(n-1)) + b_2\delta_7(y_+(n-2))) \end{aligned}$$

Les tables de coefficients restituent les valeurs des sommes binaires pour chacun des poids de 0 à 7. Le résultat est une valeur sur M bits parmi 8 configurations possibles. En effet c'est l'adressage de la table par les bits de $I(n-1), y_+(n-1), y_+(n-2)$ qui permet de réaliser l'opération de multiplication binaire.

La table sera composée de la manière suivante : $c = 2^i$

$\delta_i(I(n-1))$	$\delta_i(y_+(n-1))$	$\delta_i(y_+(n-2))$	Sortie de la table
A	B	C	
0	0	0	0
1	0	0	c.a
0	1	0	c.b1
1	1	0	c.(a+b1)
0	0	1	c.b2
1	0	1	c.(a+b2)
0	1	1	c.(b1+b2)
1	1	1	c.(a+b1+b2)

L'intérêt majeur de cette méthode est de supprimer toutes les opérations de multiplication. On est réduit à faire la somme des résultats sortant des 8 tables sous la forme d'un accumulateur. Cet accumulateur a une structure pipeline qui propage la retenue de chaque addition de même poids binaire au travers de la structure sous forme anticipée. Le résultat final revient à réaliser l'addition de deux nombres de M bits. La structure complète du filtre devient celle représentée par la Figure 3-5.

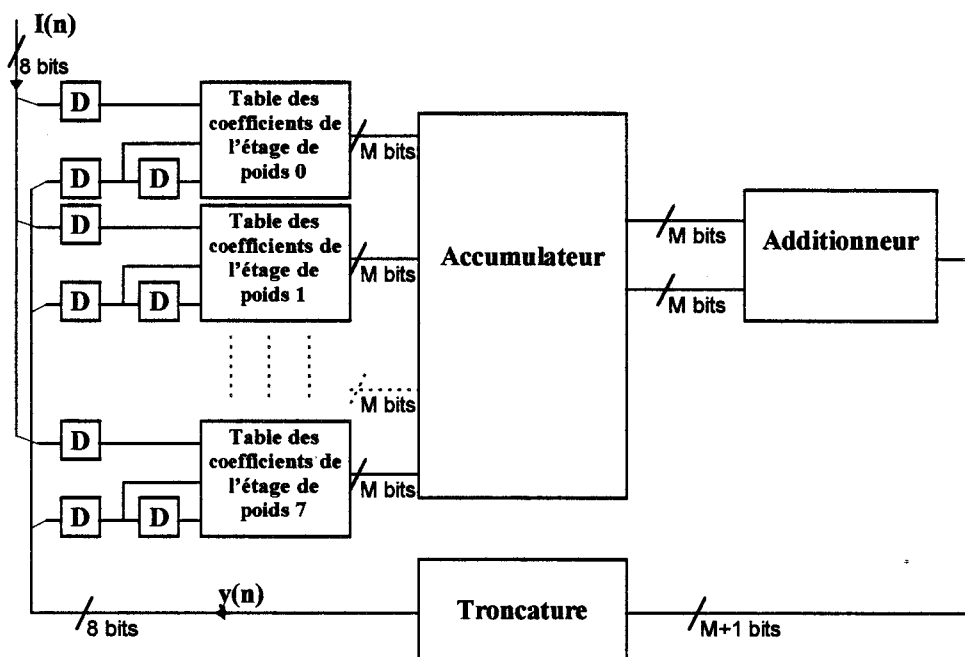


Figure 3-5 : Filtre numérique récursif utilisant l'arithmétique distribuée.

3.2.1.2.1 Exemple d'un calcul

Pour illustrer la technique on prend comme format pour les coefficients : 9 bits significatifs pour la partie entière et un bit de signe. La partie fractionnée est codée sur 6 bits.

Une valeur codée pour un rang de poids i , sera le résultat de la partie entière du produit de la valeur décimale d'un élément de la table par $2^{(n_d+i)}$.

avec n_d : nombre de bits de la partie fractionnée.

Si l'on veut le résultat : $y_+(n) = aI(n-1) + b_1y_+(n-1) + b_2y_+(n-2)$

avec pour valeurs des coefficients et des termes suivants :

$$a = 0.303239, b_1 = 0.898658, b_2 = -0.201897$$

$$I(n-1) = 76, y_+(n-1) = 111, y_+(n-2) = 136$$

$$[076]_2 = 01001100$$

les valeurs des termes en base binaire valent : $[111]_2 = 01101111$

$$[136]_2 = 10001000$$

D'où l'adressage des tables avec les valeurs restituées :

Table	A	B	C	Valeur sélectionnée en sortie de la table	Valeur sortant de la table	Valeur en base 2
0	0	1	1	b1	57	0000 0000 0011 1001
1	0	1	0	b1	115	0000 0000 0111 0011
2	1	1	0	a+b1	307	0000 0001 0011 0011
3	1	1	0	a+b1+b2	512	0000 0010 0000 0000
4	0	0	1	0	0	0000 0000 0000 0000
5	0	1	0	b1	1840	0000 0111 0011 0000
6	1	1	0	a+b1	4923	0001 0011 0011 1011
7	0	0	0	b2	63882	1111 1001 1000 1010

A,B,C sont les lignes d'adressage des tables vis à vis de l'élément au quel ils sont attachées au rang des poids des bits de chaque mot. Le résultat de la table 7 est complémentée à deux, c'est un nombre négatif.

La somme des 8 tables donne 71636.

Cette somme est soustraite de 65536 format rigide compte tenu du format retenu de 16 bits. On a donc un résultat positif de 6100. Ce nombre est divisé par 2^{16} ce qui donne 95.3125. Le résultat de $y_+(n)$ calculé sur une précision de 8 chiffres après la virgule est 95.339210. L'erreur relative, commise, est à attribuer à l'effet de troncature du système qui ne réinjecte que la partie entière de $y_+(n)$ et le format de la partie fractionnée des nombres codés dans les tables. Cette erreur est peu significative dans notre cas elle vaut 0.028%.

3.2.1.2.2 Structure de l'accumulateur

Une sommation classique de 8 mots de N bits revient à faire la somme deux à deux de chaque mot puis cascader des additionneurs qui absorberont le résultat des sommes partielles. On obtient une structure telle qu'à la Figure 3–6.

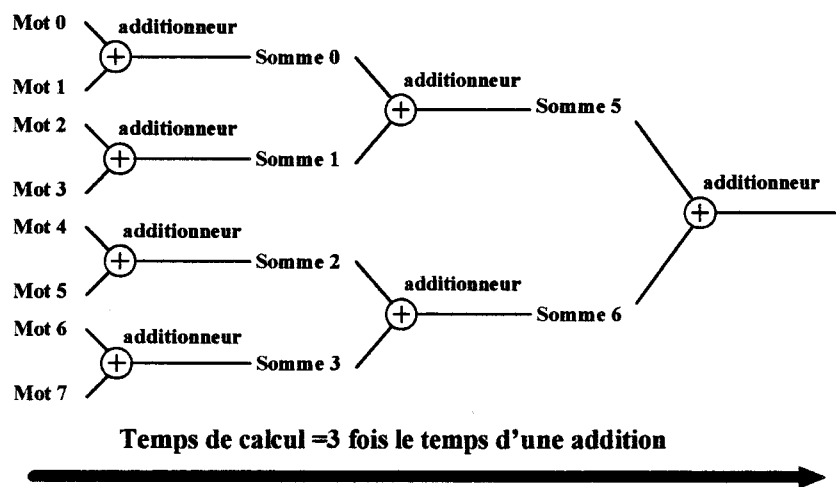


Figure 3–6 : Additions de 8 mots.

La structure de l'accumulateur permet d'anticiper sur les propagations des carry.

On additionne sur un sommateur de M bits, les bits de même poids de chaque mot.

Les sommes sont subdivisées en sous opérations, tel que l'opérateur ne demande pas une trop grande complexité de mise en oeuvre afin de gagner un temps de calcul sur chaque étage du pipeline.

La Figure 3-7 montre la construction des premières cellules de l'accumulateur. Il y a échange sur la gauche (propagation des carry), plus la construction de l'addition sur la cellule pour arriver à ne plus pouvoir exercer de subdivision. La somme est étendue à 9 mots.

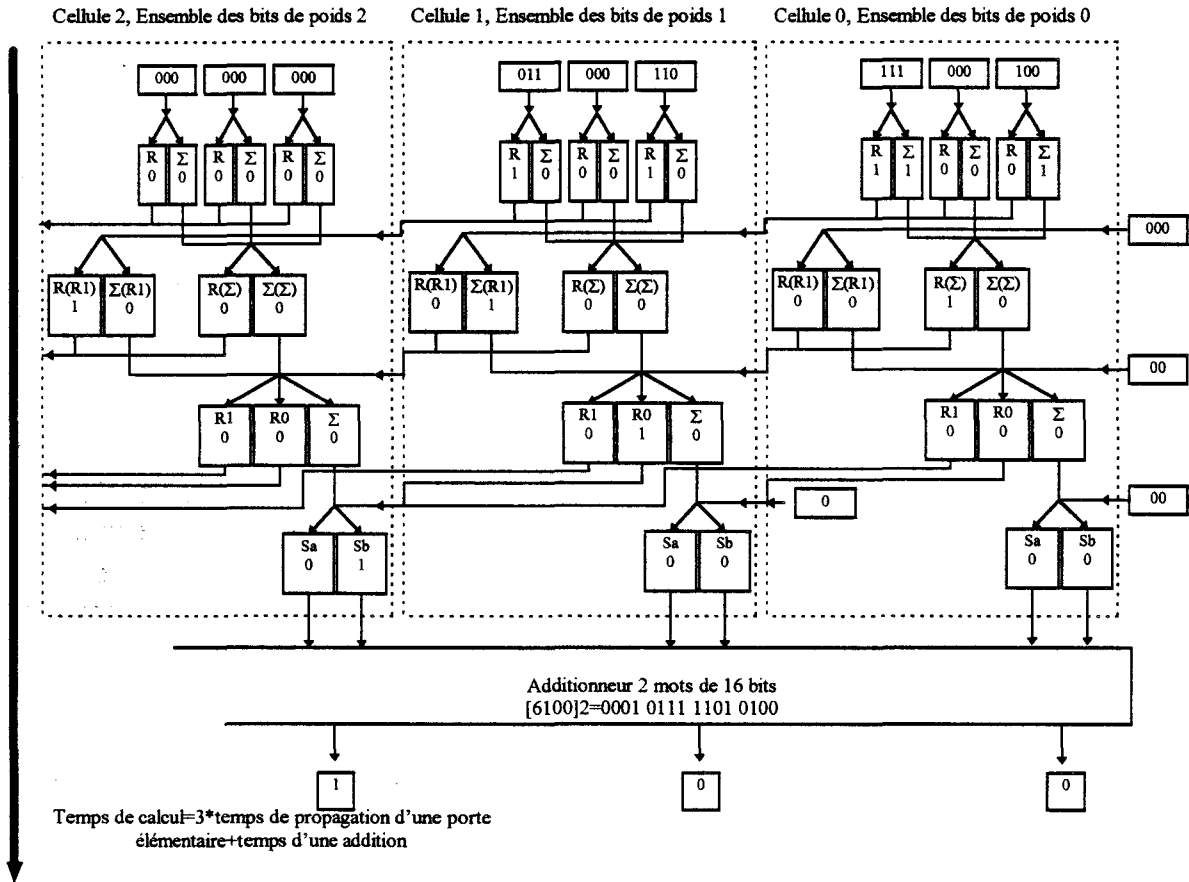


Figure 3-7 : Architecture de l'accumulateur.

Les trois multiplications du filtre original sont éliminées mais au détriment de l'utilisation de mémoires, on gagne en terme de période d'échantillonnage supérieure mais la surface occupée par les tables ne nous donne pas entière satisfaction. Il est souhaitable de parvenir à une meilleure efficacité en terme de surface occupée, c'est pourquoi on opte pour une extension vers un sur-échantillonnage. La description de cette nouvelle modification fait l'objet du paragraphe suivant.

3.2.1.3 Extension de l'arithmétique distribuée au filtre récursif échantillonné

On a vu que l'arithmétique distribuée permet de réduire les calculs à de simples additions au travers d'un accumulateur. La taille des données sortant des tables détermine la transformation sur l'architecture de l'accumulateur. Les N additions sur M bits se transforment en deux additions de M+1 bits. N est le paramètre qui définit la grandeur des données des

échantillons, ce qui implique le nombre d'étages après troncature du résultat sortant de l'additionneur. On remarque donc que l'ordre du filtre n'est pas pris en compte pour le nombre d'opérateurs dans cette technique. Seule la dimension des tables rentre en considération, ce qui implique une surface de silicium en conséquence sans perturbation sur le temps de calcul. Des filtres d'ordre important se prêtent tout à fait à cette technique de calcul.

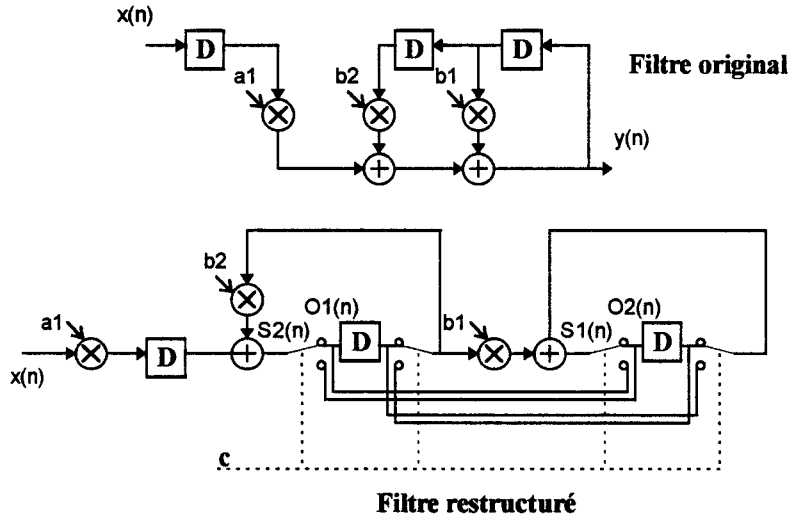


Figure 3-8 : Structure récursive modifiée permettant d'englober l'opération addition multiplication par un même opérateur d'arithmétique distribuée.

En reconsidérant le filtre récursif de Deriche on peut mettre en avant cette implantation en modifiant sa structure vers un ensemble à double récursivité (voir Figure 3-8). Ce qui implique une structure pipeline dans la chaîne directe avec des latches qui permettent un accroissement d'un facteur 2. Pour permettre ceci on opère par un échantillonnage des résultats intermédiaires vers l'une ou l'autre des boucles de retour.

Les équations récursives sont les suivantes, elles dépendent du paramètre d'aiguillage c . Pour $c=0$ on est dans la disposition indiquée à la Figure 3-9.

$$c = 0 \begin{cases} S_2(n) = O_1(n) = aI(n-1) + b_2 O_1(n-1) \\ S_1(n) = O_2(n) = b_1 O_1(n-1) + O_2(n-1) \end{cases}$$

$$c = 1 \begin{cases} S_2(n) = O_2(n) = aI(n-1) + b_2 O_2(n-1) \\ S_1(n) = O_1(n) = b_1 O_2(n-1) + O_1(n-1) \end{cases}$$

Les conditions aux limites sont :

$$y(0) = 0, y(1) = 0 \text{ et } \xrightarrow{i=1,2} O_i(0) = O_i(1) = 0$$

Le résultat du filtre récursif d'origine se trouve au point $S_1(n) = y(n-1)$ avec un différé d'un échantillon.

L'arithmétique distribuée entre dans la concaténation de l'additionneur et du multiplieur sur chacune des deux boucles par un même opérateur additionneur-multiplieur (Figure 3-9).

Sur le schéma donné Figure 3-10 on remarque que les tables de coefficients de l'arithmétique distribuée sont réduites à une seule valeur. Suivant que le bit est à 1 ou 0 le résultat de sortie se résume à 0 ou la valeur du coefficient B pondéré du poids de l'étage

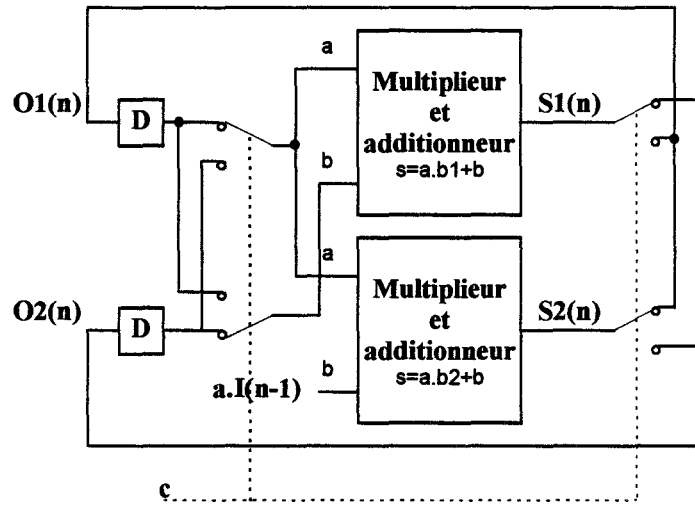


Figure 3-9 : Graphe de fluence du filtre de Deriché restructuré.

(décalage à gauche). L'accumulateur englobe la donnée à additionner dans le même temps. Cette structure optimisée en nombre d'opérateurs permet de réduire la surface des mémoires et même les éliminer. La combinaison de la ligne de commande du bit de poids i avec un opérateur logique AND, permet de masquer le coefficient lorsque le bit vaut 0 sinon il laisse transparent le coefficient. L'encombrement en surface est fortement diminué, de plus le temps de cette fonction est bien moins important que le délai d'adressage d'une mémoire.

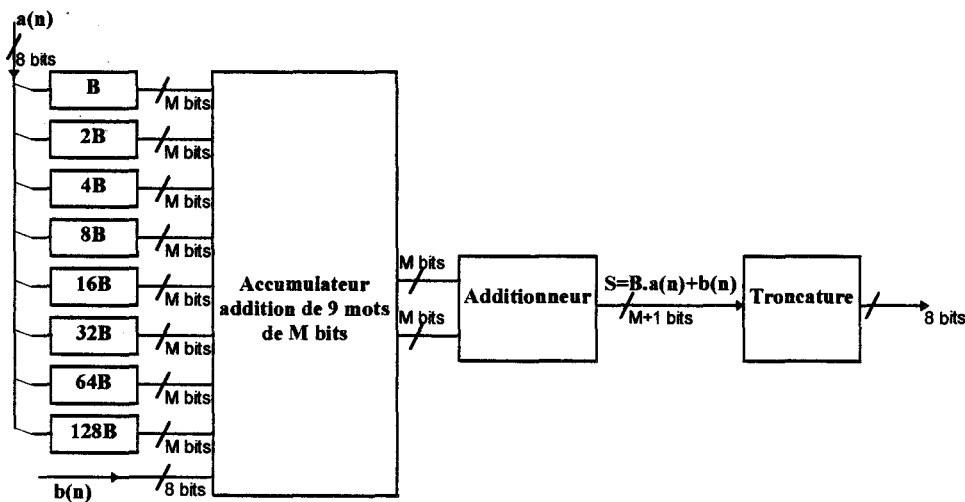


Figure 3-10 : Ensemble qui compose l'opération addition et multiplication $S=B.a+b$.

L'opération addition-multiplication demande pour temps de calcul:

$$T_{ma} = T_p + T_{acc} + T_{add}$$

T_p = Temps d'une porte AND

T_{acc} = temps d'une porte logique $\times$ profondeur du pipeline

T_{add} = temps de calcul d'un additionneur avec anticipation de carry

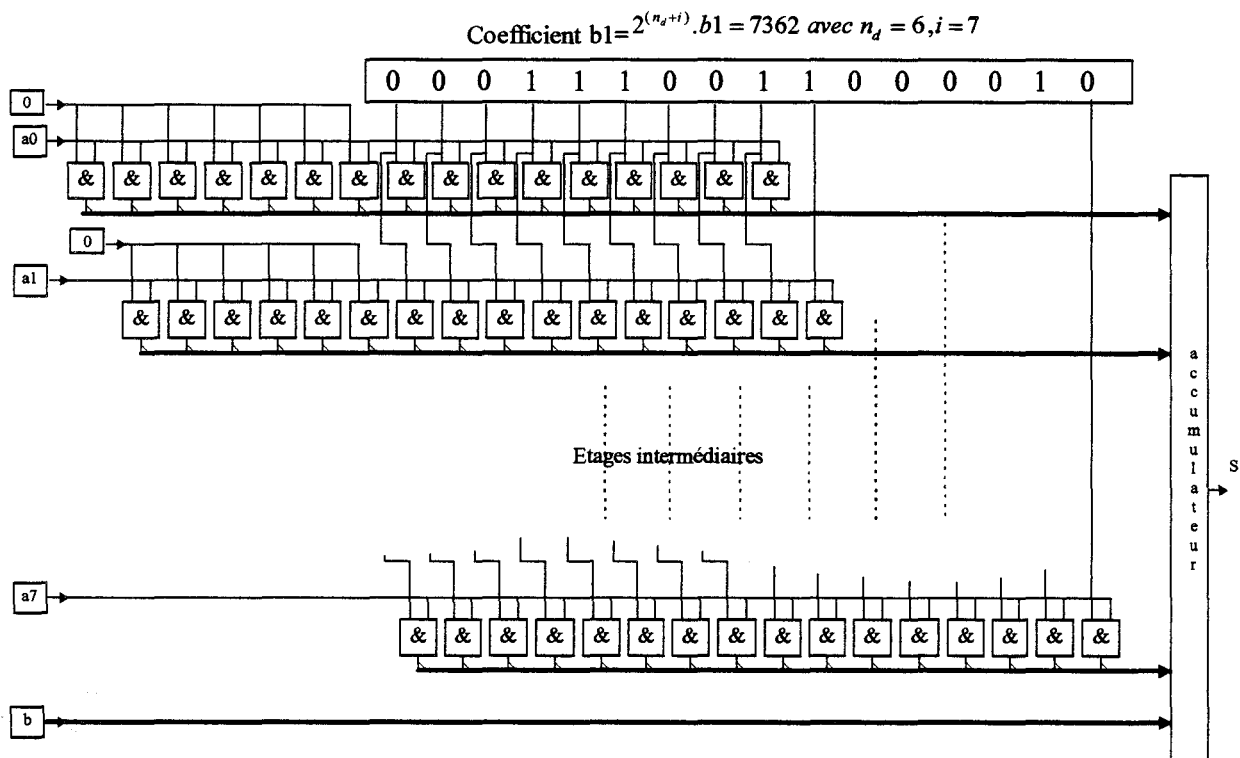


Figure 3-11: Structure sans mémoire. $S = a.b1 + b$.

L'additionneur-multiplieur que nous proposons a pour architecture la Figure 3-11.

Notre seconde démarche, après m'être sensibilisé à une possibilité d'implantation matérielle a été d'essayer de rechercher des fusions de techniques différentes permettant une implantation moins importante avec un résultat tout aussi satisfaisant.

La solution offerte par cette étude se démarque de la technique du laboratoire GERE dans le sens que la surface occupée doit être faible. Les effets de troncature du format des mots restent peu sensibles. L'efficacité de la double structure récursive montre qu'il est possible de parvenir à un résultat raisonnable en terme d'architecture au prix d'un travail de reconstruction et de conception spécifique.

3.3 Disproportion traitements - problèmes

Dans notre cas, ce qui nous intéresse : c'est la recherche de la pupille par une méthode la moins contraignante en temps de calcul. Le choix s'est porté naturellement par une localisation sommaire en employant la technique d'extraction par seuillage. Ensuite on effectue un dégrossissage des objets évalués dans cette image, ce qui permet de traiter localement par des opérateurs plus complexes. Les étapes et résultats sont exposés par la suite.

La Figure 3-12 montre l'image binarisée de l'image infra-rouge de l'oeil par la technique du seuillage bas. Nous avons la chance d'avoir une image très contrastée sur la position de la pupille noire avec la frontière de la sclérotique. L'image source peu bruitée après seuillage décrit une forme ellipsoïdale avec des trous inclus plus quelques contours de moindre importance.

Le candidat pupille s'obtient pour une forme circulaire entachée d'au moins un trou formé par les reflets cornéens. Cette attribution automatique de la forme de la pupille parmi les

autres objets contenus dans l'image est réalisée par une technique de chaînage des régions contenues dans la scène. Par une scrutation des éléments dans une liste chaînée, on peut extraire les éléments isolés et leur affecter des attributs spécifiques (localisation, forme, complexité de l'objet, trous contenus, etc.). La description de ces techniques personnelles est énoncée au chapitre suivant. Cette première étape a donc localisé la forme pupille et a estimé sa surface. Une seconde étape binarise de nouveau l'image originale dans un mode fenêtré englobant la pupille. Nous réduisons ainsi fortement le nombre de calculs pour ne se concentrer que dans une zone privilégiée. Le traitement par la technique de seuillage haut détermine la position des reflets autour et dans la pupille.

La Figure 3-13 montre l'image binarisée sur l'image complète après l'opération de seuillage haut. On s'aperçoit que de nombreuses formes viennent à se révéler d'où l'intérêt du traitement local.

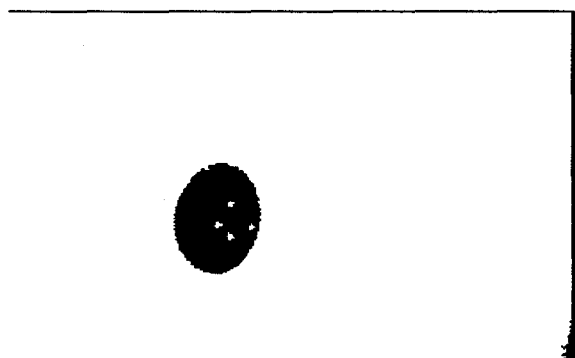


Figure 3-12 : Détection de la pupille par un seuil bas (Seuil=30).



Figure 3-13: Détection des reflets cornéens par un seuil haut (Seuil=130).

L'image illustrée par la Figure 3-14, est le résultat complet après traitement où l'on a effectivement positionné le centre de la pupille et localisé les quatre reflets cornéens.

La démarche d'extraction de la forme pupille par la technique du seuillage binaire montre des qualités tout à fait satisfaisantes au regard des résultats comparés avec la technique la plus performante de localisation de contour sous la méthodologie de Canny-Deriché (Figure 3-16).

On détermine le contour de la pupille et des reflets cornéens par un traitement localisé du filtre de Deriché sur la zone candidate à la forme pupille. On observe un contour lissé moins entaché d'erreurs de positionnement (Figure 3-15). Ce traitement local, sur une taille de fenêtre de 98*98 pixels est réduit en temps d'un facteur 27 par rapport à un traitement complet de l'image (512*512 pixels).

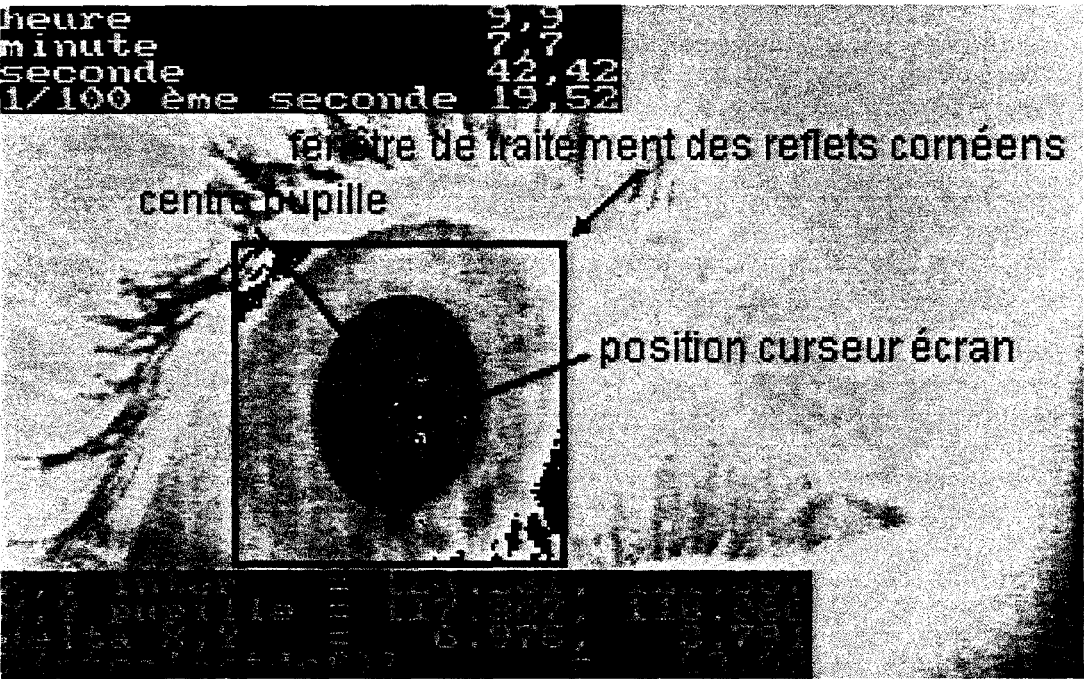


Figure 3-14 : Traitement complet. Localisation de la pupille puis sur une zone de 1,5 fois la taille de la pupille, extraction des reflets cornéens. La méthode employée d'extraction est celle d'un double seuillage adapté.

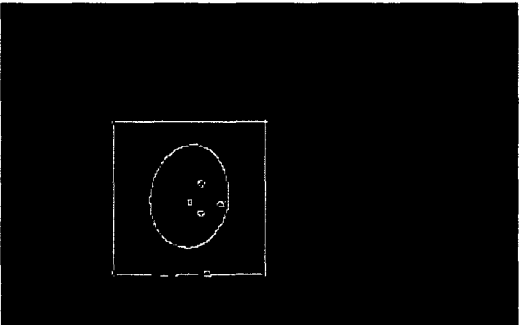


Figure 3-15 : Extraction locale des non maxima-locaux par le filtre de Deriche autour d'une zone privilégiée.

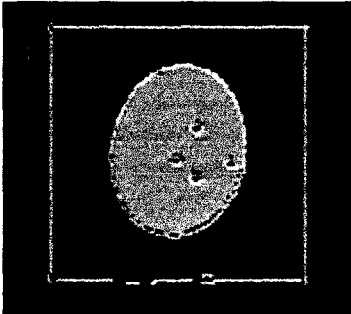


Figure 3-16 : Comparaison qualitative des contours extraits suivant la méthode du seuillage et celle de localisation de contour de Deriche.

3.4 Conclusion

Dans les premiers paragraphes on maintient un algorithme pour ses qualités évidentes, souvent au détriment d'une forte complexité de mise en oeuvre matérielle.

Notre choix n'étant pas d'être concepteur de circuits spécialisés finalisés, ce qui exige des compétences particulières, nous nous retournons vers des algorithmes moins coûteux.

Dans ce chapitre on n'a pas cherché à se contenter à de l'adaptation d'algorithmes en se renvoyant vers des traitements simplifiés, mais à vérifier qu'une image caractéristique peut également se traiter tout aussi efficacement par des techniques de segmentations moins onéreuses en terme de temps de calculs.

Chapitre 4

4. Technique originale de chaînage

Nous venons de vérifier que, dans le cadre de notre application, la technique du seuillage fournit une image binarisée d'un niveau de qualité très honorable à un coût insignifiant. Nous avons constaté que le suivi de contour sur une image binarisée ne nous convient pas car le temps de traitement varie en fonction de la complexité de l'image. Nous avons donc travaillé dans la direction ouverte par Pavlidis, à savoir le traitement séquentiel de l'image par étiquetage ligne des segments.

4.1 Introduction

Notre objectif est de repérer et caractériser la pupille dans une image video représentant une portion de visage. Il nous faut respecter les contraintes suivantes :

- ↳ le codage doit satisfaire tout niveau de complexité de l'image;
- ↳ le codage doit se faire sans perte d'information (codage réversible);
- ↳ en fin de codage, l'utilisateur doit disposer d'informations structurées lui permettant d'identifier chaque forme et d'en connaître les paramètres géométriques;
- ↳ les traitements doivent être effectués au rythme du balayage ligne de l'image video.

Dans la première partie de ce chapitre nous présentons la technique que nous avons mis au point et qui satisfait notre application sur le plan logique. Ensuite nous démontrerons que cette technique se prête particulièrement à la parallélisation et convient à un traitement « temps réel ».

4.2 Principes du contourage

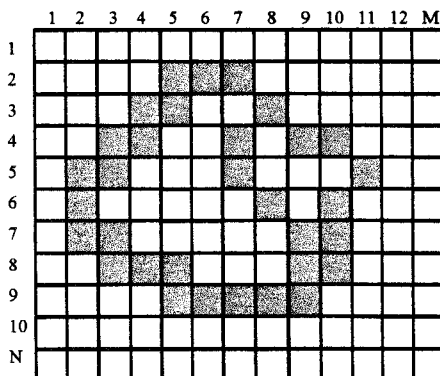


Figure 4-1: Image de référence.

Nous décrivons ici l'ensemble des traitements qui permettent de partir d'une image binarisée et d'aboutir à une liste exploitable par un traitement informatique classique. Cet ensemble de traitements est décrit pas à pas. Par la suite nous montrerons que la technique proposée convient parfaitement à un traitement « temps réel ».

Ce projet peut être résumé ainsi :

étant donnée une image, par exemple l'image de référence de la Figure 4-1, comment aboutir aux trois listes chaînées proposées dans le tableau qui suit.

N° de ligne	trait 1	trait 2	trait 3
2	H,1,Ø,5,3	Ø	Ø
3	I,1,2,4,2	H,2,1,8,1	Ø
4	I,1,Ø,3,2	I,2,3,7,1	H,3,2,9,2
5	I,1,Ø,2,2	I,2,Ø,7,1	I,3,Ø,11,1
6	I,1,Ø,2,1	I,2,Ø,8,1	I,3,Ø,10,1
7	I,1,Ø,2,2	B,2,3,Ø,Ø	I,3,2,9,2
8	I,1,Ø,3,3	Ø	I,3,Ø,9,2
9	B,1,3,Ø,Ø	Ø	I,3,1,5,5
10	Ø	Ø	B,3,Ø,Ø,Ø

(Le contenu du tableau est explicité dans la suite du chapitre)

Le remplissage automatique de ce tableau implique le passage par un certain nombre d'étapes logiques. Ces étapes sont expliquées séquentiellement ; en pratique ,et c'est le gros avantage de la technique que nous avons élaborée, ces étapes peuvent être exécutées en temps réel.

Un certain nombre de termes doivent être définis :

Ligne video : ligne horizontale d'une image. Dans le cadre de notre application, les pixels arrivent séquentiellement, en remplissant successivement toutes les lignes de l'image.

Segment : suite de pixels *objets* consécutifs sur une même ligne video. Par exemple, dans l'image de référence (Figure 4-1), la ligne 3 contient un segment de 2 pixels et un deuxième segment d'1 pixel.

Trait : Le trait est une séquence ininterrompue de segments *objets* adjacents sur une suite ininterrompue de lignes video. Il peut être constitué de pixels simples ou d'une forme quelconque complète sans évidements. Un cercle par exemple est formé de deux traits partant tous deux du haut et arrivant tous deux en bas.

Noeud : Les intersections, enfourchements, fusions ou divisions de segments sont matérialisés par des noeuds. L'image de référence comprend 4 noeuds.

Chaîne : C'est l'ensemble connexe de pixels *objet* formant l'objet dans sa totalité. L'image de référence contient une chaîne.

De nombreuses courbes sont formées de réunions de traits. Le traitement fournit un inventaire complet des traits contenus dans l'image. Ce traitement s'effectue en 5 points :

- Surcodage des pixels par fenêtres 3*2,
- codage des pixels par vecteur sémaphores,
- caractérisation des segments par balayage du vecteur « état »,
- compression et normalisation des informations liées au trait,
- chaînage des traits.

4.2.1 Surcodage du pixel par fenêtre 3*2

Ce surcodage consiste à enrichir l'information pixel. A l'information initialement contenue dans un pixel (0 ou 1 selon qu'il est *fond* ou *objet*), nous substituons une information caractérisant la fenêtre de dimension 3*2 environnant le pixel courant. Cette information est un nombre codé de 0 à 63 (2^6 configurations possibles). Le codage est effectué en pondérant chacun des 6 pixels de la fenêtre conformément aux indications de la Figure 4-2. Le pixel courant occupe la position « milieu bas ».

Avec M = nombre de pixels sur une ligne video,
et N le nombre de lignes video dans l'image.

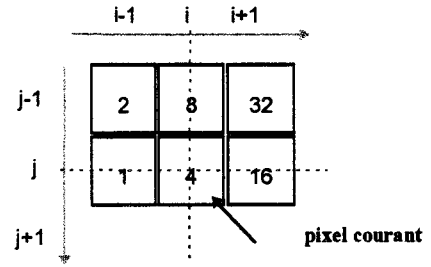


Figure 4-2: Code de pondération d'une fenêtre 3*2.

$$\begin{aligned} \text{code}[i, j] = & \text{fonction}(2^0 \cdot \delta_{\text{image}}(i-1, j) + 2^1 \cdot \delta_{\text{image}}(i-1, j-1) + 2^2 \cdot \delta_{\text{image}}(i, j) \\ & + 2^3 \cdot \delta_{\text{image}}(i, j-1) + 2^4 \cdot \delta_{\text{image}}(i+1, j) + 2^5 \cdot \delta_{\text{image}}(i+1, j-1)) \end{aligned}$$

$$\text{fonction} \in \{0, 63\}, \forall i \in [0, M-1] \text{ et } \forall j \in [0, N-1]$$

$$\delta_{\text{image}}(i, j) \in \{0, 1\} \text{ ou } \{\text{fond}, \text{objet}\}$$

Le système de pondération des pixels permet d'accélérer le codage d'un pixel connaissant le code du pixel situé à sa gauche :

$$\text{code}[i, j] = \text{code}[i-1, j] \div 4 + 2^4 \cdot \delta_{\text{image}}(i+1, j) + 2^5 \cdot \delta_{\text{image}}(i+1, j-1)$$

Il pourrait paraître plus judicieux d'étudier le pixel dans son environnement complet, en intégrant les 8 pixels voisins. Le choix d'éliminer les 3 voisins bas se justifie par la notion de « temps réel » : un certain nombre d'informations sont déjà exploitables en l'absence des 3 voisins bas. Le traitement séquentiel ligne permet de satisfaire à la continuité du codage.

4.2.2 Le vecteur « Sémaphore » des fenêtres

L'examen des 64 fenêtres révèle que ces motifs élémentaires ne sont pas compatibles avec n'importe quelle forme de courbes. Par exemple une fenêtre n° 20 () peut annoncer une courbe vers la droite ou plus bas, mais certainement pas l'achèvement d'une courbe située en haut et à gauche. Toute configuration d'une fenêtre donnée induit des possibilités et des impossibilités dans l'environnement immédiat.

Nous avons identifié un jeu de 9 sémaphores permettant de gérer tous ces cas « possibles ». Ce sont des informations booléennes que nous avons regroupé dans un vecteur dénommé « vecteur sémaphore du pixel ». Les 64 vecteurs sémaphores associés aux 64 fenêtres pixel permettent un traitement automatique du contourage de toute image binarisée.

4.2.2.1 Sémaphores liés au segment

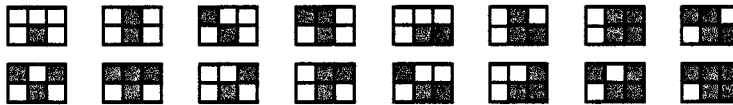
Au cours du balayage de la ligne video courante, deux variables marquent les extrémités du segment. Ces informations n'impliquent que le pixel courant et un pixel voisin : soit le gauche, soit le droit.

- **ds** : début de segment.

Il y a début de segment dès que le pixel courant est *objet* et que le pixel situé à sa gauche est *fond*.

$$ds = \overline{\delta(i-1, j)} \ \& \ \delta(i, j)$$

16 fenêtres sont dans ce cas :



- **fs** : fin de segment.

De façon symétrique, il y a fin de segment dès que le pixel courant est *objet* et que le pixel situé à sa droite est *fond*.

$$fs = \delta(i, j) \ \& \ \overline{\delta(i+1, j)}$$

16 fenêtres sont dans ce cas :



4.2.2.2 Sémaphores liés au type du segment auquel appartient le pixel

Un trait est composé d'empilements de segments sur des lignes video voisines. Ces segments peuvent avoir une place particulière dans le trait : Haut, Intermédiaire ou Bas ; on dira qu'ils ont le type H, I ou B.

- **H** : segment Haut. Le segment auquel appartient le pixel peut-il être le segment supérieur d'un trait ? Le pixel courant doit être *objet* et les 3 pixels supérieurs doivent être *fond*.

$$H = \overline{\delta(i-1, j-1)} \ \& \ \overline{\delta(i, j-1)} \ \& \ \overline{\delta(i+1, j-1)} \ \& \ \delta(i, j)$$

Parmi les 64 fenêtres possibles, 4 possèdent l'attribut H.



- **I** : segment Intermédiaire.

L'attribut I est vrai si et seulement si le pixel courant est *objet*. Il peut alors toujours être intégré dans un segment. Inversement le pixel *fond* ne peut pas appartenir à un quelconque segment ; son attribut I est donc faux.

$$I = \delta(i, j)$$

- **B** : segment Bas. Le segment auquel appartient le pixel peut-il être le segment le plus bas d'un trait ? Le segment Bas est défini sur la ligne video inférieure au segment Bas réel : c'est seulement en constatant que les pixels inférieurs au vrai segment bas sont vides que l'on peut affirmer que le bas du trait a été trouvé.

$$B = \delta(i, j-1) \ \& \ \overline{\delta(i-1, j)} \ \& \ \overline{\delta(i, j)} \ \& \ \overline{\delta(i+1, j)}$$

4 fenêtres possèdent l'attribut **B**.



4.2.2.3 Sémaphores liés aux noeuds

Au cours du balayage d'une ligne video, on rencontre des fenêtres compatibles avec une réunion de deux traits, ou une séparation en deux traits.

- **dn** : début noeud.

$$\begin{aligned} dn = & (\delta(i-1, j-1) \ \& \ \overline{\delta(i, j-1)} \ \& \ \delta(i, j) \ \& \ (\delta(i+1, j-1) + \delta(i+1, j))) \\ & + (\delta(i-1, j) \ \& \ \delta(i, j-1) \ \& \ \overline{\delta(i, j)} \ \& \ (\delta(i+1, j-1) + \delta(i+1, j))) \end{aligned}$$

6 fenêtres sont des débuts de noeud *réunion* : le pixel courant est actif, le pixel supérieur est vide, un trait peut provenir du haut gauche, un trait peut continuer sur la droite et un ou plusieurs traits peuvent continuer plus bas.



6 fenêtres sont des débuts de noeuds *séparation* : le pixel courant est vide, un trait provient du haut, un autre trait peut provenir du bas gauche, un trait peut continuer sur la droite.



- **fn** : fin de noeud.

$$\begin{aligned} fn = & (\delta(i+1, j-1) \ \& \ \overline{\delta(i, j-1)} \ \& \ \delta(i, j) \ \& \ (\delta(i-1, j-1) + \delta(i-1, j))) \\ & + (\delta(i+1, j) \ \& \ \delta(i, j-1) \ \& \ \overline{\delta(i, j)} \ \& \ (\delta(i-1, j-1) + \delta(i-1, j))) \end{aligned}$$

Les fenêtres ayant l'attribut **fn** sont celles symétriques de celles ayant l'attribut **dn**, par rapport à l'axe vertical de la fenêtre. 6 fenêtres sont des fins de noeud *réunion* :



et 6 fenêtres sont des fins de noeuds *séparation* :



4.2.2.4 Sémaphores liés aux types de noeuds

- **R** : segment réunion.

Un pixel a l'attribut Réunion lorsque deux branches distinctes peuvent arriver du haut sur le pixel courant. Le pixel courant est levé, le pixel supérieur est baissé, il y a au moins un pixel à gauche et au moins un pixel à droite.

$$R = \delta(i, j) \ \& \ \overline{\delta(i, j-1)} \\ \& \ (\delta(i-1, j-1) + \delta(i-1, j)) \\ \& \ (\delta(i+1, j-1) + \delta(i+1, j))$$

L'attribut **R** est levé dans 9 cas :



- **S** : segment Séparation.

Un pixel a l'attribut Séparation lorsque deux branches distinctes peuvent descendre de part et d'autre du pixel courant. Le pixel courant est baissé, le pixel supérieur est levé, il y a au moins un pixel à gauche et au moins un pixel à droite.

$$S = \overline{\delta(i, j)} \ \& \ \delta(i, j-1) \\ \& \ (\delta(i-1, j-1) + \delta(i-1, j)) \\ \& \ (\delta(i+1, j-1) + \delta(i+1, j))$$

L'attribut **S** est levé dans les 9 cas symétriques des cas de réunion par rapport à l'axe horizontal du pixel :



Deux branches distinctes peuvent se séparer au dessus du pixel courant et descendre de part et d'autre du pixel courant.

En résumé, les 64 fenêtres possibles ont chacune un vecteur « sémaphore » de dimension 9 ; par exemple le vecteur sémaphore de la fenêtre 13 () vaut 001010000, ce qui se décode ainsi de gauche à droite : H=0 ; B=0, I=1 ; ds=0 ; fs=1 ; R=0 ; S=0 ; dn=0 ; fn = 0.

En pratique, le code de la fenêtre ne sert qu'à la recherche rapide du vecteur sémaphore de la fenêtre. Les sémaphores sont donnés en annexe 1.

4.2.3 Recherche du type d'un segment par balayage par le vecteur ETAT

Au cours de cette étape, nous raisonnons *trait*. Lors de l'analyse d'une ligne video, on rencontre des segments. Quelle place occupe chaque segment dans un trait : le Haut, le Bas ou un niveau Intermédiaire. On rencontre également des fenêtres où toutes les conditions sont réunies pour avoir un noeud : une Réunion ou une Séparation. En rencontrant le premier pixel du segment, aucune de ces cinq informations n'est déterminée (sauf cas particuliers).

Nous sommes partis du principe qu'au départ un segment avait les 3 potentialités : Haut, Bas et Intermédiaire ; de même lors de la rencontre d'un profil de noeud, ce noeud avait les 2 potentialités, Réunion et Séparation. Au fur et à mesure qu'avance l'exploration du segment ou du noeud, en fonction des sémaphores des pixels rencontrés, ces potentialités disparaissent chaque fois qu'un sémaphore vient contredire la potentialité. En fin de segment seul reste un sémaphore concernant le type de segment (H, I ou B) ; à la fin d'une configuration de type noeud, soit il s'est avéré qu'il n'y avait pas de noeud, soit un seul sémaphore reste : R ou S.

Cette décantation des potentialités s'effectue par le vecteur état.

Ce vecteur état contient 7 informations booléennes :

Les 5 informations de base : Haut, Bas, Intermédiaire, Réunion et Séparation.

Une information **rs** (rupture de segment) indiquant au pixel suivant que le segment vient de s'achever.

Une information **rn** (rupture de noeud) indiquant au pixel suivant que le noeud vient de s'achever.

Ce vecteur état démarre initialisé à la gauche d'une nouvelle ligne ; les effets de bord de l'image sont neutralisés par la création d'un anneau de garde : un cadre de pixels virtuels *fond* est ajouté à l'image.

Pixel après pixel le vecteur état se déplace vers la droite en opérant avec les sémaphores des pixels. Chaque fois que le vecteur état opère sur un pixel, il affecte à ce pixel le vecteur état résultant, puis passe au pixel suivant.

L'opération *Vecteur état amont* par *Vecteur sémaphore* produit un *vecteur état aval*. L'opération s'effectue selon les règles suivantes :

(les indices e, pour entrée, désignent la valeur du vecteur amont ; les indices s, pour sortie, désignent la valeur du vecteur aval, les indices p, pour pixel, désignent la valeur du sémaphore)

$$H_s = (ds_p + H_e \& H_p) \& \overline{rs_e}$$

$$I_s = (ds_p + I_e \& I_p) \& \overline{rs_e}$$

$$B_s = (ds_p + B_e \& B_p) \& \overline{rs_e}$$

$$R_s = (dn_p + R_e \& R_p) \& \overline{rn_e}$$

$$S_s = (dn_p + S_e \& S_p) \& \overline{rn_e}$$

$$rs_s = fs_p$$

$$rn_s = fn_p$$

4.2.4 Compression et Normalisation des informations

Chaque pixel s'est enrichi du vecteur état résultat du *vecteur état amont* avec son propre sémaphore. Le travail de collection des informations peut démarrer. Sur chaque ligne, pour chaque segment on effectue les recherches suivantes :

chercher et enregistrer dans l'ordre la présence de Réunions ou Divisions.

Noter le type du segment (enregistré dans le vecteur état du dernier pixel du segment), sa position initiale et sa position finale.

Le résultat de ce travail est donné dans l'exemple suivant (Figure 4-3):

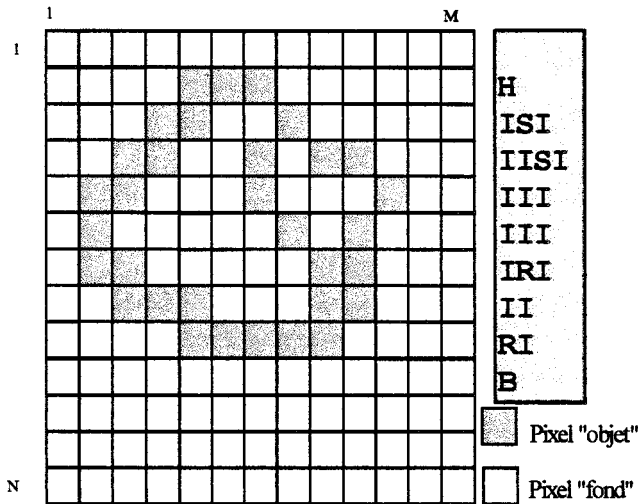


Figure 4-3 : Exemple.

Derrière chaque code H, I et B sont cachées ici deux informations numériques : l'abscisse du début du segment (par rapport au bord de l'image) et la largeur du segment.

A titre de comparaison, voici le résultat (Figure 4-4) des traitements précédents pour une image présentant la particularité d'un trait relié uniquement par sa partie basse :

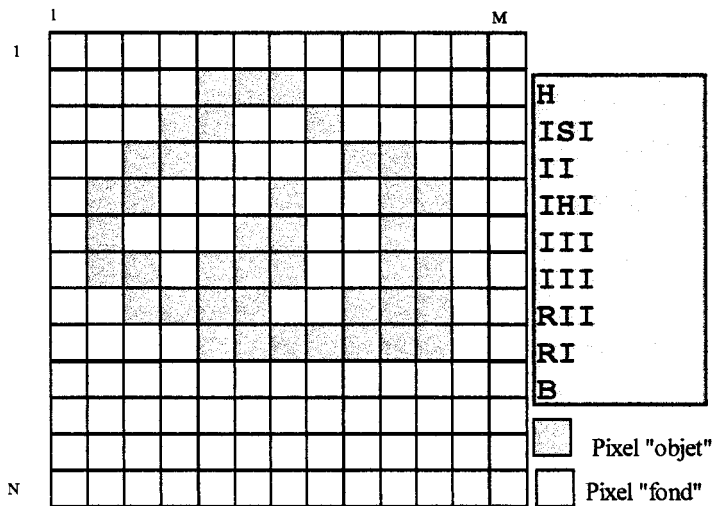


Figure 4-4 : forme avec concavité interne.

En fin de ce traitement, tous les traits sont reconstitués. Il reste la partie la plus importante : chaîner ces traits pour retrouver les vraies courbes.

4.2.5 Chaînage des traits

Un algorithme très simple permet maintenant de chaîner les traits de manière à pouvoir suivre toute courbe de bout en bout. L'étape précédente a fourni :

- Le type du segment (H, I ou B), avec les informations abscisse et largeur.
- Eventuellement l'information Séparation ou Réunion.

Il faut maintenant exploiter les informations Séparation et Réunion pour effectuer les divers chaînages.

Le résultat final sera enregistré dans un tableau final de dimension M (nombre de pixels par ligne) sur N (nombre de lignes). Cette forme d'enregistrement est rustique, ce qui permet aux utilisateurs électroniciens, automaticiens ou informaticiens de déployer leurs propres techniques de travail dans leurs propres contextes.

Les colonnes sont numérotées de 1 à M. La colonne de gauche est réservée aux traits les plus à gauche dans l'image. Au fur et à mesure que l'on trouvera de nouveaux traits sur une même ligne video, les colonnes à droite se rempliront et les traits recevront des numéros correspondant aux colonnes.

Chaque segment est décrit par une structure à 5 éléments :

- Type du segment (H, I ou B).
- Numéro du trait.
- Numéro du trait de chaînage (ou vide).
- Abscisse par rapport au bord gauche de l'image.
- Largeur du segment.

Pour l'image de référence donnée par la Figure 4-1, voici le résultat obtenu. Une séquence ISI engendre un nouveau trait, avec échange des numéros de chaînage. Une séquence RI ferme le trait de gauche, avec échange des numéros de chaînage. Un chaînage avec le trait « zéro » signifie « non chaînage ».

N° de ligne	ancien code	trait 1	trait 2	trait 3
2	H	H,1,0,5,3	0	0
3	ISI	I,1,2,4,2	H,2,1,8,1	0
4	IISI	I,1,0,3,2	I,2,3,7,1	H,3,2,9,2
5	III	I,1,0,2,2	I,2,0,7,1	I,3,0,11,1
6	III	I,1,0,2,1	I,2,0,8,1	I,3,0,10,1
7	IRI	I,1,0,2,2	B,2,3,0,0	I,3,2,9,2
8	II	I,1,0,3,3	0	I,3,0,9,2
9	RI	B,1,3,0,0	0	I,3,1,5,5
10	B	0	0	B,3,0,0,0

Il est toujours possible de proposer des images de plus en plus complexes. Nous nous bornons à traiter la seconde image proposée plus haut, qui présente la particularité d'une concavité interne.

N° de ligne	Ancien code	chaîne 1	chaîne 2	chaîne 3
2	H	H,1,0,5,3	0	0
3	ISI	I,1,2,4,2	H,2,1,8,1	0
4	II	I,1,0,3,2	I,2,0,9,2	0
5	IHI	I,1,0,2,2	I,2,0,10,2	H,3,0,7,1
6	III	I,1,0,2,1	I,2,0,10,1	I,3,0,6,2
7	III	I,1,0,2,2	I,2,0,10,2	I,3,0,5,3
8	RII	B,1,3,0,0	I,2,0,9,3	I,3,1,3,4
9	RI	0	I,2,3,5,7	B,3,2,0,0
10	B	0	B,2,0,0,0	0

L'ordre naturel d'apparition des traits n'est plus conservé dans le tableau ; cela peut gêner un opérateur humain, mais pas les outils de navigation automatique dans de telles chaînes.

En conclusion, l'outil de traitement ainsi constitué s'est adapté à toute forme d'image, jusqu'à la forme extrême du damier. Sur les listes chaînées ainsi fabriquées, nous avons pu effectuer de nombreux traitements : repérages, contourages extérieurs, calculs de barycentres et de surfaces, suppressions ou fusions de chaînes, etc ... Une représentation classique sous forme d'une liste chaînée est proposée en annexe 2 avec des notions de comptage des évidements et possibilité de circulation sur le contour extérieur d'une forme quelconque.

4.3 Implantation algorithmique sous forme parallèle

Dans une recherche d'accélération du traitement spécifique du chaînage on se propose d'étudier une structure on l'on puisse mettre en avant le parallélisme potentiel contenu dans cet algorithme.

Le but est d'allouer les calculs qui sont indépendants sur des structures matérielles qui sont mises en parallèles afin de multiplier le facteur d'efficacité de calcul. Un calcul d'un temps d'exécution T peut se voir diminuer au maximum de T/N , N étant le nombre d'opérateurs mis en parallèles. L'efficacité d'une architecture parallèle n'atteint que rarement le facteur T/N . Les causes principales des pertes sur le facteur d'efficacité sont d'une part les temps de communications entre opérateurs parallèles et d'autre part que certains calculs sont indissociables.

4.3.1 Notion de temps réel

Un système réactif est régi par des contraintes de temps sur l'évolution des paramètres influant sur l'état du programme.

L'étude de la mise en oeuvre d'une application *temps réel* revient à maîtriser le comportement temporel de l'architecture. Le temps d'exécution sur chacun des opérateurs permet d'estimer la cadence maximale de l'horloge que peut supporter la structure entière avec l'espoir que le circuit obéit aux contraintes temporelles.

La cadence maximale est obtenue par l'obtention du temps qui offre un résultat stable à la sortie d'un module ou opérateur élémentaire. Ce temps associé avec sa structure doit être en relation directe avec le temps de latence du dispositif afin de conserver la synchronisation entre les échanges de signaux internes à l'architecture.

Dans une configuration parallèle vraie, il faut pouvoir vérifier la convergence des résultats séparés afin de permettre d'effectuer des opérations élémentaires avec une présence des signaux sans faille. Chaque opération a un temps d'exécution propre, d'où l'impossibilité d'échanges physiques des données de manière anarchique. Le travail de conception sur une structure matérielle repose en partie sur des associations d'opérations élémentaires en vue d'atteindre un pseudo-synchronisme temporel qui vérifie à coup sûr la préservation des informations ne s'échangeant pas au même rythme.

La conservation d'une structure asynchrone est dangereuse pour l'étude d'une architecture dédiée et n'est conservée que dans le cas restrictif d'évolution intemporelle. Il est évident que l'absence de synchronisme ne permet pas d'observer l'état du dispositif à un instant donné puisque les échanges se font sans contrôle et de manière désordonnée.

Le fait qu'un circuit soit dépourvu de cadencement séquentiel n'implique pas qu'il faille le classer dans la catégorie des circuits dit « asynchrones ». L'état du process peut être pris en compte par un ordonnancement à des instants précis.

La notion de contrainte temporelle est la seule qui influe sur l'étude d'une architecture d'un circuit temps réel. Au niveau du concept, on doit pouvoir vérifier par simulation ce qui peut être englobé dans un synchronisme des données évoluant dans le graphe.

La démarche générale est de spécifier l'application vérifiant des propriétés logiques avant toute étude du support matériel qui fusionnera la structure, ayant lui même des contraintes physiques spécifiques.

4.3.2 Recherche d'un traitement parallèle

Avant toute proposition de disposition parallèle il faut pouvoir décrire le dispositif sous forme fonctionnelle.

4.3.2.1 Description fonctionnelle

La description d'une application peut se représenter sous forme de tâches et de ses relations de dépendances.

Une tâche peut se composer d'opérateurs élémentaires (non divisibles) représentés sur un niveau inférieur et concaténés dans une structure globale. Il existe des dépendances entre ces opérateurs par des relations de précédence ou descendance.

La description fonctionnelle d'un algorithme peut être donc établit par un graphe de dépendance entre tâche [ROY,70][□].

4.3.2.2 Graphe de dépendance

Le graphe se décompose en éléments constituant les tâches (fonctions) par des sommets et des arcs (échange de signaux) orientés reliant les sommets. La construction d'un graphe et les notions s'y reportant sont documentées dans [CHO,94][○]. Un exemple de construction de graphe de dépendance vient illustrer les propos (Figure 4-5).

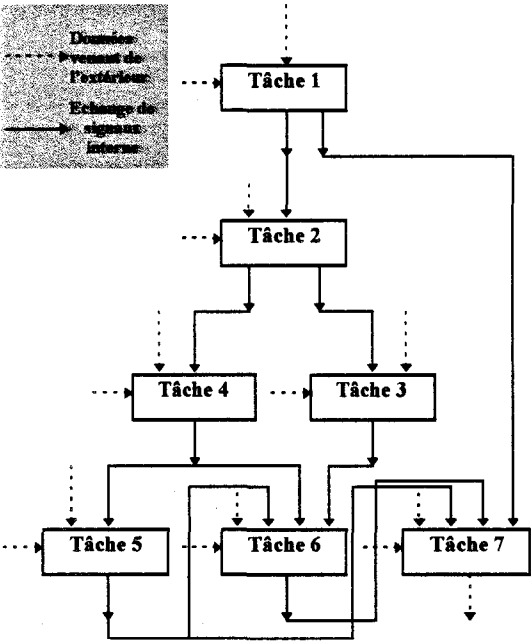


Figure 4-5 : Description fonctionnelle sous forme d'imbrication de tâches.

L'avantage de cette représentation est de pouvoir distinguer les tâches pour un sommet donné qui lui sont précédantes ou succédantes.

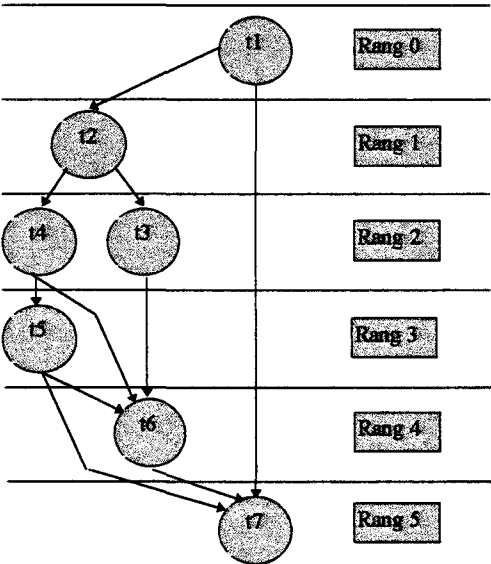


Figure 4-6 : Réseau d'activité.

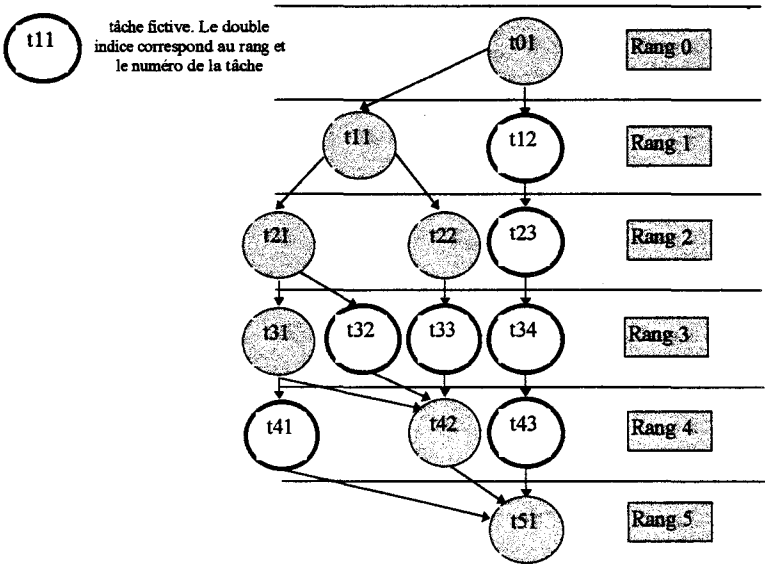


Figure 4-7 : Graphe de dépendance.

La notion de rang indique le nombre de sommets qui précèdent une tâche. La représentation sous forme d'un réseau d'activité indique la hiérarchie qui s'impose dans la décomposition de l'algorithme (Figure 4-6).

L'ajout de tâches fictives permet de révéler un synchronisme temporel total, à chaque rang du graphe de dépendance (Figure 4-7).

Le placement et l'ordonnement sont facilités grâce à la représentation du graphe de dépendance et peuvent s'opérer automatiquement par des outils tels que (SYNDEX) [LAV,89] [⊗], [LAV,94] [⊗] vers une structure dédiée. L'outil permet connaissant le temps d'occupation sur chaque opération s'il faut fusionner ou non des opérateurs afin d'atteindre une meilleure adéquation sur les contraintes temporelles (temps réel) ou matérielles (nombre de processeurs) du circuit.

La concaténation de fonctions qui englobent plusieurs opérateurs élémentaires ou fonctions, nous permet de décrire le graphe sous des niveaux supérieurs (granularité), tout en conservant la notion de rang afin de préserver une exécution synchrone.

L'optimisation de complexité d'architecture est permise afin d'évaluer un temps d'exécution optimisé ou un nombre de ressources réduites.

Dans les descriptions à venir, on conserve une approche de spécification synchrone. Les actions élémentaires ont une durée d'exécution instantanée. Dans ce cas, la description permet de conserver le rythme de l'entrée à la sortie de l'opérateur qui n'a qu'un aspect transformationnel. Cette description est totalement différente de celle des graphes de tâche, la notion de temps est exclue.

4.3.2.3 La formulation flot de donnée

La méthodologie *flot de donnée* est exploitée au sein de l'équipe de travail GT6 du GDR 134 auxquels nous participons sur l'aspect plus particulier de *l'adéquation algorithme et architecture* sur les algorithmes de traitement d'images.

La représentation *flot de données* s'effectue par une représentation symbolique de sommets et d'arcs de transfert orientés. Les sommets sont les fonctions et les arcs les transferts de données entre sommets. L'avantage de cette description est de révéler le parallélisme potentiel de l'algorithme.

La formulation flot de données fait apparaître la mémoire d'état mais également la mémoire de données de l'algorithme de façon explicite.

Le flot de données permet d'intégrer implicitement la notion d'itération et le flot des entrées la notion d'événement.

Pour une représentation flot de contrôle en version organigramme ou automate, les données sont manipulées par des actions et ne sont pas déclarées explicitement. Cette modélisation de l'algorithme induit un ordre total et sera différenciée à la représentation flot de données qui aboutit à un ordre partiel.

Le langage synchrone SIGNAL [BEN,90] [⊗] est un langage *flot de données*, construit autour d'opérateurs de base, sur lequel nous nous sommes inspiré pour définir un outil de spécification de nos algorithmes.

4.3.3 Le concept flot de donnée

Dans un concept *flot de donnée* les instructions sont actives sur les sommets lorsque les données sont disponibles.

L'exécution est représentée par un graphe orienté qui met en évidence le flot de données entre les instructions.

L'instruction contient un opérateur, des opérandes et des résultats. La particularité du flot de donnée est qu'il n'y pas de sauvegarde des données partagées comme dans le cas de manipulation de variables dans une description de flot de contrôle (séquentiel).

Les deux dispositifs, flot de contrôle et flot de donnée sont illustrés par un exemple. On voit qu'il n'y a plus de partage des résultats conservés dans une mémoire partagée pour la représentation flot de donnée (Figure 4-8).

Notre recherche est de parvenir au parallélisme maximum. Le séquencement des blocs d'instructions est uniquement lié à la dépendance des données, sans tenir compte des réalités de synchronisme temporel car il n'y a plus d'état d'attente.

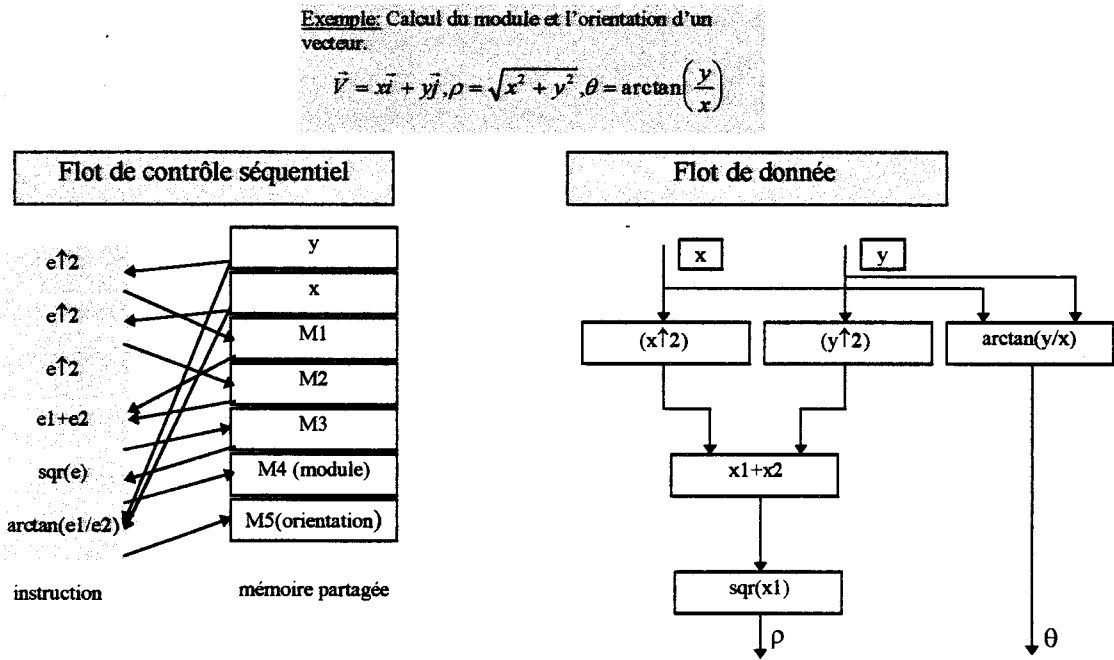


Figure 4-8 : Calcul flot de contrôle et flot de donnée.

4.3.3.1 Le langage SIGNAL

La syntaxe de SIGNAL permet de décrire des relations entre des suites d'événements à des instantanés de présence du signal.

Les relations sont définies sur la base de relations de précedence entre les événements, dépendance fonctionnelle et simultanéité entre événements sur des signaux différents. Signal décrit également les relation de dépendances opératoires, dont certaines peuvent être conditionnées, qui définissent les signaux de sorties représentant les résultats d'opérations exécutées sur des signaux d'entrée.

Le synchronisme est vérifié quand les événements de deux signaux sont simultanés. Ce qui est implicite en langage synchrone puisque l'on suppose que les sorties produites par une dépendance opératoire sont simultanées avec ses entrées. Les caractéristiques matérielles des opérateurs ne sont pas pris en considération puisque l'on fait abstraction du temps.

On dit que deux signaux sont synchrones si leurs événements sont deux à deux simultanés. Cette définition conduit à une relation d'équivalence. La classe d'équivalence d'un signal est son horloge qui précise la présence ou l'absence relative de ce signal à d'autres. Les relations temporelles de base sur les signaux sont déterminées par du calcul ensembliste.

On dispose de quatre opérateurs de base et un opérateur de composition réalisant des connexions par identité de nom de signaux identiques.

Ce concept de description permet de construire l'algorithme sous la forme d'une programmation modulaire hiérarchisée.

L'adéquation algorithmes vers l'architecture exige d'exploiter avec une méthodologie adaptée la recherche des parallélismes spatiaux et temporels d'un traitement en vue d'une architecture parallèle. Les langages de spécification type SIGNAL, langage synchrone, sont largement utilisés dans le cadre d'une description d'algorithmes de traitement du signal et de l'image.

Ne maîtrisant pas l'outil SIGNAL, par une infrastructure matérielle ne nous permettant pas son exploitation, nous nous sommes permis de définir une syntaxe proche de celle du langage SIGNAL pour vérifier l'efficacité de notre description parallèle. La validation de notre structure a pu être vérifiée par le développement de cette architecture en un programme en langage C, contraint à un synchronisme sur chaque étage d'un réseau multi-pipeline.

4.3.3.1.1 Syntaxe et sémantique

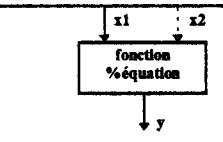
La syntaxe de description est composée d'identificateurs symboliques qui affectent une propriété au commentaire qui la suit. Les différents symboles sont désignés entre parenthèses :

- ⇒ (?) une entrée.
- ⇒ (!) une sortie.
- ⇒ (%) commentaires.
- ⇒ (|) association d'opérateurs dans un processus.

4.3.3.1.2 Fonctions

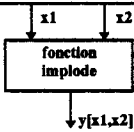
✓ Les fonctions immédiates

Ce sont des fonctions transformationnelles (pas de mémoire d'état). Les valeurs de sorties ne dépendent que de la présence ou non du flot d'entrée. Le rythme des entrées et des sorties sont tous égaux. Les opérateurs arithmétiques et logiques sont des exemples de fonctions immédiates.

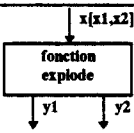
symbole	
Equation	exemple: $y:=x1-x2$
Ecriture	Fonction(?x1,x2!y)

Des notions de structures de flot d'entrées et de sorties sont exprimés par les fonctions immédiates agrégation et désagrégation:

* Agrégation: Regroupement des flots d'entrées de même type vers un flot de sortie structurant l'ensemble.

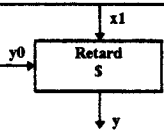
symbole	
Ecriture	Implode(?x1,x2!y)

* Désagrégation: démantèlement d'une structure d'un flot d'entrée vers plusieurs flots de sortie de même type.

symbole	
Ecriture	Explode(?x!y1,y2)

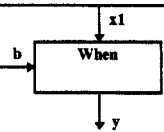
✓ Fonction retard

C'est le seul opérateur qui fait appel de l'état passé. Le rythme et le type de la sortie sont égaux au flot d'entrée mais décalé d'une donnée, avec possibilité d'une valeur initiale.

symbole	
Ecriture	$\$(?x1!y) \text{ init } y0$

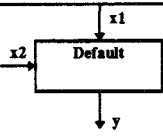
✓ Fonction extraction ou condition

Les rythmes des deux entrées peuvent être différents. Le type du flot d'entrée et de sortie est équivalent. Seule la commande est un booléen. Le flot de sortie est un sous-échantillonnage du flot d'entrée pour les données vraies du signal de commande.

symbole	
Ecriture	When(?x1,b!y)

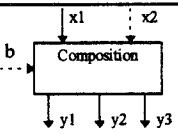
✓ Fonction mélange

Le rythme du flot de sortie est l'union des rythmes des deux flots d'entrée, avec une priorité donnée à x1. Tous les types de signaux sont égaux.

symbole	
Ecriture	Default(?x1,x2!y)

✓ Opérateur de composition

La composition correspond à l'union de systèmes d'équations, l'opérateur de composition est commutatif et associatif. Il est possible de spécifier des comportements ne faisant appel qu'à un seul indice temporel.

symbole	
équations	exemple : $S:=x1+x2$, $D:=x1-x2$, $y1:=S.x1$, $y2:=D.x2$, $y3:=y1$ when b
Ecriture	$S=\{?x1,x2,b!y1,y2,y3\}$ $(S:=x1+x2 D:=x1-x2$ $ y1:=S.x1$ $ y2:=D.x2 $ $ y3:=y1 \text{ when } b)$

4.3.4 Application au codage d'une image binarisée

La solution apportée est l'étiquetage des segments suivant notre méthodologie afin d'opérer le chaînage des traits suivant une structure de liste chaînée.

Le rôle de cette étude est de montrer qu'il tout à fait possible de générer une liste d'étiquettes au rythme de l'échantillonnage ligne [CHR,94][⊗]. Ce qui implique un facteur de rapidité accrue de la taille de la largeur de la ligne. L'architecture a une structure en réseau multi-pipeline. Tous les processus pixels ont des liens de communication ligne. Les communications se font à chaque niveau du pipeline, ce qui permet de recevoir un flot d'entrée du processeur voisin sur un rang du pipeline supérieur et envoyer à son tour après un traitement local vers le deuxième voisin sur un rang de niveau supérieur.

La description suivante exprime le parallélisme en M sous-graphes parallèles attachés chacun à un pixel sur la ligne de référence (Figure 4-9).

Le flot de données d'entrée est de type ligne produit par le sommet lignej :

lignej;retard(?lignej !lignej-1) ligne0;

explode(?lignej !p1..pn)

explode(?lignej-1 !p'1..p'n)

fonction(?p2,p'2,p1,p'1,0,0!code1),fonction(?p3,p'3,p2,p'2,p1,p'1!code2),,,,,fonction(?0,0,pn,p'n,pn-1,p'n-1!coden+1);

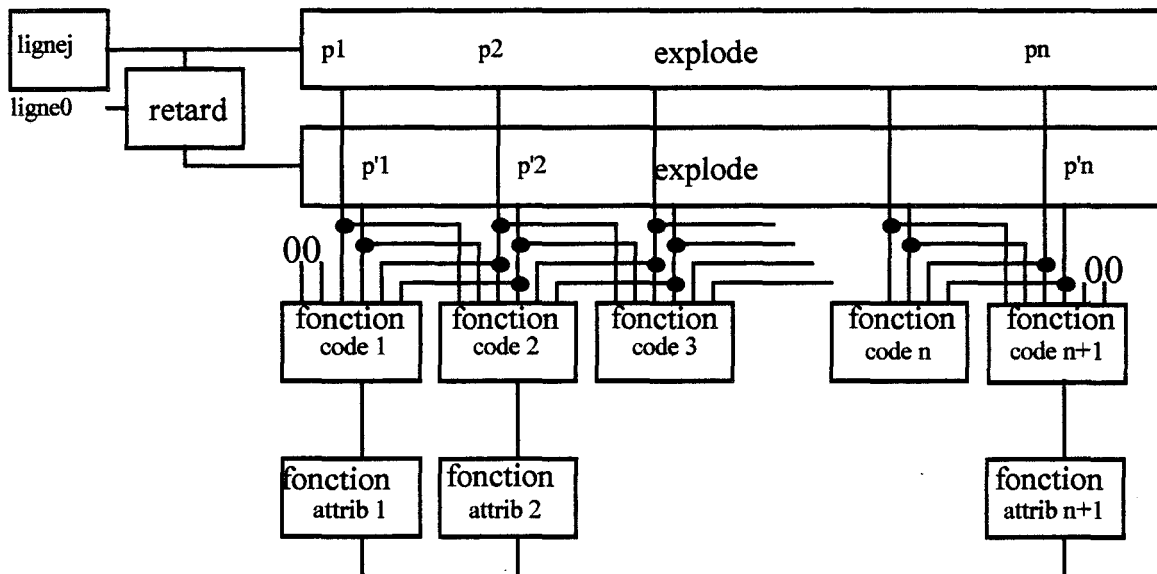


Figure 4-9 : Sommet du graphe qui met en évidence le traitement parallèle de M sous-graphes.

La Figure 4-9 met en relief l'aspect du surcodage des pixels au travers des notions de voisinage des éléments de la fenêtre 3*2. Il est ainsi possible d'opter au codage du pixel attaché à son graphe par son vecteur sémaphores.

Afin d'exprimer au mieux les effets des fonctions contenues dans l'architecture, on ajoute deux expressions composées sur la base des fonctions primaires du langage Signal flot de donnée.

4.3.4.1 Expression de deux sommets ajoutés à la liste des sommets habituels

* Fonction de sous-échantillonnage conditionnée par un booléen actif complémenté faux.

écriture : Whenc (?entrée,booléen!sortie).

* Fonction d'aiguillage.

écriture: Multiplex(?entrée1,entrée2,booléen!sortie) Ensemble d'un mixage avec un booléen de contrôle, priorité à l'entrée1.

4.3.4.2 Explication de l'architecture du graphe

Nous avons un parallélisme au niveau pixel. La ligne est éclatée en M sous-graphes (Figure 4-10) absorbant à leur sommet les attributs renvoyés par la table de correspondance de la fonction *attrib*. Cette table est identique à celle de l'annexe.

Le graphe pixel laisse apparaître deux traitements distincts selon que l'on opère sur un segment ou sur un noeud.

① Attributs du segment: $\{ds, fs, H, B, I\}$.

② Attributs du noeud: $\{dn, fn, R, S\}$.

Les sommets *When* et *Whenc* supérieurs conditionnent l'exécution des sommets inférieurs pour un codage du segment.

Dans le cas où le segment est de la taille d'un pixel les booléens extrémités début et fin de segment sont vrais pour le même pixel, le sous-graphe va produire un flot de sortie segment.

Pour un segment de taille supérieure à un pixel, plusieurs graphes vont intervenir dans l'identification du segment pour produire un flot de sortie unique à l'endroit où le segment débute. L'extrémité *fin de segment* repérée au sous-graphe positionné en $x(fin_seg)$ va transmettre ses attributs à son voisin de gauche qui absorbera les données en différé d'une donnée. Le sous-graphe contacté au sommet de son graphe traite déjà la ligne suivante, le flot reçu à sa droite est analysé à la couche inférieure. Il opère de la même manière que le graphe précédent en contactant son voisin avec le résultat des informations de la ligne courante, plus la comparaison de son état précédent avec les attributs voisins transmis. Le résultat est absorbé par le graphe en $(x(fin_seg)-2)$ sur la couche inférieure -2.

Un pixel en position (X) peut être l'élément d'un segment d'une largeur (M+1-X.). Il doit pouvoir recevoir à la couche (M+1-X) le flot *attribut* envoyé par le graphe (M+1) transmis de graphe en graphe en différé de (M+1-X) donnée.

Cette étape constitue la caractérisation du segment par échange des sous-graphes du vecteur état.

Un mode opératoire supplémentaire permet de véhiculer de graphes en graphes la position du pixel *extrémité fin de segment* vers le graphe *début de segment* afin de connaître la longueur du segment.

Le flot de sortie *type segment* déterminé par les fonctions *seg* se calcule sur une diagonale (Figure 4-11). Afin de synchroniser toutes les informations d'une même ligne dans une agrégation on décale les données vers la gauche en écrasant les sous-graphes qui ne génèrent pas de flot de sortie de leur sommet *seg*.

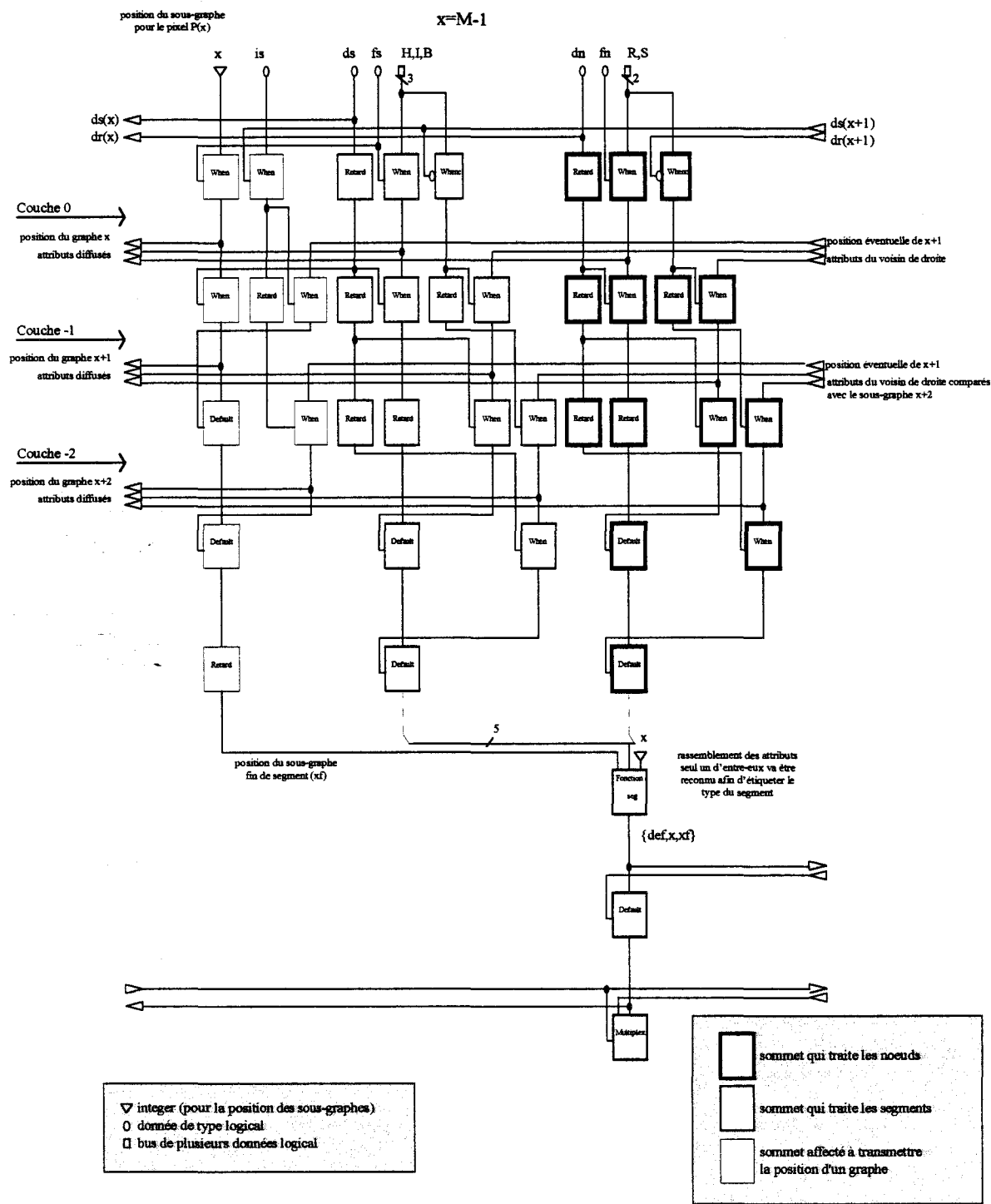


Figure 4-10 : Graphe flot de donnée d'un sous-graphe attaché à un pixel en position $m-1$.

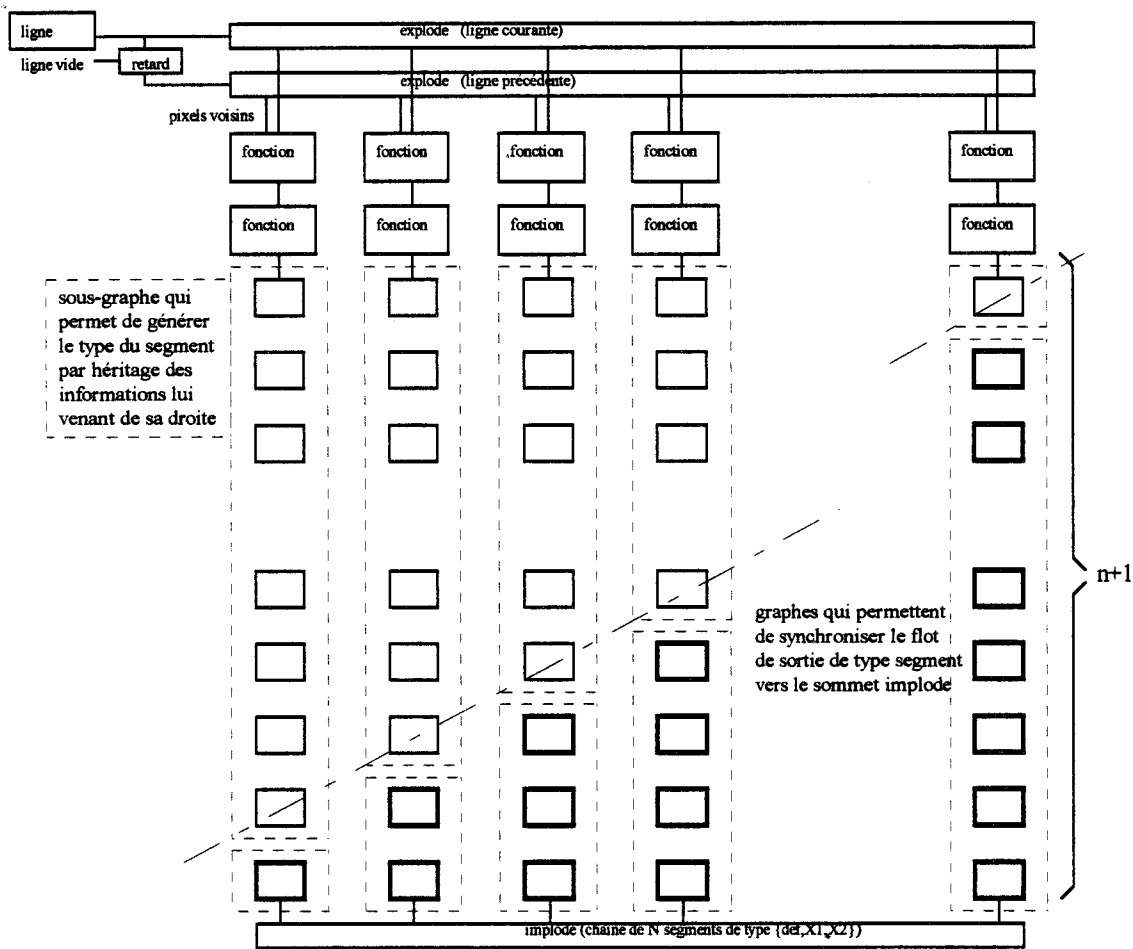


Figure 4-11 : Architecture complète sans les communications internes à chaque sous-graphes.

Le réseau multi pipeline avec des communications transversales à chaque rang ne permet pas une représentation explicite de la structure car de nombreuses liaisons encombreraient le dessin de la structure. Néanmoins cette figure montre l'aspect massivement parallèle et l'implication des fonctions suivant le rang du sous-graphe mis en jeu.

Une construction plan par plan est détaillée pour un traitement d'une image test à la Figure 4-12.



Figure 4-12 : Visualisation des modifications temporelles afin de construire le tableau de sortie des segments étiquetés.

La Figure 4-12 montre le cheminement des opérations telles qu'elles seraient réalisées pour un traitement ligne détaillé au travers de la structure parallèle avec conservation des données à chaque étage des graphes. Puisqu'il nous faut communiquer des éléments qui se produisent en sommet de chaque sous-graphes de manière simultanée, il faut procéder à un retard implicite afin de communiquer au travers de la structure des éléments ayant même origine mais à des temps de traitement ligne différés, ce qui revient à exercer une rotation de l'image d'origine d'un pixel différé sur chaque pixel voisin de la ligne.

De ce nouveau plan image, il suffit de procéder à l'étape du codage du vecteur état. La compression des codes retenus sur l'extrémité gauche d'un segment par une étiquette spécifique revient à faire glisser sur la diagonale chacun des éléments en occupant les espaces libres par un éventuel élément existant sur la droite.

Soit, la ligne est vide, d'où non présence de segments à chaîner ou apparition d'une séquence d'étiquettes représentatives de la forme codée et générée au travers d'une liste permettant une exploitation de primitives facilitant la recherche d'une forme spécifique.

4.3.4.3 Résultat du traitement d'une image test

numéro	'pixel fond',X 'pixel objet'												ligne traitée
ligne	graphe												position pixel
	0	0	0	0	0	0	0	0	1	1	1		chaîne de sortie
	2	3	4	5	6	7	8	9	0	1	2		0000000001111
													1234567890123
1													
2													...XXX...
3													..XX...X....
4													..X...X.....
5													..X..X.....
6													..X...X.....
7													..X...X.....
8													..XXX...X....
9													..X...X.....
10													..XXX...X....
11													..XXX...X....
12													..XXX...X....
13													
14							I						
15													...XXX...
16							H	I					..XX...X....
17							S						..XX...X.XX..
18							I	I	I				..XX...X...X..
19							I						..X...X.X...
20							I	I	I				..X...XX...
21							I						..XX...XX...
22													..XXXXXX...
23							I	I					..XXXXXX...
24							R						..XXXXXX...
25							I	I					..XXXXXX...
26							I	I	I	B		I	..XXXXXX...
27							I	I					..XXXXXX...
28							S	I					..XXX.....
29							H	S	I				..XX...X....
30							I	I	T				..X...X.....
31							I			I			..X...X.....
32													..X...X.....

Tableau 4-1 : Résultat d'un traitement parallèle sur des images de références.

Le Tableau 4-1 est la représentation des résultats à chaque nouvelle ligne du flot d'entrée et à chacun des étages du réseau pipeline. Ce calcul a été obtenu par le développement personnel d'un pseudo programme synchrone sur la base de la syntaxe Signal.

Chaque étage est traité de bas en haut à l'intérieur du pipeline par rapport à l'événement antérieur d'une ligne traitée précédemment. On s'impose un rythme du flot de données de la ligne vidéo qui cadence les flots internes par un synchronisme établi à chaque rang du réseau. La présence ou non d'une donnée est vérifiée par un troisième niveau logique par un flag de présence. Ce programme, écrit en langage C, a le mérite de vérifier la bonne communication interne dans le réseau grâce à la visualisation des événements à l'intérieur du pipeline.

4.3.4.4 Conclusions sur l'efficacité d'une implantation massivement parallèle

Le graphe flot de données montre bien le parallélisme complet d'une ligne éclatée en $M+1$ graphes attachés à un pixel. Le rythme d'entrée de type ligne est conservé, on produit un flot de sortie identifiant les segments avec l'héritage de la définition des chaînes précédentes (vecteur état).

La profondeur du graphe est de la taille du nombre de pixels sur la ligne ce qui signifie un résultat chaîné en différé de $(M+1)$ vis à vis de l'entrée. La technique employée s'adapte facilement à un traitement parallèle mais elle est coûteuse en surface de silicium. Il nous faut encore trouver une architecture matérielle des différents opérateurs qui puisse laisser envisager une implantation *temps réel* sur VLSI.

4.4 Conclusion

La technique retenue de chaînage nous permet d'envisager les traitements de localisation et de reconnaissances des formes contenues dans l'image sans avoir à procéder à des calculs supplémentaires fastidieux. Le positionnement des objets, la détermination des surfaces, le dénombrement des évidements internes sont intégrés dans le processus de chaînage. La description du parcours de la frontière extérieure délimitant la forme est accélérée par un codage de l'objet en listes chaînées. On ne survole que les éléments de sommets, vallées et enfourchements du squelette de la forme impliquée. Le marquage et le suivi de contours qui impliquent des traitements en relation avec la complexité de l'image codée sont exclus.

Dans cette étude, on propose d'élargir le concept en intégrant les paramètres spatiaux et temporels du chaînage afin de permettre un calcul sur la ligne complète. Le formalisme flot de données nous a permis de dégager le parallélisme potentiel de notre algorithme et à formaliser un graphe synchrone. On propose une structure massivement parallèle sous la forme d'un réseau multi-pipeline qui supporte un codage des étiquettes segments lignes dans le rythme de la ligne. La validité de cette structure a été établie par une simulation à partir d'un programme écrit en langage C qui visualise le bon comportement de l'architecture face aux différents événements.

Chapitre 5

5. Dispositif de pointage sur un écran informatique

5.1 Introduction

Comme présenté dans le premier chapitre, notre objectif est de concevoir un dispositif s'appuyant sur les propriétés des mouvements oculaires. Le dispositif de pointage informatique doit permettre de se substituer à la souris manuelle classique par un traitement qui permet d'analyser les mouvements oculaires. Cette application vient s'insérer comme nouvel outil d'interaction homme-machine afin d'offrir une nouvelle dimension à la communication : la commande visuelle.

Notre souci majeur est de laisser une grande liberté à l'utilisateur de ce genre d'interface. La tête est libre de tout mouvement quand la prise d'informations oculaires n'est plus effective. Lorsque le patient souhaite pointer le curseur sur une zone à désigner, il présente son oeil dans une aire d'acquisition, délimitée au moyen d'une caméra fixe proche de l'oeil, et porte son regard sur l'écran.

L'action de ce dispositif est limitée à un pointage absolu sur un support visuel fixe ce qui permet d'atteindre quelques libertés de fonctionnement. Contrairement aux différents systèmes présentés dans le premier chapitre nous ne sommes plus en surveillance continue de l'activité oculomotrice (surveillance saccadique exclue). Pour mettre en mouvement le curseur il faut que l'utilisateur surveille une zone d'attention particulière pendant une durée de fixation qui valide cet emplacement d'observation. Quand l'activité oculaire n'est plus forcément contrôlée par un désir de pointage, le mécanisme de surveillance peut très bien se mettre en veille.

La composition de ce chapitre qui décrit le dispositif « souris oculaire » est la suivante :

Dans un premier temps on présente brièvement le dispositif de prise de vue et d'acquisition d'une image numérisée, qui nous a permis d'exploiter les images de l'oeil suivant la technique du reflet cornéen. L'information de la direction du regard tête libre par ce type de mesure ne convient pas, d'où la proposition d'une nouvelle méthode adaptée au cas d'un support visuel fixe. On éclaire l'oeil par des sources ponctuelles infra-rouges disposées sur le pourtour de l'écran du moniteur informatique qui sert de support de communication visuelle cible.

Notre dispositif d'éclairement et de prise de vue, nous donne des images types de l'oeil qu'il faut analyser soigneusement. On décrit comment on localise l'image de la pupille discriminée dans l'ensemble des formes révélées par une technique de binarisation par un seuillage adapté au sujet et à l'éclairage ambiant, puis on effectue au positionnement des différents reflets cornéens. On déduit de cet ensemble d'informations le positionnement du pseudo vecteur cornéen. On montre par une phase d'initialisation qu'il y a conservation d'un paramètre intrinsèque au dispositif optique et aux données anatomiques de l'oeil, qui permet de désigner directement sur l'écran l'endroit observé attentivement par l'utilisateur.

Le travail contenu dans ce dernier chapitre fait état d'un important travail d'expérimentation et de développement de logiciels spécifiques.

Ne disposant pas d'une plate forme d'analyse de scènes dynamiques, le banc de mesures ne permet que de vérifier la faisabilité d'un dispositif de pointage à partir d'images fixes.

La synthèse et l'analyse des images sont réalisées sur P.C. équipé de processeurs 486. L'algorithme de chaînage sur image binarisée est traduit en un traitement séquentiel par un programme écrit en langage C. Toute l'étude amont qui consiste à opter à un parallélisme massif est de ce fait mise en marge mais nous a permis de valider une structure originale à implanter sur circuit ASIC ou cartes équipées de plusieurs DSP.

Les perspectives de nos recherches seront effectivement d'appliquer les structures parallèles que nous avons validées sur un prototype câblé fonctionnant au rythme de 50 trames par seconde.

5.2 Le système d'acquisition d'une image

La chaîne d'acquisition est décrite Figure 5-1.

L'acquisition est effectuée au moyen d'une caméra fixe couplée à un numériseur d'image à mémoire. La scène observée est éclairée par une source infra-rouge contrôlée (voir Figure 5-2).

Le numériseur est une carte PIP 1024B de matrox. La caméra est à encombrement réduit 90*56*23 mm, équipée d'un capteur CCD 1/3 de pouces de 300000 pixels et de sensibilité 1 lux.

L'échantillonnage spatial de l'image est de 512*512 pixels. Le champ couvert par la caméra est de 7/5.25 cm à une distance d'observation de 10 cm. On obtient une surface

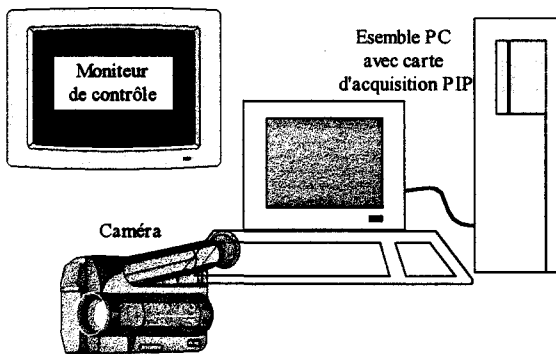


Figure 5-1 : Equipement de la chaîne d'acquisition.

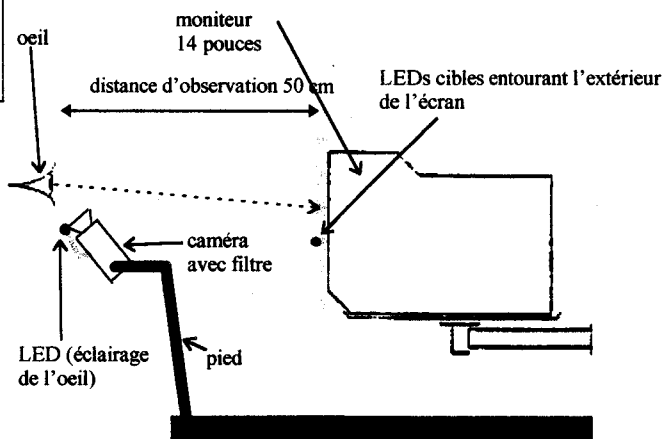


Figure 5-2 : disposition des éléments du banc d'expérimentation.

élémentaire de traitement approximative de 0.02 mm^2 .

Le codage de l'image est réalisé sur 8 bits, ce qui offre une plage de 256 niveaux de gris. Un contrôle sur le gain et l'offset de l'amplificateur vidéo de la carte PIP permet d'obtenir une image de qualité (grande dynamique) pour une ambiance lumineuse faible. Le traitement s'opère sur l'image infra-rouge au moyen d'un filtre.

5.2.1 Calibrage de la caméra

L'optique de l'objectif de la caméra désigne directement la focale et l'angle d'ouverture du champ de la scène observée. Il nous faut maîtriser les données de distance focale et coefficients des composantes pixels horizontaux et verticales normalisées afin de retrouver les dimensions réelles d'une forme à une distance d'observation donnée.

Le passage d'un plan de pixels au positionnement réel s'opère par conversion avec deux facteurs d'échelle. Un pixel revient à avoir une surface élémentaire sur le capteur de $k_x.k_y \text{ cm}$. En projection perspective on a la transformation du couple x_1, y_1 en u_1, v_1 par les relations :

$$u_1 = -\frac{f}{z_1}.x_1 \text{ et } v_1 = -\frac{f}{z_1}.y_1$$

Pour le couple u_1, v_1 on effectue un changement d'échelle vertical et horizontal tenant compte de l'échantillonnage spatial en éléments discrets dits pixels. Les facteurs d'échelles exprimés en cm/pixel permettent donc le passage du plan image pixel aux éléments photosensibles sur le capteur CCD.

Les composantes normalisées k_x et k_y pour une distance focale f donnée, valent :

$$k_x = -\frac{u_1}{f.m} \text{ et } k_y = -\frac{v_1}{f.n} \text{ avec } m = m_0 - m_1 \text{ et } n = n_0 - n_1$$

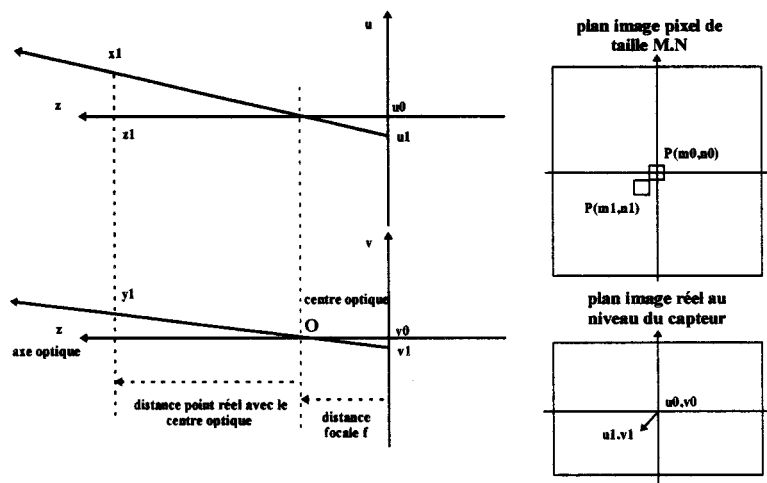


Figure 5-3 : Transformation image caméra.

Bien souvent il est difficile de connaître avec certitude la distance focale, c'est pourquoi on opère un calibrage de la caméra. Si l'on connaît avec suffisamment de précision les dimensions d'un objet présenté dans l'axe optique de la caméra, on peut retrouver la distance focale à partir des dimensions du capteur et de la distance à laquelle est observé l'objet. La distance d'observation peut être déduite si l'objet présente deux formes caractéristiques à des distances différentes. Nous avons opté pour l'observation d'un cylindre dans l'axe optique du capteur. Le positionnement de l'objet présentant ces deux faces parallèlement au plan de projection et dans l'axe optique est obtenu par conservation des axes verticaux et horizontaux orthogonaux et par obtention du reflet du centre de l'objectif de la caméra dans le centre de l'image recueillie (Figure 5-4).

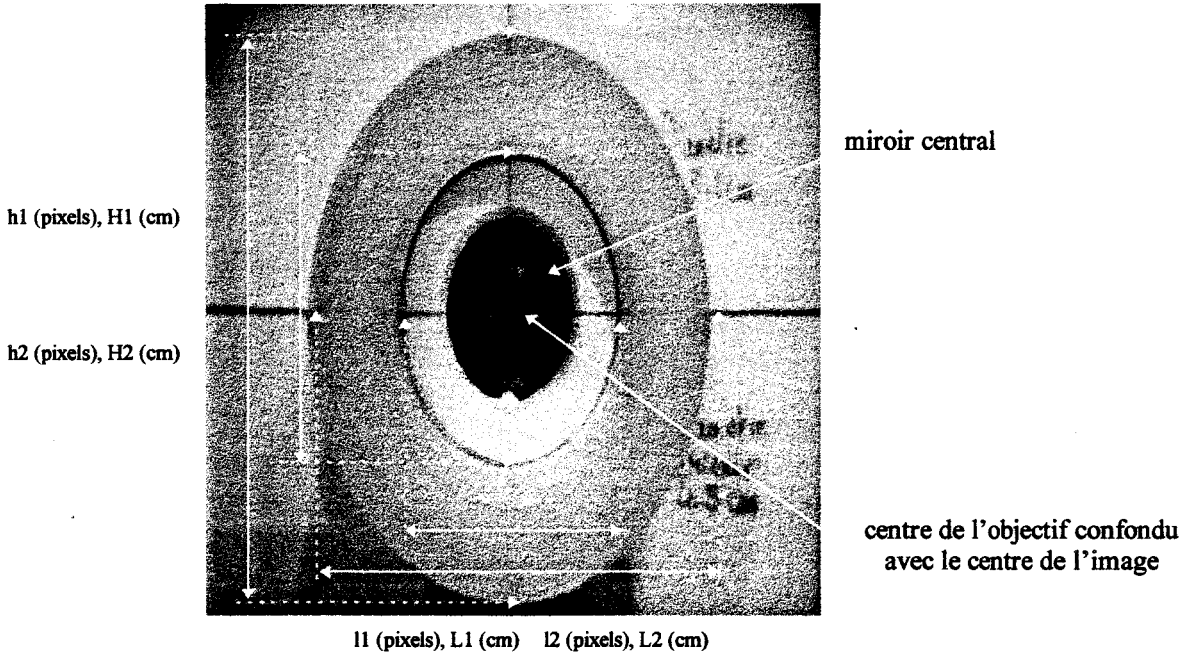


Figure 5-4 : Image du cylindre présenté perpendiculairement à l'axe optique de la caméra.

On détermine les facteurs d'échelles k_x et k_y par :

$$k_x = \frac{1}{P_0} \left\{ \frac{L1}{l1} - \frac{L2}{l2} \right\}$$

$$k_y = \frac{1}{P_0} \left\{ \frac{H1}{h1} - \frac{H2}{h2} \right\}$$

De là on détermine la distance focale de l'objectif :

$$f = -\frac{u1}{k_x \cdot l1} = -\frac{Lc \cdot \frac{l1}{M}}{k_x \cdot l1} = -\frac{Lc}{M \cdot k_x \cdot l1} \quad \text{avec } Lc$$

largeur du capteur en cm et M nombre de pixels sur une ligne complète.

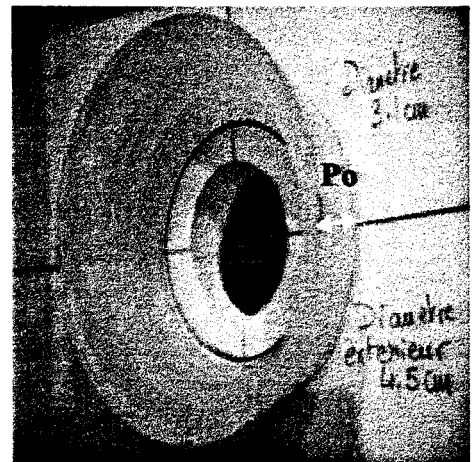


Figure 5-5 : prise de vue de l'objet sur le coté.

5.2.2 Positionnement de la caméra

Le résultat de l'image captée par la caméra vient de la transformation par projection perspective d'un objet situé à une distance donnée, observé suivant un axe optique orienté. Si l'on considère que le repère physique est construit à l'origine sur le centre optique et orienté suivant l'axe optique, on doit pouvoir exprimer l'emplacement des points matériels dans ce nouveau repère. Il est très délicat de connaître avec exactitude l'emplacement d'une source lumineuse car on perd la propriété de profondeur dans le cas de l'analyse d'une image. Pour remédier à ce problème on construit l'image du reflet d'une source par rapport au glissement d'un miroir perpendiculaire à l'axe optique (Figure 5-6). Le plan miroir est parfaitement parallèle au plan de projection.

On va avoir un déplacement du reflet sur le miroir à une distance z_p donnée suivant les relations suivantes :

$$x_p = \frac{z_p \cdot x_l}{2z_p - z_l}$$

$$y_p = \frac{z_p \cdot y_l}{2z_p - z_l}$$

$$x_p \xrightarrow{z_p \rightarrow \infty} \frac{x_l}{2}$$

$$y_p \xrightarrow{z_p \rightarrow \infty} \frac{y_l}{2}$$

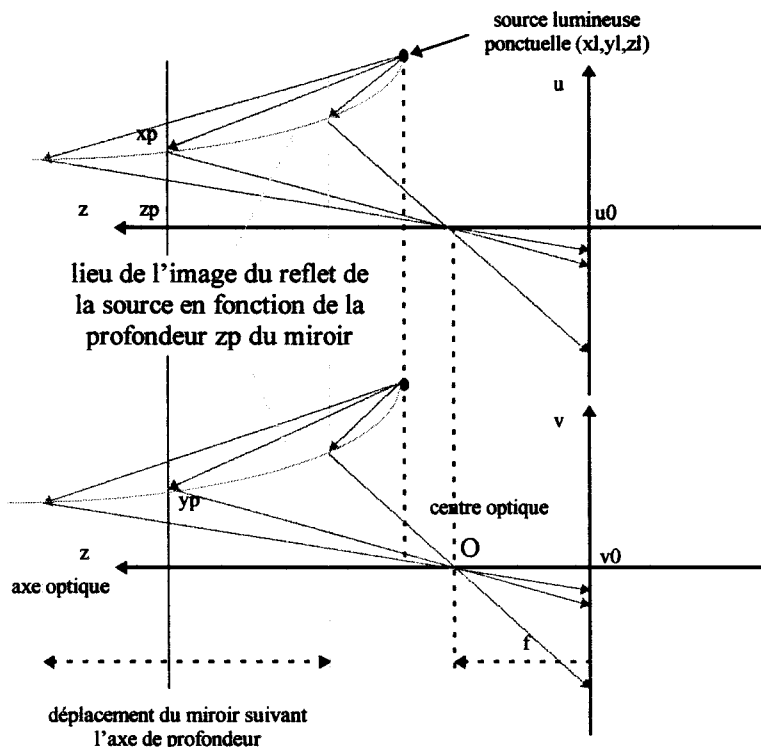


Figure 5-6 : Image du reflet d'une source sur un miroir plan.

Pour reconstruire le positionnement réel de la source par ses coordonnées (x_l, y_l, z_l) , il suffit d'opérer par mise en relation avec des composantes connues, tel que des couples de sources pour lesquelles on connaît leurs dispositions spatiales. L'image de la Figure 5-7 montre un exemple de sources ponctuelles alignées et espacées régulièrement d'un pas P .

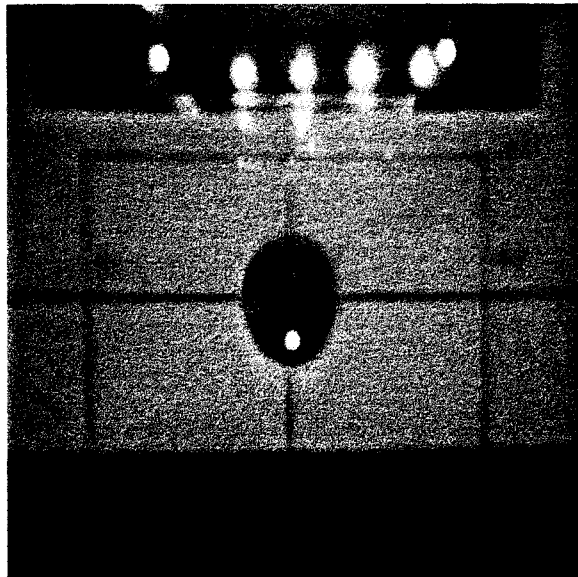


Figure 5-7 : Observation d'une ligne de LED sur un miroir plan.

Le calibrage de la caméra ne s'effectue plus par rapport à un environnement de référence extérieur mais au contraire on choisit d'opter pour des relations où tous les objets sont ramenés au centre optique du système de prise de vue. On élimine ainsi la délicate tâche de détermination de l'orientation de l'axe optique

5.3 Utilisation d'un ensemble de sources ponctuelles délimitant le champ visuel

Le but recherché par cet ensemble de LEDS est de former un point infrarouge virtuel sur le moniteur d'un système micro-informatique. Etant entendu qu'il est impossible de placer une matrice de LEDS sur l'écran, on simule l'éclairement du point en position $P_v(x,y)$ par l'intersection des droites horizontales et verticales qui passent par ce point. Ces deux droites sont obtenues par le passage aux points $p_0(y_l)$ et $p_2(y_l)$ ainsi que par le couple de points $p_1(x_c)$ et $p_3(x_c)$ (voir Figure 5-8).

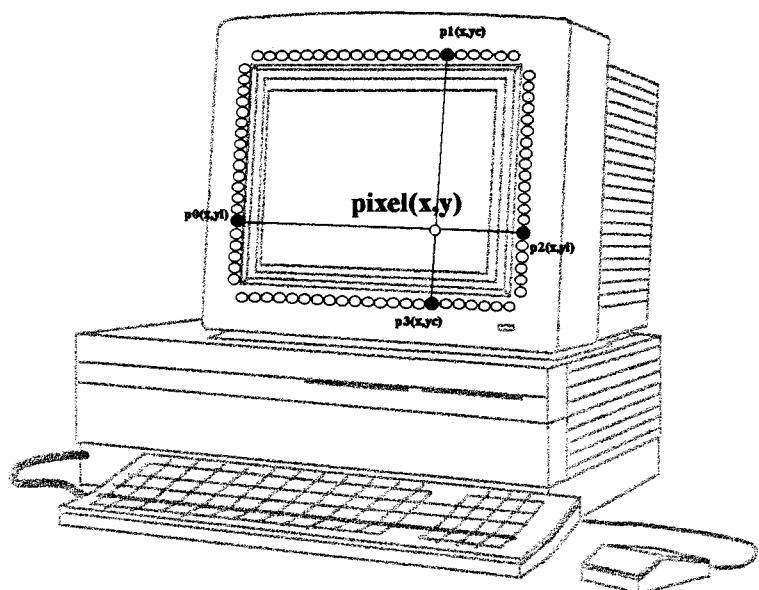


Figure 5-8 : Délimitation des LEDs autour de l'écran et obtention du point virtuel.

Les LEDs sont disposées en ligne sur la largeur et hauteur d'un écran de diagonale 14 pouces. Afin de se limiter à un nombre raisonnable de LEDs infra-rouges, nous en disposerons 33 horizontalement et 25 verticalement. Le choix d'un nombre impair d'émetteurs permet la sélection du point central de l'écran. Le rapport 33/25 tient compte du format 4/3 des écrans actuels.

La gestion de la sélection des sources I.R, se fait par le PC via liaison parallèle 8 bits Centronics, en niveau TTL, interfacé à une carte annexe. Cette carte reçoit les informations, elle les mémorise et elle distribue sur les différents bargraphes le signal qui allumera les LEDs permettant de construire le point de référence sur l'écran (Figure 5-9).

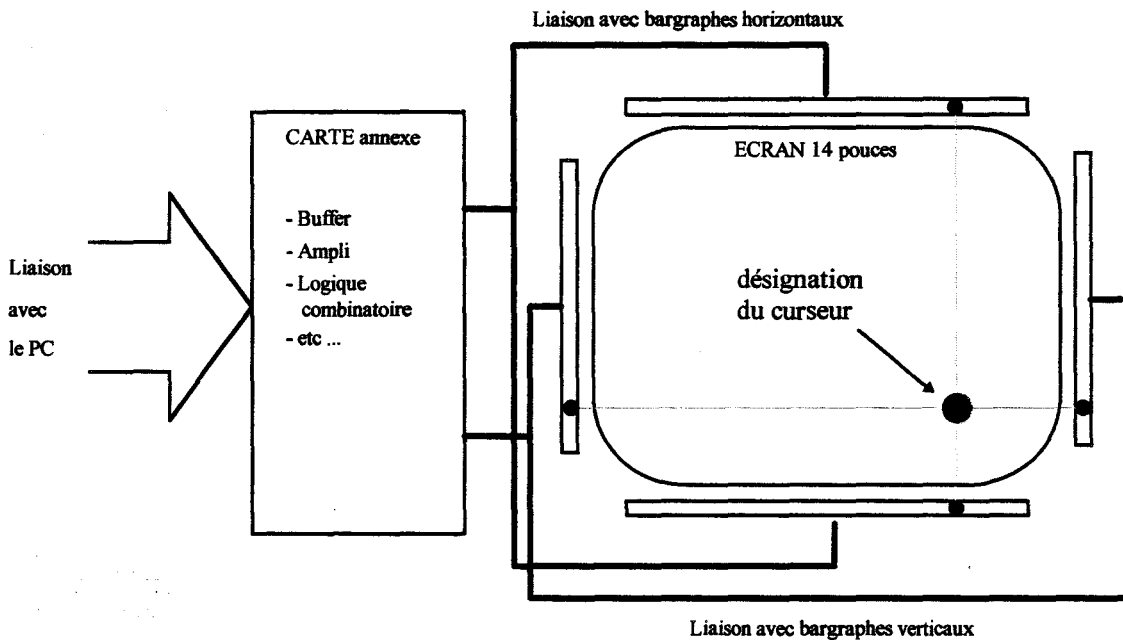


Figure 5-9 : Contrôle des sources qui éclairent l'oeil.

Cette disposition permet d'asservir et de construire un point de référence à n'importe quel emplacement de l'écran par un quadrillage régulier. Cette démarche est essentielle dans le mécanisme qui permet la détermination du regard.

5.4 Caractérisation des données anatomiques de l'oeil

L'environnement dans lequel prend place ce système est un ensemble informatique type PC composé, d'un écran, d'un clavier et unité centrale. On cherche à connaître avec une précision suffisante la zone observée sur le moniteur par un utilisateur dont on observe l'oeil par une caméra dans le spectre infra-rouge.

5.4.1 Image de l'oeil éclairé par une source I.R proche de l'axe optique de la caméra

Lorsque le regard est dirigé directement vers l'axe optique de la caméra on observe l'image telle qu'à la Figure 5-10. Ce cas précis permet de mettre en évidence la construction du reflet de la source ponctuelle sur la cornée. Les données anatomiques de l'oeil font que la construction est fonction de la position du centre de courbure de la cornée, du rayon de courbure et de l'orientation de l'axe du regard. Le passage par le centre de la pupille, le centre de courbure et le centre du globe oculaire définit l'axe optique du regard. La Figure 5-11

permet de visualiser le parcours des rayons de la source ponctuelle pour un positionnement connu de l'oeil orienté vers un regard centré sur l'objectif de la caméra.

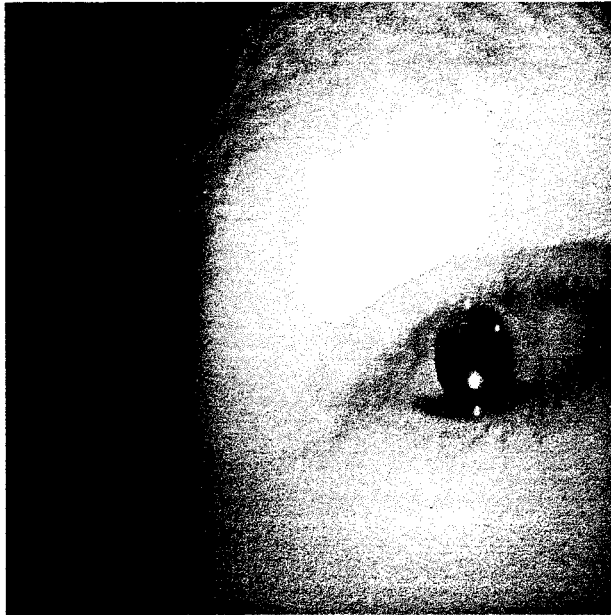


Figure 5-10 : Image de l'oeil pour un regard dirigé vers le centre focale de la caméra.

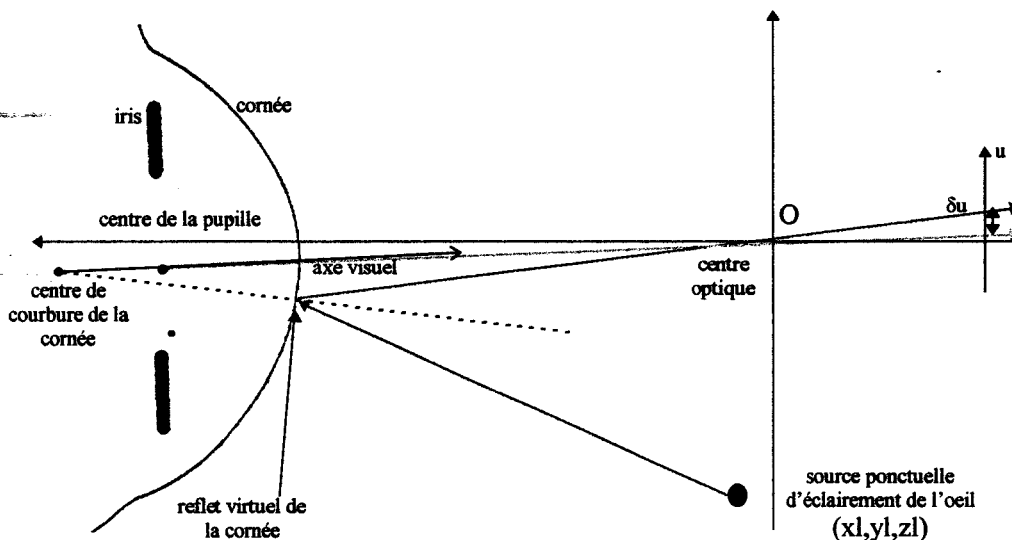


Figure 5-11 : Reflet de la source lorsque l'oeil regarde le centre optique.

L'image du reflet est construite par réflexion sur la cornée du faisceau incident coïncident avec le centre optique pour une position précise du centre de courbure de la cornée. Il n'existe qu'une seule solution pour laquelle on retrouve l'image de la pupille et du reflet construit comme à la Figure 5-11 et représenté par l'écart δu . Cet écart dépend principalement des caractéristiques anatomiques de l'oeil que l'on peut déterminer par obtention de la distance d'observation du globe oculaire. On rappelle que le rayon de courbure est de l'ordre de 5.5 mm et la distance pupille centre de courbure de 4.5 mm. Une démarche similaire permet de caractériser la construction du reflet pour une observation du regard dirigé sur la source d'éclairement. Dans ce cas on a le trajet du rayon incident qui se réfléchit sur la cornée pour être en coïncidence avec le centre optique suivant la construction illustrée Figure 5-12.

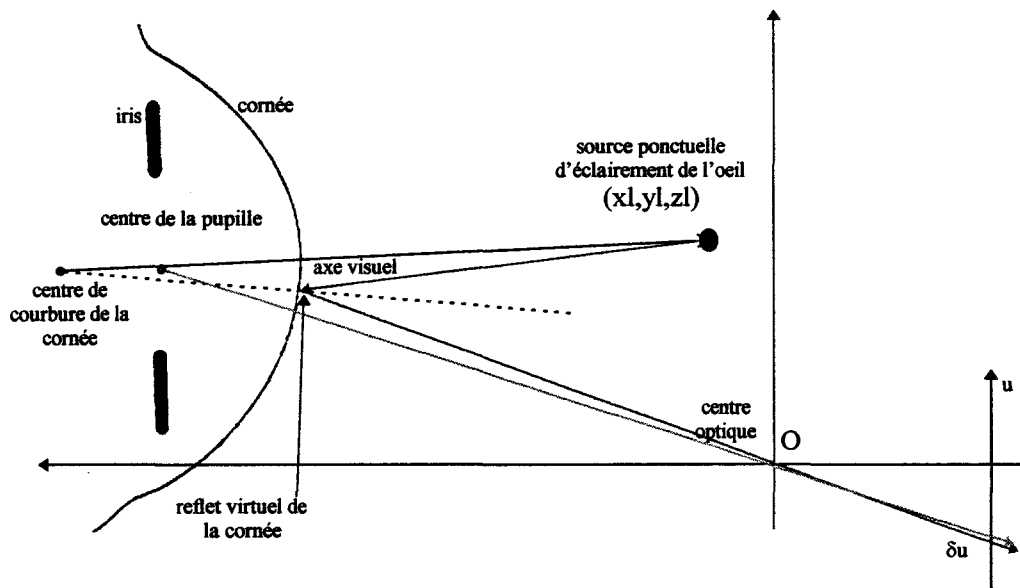


Figure 5-12 : Reflet cornéen lorsque l'oeil regarde la source.

Du fait que l'on récupère une image virtuelle sur la cornée, on voit qu'il n'est malheureusement pas possible d'obtenir le même point entre le centre de la pupille et l'image du reflet. Ce n'est qu'en connaissant parfaitement la position de la source et les données anatomiques de l'oeil que l'on peut désigner un nouveau point de référence sur lequel on pourra engager une mesure réelle de la direction du regard.

A tête fixe, on cherche à mettre en relation le point de référence de l'image de la source que l'on déplacerait afin d'avoir confusion des deux images, ce qui impliquerait la connaissance de la direction du regard.

Dans une démarche similaire d'adaptation de l'éclairement d'une source avec un souci de minimiser les calculs et de faire abstraction des mouvements de la tête, on s'appuie sur une technique mise en place par la suite. C'est ainsi que les outils de traitement de l'image de l'oeil développés et décrits dans les deux chapitres précédents vont nous faciliter les calculs et permettront une surveillance automatique de l'oeil. Ils permettront lorsque l'utilisateur présentera son oeil dans la zone d'acquisition de retrouver dans l'image numérique l'emplacement de la pupille et des différents reflets qui constituent l'image type. De même si l'image ne correspond pas pour des causes diverses à celle de la pupille et des quatre reflets, le dispositif se met en veille automatiquement pour reprendre dès que l'oeil réapparaît.

5.5 Exploitation des images vidéo pour déterminer le centre du regard

5.5.1 Détection de la pupille noire

La pupille de forme ellipsoïdale est détectée par une binarisation classique de l'image suivant la technique du seuillage. La valeur de ce seuil dépend de l'éclairage ambiant et des réglages opérés sur le gain et l'offset de l'amplificateur vidéo d'entrée sur la carte de numérisation. L'image peut donc être segmentée directement par classification des pixels en objet ou fond suivant que le niveau de gris est supérieur ou inférieur au seuil pupille. La transition entre l'iris et la pupille permet d'avoir le découpage d'une forme pupillaire avec des



Figure 5-13 : Image binarisée par le seuil pupille.

contours nets. Le cas le plus défavorable des yeux de couleur brun, transition la plus faible, est admis sans problème majeur de détection. Le procédé de chaînage génère une liste chaînée (voir annexe) qui permet de retrouver rapidement le candidat pupille parmi toutes les autres formes détectées. En effet l'image de l'oeil est en majeure partie supérieure au seuil de la pupille sauf sur des éléments distincts tel que les cils, les sourcils et le fond qui n'appartient pas au visage (voir Figure 5-13). Le nombre de formes étant minime, l'analyse des en-têtes de liste

rapportant les caractéristiques géométriques de positionnement, centre de gravité et surface suffisent à trouver la forme pupillaire. On sait également d'une image à l'autre que le positionnement de la pupille ne varie pas brutalement, d'où une plus grande sensibilité de détection à l'intérieur d'une fenêtre de taille adaptée à la surface de la pupille et une plage de tolérance qui tient compte des déplacements lents de la tête.

5.5.2 Détection des reflets

Les reflets sur la cornée sont localisés autour de la forme pupillaire (Figure 5-14). Il est aisé de procéder par un traitement secondaire local, cette fois par un seuillage haut sur les éléments qui renvoient des pics d'intensité lumineux. La valeur du seuil haut dépend de la qualité de la réflexion sur la cornée, de la distance d'observation et du contrôle de l'éclairage des différentes sources mises en cause. L'intérêt majeur d'opérer par une succession de deux seuils de déclenchement pupille et reflet permet d'exclure un grand nombre de candidats reflets sur l'image complète. Les formes retenues dépendent des surfaces détectées, le reflet étant ponctuel, la surface en nombre de pixels doit être petite. La géométrie des reflets doit être conservée suivant l'emplacement où les sources se positionnent malgré la surface courbe de la cornée.

Il est plus facile de déterminer les formes reflets, du fait que l'on a connaissance sur la disposition spatiale des sources et un a priori sur leur positionnement autour du centre de la forme pupillaire. Le contrôle des LEDs émettrices est affecté à la carte annexe qui gère les différents bargraphes.

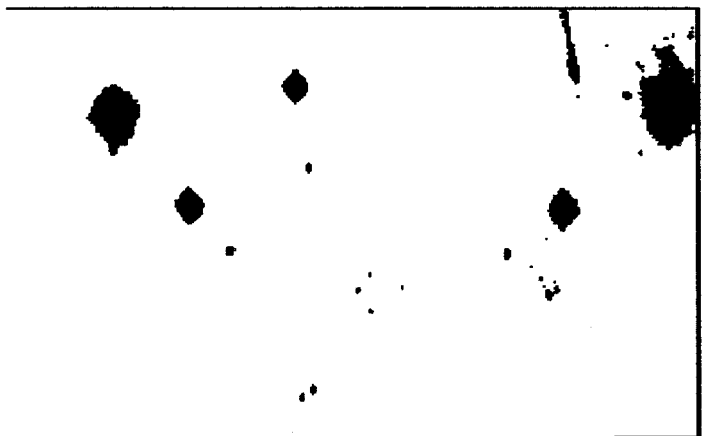


Figure 5-14 : Détection des reflets par un seuil de binarisation haut.

Les courbes de niveau de gris montrent la transition iris-pupille et reflet-pupille qui sont à prendre en considération sur les valeurs des deux seuils qui permettront de segmenter les images en pupille et reflet (Figure 5-16 & Figure 5-15).

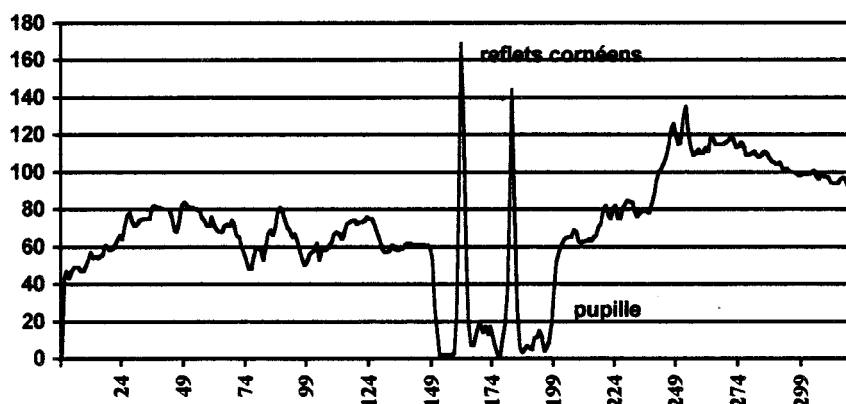


Figure 5-16 : Courbe de niveaux de gris sur une ligne de l'image de l'œil

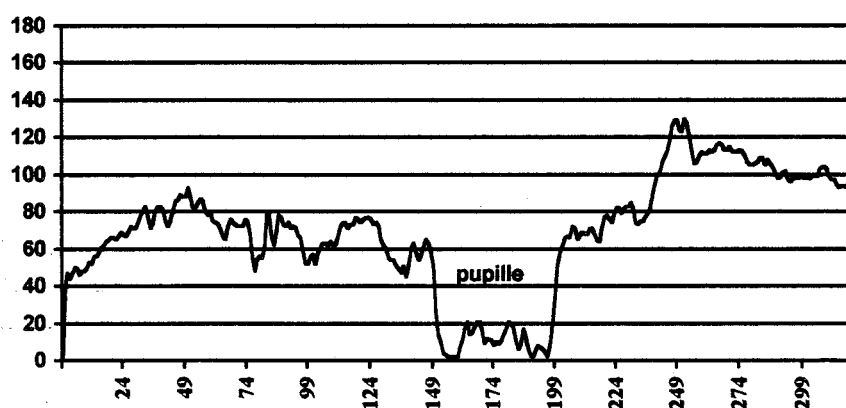


Figure 5-15 : Courbe de niveaux de gris sur une ligne n'interceptant pas de reflets.

5.5.3 Procédure automatisée de la détection de la pupille et des reflets

On s'arrange par l'intermédiaire d'un étalonnage sur le gain et l'offset de l'amplificateur à numériser une image fortement contrastée.

Le traitement s'effectue en deux phases:

Dans un premier temps on cherche à localiser la pupille. Une technique de binarisation par seuillage adapté, traite l'image au complet. De là l'algorithme de chaînage (chap 4) permet d'identifier et classer les différents objets contenus dans l'image. Les attributs (surface, barycentre, taux de complexité, trous, etc.) nous indiquent s'il y a vraisemblance ou non avec le modèle défini pour la pupille. Dans le cas où la pupille est détectée nous passons à une deuxième phase de traitement local. On ouvre une fenêtre de travail centrée sur la pupille et de taille adaptée à la surface pupillaire afin de déterminer les quatre reflets cornéens (Figure 5-17).

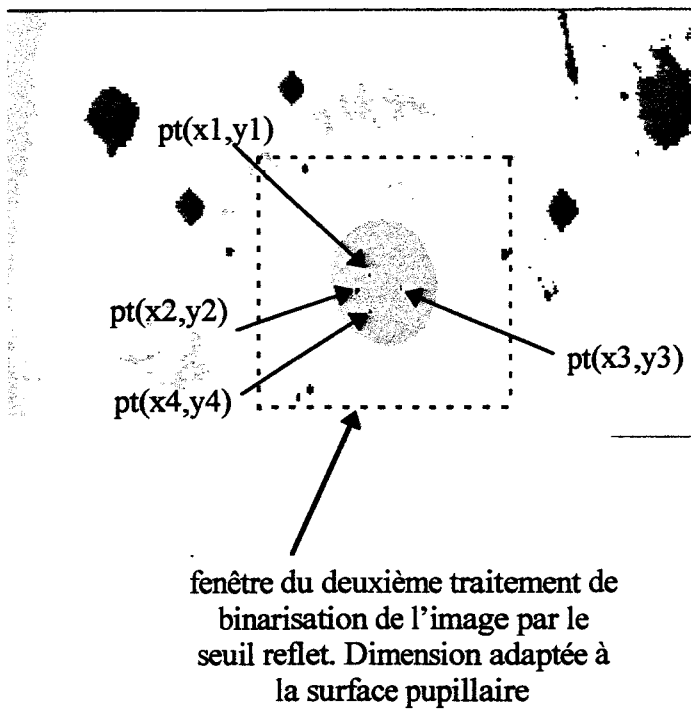


Figure 5-17 : fenêtre de traitement des reflets cornéens.

Lorsque nous sommes en possession des quatre couples de coordonnées $pt(x_i, y_i)$, on calcule le point d'intersection des droites horizontales et verticales passant par les reflets. Le point calculé P_i est la représentation du point de l'écran où se trouve le curseur dans le plan de projection du capteur vidéo. La position relative du barycentre de la pupille P_p peut de ce fait être référencée à P_i dans le repère physique de l'écran. L'écart obtenu entre les deux points nous informe s'il y a correspondance ou non entre la position du regard et celle du curseur. Le résultat après traitement sur image fixe est représentée Figure 5-18.

L'asservissement du curseur en relation avec l'état des LEDs sur les différents bargraphes permet de chercher une relation qui permettra de connaître la direction absolue du regard.



Figure 5-18 : traitement vidéo de l'image infra-rouge de l'oeil.

Le système reste maître sur le positionnement du curseur mais le regard s'il est suffisamment stable lors d'une attention particulière sur une zone précise de l'écran permet d'atteindre à la coïncidence parfaite entre le positionnement du curseur (changement des LEDs actives par la carte annexe) et la direction absolue du regard.

Ce principe limite le contrôle de l'activité oculaire aux durées de fixation du fait que l'on connaît parfaitement la direction du regard uniquement pendant les durées de fixation qui stabilisent le dispositif asservi par une boucle de réaction. Dans notre contexte, les mesures exploitées sur images fixes, nous imposent de travailler sur le dispositif en boucle ouverte, la contrainte temps réel n'étant pas atteinte.

5.6 Technique de la mesure absolue de la direction du regard

Nous avons opté par un asservissement de l'emplacement du point virtuel, de telle façon que le contrôle du positionnement permet d'annuler le pseudo vecteur cornéen.

Le pseudo vecteur cornéen répond à la description de la construction comme à la Figure 5-12 où on avait remarqué le fait que les images virtuelles d'une source ponctuelle ne coïncident pas avec le centre de la pupille pour un axe visuel orienté vers cette même source. Un point de référence marquait cette légère différence à partir des données anatomiques de l'oeil et le positionnement du globe oculaire. Le pseudo vecteur cornéen s'identifie donc à la mesure de l'écart entre le centre pupillaire et ce point de référence et de même à l'évaluation de l'écart entre le point calculé d'intersection des quatre reflets et du point de référence. Ainsi si le pseudo vecteur est nul, on sait que nous sommes directement en relation entre le curseur et l'axe visuel, matérialisant ainsi la zone d'observation particulièrement attentive de l'utilisateur du pointeur de regard.

Puisque la position du curseur est commandée par les positions des bargraphes, il suffit de ramener le point P_i en concordance avec le centre de la pupille. Dans le cas où les points sont confondus on a la direction du regard matérialisée par le curseur. Le dispositif s'efforce à minimiser la distance $D(P_p, P_i)$ lors de chaque mouvement oculaire.

La construction optique des différents reflets est schématisée Figure 5-19.

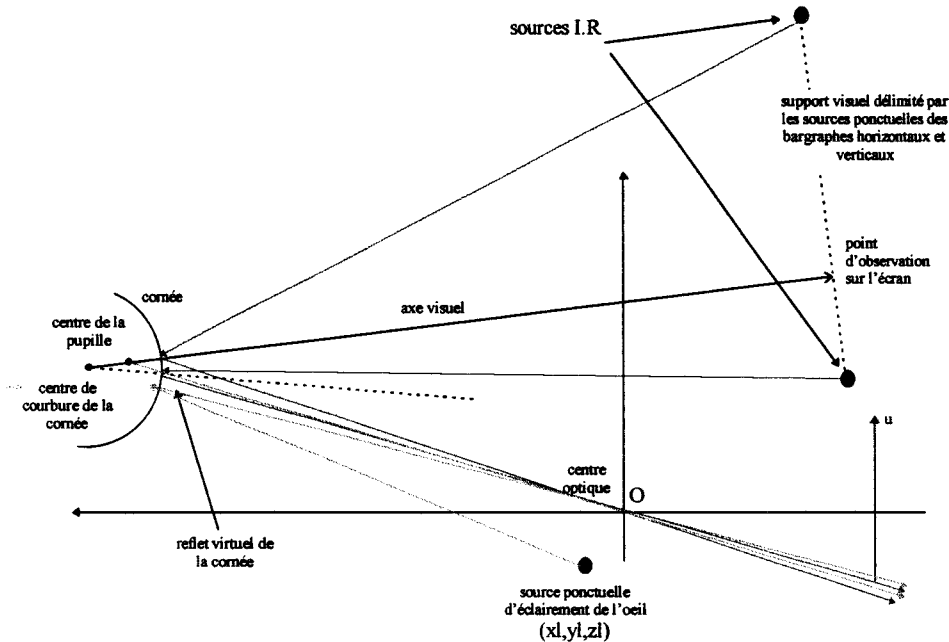


Figure 5-19 : construction des reflets sur la cornée en fonction de l'orientation du regard.

5.6.1 Procédure expérimentale de calibrage

Comme nous l'avons vu dans les différentes constructions, notre problème est d'évaluer le point de référence qui tient compte de la combinaison orientation axe optique de la caméra et des différents paramètres anatomiques de l'oeil. Lorsque l'utilisateur observe le point virtuel formé par les quatre LEDs, l'image du centre de la pupille ne se forme pas sur l'image du reflet du point concerné mais légèrement décalé sur un point dit de « référence ». Le but est donc d'évaluer pour une position de la tête le point commun pour différentes orientations du regard grâce au contrôle des différents bargraphes.

La procédure d'étalonnage consiste à ordonner différents points de regard sur une mire de calibrage composée de six repères distincts. A chaque repère l'utilisateur conserve une attention particulière afin que le système exploite la mise en correspondance du repère virtuel composé à partir de la sélection active des sources I.R sur les quatre bargraphes qui encadrent l'écran du moniteur. Le point en adéquation qui permettra d'évaluer le pseudo-vecteur cornéen étant celui situé directement sur le repère physique dicté par la phase de calibrage.

En balayant les six repères on obtient une évaluation suffisamment précise du décalage induit par le système optique pour déterminer avec certitude la direction du regard en harmonie avec les deux couples de LEDs permettant la conservation du même écart relatif entre le centre pupillaire et le point de calcul du complexe reflets cornéens.

La mire se présente suivant la Figure 5-20.

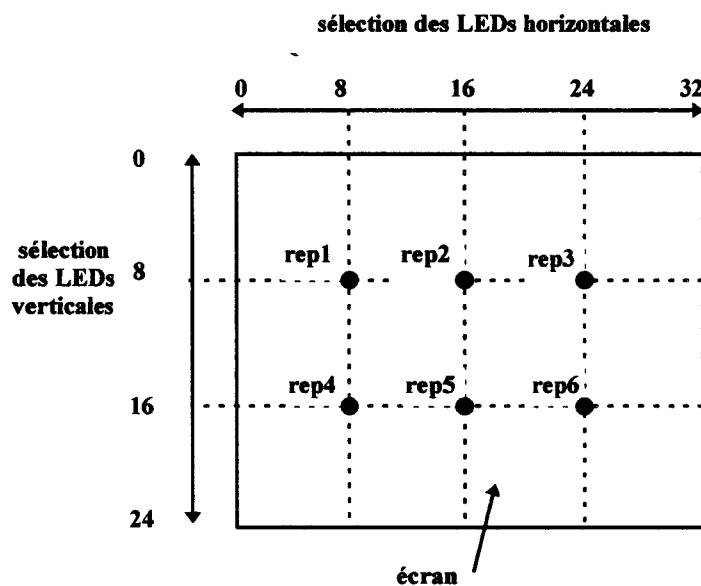
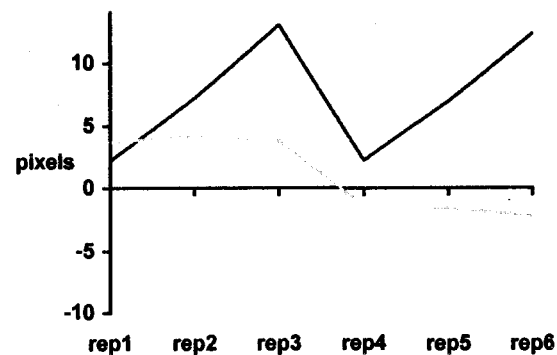


Figure 5-20 : Mire de calibrage.

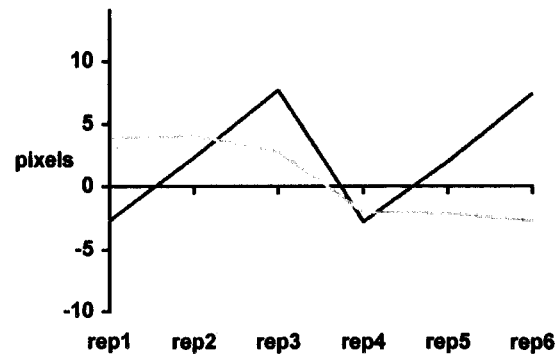
5.6.2 Expérimentation sur un individu

— déplacement horizontal
- - - déplacement vertical

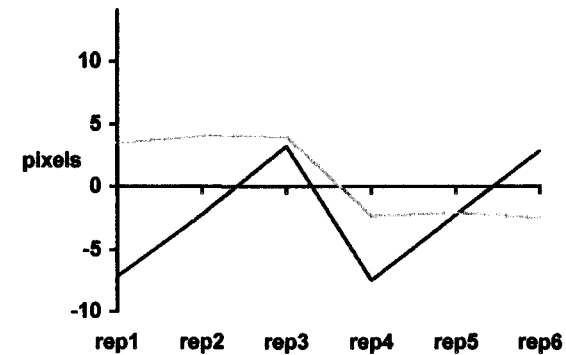
Sélection de Rep1 LEDs (8,8)



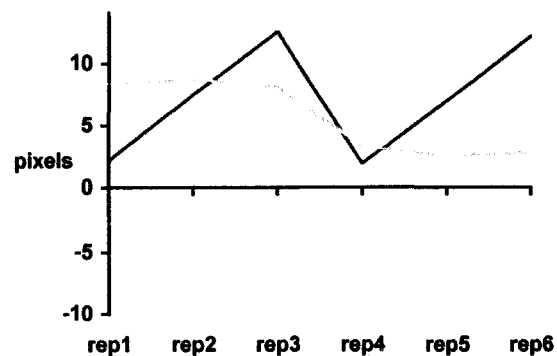
Sélection de Rep2 LEDs (16,8)



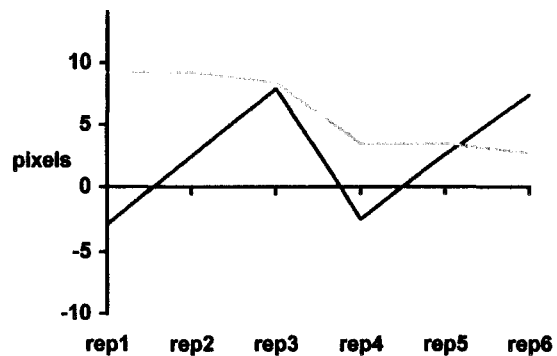
Sélection de Rep3 LEDs (24,8)



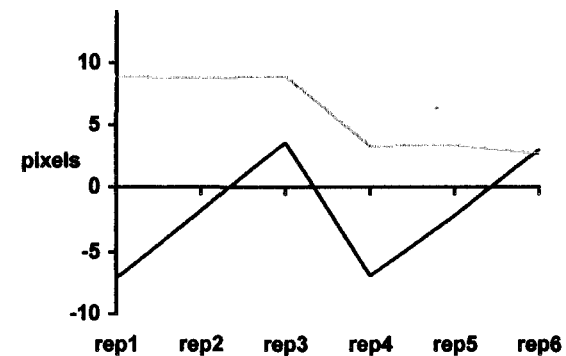
Sélection de Rep4 LEDs (8,16)



Sélection de Rep5 LEDs (16,16)



Sélection de Rep6 LEDs (24,16)



L'ensemble de six tableaux compose la source d'étalonnage du dispositif pour une personne spécifique. Il est entendu que les données anatomiques de l'oeil varient en fonction des individus et cette opération est à renouveler à chaque changement de sujet.

Chacun des tableaux est établi suivant un ordre de direction du regard allant du rep1 au rep6 pour un point matérialisé indiqué en entête du tableau. Le premier tableau correspond à la sélection des LEDs du rep1 couple (8,8). Le déplacement horizontal et vertical est en fait l'écart entre le centre de la pupille calculé et l'image du point d'intersection des quatre reflets sur la cornée.

Si l'on observe la progression du rep1 à rep3 ou du rep4 à rep6, on remarque un déplacement vertical constant et une progression linéaire du déplacement horizontal. L'explication de ce comportement est le résultat du balayage de l'axe visuel dirigé d'une ligne horizontale allant de la LED 8 à 24.

L'effet de la marche d'escalier du déplacement vertical permet de mettre en évidence le changement de ligne où se trouvent positionnés les repères 1 à 3 et 4 à 6.

Pour le premier tableau le regard référencé au rep1 est en fait la seule information qui permet de recueillir le point de référence, car dans cette seule configuration il y a coïncidence entre l'éclairement des LEDs et l'emplacement observé à l'écran.

Dans les autres tableaux où la sélection correspond avec la direction du regard (repère matériel avec la bonne combinaison des LEDs) on remarque que les quantités déplacement vertical et horizontal sont sensiblement identiques. On a bien donc conservation du pseudo vecteur cornéen pour établir l'orientation du regard. Les images des combinaisons en harmonie avec le point matérialisé à l'écran sont représentées Figure 5-21.

Il revient donc après cette étape d'opérer à chaque mesure de la direction du regard

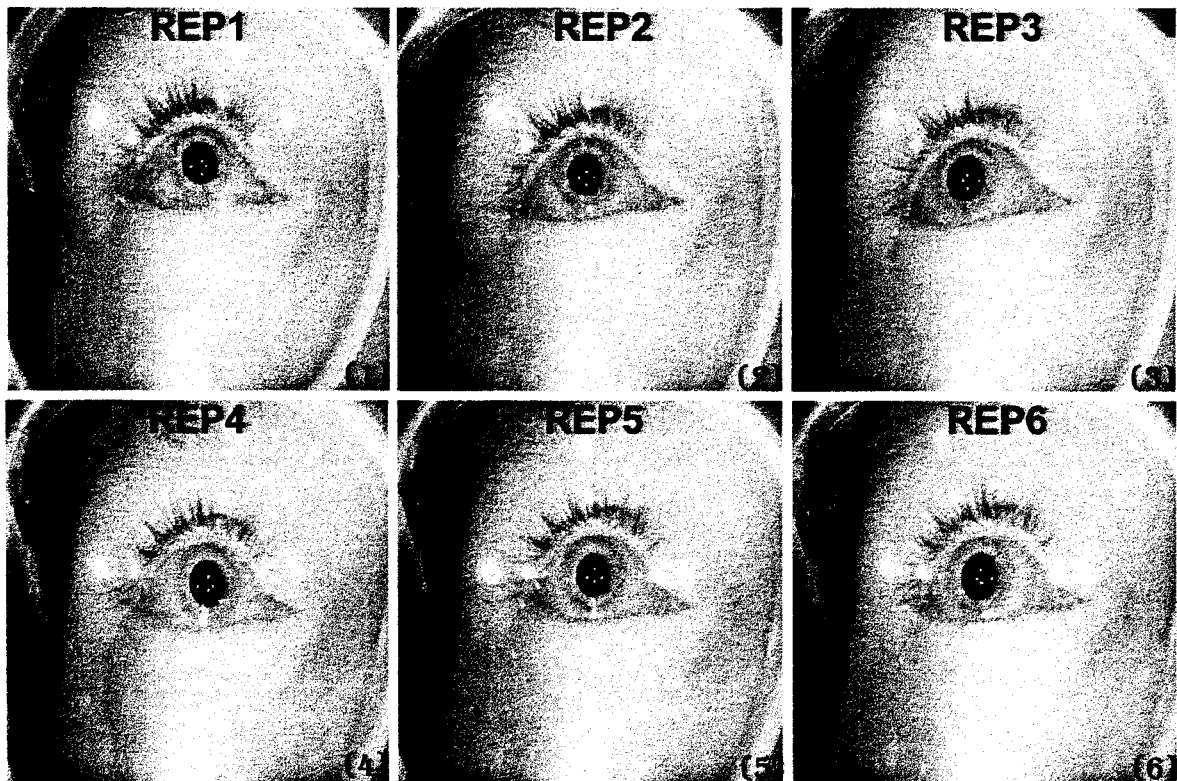


Figure 5-21 : Les six regards en coïncidence avec le repère matériel.

désiré, d'adapter l'éclairage des LEDs de telle façon à retrouver ces mêmes quantités.

Ainsi sans même à avoir à calculer les caractéristiques anatomiques de l'oeil ou de lancer des calculs fastidieux, on peut facilement après itération de plusieurs mesures connaître l'emplacement exact où porte l'attention d'un sujet face à la caméra grâce à un asservissement des LEDs à sélectionner.

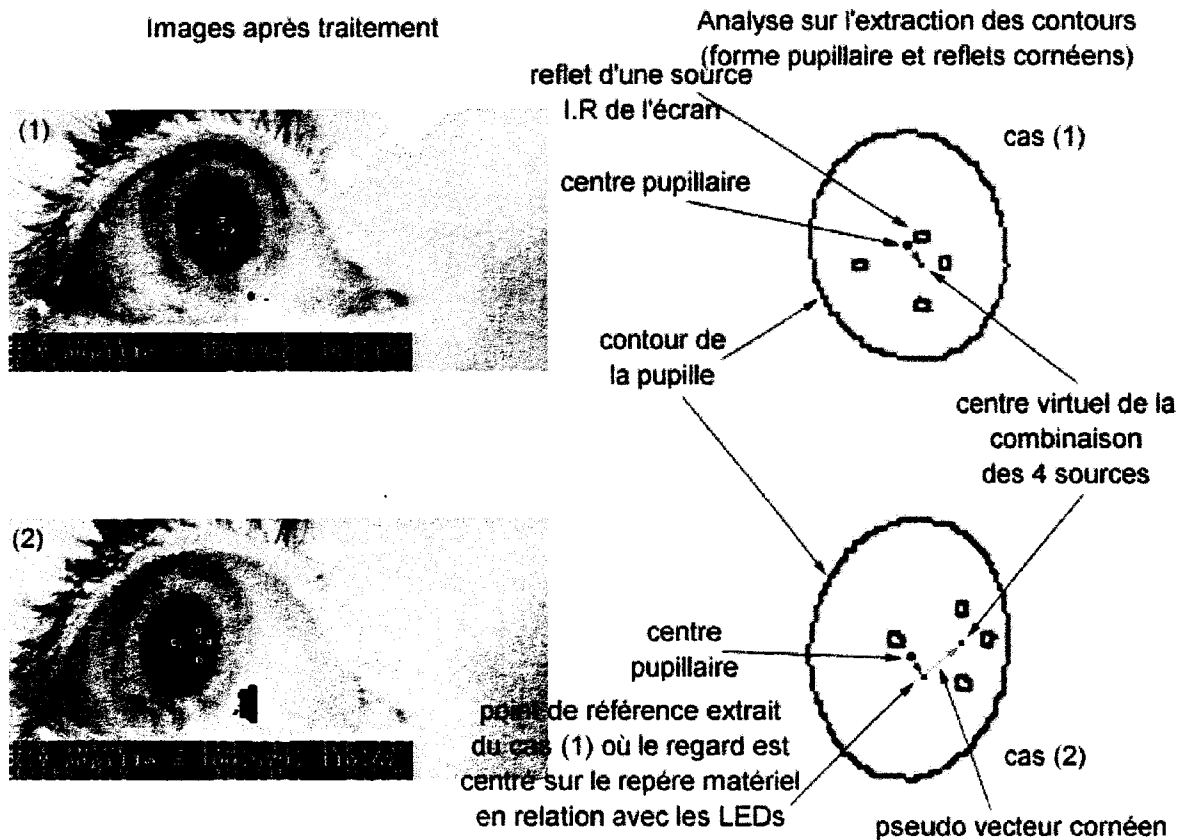


Figure 5-22 : Analyse des différences déplacement des reflets pour deux mouvements oculaires opposés.

La Figure 5-22 reprend l'image du repère 1 où le regard est centré sur la cible en relation avec les LEDs, cette combinaison permet de déduire le point de référence. Ce point de référence mis en relation avec une seconde image de l'oeil où le regard est excentré en bas à droite vers le repère 6 avec conservation des LEDs sélectionnées au cas (1) montre que l'on a une quantité qui traduit le désaccord entre le positionnement du regard et le point virtuel. Cette quantité est représentée par le pseudo vecteur cornéen qui doit être toujours nul pour déduire la direction absolue du regard.

5.7 Conclusion

Cette méthode peu contraignante permet de désigner sur un écran cible la direction du regard par simple adaptation de l'éclairage de deux couples de LEDs au travers d'une sélection sur des bargraphes.

Le fait d'utiliser une méthode adaptative par un asservissement qui cherche à conserver un pseudo vecteur cornéen nul simplifie nettement le traitement.

La connaissance des LEDs éclairées par un dispositif toujours maître sur la commande oculaire permet d'avoir en permanence un retour d'information cohérent avec le fait de déplacer le curseur continuellement sur l'écran (pas de sauts brutaux).

L'utilisateur peut également débrayer le dispositif de traitement de façon automatique lorsqu'il ne présente plus son oeil dans la zone de surveillance de la caméra.

L'aspect de commande prend sa vraie dimension vis à vis d'un dispositif plus classique qui reste accroché à un traitement spécifique sans connaissances a priori des éléments entrant dans la composition de l'image. Le fait de pouvoir distinguer les formes et de s'adapter à la scène observée (absence pupille, manque un reflet, etc..) facilite le processus de commande. Il n'est plus nécessaire qu'un opérateur humain vienne prendre le relais lors d'un décrochage d'un traitement sur une image qui ne serait plus exploitable.

La sensibilité du dispositif monte à 1.5 cm pour une distance d'observation de 50 cm. Ce qui implique un angle de rotation de 1.7° sans aucun appareillage spécifique sur la tête.

La mesure de l'activité oculaire sans contact par ce dispositif pourra être améliorée sensiblement par un choix spécifique du capteur CCD et de la qualité de l'objectif.

Le fait de rester avec un traitement sur une image numérique de taille 512*512 pixels améliore le temps de calcul pour une application dite « temps réel ». Augmenter l'échantillonnage spatial et ajouter un contrôle sur la distance focale, augmenterait le champ de couverture de prise de vue; La technique de suivi de l'oeil par un asservissement mécanique des mouvements de la caméra étant exclue d'après les essais peu concluants de certains dispositifs.

Ce traitement est réalisé sur images fixes à partir de logiciels de traitements d'images spécifiques. L'analyse de scène est effectuée par la technique originale de chaînage décrite dans le quatrième chapitre. L'acquisition des images numérisées à partir de la carte PIP n'autorise pas des traitements temps réels car sa mémoire de trame ne permet pas des transferts à haut débit.

Il est néanmoins établi que les procédés retenus doivent être poursuivis et implantés sur des machines spécifiques ou circuits spécialisés.

Le mécanisme d'asservissement peut paraître handicapant pour une mesure de la direction du regard lors des temps de désaccord entre disposition des LEDs et les mouvements de saccades de l'oeil. Notre souci étant de concevoir une commande visuelle, les temps de fixation de l'oeil d'environ dix images successives, permet de pallier au problème d'évaluation par un processus d'asservissement avec un temps de réponse suffisamment rapide.

Conclusions et perspectives

Nous avons proposé une méthode originale de chaînage et d'étiquetage d'une image binarisée dont on a pu vérifier la bonne tenue sur les images spécifiques de notre dispositif de pointage sur écran. La solution retenue permet des extensions au codage du cas typique du double seuillage par hystérésis après suppression des non maxima locaux pour le traitement récursif de Canny-Deriche. Il suffit à moindre coût, d'intégrer dans le traitement du seuillage la recherche des niveaux maxi et mini du niveau de gris sur l'ensemble connexe de la forme chaînée au complet.

C'est après une étude spécifique des différents traitements sur l'image infra-rouge de l'oeil que nous avons pu mettre en place un traitement séquentiel suffisamment simple afin de parvenir à une méthodologie séquentielle supportant un traitement cadencé au rythme de la vidéo. On a suivi deux démarches pour résoudre le traitement.

La première s'adapte facilement vers des structures personnelles sous forme de circuits VLSI, la deuxième convient mieux à un partage sur différents processeurs suivant des machines spécialisées et modulables.

Un traitement séquentiel se met facilement sous forme parallèle pour un coût souvent prohibitif vis à vis d'une machine cible et où les temps de communication entre processeurs limitent son efficacité. C'est pourquoi la description flot de données a retenu notre attention afin de parvenir à exprimer le parallélisme contenu naturellement dans l'algorithme. Les échanges et le contrôle des flots gérés par la présence ou non des données éliminent tout dispositif de commande externe. L'architecture régule elle-même son flot de sortie sans l'intermédiaire d'un système hôte. L'une des perspectives pour valider les structures massivement parallèles proposées est de concevoir les opérateurs élémentaires permettant de réaliser notre ASIC.

Notre ambition était de tenir le concept « temps réel » et d'être en adéquation avec une structure maîtrisée. Le chaînage au rythme de la ligne a pu être vérifié par une simulation d'un flot ligne, reste à ouvrir la voie qui permettra notamment de matérialiser les opérateurs formulés sous la description adaptée du langage Signal.

La description du chaînage ligne peut apparaître superflu pour l'application du suiveur de regard, par contre les possibilités offertes par cette structure pour les capteurs vidéo linéaires sont indéniables. Plutôt que de sérialiser les échantillons pixels au travers d'un registre de décalage, on pourrait calculer la chaîne de la ligne active directement afin d'augmenter considérablement la fréquence ligne. Le rythme n'est plus de l'ordre de l'échantillonnage du pixel, mais de la ligne complète. La seule contrainte est la profondeur du pipeline qui n'offre le résultat final qu'après l'équivalent en temps du nombre de pixels qui composent la ligne.

Le suiveur de regard bénéficie des avantages apportés par le traitement de l'image fournissant une liste chaînée structurée. Il n'est plus nécessaire qu'un opérateur humain se préoccupe du bon déroulement des mesures lorsque la mesure n'est plus possible (EX: oeil en dehors du champ couvert par la caméra). La qualité du codage de la liste chaînée sur une image suffisamment typée permet d'avoir du recul en terme de présence ou non de la pupille de l'oeil. Les possibilités offertes d'adaptation aux changements environnementaux et la sélection des sources Infra-rouges autour de l'écran offre un retour d'informations maîtrisées et recueillies sous formes de reflets désignés distinctivement.

Les possibilités futures à partir d'une conception câblée « temps réel » permettront d'optimiser les mesures et de profiter pleinement de la désignation sans calculs à base de conservation du pseudo vecteur par l'intermédiaire de l'asservissement de la sélection des LEDs I.R. formant le point de référence en association avec le curseur du moniteur.

Les verres correcteurs ne sont pas gênants pour les mesures mais peuvent perturber la désignation des reflets cornéens dans certains cas de figure. Des confirmations sur la disposition des reflets sur le verre correcteur auprès des spécialistes du domaine ophtalmologique devraient permettre des mesures intéressantes en localisation spatiale du globe oculaire.

La sensibilité du dispositif peut être améliorée par un échantillonnage spatial supérieur mais au détriment d'un temps de calcul forcément supérieur. Le contrôle de la saisie trame avec une synchronisation sur le temps et moment d'éclairement des LEDs, éliminerait les effets d'intégration du capteur CCD lors de mouvements rapides de l'oeil.

Il est à signaler en dernier lieu que le dispositif de pointage dégage certains degrés de liberté qui permettent notamment un confort d'utilisation supérieur à ceux qui nécessitent une contention de la tête rigoureuse. Le fait de ne pas opter pour une surveillance continue est également source de confort.

Notre choix d'une surveillance proche de l'oeil par une caméra miniature n'affecte pas le champ visuel. mais encombre l'espace de travail par un pied qui supporte le dispositif de prise de vue. Il est possible d'améliorer l'encombrement par l'utilisation de caméra avec capteur déporté sous forme d'un flexible. Ces modèles de caméra arrivent avec l'entrée de la visio conférence dans le secteur informatique.

Il n'est pas encore envisageable à ce stade d'expérimentation de parvenir à une résolution équivalent à la capture d'une lettre à l'écran en mode texte. Pour y parvenir il faudrait une détection d'un angle de rotation de 0.3° pour une observation d'un écran de 14 pouces à une distance de 60 cm.

La sensibilité du prototype à ce jour est d'environ 1.5 cm, ce qui offre la possibilité de sélection à l'écran de 18×14 zones distinctes.

6. Annexes

6.1 Combinaison des 64 cas qui peuvent se présenter dans la fenêtre glissante

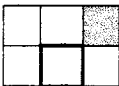
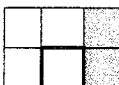
							
0	1	2	3	4	5	6	7
0	0	0	0	1001100001	1010100001	0011100001	0010100001
							
8	9	10	11	12	13	14	15
0101100001	0	0100100001	0	0011100001	0010100001	0011100001	0010100001
							
16	17	18	19	20	21	22	23
0	0	0	0	1101100000	1010001001	0011001101	0010001101
							
24	25	26	27	28	29	30	31
0	0000010111	0000010011	0000010111	0011000001	0010000001	0011000001	0010000001
							
32	33	34	35	36	37	38	39
0	0	0	0	0011100001	0010101011	0011101111	0010101111
							
40	41	42	43	44	45	46	47
0101000001	0000010101	0100010001	0000010101	0011100001	0010100001	0011100001	0010100001
							
48	49	50	51	52	53	54	55
0	0	0	0	0011000001	0010001011	0011001111	0010001111
							
56	57	58	59	60	61	62	63
0	0000010111	0000010011	0000010111	0011000001	0010000001	0011000001	0010000001

Tableau 6-1 : Codes associés aux fenêtres 3*2 pixels.

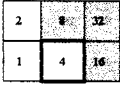

56 = code de la fenêtre code[i,j] 0000000000 = attributs (b9..b0) H,B,I,ds,fs,S,R,dn,fn,is

Tableau 6-2 : Affectations des différents éléments contenus dans le cadre.

6.2 Composition d'une liste chaînée structurée

La description des chaînes exige un ordonnancement dynamique à l'intérieur du tableau. En effet lors de la composition du codage de l'image pour des formes complexes, on peut arriver à devoir opérer des permutations des éléments dans une colonne spécifique qui désigne une chaîne.

Le problème de la description d'une image avec de nombreuses formes à coder et des niveaux d'imbrication élevés, est principalement l'occupation mémoire réservée au tableau. Le tableau a une dimension statique qui autorise le codage de Mc chaînes en simultanément sur une ligne. La dimension du tableau devient Mc colonnes sur N lignes.

La structure que nous avons retenue se présente de la manière suivante (Figure 6-1).

La composition des éléments chaînés est décrite au travers des définitions qui suivent ce texte.

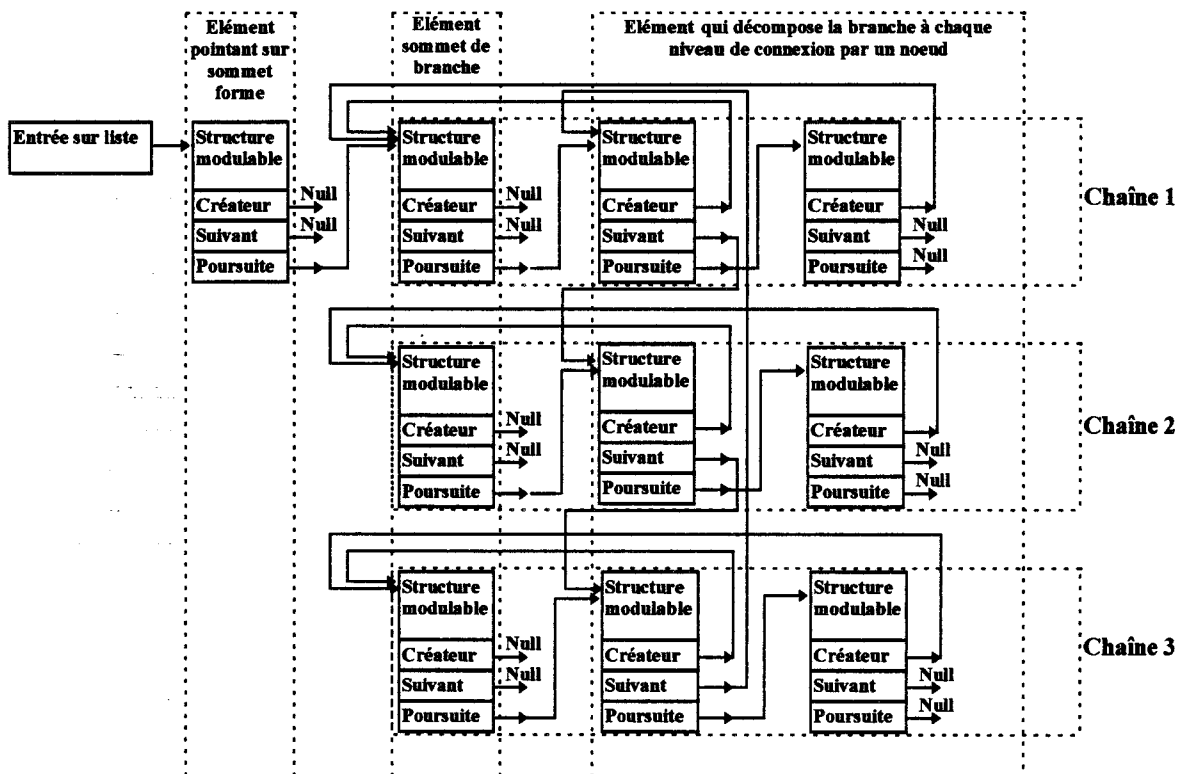


Figure 6-1 : Exemple de liste chaînée. (une forme sans évidement).

6.2.1 Le maillon

Pour parcourir la liste on crée un élément qui peut pointer sur d'autres éléments qui lui sont semblables.

Le point d'entrée dans la liste permet de scruter au niveau le plus haut la liste des objets contenus dans l'image. Cette première liste survole d'élément en élément chacun des sommets de forme objet tant qu'il existe un élément qui puisse être contacté par le pointeur *Suivant*. La structure interne du maillon sommet forme contient le nombre de noeuds qui composent la forme en totalité, le nombre de trous de la forme, la surface, le centre de gravité, et d'autres paramètres suivant les options retenues dans l'application. Uniquement en balayant cette liste on peut estimer l'emplacement de la forme recherchée suivant les attributs classiques comparés

aux éléments mémorisés dans le maillon. Le nombre d'objets distincts est dénombré par le nombre de maillons que compose la liste sur les sommets de formes.

Un deuxième niveau de parcours permet de scruter la forme en totalité en naviguant de noeud en noeud suivant des règles que chacun déterminera par rapport à la fonctionnalité recherchée.

6.2.1.1 *L'identificateur créateur*

Ce pointeur permet de désigner le retour sur le sommet du trait à chaque noeud qui compose le trait. C'est un élément qui permet d'accélérer le processus de parcours en décrivant l'image de bas en haut, alors qu'initialement elle est codée de haut en bas.

6.2.1.2 *L'identificateur suivant*

Ce pointeur permet de scruter les différents traits qui sont attachés soit par une fusion ou soit par une division de segment, pour une description ligne. Le parcours s'interrompt lorsque le pointeur *Suivant* retourne sur le premier maillon qui a généré la connexion. Si ce pointeur ne désigne pas de maillon, c'est que le maillon est un sommet début de trait isolé ou une fin de trait isolé (description symbolique: haut ou bas de forme).

6.2.1.3 *L'identificateur poursuite*

Ce pointeur permet de poursuivre la description de la chaîne de haut en bas à chaque niveau de connexion sur le même trait. S'il n'y a plus de maillon pointée c'est que nous sommes sur l'élément terminal du trait.

6.2.1.4 *Contenu de la structure du maillon*

Pour le premier niveau, ce sont les données essentielles qui caractérisent la forme de l'objet, sa position, etc..., qui composent les différents champs de la structure. Pour les maillons décrivant la forme, ils contiennent l'indice ligne et colonne du tableau statique afin de connaître l'emplacement du tableau qui nous permettra de positionner les segments intermédiaires entre les noeuds.

Le gain apporté par une description sous forme d'une liste chaînée noeudale par rapport à une liste chaînée complète est très appréciable. Il est évident qu'il est plus rapide de naviguer en dehors des noeuds sur une colonne d'un tableau statique plutôt que de se cheminer d'éléments en éléments pointés rangés de façon non contigus.

6.2.2 *Construction d'une liste graphe objet*

Pour illustrer la composition de la liste chaînée, on suit les étapes de construction par rapport à une image de référence (Figure 6-2).

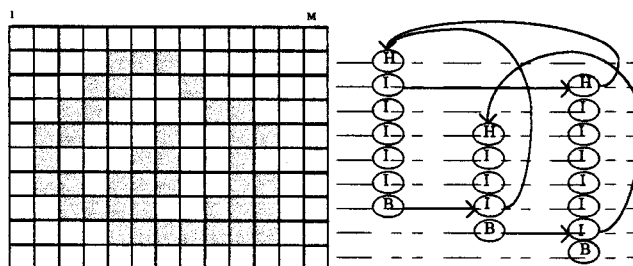


Figure 6-2 : Objet avec un évidement.

Séquence des étiquettes	N° de ligne	chaîne 1	chaîne 2	chaîne 3
H	2	H,1,Ø,5,3	Ø	Ø
I,S,I	3	I,1,2,4,2	H,2,1,8,1	Ø
I,I	4	I,1,Ø,3,2	I,2,Ø,9,2	Ø
I,H,I	5	I,1,Ø,2,2	I,2,Ø,10,2	H,3,Ø,7,1
I,I,I	6	I,1,Ø,2,1	I,2,Ø,10,1	I,3,Ø,6,2
I,I,I	7	I,1,Ø,2,2	I,2,Ø,10,2	I,3,Ø,5,3
R,I,I	8	B,1,3,Ø,Ø	I,2,Ø,9,3	I,3,1,3,4
R,I	9	Ø	I,2,3,5,7	B,3,2,Ø,Ø
B	10	Ø	B,2,Ø,Ø,Ø	Ø

Tableau 6-3 : Tableau de sortie de l'image codée, accompagnée de la séquence ligne des étiquettes segments.

Pour cet exemple on peut remarquer la non concordance entre l'ordre d'apparition du segment et son emplacement dans le tableau. A la ligne 6 le segment attribué à la chaîne 3 est en fait étiqueté avant l'élément de la chaîne 2. Pour ce cas précis il faut mémoriser l'apparition du code, affecter le nouveau segment à la première chaîne libre et transposer le contenu de la mémoire en fonction des affectations sur les traits.

La liste chaînée est construite à partir des étiquettes uniquement lorsque l'on rencontre un sommet, une vallée ou un noeud. Ailleurs seules les valeurs de position et largeur de segment sont stockées dans une table statique (Figure 6-3).

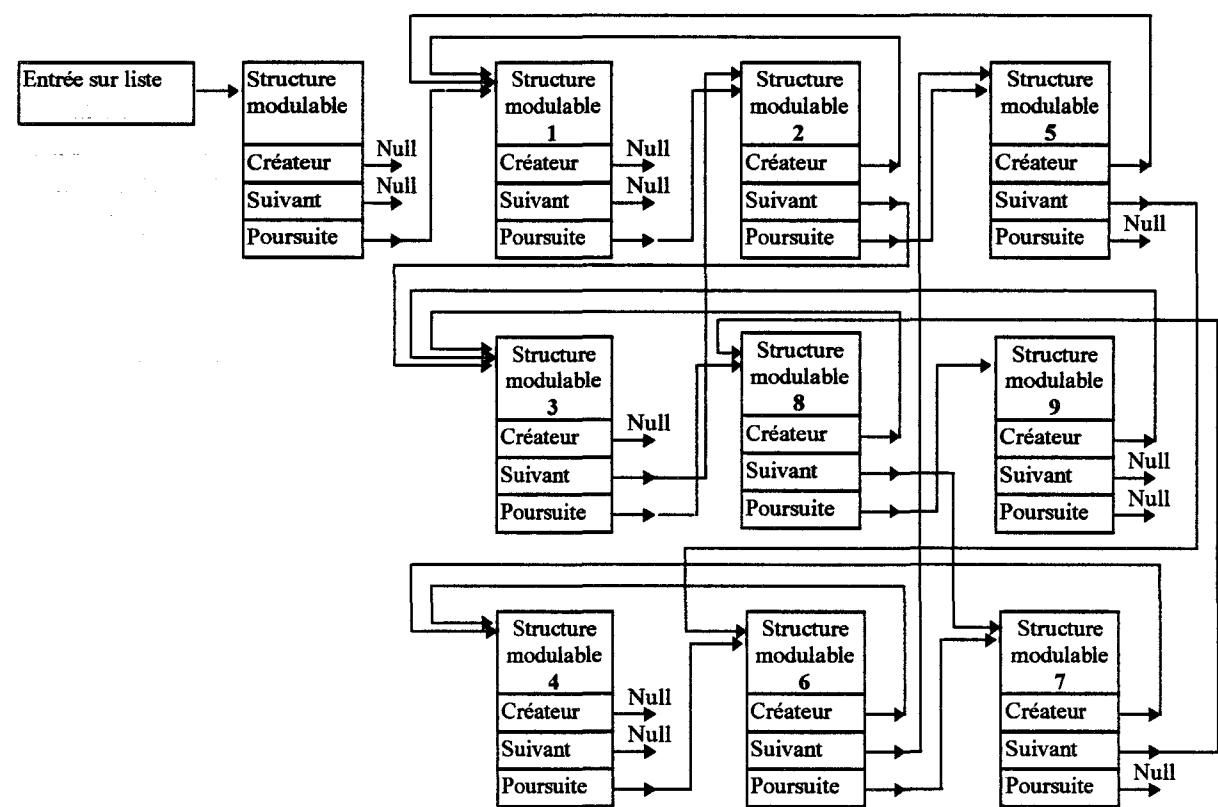


Figure 6-3 : Liste chaînée de l'image de référence.

Afin d'illustrer la construction de la liste chaînée on détaille ligne par ligne les interventions dans la structure chaînée. Les chiffres mis entre parenthèses signalent l'ordre d'apparition dans la liste et permettent de localiser le maillon en jeu dans la construction Figure 6-3.

A la ligne 2, on a le début d'un premier sommet. Dans ce cas l'image contient au moins une forme, on génère le maillon sommet forme. Il est supposé être seul à ce moment de la description, donc le pointeur suivant est nul. Une nouvelle forme implique l'attribution d'une chaîne dédiée à la description du trait d'où création du sommet trait (1).

A la ligne 3, on a la division de deux segments dépendant du sommet de la ligne précédente. Création d'un maillon poursuivant la première chaîne (2) et création d'un nouveau sommet trait sur la chaîne 2 libre(3). On opère aux échanges des pointeurs.

A la ligne 4, poursuite normale des deux branches, il n'y a pas d'intervention dans la liste chaînée car pas de noeuds.

A la ligne 5, début d'un nouveau segment compris entre les deux branches déjà existantes. Ce sommet à ce moment de la description est isolé, on crée donc un maillon sommet trait isolé (4). Les deux autres chaînes ne sont pas modifiées.

A la ligne 6 et 7, on effectue une poursuite normale de chaque branche, pas d'intervention dans la liste.

A la ligne 8, on opère à la réunion des deux premiers segments fusionné en un seul segment. On met fin à la chaîne 1 par un maillon fin de trait (5). Son pointeur poursuite est nul automatiquement. Le pointeur suivant désigne la deuxième branche qui prend le relais, d'où création d'un nouveau maillon poursuite de trait (6) sur la chaîne 3. Cet élément n'est pas un sommet de trait d'où opération d'échange des pointeurs de la branche afin de connecter les deux éléments. Le sommet qui à la ligne 5 était isolé devient élément à part entière de la forme décrite.

A la ligne 9, même schéma de construction qu'à la ligne précédente mais avec les chaînes 3 et 2. Création d'un maillon fin de trait (7) et un maillon poursuite de trait sur la chaîne 2 (8).

A la ligne 10, fin de la forme. Le dernier maillon de la forme a la même structure qu'une fin de trait (9) sauf que la progression dans la chaîne est stoppée.

De façon générale la succession des étiquettes permet de calculer toutes les caractéristiques dans la phase de chaînage, sinon il faut procéder à un parcours sur l'ensemble connexe de toutes les chaînes qui composent l'objet. Le calcul de centre de gravité, le calcul de surface sont possibles dans la même passe. Pour le calcul du périmètre on parcourt la courbe frontière externe suivant un concept décrit par la suite qui accélère le traitement vis à vis d'une description conventionnelle type Freeman.

6.2.3 Comparaison de deux listes suivant que l'une décrit un évidement interne et l'autre non.

A la ligne 8 de l'image de référence on échange un pixel objet en pixel fond, ce qui a pour incidence de ne plus avoir d'évidement interne.

La détection d'un trou se réalise par contamination d'une marque au moment d'une division de segment. Il faut que le trait marqué en son sommet soit de nouveau contacté par la marque véhiculée de noeud en noeud sur les autres chaînes afin de valider la présence d'un évidement.

A la Figure 6-4 on voit le parcours qu'il faut mettre en place pour de nouveau tomber sur la marque générée par le maillon (2). Par contre à la Figure 6-5 le cheminement de la marque s'interrompt sur le maillon fin de trait de la chaîne 1, puisqu'il n'a plus de contact avec la branche comme auparavant dans le cas de l'évidement.

La démarche de détection et de dénombrement d'évidement interne s'effectue dans la progression normale du chaînage. Il n'est plus nécessaire de parcourir la forme en différé pour que ce traitement puisse être effectif. L'ajout d'un jeton n'encombre pas le codage et permet de satisfaire pleinement à cette tâche.

Les figures suivantes (Figure 6-4 & Figure 6-5) sont accompagnées de tableaux indicatifs du parcours et de la formation des échanges des marques à partir des connexions entre formes internes.

Si l'on veut décrire le chemin qui permet d'identifier un évidement, il suffit de mémoriser les liens entre noeuds lorsque le parcours est accompli pleinement lors de la détection d'un évidement, sinon cette phase peut-être masquée.

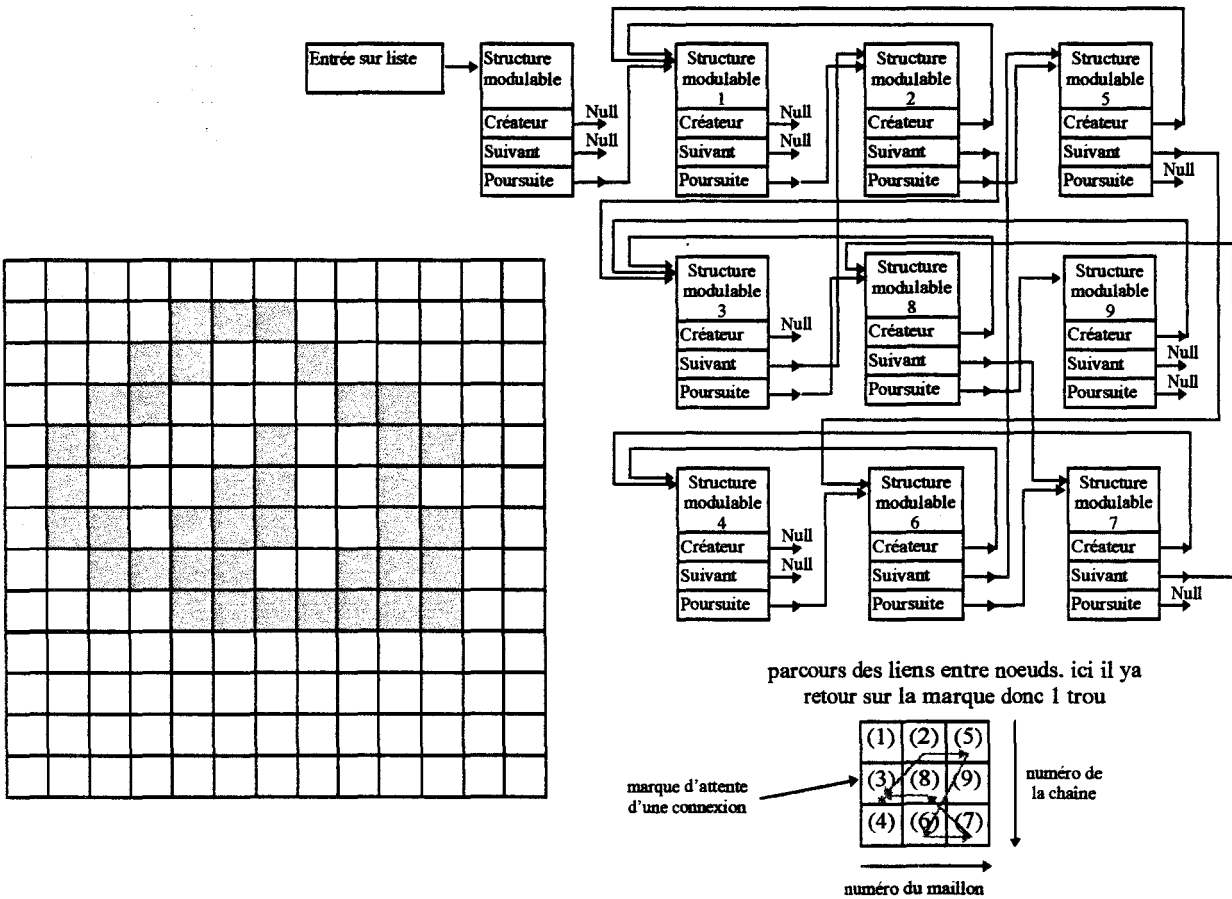


Figure 6-4 : Image avec un évidement.

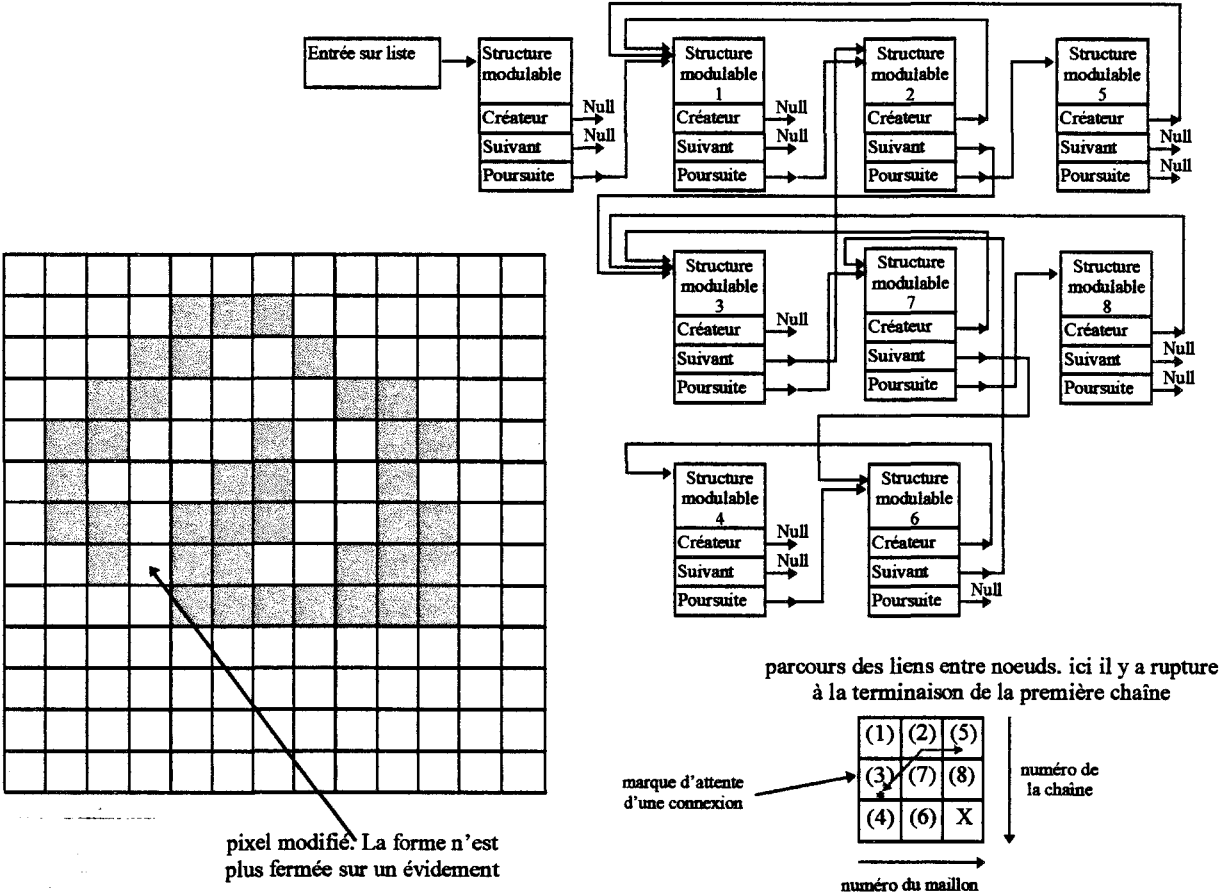


Figure 6-5 : Image sans évidement.

6.2.3.1 Règles de parcours sur le graphe afin de chaîner le contour extérieur

La méthode proposée pour construire le contour extérieur est de retenir pour chaque segment un point qui soit le plus représentatif du contour. La solution retenue est de décrire pour les sommets et vallées, le point central du segment. Dans le cas d'une progression descendante sur le contour (segment continuité) on retient le point extrémité gauche, à l'inverse pour la progression montante le point le plus à droite (voir Figure 6-6).

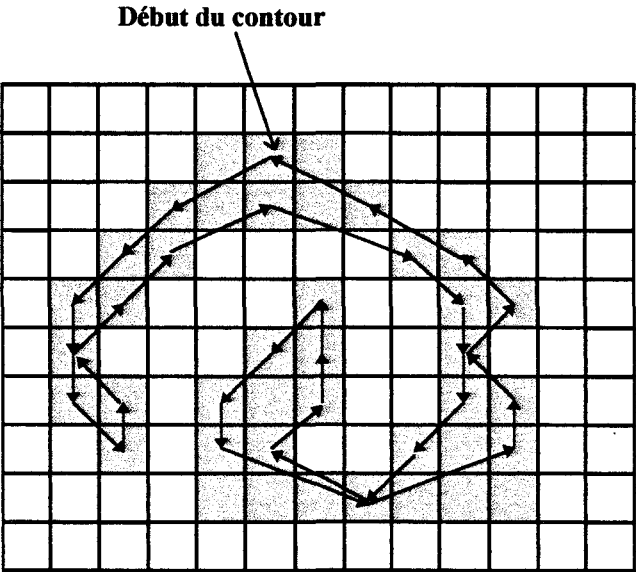


Figure 6-6 : Description du contour extérieur de la forme.

La progression du parcours s'effectue par convention dans le sens trigonométrique. L'intérêt de la description sous forme de liste chaînée noeudale est de se contenter d'aller pointer uniquement sur les noeuds pour connaître la progression future suivant la configuration de la forme à cet endroit.

On retient l'élément débutant dans la liste chaînée qui permettra de fermer la progression du contour. Puisque toute forme débute en son sommet par un segment début de trait isolé, c'est à cette position que débute le contour.

Dans notre description il n'y a plus de suivi de contour puisque le codage fait état de l'aspect de la forme et de la connexité de l'objet.

Dans une première phase on passe directement au premier noeud du trait qui compose la forme. Plusieurs cas peuvent se produire:

- ✎ Le maillon est terminal et isolé fin de trait. On décrit le contour en inversant le sens de progression.

- ✎ Le maillon est un noeud.

Si le trait est l'élément le plus à gauche pour une progression descendante, on poursuit dans le même sens. Dans le cas négatif, on inverse la progression et le trait parcouru est le premier à gauche.

Si le trait est l'élément le plus à droite pour une progression montante, on poursuit dans le même sens. Dans le cas contraire, on inverse la progression et le trait parcouru est le premier à droite.

- ✎ Le maillon est terminal et connecté. On vérifie s'il n'est pas possible de poursuivre sur le trait attaché dans le même sens de progression.

- ✎ Le maillon est un sommet connecté. On vérifie s'il n'est pas possible de poursuivre sur le trait attaché dans le même sens de progression.

Il est mis fin à la progression, dans le cas où le sommet est de nouveau atteint.

La Figure 6-7 permet de voir le cheminement des traitements sur les noeuds par rapport aux règles simples énoncées.

La description du contour pour l'image de la Figure 6-7 s'obtient en 10 progressions sur les noeuds.

On comprend mieux l'intérêt de la description structurée de la liste chaînée qui ne survole que les éléments noeud, sommet et fin de trait.

L'avantage est considérable par rapport à une description conventionnelle sur le gain de temps qu'apporte cette solution. Elimination de l'approche de suivi de contour, les marquages n'existent plus, le passage sur un même pixel ne pose aucun problème et les croisements sont autorisés.

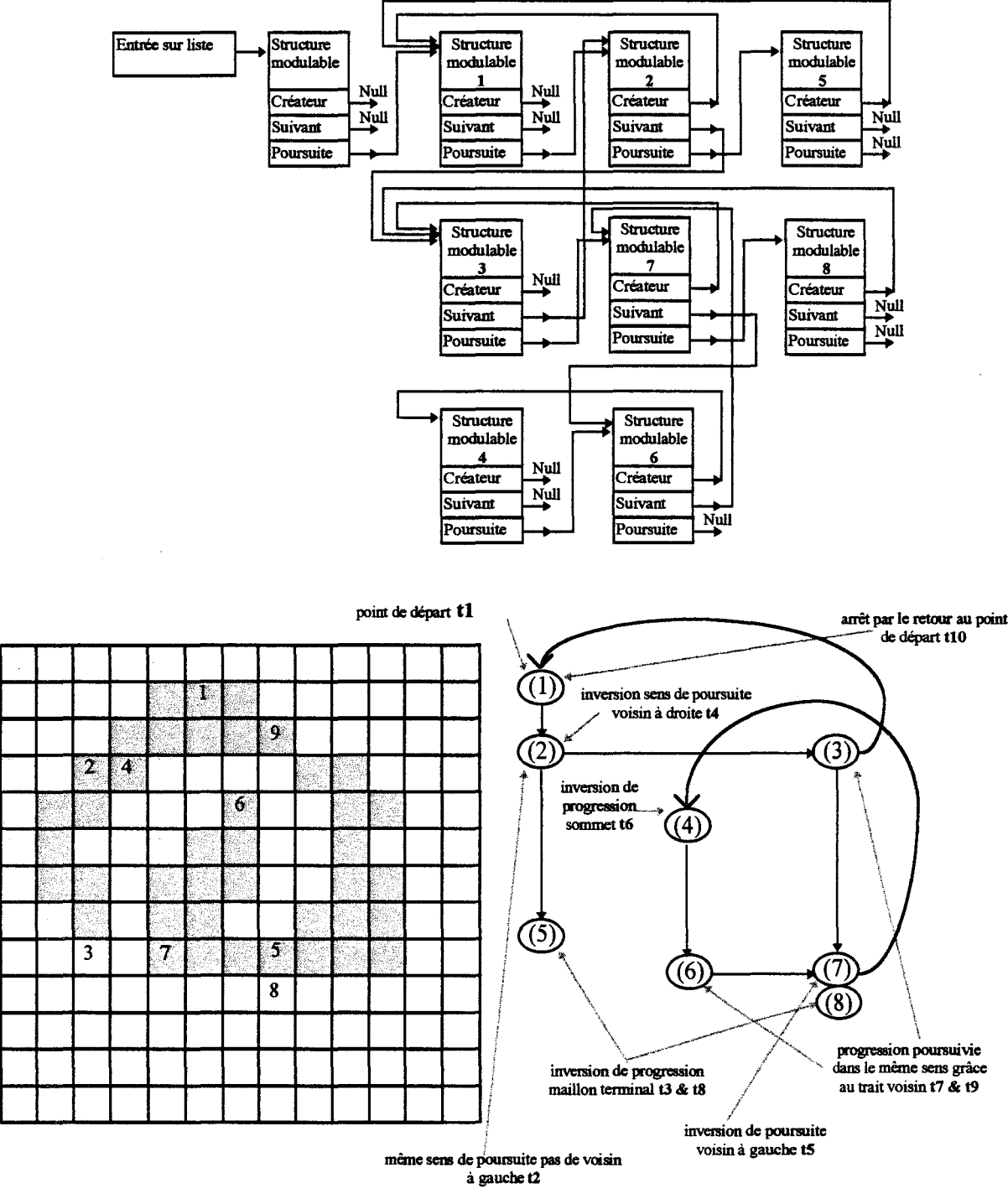


Figure 6-7 : Parcours sur le graphe du contour extérieur à l'objet.

Table des illustrations

Chapitre 1

Figure 1-1 : Coupe de L'oeil.	4
Figure 1-2 : Oeil et orbite.	4
Figure 1-3 : Axe optique (ligne passant par la fovéa et le centre pupillaire).	5
Figure 1-4 : Zones sensibles de la vision.	6
Figure 1-5 : Disposition des électrodes cutanées.	7
Figure 1-6 : Système expérimental DMI.	8
Figure 1-7 : Anneau scléral.	9
Figure 1-8 : Présentation du dispositif DMI pour l'homme.	9
Figure 1-9 : Méthode de variation de réflectance iris-sclérotique.	10
Figure 1-10 : Principe de récupération du reflet cornéen.	11
Figure 1-11 : Technique du miroir scléral (machine à écrire à commande visuelle).	12
Figure 1-12 : Double image de Purkinje.	13
Figure 1-13 : Principe du reflet cornéen.	13
Figure 1-14 : Dispositif de la détermination de la position de la pupille au travers d'une fenêtre circulaire.	14
Figure 1-15 : Dispositif optique de l'analyse des mouvements oculaires par la méthode du reflet cornéen.	15
Figure 1-16 : Exemples d'images I.R et illustration de l'image patron recherchée.	16
Figure 1-17 : Oculomètre NAC EYE MARK RECORDER 4.	17
Figure 1-18 : Principe optique du capteur NAC 4.	17
Figure 1-19 : Principe du vecteur cornéen.	18
Figure 1-20 : Dispositif optique qui laisse le champ de vision libre.	19
Figure 1-21 : Image du reflet pupillaire et du reflet cornéen, tache lumineuse réduite et d'intensité plus importante.	19
Figure 1-22 : Effets des mouvements de translation (tête) et rotation (oeil).	20
Figure 1-23 : Présentation du dispositif en milieu expérimental.	21
Figure 1-24 : Masquage partiel de la pupille par la paupière et les cils pour un déplacement oculaire vertical.	21
Figure 1-25 : Reflets infra-rouge du dispositif de discrimination des reflets parasites.	21
Figure 1-26 : Décomposition matérielle du Suiveur de regard.	22
Figure 1-27 : Oculomètre ASL 4000.	23
Figure 1-28 : Ensemble optique du dispositif ERICA.	24
Figure 1-29 : Description matérielle de la commande oculaire ERICA.	24
Figure 1-30 : Dispositif tenant compte de la position de la tête.	24
Figure 1-31 : Repérage du vecteur tête (origine centre image, extrémité intersection des axes verticaux et horizontaux passant par les deux reflets cornéens).	25

Chapitre 2

Figure 2-1 : Image découpée par le réseau rectangulaire.	27
Figure 2-2 : Pyramide des traitements d'images numériques.	28
Figure 2-3 : Paramètres caractérisant un contour à deux dimensions.	30
Figure 2-4 : Différents profils de contours.	30
Figure 2-5 : Opérateur RII filtre de dérivation et de lissage.	41
Figure 2-6 : Synoptique de la détection de contours par les opérateurs de Deriche.	43
Figure 2-7 : Interpolation du gradient.	43
Figure 2-8 : Exemple de formes imbriquées. L'arbre permet de connaître le niveau d'imbrication pour chacune des formes.	45
Figure 2-9 : Espace d'un voisinage de huit voisins.	46
Figure 2-10 : Exemple permettant d'exprimer les différentes relations précédentes.	48
Figure 2-11 : Illustration du codage de Freeman.	50
Figure 2-12 : Les cinq caractérisations des liaisons inter-segments.	51
Figure 2-13 : Exemple de suivi de contour par la règle du Left Most Looking.	52
Figure 2-14 : Algorithme général [BOM,91].	53
Figure 2-15 : Exemples des 4 configurations possibles.	54
Figure 2-16 : Image originale.	55
Figure 2-17 : Image norme du gradient sur l'image originale.	55
Figure 2-18 : Contours après suppression des non maxima-locaux (image originale).	56
Figure 2-19 : Image noyée dans un bruit gaussien.	56

Figure 5-10 : Image de l'oeil pour un regard dirigé vers le centre focale de la caméra.	105
Figure 5-11 : Reflet de la source lorsque l'oeil regarde le centre optique.	105
Figure 5-12 : Reflet cornéen lorsque l'oeil regarde la source.	106
Figure 5-13 : Image binarisée par le seuil pupille.	107
Figure 5-14 : Détection des reflets par un seuil de binarisation haut.	107
Figure 5-15 : Courbe de niveaux de gris sur une ligne n'interceptant pas de reflets.	108
Figure 5-16 : Courbe de niveaux de gris sur une ligne de l'image de l'oeil	108
Figure 5-17 : fenêtre de traitement des reflets cornéens.	109
Figure 5-18 : traitement vidéo de l'image infra-rouge de l'oeil.	109
Figure 5-19 : construction des reflets sur la cornée en fonction de l'orientation du regard.	110
Figure 5-20 : Mire de calibrage.	111
Figure 5-21 : Les six regards en coïncidence avec le repère matériel.	113
Figure 5-22 : Analyse des différences déplacement des reflets pour deux mouvements oculaires opposés.	114

Annexes

Figure 6-1 : Exemple de liste chaînée. (une forme sans évidement).	119
Figure 6-2 : Objet avec un évidement.	120
Figure 6-3 : Liste chaînée de l'image de référence.	121
Figure 6-4 : Image avec un évidement.	123
Figure 6-5 : Image sans évidement.	124
Figure 6-6 : Description du contour extérieur de la forme.	124
Figure 6-7 : Parcours sur le graphe du contour extérieur à l'objet.	126

Bibliographie

[AAA,92] E. MARTIN, J.L. PHILIPPE, R. REYNAUD, Y. SOREL - « Adéquation Algorithme Architecture. Outils d'aide à la conception et à la programmation d'architectures dédiées au traitement du signal et de l'image » - Document de synthèse de la conférence AAA de Lannion 14-15 Septembre 1992. Actes ENSSAT, GDR 134, 1992.

[ALI,88] S.M. ALI, R. BURGE - « A new algorithm for extracting interior of bounded region based on chain coding » - Computer vision graphics and image processing, Vol. 43, pp256-264, 1988.

[BAR,86] D. BARBA, J.F. BERTRAND - « Automatic region construction by edge detection and contour following in image segmentation » - 8 th International conference on pattern recognition, Vol. 2, Paris, pp 681-683, 1986.

[BAU,73] C.R. BAUGH, B.A. WOOLEY - « A two's complement parallel array multiplication algorithm » - IEEE Transactions on computer, Vol. C-22, n° 12, December 1973.

[BEN,90] A. BENVENISTE, P. LE GUERNIC - « Hybrid dynamical systems theory and the SIGNAL language » - IEEE Transactions on automatic control, Vol. 35, n° 5, pp 535-546, May 1990.

[BOM,90] V. BOMBARDIER, E. LEVRAT, J. BREMONT - « Un algorithme robuste de suivi de contour. Application au comptage de fibres nerveuses » - ISMM International symposium, Lugano, 19-21 Juin 1990.

[BOM,91] V. BOMBARDIER - « Codage en chaîne par suivi de contours en segmentation d'images binaires. Apport d'un opérateur flou pour une extension aux images monochromes » - Thèse de doctorat en automatique, Université de Nancy 1, 1991.

[BOU,84] L.J. BOUR, J.A.M. VAN GISBERGEN, J. BRUIJNS, F.P. OTTES - « The double magnetic induction method for measuring eye movement - Results in monkey and man » - IEEE Transactions on biomedical engineering, Vol. 31, n° 5, May 1984.

[BOU,93] E. BOURENNANE, M. PAINDAVOINE, F. TRUCHETET - « Amélioration du filtre de Canny Deriche pour la détection des contours sous forme de rampe » - Traitement du signal, Vol. 10, n° 4, pp 297-310, 1993.

[BOU,94] E. BOURENNANE, C. MILAN, M. PAINDAVOINE, M. ROBERT - « Implantation du filtre Canny-Deriche optimisé sous forme parallèle sur un ASIC » - Conférence adéquation algorithmes architectures, Workshop, Grenoble, pp 179-185, 20 et 21 Janvier 1994.

[BUQ,88] C. BUQUET, J.R. CHARLIER, V. PARIS - « Museum application of an eye tracker » - Med. bio. engineering and computing, Vol. 26, pp277-281, May 1988.

[BUQ,93] C. BUQUET, J.R. CHARLIER, G. DHELIN, S. TOUCAS, M. QUERE, I. INGSTER-MOATI - « Système de mesure des mouvements oculaires par traitement d'image pour les applications clinique ophtalmologique et neurologique » - Innov. tech. biol. med, Vol. 14, n° 3, 1993.

[CAN,86] J. CANNY - « A computational approach to edge detection » - IEEE Transactions Vol. PAMI-8, n° 6, pp 679-698, November 1986.

[CHA,82] J.R. CHARLIER, J.C. HACHE - « New instrument for monitoring eye fixation and pupil size during the visual field examination » - Med. bio. engineering and computing, Vol. 20, pp23-28, 1982.

[CHA,87] M. CHAMMAM - « Contribution à la conception et à la réalisation d'un oculomètre pour la mesure et le traitement automatique de la direction absolue du regard d'un opérateur humain au travail » - Thèse de doctorat en automatique et informatique industriel, Université de Valenciennes, 1987, n° ordre 87-09.

[CHE,87] B.D. CHEN, P. SIY - « Forward/backward contour tracing with feedback » - IEEE Transactions Vol. PAMI-9, n° 3, pp 438-446, May 1987.

[CHO,94] J.B. CHOQUEL - « Contribution à l'étude de la méthodologie de conception d'architectures parallèles par flot de données, pour le traitement temps réel de la représentation d'objets en trois dimensions » - Thèse de doctorat en électronique, Université de Lille 1, Février 1994.

[CHR,94] P. CHRISTIAEN, P. LOOSFELT - « Chaînage de contours » - Conference adéquation algorithmes architectures, Workshop, Grenoble, pp 21-28, 20 et 21 Janvier 1994.

[DEC,92] D. DECOMBLE - « Etude et réalisation d'un système de reconstruction et de visualisation 3D de coupes scanner IRM » - Mémoire CNAM, Lille, Septembre 1992.

[DER,87] R. DERICHE - « Using Canny's criteria to derive a recursively implemented optimal edge detector » - International journal of computer vision, Vol. 1, n° 2, pp 167-187, May 1987.

[DER,87] R. DERICHE, J.P. COCQUEREZ - « Separable recursive filtering for efficient multiscale edge detection » - IEEE International Workshop on mach. vision and mach. intelligence, Tokyo, December 1987.

[DER,90] R. DERICHE - « Fast algorithms for low level vision » - IEEE Transactions Vol. PAMI-12, n° 1, pp 78-87, January 1990.

[DER,91] G. DERICQUEBOURG - « Contribution à l'élaboration d'une méthodologie de conception d'interfaces opérateur incluant une commande visuel » - Thèse de doctorat en automatique, Université de Valenciennes, 1991, 91 VALE 0017.

Résumé :

Ce travail présente la conception d'un suiveur de regard utilisable comme outil de désignation sur un poste de travail informatique, sans contraintes pour l'utilisateur. Le dispositif utilise comme information l'image vidéo d'une partie du visage comprenant au moins un oeil. Quatre sources lumineuses infra-rouges se reflètent sur la cornée et sont visibles sur l'image vidéo. L'analyse de l'image permet de définir la zone observée sur l'écran. La première partie du travail consiste à extraire de l'image les informations « pupille » et « reflets » ; nous avons élaboré un procédé permettant d'effectuer rapidement cette extraction sur l'image binarisée à partir d'une fenêtre glissante 3x2 pixels ; ce procédé fournit une liste chaînée et structurée ; cette liste contient autant de sous-listes qu'il y a de formes distinctes. Il est alors possible de repérer les informations liées à la pupille et aux reflets. La technique de reconnaissance de forme s'avère robuste, et le procédé ainsi élaboré convient à un traitement automatique. Il convient également à un traitement hautement parallélisé, ce qui est validé par sa représentation sous forme d'un graphe flot de donnée, au rythme de la ligne. La construction de la liste chaînée est indépendante du niveau de complexité de l'image. En conséquence, ce procédé supporte le temps réel. Un traitement classique permet de déterminer la zone de l'écran visée par l'oeil. Pour améliorer la précision de cette zone, nous avons introduit un dispositif de LEDs contrôlées sur le pourtour de l'écran ; nous pouvons ainsi asservir le système de repérage, de manière à compenser les mouvements de la tête. Nous atteignons ainsi une sensibilité de 1,5 cm sur écran.

MOTS CLES :

- Traitement d'images
- Mouvements oculaires
- Concept flot de donnée
- Traitement parallèle
- Souris oculaire

Abstract :

This work presents the description of an eye gaze system for absolute designation on computer work station, without constraints for user. The device uses the video picture information of a part of human face with one eye at least. Four infrared light-targets with glints on the corneal are seen on the video picture. The determination of the absolute gaze on the monitor is possible by numerical picture analysis.

In the first time, we have made a pattern recognition method to extract « pupil » and « glints » ; a fast extracting proceeding on binary picture with a slipped window of 3*2 pixels has been developped. A structured chainlist is produced. This list contains as much sub-list as separated patterns. It's easy to take some informations on the pupil and glint patterns.

It's turns out that original method of pattern recognition is robust for an automatic treatment. A massive parallel treatment is also possible. We show that the data flow concept expresses a valid parallel architecture at the rate of the video line. The chainlist building is independent of picture level complexity. Consequently this method takes place in real time process.

A classical treatment is used for determining the absolute gaze. The contribution of enslavement of lights LEDs around the monitor allows to reach a resolution of displacement with a precision of 1,5 cm on the monitor. The system compenses the head movements.

KEYWORDS :

- Image processing
- Eye movements
- Data flow concept
- Parallel treatment
- Eye gaze computer interface