



FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE MAGISTER

En **Electronique**,

Option : **Téledétection**

Présenté par :

Mr. BELGUESMIA Said

Sujet :

***Modèle basé sur le filtre de Deriche et sur la
Transformée de Hough en Reconnaissance des
formes, Application à la vision robotique***

Devant le jury :

Mr AMEUR Soltane	Professeur à l'UMMTO	Président
Mr BOUTARFA Abdelhalim	Maître de conférence à l'université de Batna	Rapporteur
Mr ADANE Abdelhamid	Professeur à l'USTHB	Examineur
Mr DIAF Moussa	Professeur à l'UMMTO	Examineur
Mr ZIANI Arezki	Maître de conférences à l'UMMTO	Examineur

Remerciements

Ce travail a été effectué dans le cadre d'une collaboration étroite de plusieurs mois avec :

- Le laboratoire d'électronique avancée de l'Université de Batna;
- Le laboratoire de Vision Robotique et d'Intelligence Artificielle au Centre des Technologies Avancées de Baba-Hassen à Alger;
- Le laboratoire LAMPA de l'Université Mouloud Mammeri de Tizi-ouzou.

Je tiens tout d'abord à remercier mon Encadreur, Docteur Boutarfa Abdelhalim, Maître de Conférences classe A à l'Université El-Hadj-Lakhdar de Batna pour l'intérêt qu'il a porté à mon travail depuis mes débuts dans ce monde fascinant de la recherche, pour les discussions constructives et enrichissantes qu'il m'a apportées et aussi pour avoir été là dans les moments de doute. Ses conseils m'ont été particulièrement précieux durant ma vie scientifique dont il a assuré une direction motivante depuis plus de six mois. Qu'il trouve ici toute ma reconnaissance et ma gratitude.

Je remercie vivement Dr. Ameer Soltane, Professeur à l'Université Mouloud Mammeri de Tizi-ouzou pour m'avoir fait l'honneur de présider mon jury de thèse.

Je remercie également Professeur Adane Abdelhamid de l'Université Houari Boumediene, Professeur Diaf Moussa et Docteur Ziani Arezki de l'Université Mouloud Màmmeri de Tizi-ouzou pour l'intérêt qu'ils ont bien voulu porter à mon travail en acceptant la lourde tâche d'examineur. Merci infiniment pour leurs regards critiques et pertinents.

Je tiens aussi à exprimer toute ma gratitude à Monsieur Djakoun Oualid, Chargé de recherches à la Division Productique-Robotique au Centre des Technologies Avancées (CDTA) d'Alger, pour l'accueil chaleureux qu'il m'a prodigué et l'intérêt dont il m'a témoigné en m'orientant dans mon travail de recherche.

Je voudrais exprimer ma reconnaissance au Professeur Bouguechal Noureddine, Directeur du laboratoire d'électronique avancée de l'Université de Batna pour l'accueil qu'il m'a réservé au sein de son laboratoire, pour sa contribution scientifique, pour ses encouragements et pour les remarques judicieuses dont il a fait part lors de la lecture de ce

manuscrit.

Je tiens également à remercier tout particulièrement tous mes enseignants de mon Département et tout le mérite leur revient pour ce que je suis aujourd'hui sans omettre de citer tous mes collègues de promotion pour l'amitié et le soutien qu'ils m'ont témoignés.

Je réserve mes derniers remerciements à toute ma famille père, mère, frères, sœurs et ma femme pour leur dévouement, leurs sacrifices et leur soutien indéfectible.

Je tiens à remercier tout mes amis M'henna, Mourad, Boussad, Idir, Djamel, Areski, Djouhar, Fazia et tout le staff de Font office MOBILIS à Alger.

Table des Matières

INTRODUCTION GENERALE.....	1
1 Historique.....	2
2 Problématique.....	3
3 Cadre et hypothèses de notre travail de recherche	4
4 Présentation de notre travail de recherche	4
CHAPITRE 1	
LA VISION ARTIFICIELLE.....	6
1.1 Qu'est-ce que la vision ?.....	6
1.2. Comprendre la vision	7
1.3 Une théorie de la vision	7
1.4 Le paradigme de David Marr	9
1.5 Vision pour la navigation de robots mobiles.....	10
1.5.1 Navigation visuelle en milieu intérieur	11
1.5.2 Navigation visuelle en milieu extérieur	12
1.6 Reconnaissance d'objets (l'aspect général)	12
1.6.1 Choix des primitives	14
1.6.2 Définitions principales	16
1.6.2.1 Systèmes de Vision pour la Reconnaissance	16
1.6.2.2 Définition de l'image.....	17
1.6.2.3 Acquisition d'image	18
1.6.2.4 Le Capteur CCD.....	18

1.6.2.5 Principe	18
1.6.2.6 Caractéristiques	18
1.6.2.7 Avantages.....	18
1.6.2.8 Inconvénients	19
1.7. DISCUSSION.....	19

CHAPITRE 2

SEGMENTATION D'IMAGES 20

2.1 Introduction.....	20
2.2 Approche frontière	20
2.2.1 Extraction des contours	21
2.2.1.1 Méthodes surfacique	22
2.2.1.2 Méthode morphologique	22
2.2.1.3 Méthodes dérivative	22
2.3 Filtrage et lisage de bruit	23
2.3.1 Filtrage linéaire d'une image.....	23
2.3.1.1 Filtre moyen	24
2.3.1.2 Filtre Gaussien	24
2.3.2 Filtres non linéaires	25
2.3.2.1 Filtre médiane	25
2.4 Différents approche dérivative	25
2.4.1 les approches gradients	25
2.4.2 Les approches Laplaciennes	26

2.5	Application des approches dérivatives aux images numérique	26
2.5.1	Opérateurs Dérivatifs de premier ordre	26
2.5.1.1	Opérateur de Prewitt et Sobel	26
2.5.1.2	Opérateur MDIF	27
2.5.1.3	Opérateur NAGDIF	27
2.5.2	Opérateur deuxième ordre	28
2.5.2.1	Opérateurs de Marr et Hildreth	28
2.6	Approche par filtres optimaux	29
2.6.1	Approche de Canny	29
2.6.2	Filtres de Shen-Castan	31
2.6.2.1	Implémentation des opérateurs bidimensionnels de Shen-Castan	32
2.6.2.1.1	Lissage	32
2.6.3	Filtre de Deriche	33
2.6.3.1	Utilisation de filtre de Deriche	34
2.6.3.2	Calcul de gradient	34
2.6.3.3	Discretisation de filtre de Deriche	35
2.6.3.4	Discretisation de filtre de lissage	35
2.7	Application aux images	36
2.7.1	Suppression des non maxima locaux	36
2.7.1.1	Etapas de l'algorithme	37
2.7.2	Seuillage par hystérésis	38
2.7.3	Affinage des contours	38

2.7.4 Chaînage des contours.....	39
2.8 Discussion	43
CHAPITRE 3	
LA TRANSFORMEE DE HOUGH.....	44
3.1 Introduction	44
3.2 Classification des méthodes	44
3.2.1 Méthodes locales.....	44
3.2.1.1 Détection de droite à partir d'une image de gradient	44
3.2.1.2 Regroupement des segments élémentaires de droites.....	45
3.2.2. Méthodes globales.....	45
3.2.2.1 La Transformée de Randon	45
3.2.2.2 La Transformée de Hough (TH).....	47
3.2.2.2.1 Développement.....	47
3.2.2.2.2 Dimension des paramètres ρ_0 et θ_0	50
3.2.2.2.3 Propriétés de la Transformée de Hough (cas d'une droite)	51
3.3 Application de la Transformée de Hough.....	51
3.3.1 Implémentation de la Transformée de Hough	52
3.3.2 Quantification du plan des paramètres.....	52
3.4 Etat de l'art de la Transformée de Hough.....	55
3.4.1 Transformée de Hough probabiliste.....	55
3.4.2. Transformée de Hough aléatoire	55
3.4.3. Transformée de Hough hiérarchique	55
3.4.4. Transformée incrémentale	56
3.4.5 Transformée de Hough Additive.....	56
3.5 Parallélisation de la Transformée de Hough.....	57
3.5.1 Calcul de la TH locale	57
3.5.2 Calcul de la TH globale.....	57
3.5.3 Avantage de Parallélisme	58
3.5.4 Accélération matérielle de l'ATH	58
3.6 Discussion	59

CHAPITRE 4

APPLICATION A LA VISION ROBOTIQUE.....	60
4.1 Introduction.....	60
4.2 Application du filtre de Deriche.....	60
4.2.1 Organisation originale 2D.....	61
4.2.2 Organisation parallèle d'un opérateur 1D.....	62
4.2.3 Complexité.....	63
4.3 Optimisations algorithmiques.....	63
4.3.1 Forme cascade des opérateurs 1D.....	63
4.3.2 Nouvelle expression du Lisseur.....	64
4.3.3 Nouvelle Organisation 2D.....	64
4.3.4 Précision des Calcul.....	66
4.4 Calcul de gradient.....	67
4.4.1 Détection des Maximas Locaux.....	67
4.4.2 Seuillage par hystérésis.....	67
4.5 Implémentation de La Transformée de Hough.....	68
4.5.1 Parallélisme.....	68
4.5.1 Extraction des droites significatives.....	68
4.6 Tests et résultats.....	70
4.6.1 Interprétations des résultats.....	73
4.6.2 Application de notre modèle Rdf.....	74
4.6.3 Présentation de l'organigramme de notre modèle Rdf.....	77
4.7. Résultats expérimentaux.....	77
4.8 Discussion.....	79
CONCLUSION GENERALE ET PERSPECTIVES	83
ANNEXES	87

Annexe A : ROBOT MOBILE (CESA)	87
Annexe B : SYSTEME DE VISION.....	89
REFERENCES BIBLIOGRAPHIQUES	93

Liste des Figures

Figure 1.1	Stratégies adaptées par la vision par ordinateur.....	8
Figure 1.2	Synoptique général de la reconnaissance d'objets	13
Figure 1.3	Cônes et cylindres généralisés	15
Figure 1.4	Chaîne d'une vision artificielle	17
Figure 2.1	Quelques modèles de contours.....	21
Figure 2.2	les étapes d'extraction des contours	23
Figure 2.3	filtre gaussien	24
Figure 2.4	Orientation des repères dans l'image numérique	26
Figure 2.5	contour de la perturbation par un bruit gaussien.....	29
Figure 2.6	Filtre de Shen-Castan	32
Figure 2.7	Interpolation dans la direction du gradient	37
Figure 2.8	Principe d'affinage des contours.....	39
Figure 2.9	Principe de chaînage.....	40

Figure 2.10	(a) : image originale, (b) : gradient en x (filtre de Deriche), (c) gradient en y (filtre de Deriche), (d) extrema locaux de la norme du gradient dans la direction du gradient.....	41
Figure 2.11	image maxima locaux.....	42
Figure 2.12	Contours obtenus après seuillage par hystéresis	42
Figure 3.1	Paramétrisation de la droite p avec les paramètres ρ^* et θ^*	46
Figure 3.2	La Transformée de Randon d'une droite.....	46
Figure 3.3	Transformée de Hough (a): plan cartésien, (b): plan des paramètres (a,b).	48
Figure 3.4	Quantification du plan des paramètres (a,b).....	48
Figure 3.5	Une droite avec un paramétrage polaire	49
Figure 3.6	Transformée de Hough pour un paramétrage polaire.....	49
Figure 3.7	Paramètres polaires de deux droites opposées	50
Figure 3.8	Champ de la dimension de ρ	50
Figure 3.9	Le champ de dimension de ρ après décalage de l'origine	51
Figure 3.10	Illustration des différents seuils sur une droite	52
Figure 3.11	Organigramme de la TH.	54
Figure 3.12	Structure pyramidale de la TH hiérarchique.....	55
Figure 3.13	illustration de la propriété de la TH.....	56
Figure 3.14	Image 8*8 subdivisée en 4 blocs.....	57

Figure 4.1	Organisation classique des calculs du filtre de Deriche	61
Figure 4.2	Forme parallèle de l'opérateur récursif DH.....	62
Figure 4.3	Forme cascade de l'opérateur récursif DH.	63
Figure 4.4	Nouvelle organisation 2D du filtre de Deriche	65
Figure 4.5	Détail de la nouvelle organisation 2D du filtre de Deriche.	66
Figure 4.6	Organigramme de la TH pour un seul bloc.....	69
Figure 4.7	a-Image en niveaux de gris b. b-Image contours c- Image résultat_TH5....	71
Figure 4.8	Tableau accumulateur de l'image résultat_TH5 Balise de classe II.....	72
Figure 4.9	a-Image en niveaux de gris Balise de classe I. b-Image résultat Deriche c- Image résultat_TH5.....	72
Figure 4.10	Tableau accumulateur de l'image résultat_TH5 Balise de classe I.....	73
Figure 4.11	Image robot avec caméra embarquée.	75
Figure 4.12	Image robot aménagé	75
Figure 4.13	Image robot pendant l'acquisition et le traitement (Distance = 2m).....	75
Figure 4.14	Image robot pendant l'acquisition et le traitement (Distance = 0.8m).....	76
Figure 4.15	Image robot à sa sortie de salle Fin de tâche..	76
Figure 4.16	L'Organigramme de notre modèle de RDF.....	77
Figure 4.17	Image en niveaux de gris balise I.....	77
Figure 4.18	Image en niveaux de gris balise II.....	78

Figure 4.19	a- Image originale Balise de classe II. b-Image résultat.....	81
Figure 4.20	a- Image originale Balise de classe I. b-Image résultat	81
Figure 4.21	a- Image bureau. b-TH Image bureau.....	81
Figure 4.22	a- Image couloir labo LAMPA. b-TH Image couloir.....	81

Liste des Tableaux

Tableau 4.1	Coordonnées polaires des pics dans le tableau accumulateur de l'image résultat_TH5(baliseII)	71
Tableau 4.2	Coordonnées polaires des pics dans le tableau accumulateur De l'image résultat_TH5 (balisI)	73

Introduction

L'intérêt indéniable de la robotique mobile est d'avoir permis d'augmenter considérablement nos connaissances sur la localisation et la navigation de systèmes autonomes. Un robot doté d'un organe de vision a besoin d'identifier les objets qui l'entourent, ceux-ci sont repérés dans l'image qu'il a acquise.

Même si les robots sont entièrement autonomes dans leurs fonctionnalités, ils sont reliés à un réseau pour permettre de visualiser leurs états, sauvegarder leurs données pour des études off-line, pour qu'ils puissent interagir avec l'environnement (naviguer, se localiser où démarrer des démonstrations) et pour qu'ils sachent quelle visite n'est pas déjà occupée par un autre robot. On distingue sans trop d'ambiguïté un certain nombre de problèmes en robotique mobile, l'aspect matériel, qui consiste à choisir et dimensionner aussi bien la structure mécanique du système que sa motorisation, son alimentation et l'architecture informatique de son système de contrôle-commande apparaît comme le premier point à traiter. Bien évidemment, l'architecture du robot dépend fortement de l'offre et des choix technologiques du moment.

Le modèle que nous avons élaboré à cet effet est basé sur le Filtre de Deriche [Der 87a] et la Transformée de Hough additive [Kit 88], [Tzv 91], [Ati 92], [Tag 01] [Ach 04], [Kar 05], [Sat 08] afin de détecter des balises (amers) dans l'image. La Transformée de Hough additive est bien adaptée aux problèmes ayant un nombre important de données par rapport au nombre des paramètres à estimer. En effet, si les données et les inconnues sont de taille équivalente, il est difficile de trouver un accumulateur prépondérant par rapport aux autres. En plus, dû à la discrétisation et au bruit, il est possible que l'optimum soit délocalisé. Finalement, cette technique est capable de segmenter les données en plusieurs populations qui vérifient le modèle de référence. L'implémentation de la transformée de Hough sur un système informatique a l'avantage de présenter un large éventail de tâches à exécuter en parallèle. Ce dernier peut exécuter d'autres algorithmes en plus de la transformée de Hough et être capable de donner une performance aux calculs antérieurs tels qu'une segmentation ou aussi une détection de contours.

Cette dernière dispose de ressources dédiées aux opérations arithmétiques et aussi dans le domaine de traitement de signal. Le circuit donne de très bons résultats en termes de performances temporelles.

L'idée d'utiliser la TH est due à sa robustesse dans l'analyse de l'extraction des primitives d'objets et aussi à ses propriétés d'implémentation hardware (dans notre cas sur un circuit FPGA).

La Transformée de Hough Additive (THA) permet d'optimiser en temps et en espace le calcul global de la TH, ce qui justifie notre choix de l'étudier dans notre projet.

Introduction

1. Historique

Souvent imité, mais jamais égalé, l'oeil, roi des organes sensoriels, permet aux êtres vivants de détecter, de reconnaître et de se localiser en dynamique dans un environnement. Il est également complémentaire des autres organes extéroceptifs qu'utilisent les humains.

L'endroit qu'il occupe est une place privilégiée chez les êtres vivants. Il n'est pas étonnant que les scientifiques essaient de le reproduire artificiellement à l'aide de caméras, d'ordinateurs et d'algorithmes. La vision par ordinateur est un des axes majeurs utilisés pour la robotique et pour toutes les applications d'analyse et de traitement d'images.

Le rôle de la vision est d'apporter une perception dynamique de l'environnement évolutif ou non. Le robot évolue dans un environnement dont il fait partie et doit sans cesse l'observer et en relater les différents changements. Grâce au progrès de l'électronique et de l'informatique, la vision artificielle s'est considérablement améliorée au point de devenir un capteur intelligent.

Le domaine de la vision artificielle est vaste. Bon nombre de corps de métier s'y intéressent de près comme de loin, cela va des militaires, à la médecine afin d'assister les chirurgiens dans les opérations de microchirurgie ou de la reconstruction anatomique du corps et de la robotique mobile [Fla 01].

Quel que soit le domaine d'application, le principe général est identique. A partir d'une image ou séquence d'images, sont analysées des informations issues des scènes tridimensionnelles, afin d'informer le système et de l'aider à la prise de décision dans le cas d'une intelligence artificielle [Eng 03], [Oso 05].

Le traitement numérique des images ne s'est développé que de façon relativement récente, à partir des années 1960 avec l'apparition des calculateurs de la troisième génération.

En effet nous verrons qu'une image, même de dimensions domestiques, véhicule une quantité d'informations importantes et ces calculateurs ont permis, avec les progrès qu'ils ont apporté en capacité de stockage et en vitesse de calcul, de développer des algorithmes de traitement pour de multiples applications.

Parallèlement les domaines d'application du traitement numérique des images se sont progressivement mais considérablement élargis : médecine (radiographies, tomographies, scintigraphies), biologie, cartographie, météorologie, archéologie, astronomie, physique (spectroscopie, physique des plasmas, microscopie), applications militaires, applications industrielles (radiographies de matériaux, contrôles non destructifs, etc.).

De même que les sujets concernés, la nature des images traitées est extrêmement variée : photographies aériennes, photographies satellites, radiographies, spectre de physique, images radar, photos prises au microscope électronique. Aujourd'hui la liste ne se limite plus aux disciplines

Introduction

scientifiques et techniques. On peut dire qu'aucun domaine n'échappe ou n'échappera au besoin d'utiliser le traitement des images. Citons par exemple les métiers du spectacle, de la vidéo, de la mode et tous les secteurs de la création artistique.

De nouvelles applications pratiques sont maintenant en vue, allant des diagnostics médicaux à la vision par ordinateur appliquée à la robotique et à l'intelligence artificielle : reconnaissance automatique de formes d'objets par une machine appelée à remplacer l'être humain. Cette abondance des utilisations du traitement d'images a eu comme revers des multiplications de logiciels d'application spécialisés ainsi que de logiciels standards généralement écrits en langage évolué et aisément transportables sur n'importe quel type de calculateur.

2. Problématique

On peut dire que l'on traite une image dès lors que l'on extrait une information de cette image. Cette façon de voir n'a aucun rapport avec la création d'images purement synthétiques à l'aide d'un ordinateur. Ce domaine, la création d'images artificielles ou images de synthèse, ne fait donc pas partie de ce qu'on a coutume d'englober sous le vocable « traitement d'images » et nous n'en parlerons guère dans ce manuscrit. Ceci étant précisé, il existe de très nombreuses façons de traiter des images.

Dans une image, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes ; il s'agit de contours de type « saut d'amplitude ». Un contour peut également correspondre à une variation locale d'intensité présentant un maximum ou un minimum, il s'agit de contour « en toit ».

Les approches dérivatives sont les plus immédiates pour détecter et localiser les variations du signal. Elles sont fondées sur la recherche des points de l'image présentant un fort gradient, ou de dérivée seconde nulle. Dans les approches surfacique et morphologique, l'image des intensités est considérée comme une surface. Dans les méthodes surfaciques, la transition entre deux régions est modélisée par un gabarit utilisable à deux fins :

- un contour est présent quand la mise en correspondance entre le gabarit et une zone de l'image est bonne [Der 87b] ;
- l'approximation par facettes de la surface fournit une équation analytique locale qui permet de calculer de manière très précise (sub-pixel) la position du contour et ses caractéristiques.

3. Cadre et hypothèses de notre travail de recherche

- On peut chercher à détecter la présence de certaines formes, certains contours ou certaines textures du modèle connu, sans vouloir préserver les autres informations contenues dans l'image. Il s'agit là de problèmes de détection, classiques en théorie du signal : on veut extraire un signal de gabarit connu noyé dans un bruit constitué par tout ce qui n'est pas le signal utile.
- Le traitement d'images concerne aussi l'analyse et la compréhension de l'image dont un des buts est de doter les machines de capacité visuelle : c'est la vision par ordinateur qui débouche surtout sur la Robotique. D'ailleurs notre sujet s'est inspiré de cette philosophie. En effet l'analyse de l'image (appelée analyse de scènes) cherche à extraire les informations contenues dans les divers objets de la scène sans toutefois fournir d'interprétation. Les techniques de base sont essentiellement l'extraction d'attributs [Che 05], c'est à dire de « formes » (contours, textures) et la segmentation de l'image en zones présentant des caractéristiques spécifiques grâce à l'utilisation de seuillages, extractions et poursuites de contours (dans notre étude on a utilisé la détection par le filtre de Deriche). La classification des formes dans des catégories préétablies (le cas de nos balises) utilisera dans notre étude les principes de la morphologie mathématique (la Transformée de Hough additive) et permet ainsi la reconnaissance des formes. Puis, à un degré plus élevé, on utilise divers modèles de la connaissance afin de prendre les décisions en temps réel. Une base de données implémentée dans une Eprom permet à notre robot de faire la mise en correspondance adéquate en un temps record. Cette compréhension de l'image est en fait le point de départ de la vision automatique et des techniques qui relèvent de l'intelligence artificielle.

4. Présentation de notre travail de recherche

Notre travail s'organise comme suit :

Ce mémoire est subdivisé en quatre chapitres, en sus de références bibliographiques, la conception et l'expérimentation du logiciel (un programme source en langage Cpp) destiné à donner au robot une information de traitement parmi les autres informations nécessaires dont il a besoin pour atteindre une vision de haut niveau et de prendre les décisions utiles concernant par exemple sa localisation relative.

Dans le premier chapitre, nous nous sommes initiés à la vision artificielle et son côté applicatif.

Le deuxième chapitre, porte sur l'analyse des outils de détection de contours tel que le filtre de Sobel, le filtre du Laplacian, l'impact visé est de justifier notre choix qui s'est porté sur le filtre de Deriche.

Le troisième chapitre est consacré à la présentation de la Transformée de Hough et ses propriétés techniques associées à notre processus de reconnaissance de formes.

Introduction

Le Quatrième chapitre relate une application du modèle Rdf proposé en robotique mobile. Ce dernier à partir d'une acquisition d'images en niveaux de gris en entrée permet au robot séquentiellement d'avoir une bonne reconnaissance des attributs en sortie. Ce qui lui vaut d'avoir une vision de haut niveau.

Nous exposons ainsi ses performances sur des images en niveaux de gris de taille 256*256 et de format 8 bits/pixel. Enfin, nous finalisons notre mémoire par une conclusion générale en évoquant les points forts de notre méthode, ses limitations ainsi que quelques perspectives futures.

Chapitre 1

Initiation à La Vision Artificielle

1. Qu'est-ce que la vision ?

L'espace qui nous entoure ou à qui on est sensible a une structure tridimensionnelle (3D). Lorsque l'on demande à une personne de décrire ce qu'elle voit, elle n'éprouve aucune difficulté à nommer les objets qui l'entourent : téléphone, table, livre, etc. Et, pourtant l'information qui est réellement disponible sur la rétine de ses yeux n'est qu'une collection de points (environ un million !). En chaque point ou pixel il y a tout simplement une information qui donne une indication quant à la quantité de lumière et la couleur qui proviennent de l'espace environnant et qui ont été projetées à cet endroit de la rétine. Guidé à la fois par l'information codée dans l'image (ou la rétine) et par ses propres connaissances, le processus visuel construit des percepts.

Le téléphone ou le livre sont la réponse finale, résultant d'un processus d'interprétation qui fait partie intégrante du système de vision. De plus, il n'y a pas de correspondance terme à terme entre l'information sensorielle (la lumière et la couleur) et la réponse finale (des objets 3D). Le système de vision doit fournir les connaissances nécessaires afin de permettre une interprétation non ambiguë.

Depuis très longtemps, les chercheurs ont été fascinés par la capacité du système de vision humain à percevoir l'espace qui l'entour sans aucune difficulté. Avec la naissance de machines de plus en plus puissantes et les progrès techniques au niveau du traitement automatique des images, une nouvelle discipline est apparue sous le nom : « Vision par Ordinateur ».

Ce nouveau paradigme avait pour ambition de réaliser un système de vision artificiel capable de reproduire certaines fonctionnalités de la vision humaine au travers l'analyse des images de la scène captée par un dispositif d'acquisition. Pour réaliser un tel système, le traitement visuel a été formalisé d'une manière scientifique ce qui a donné naissance à la théorie de vision.

2. Comprendre la vision

La vision artificielle est la dernière née des approches de la recherche sur le mécanisme de la perception visuelle. Elle rejoint ainsi les écoles suivantes :

➤ La Neurophysiologie

Le neurophysiologue tente de comprendre le fonctionnement de l'acquisition et des mécanismes neuronaux des systèmes biologiques.

➤ La Psychologie Perceptive (ou psychophysique)

Le psychophysicien tente de comprendre les mécanismes sous-jacents à la perception visuelle, incluant les mécanismes intelligents. Son outil est l'analyse de la réaction à des situations particulières.

Il n'est pas suffisant de constater qu'un problème est complexe, Encore faut-il essayer de le comprendre dans ces moindres détails et de proposer une solution.

La vision a suscité l'intérêt de nombreux scientifiques et philosophes depuis déjà très longtemps, Parmi ceux-ci, les neurobiologistes mènent des recherches théoriques et expérimentales afin d'essayer de comprendre l'anatomie et le fonctionnement du cerveau dans son ensemble. Ils ont découvert une structure très complexe qui est loin de leur avoir révélé tous ses secrets, La tâche des neurobiologistes semble être à la fois grandiose et illusoire. Il reste et restera pour longtemps le bastion encore inconnu que les sciences humaines se proposent de conquérir, illusoires, car on est loin de connaître ses limites.

Avec la naissance de machines de calcul de plus en plus sophistiquées un certain nombre de scientifiques se sont attaqués au problème de la vision : est-il possible de construire un modèle computationnel pour la perception visuelle, il ne s'agit pas de fournir une explication, de comment marche la vision biologique mais de créer un modèle qui vu de l'extérieur possède des propriétés semblables.

Ce modèle artificiel peut-il être d'une utilité quelconque quant à la vision biologique?

Peut-il constituer la base d'une nouvelle technologie des machines qui voient? Il est certainement trop tôt pour répondre à ces questions et pour tirer des conclusions. Malgré les efforts non négligeables, il y a très peu de résultats convaincants.

3. Une théorie de la vision

L'élaboration d'une théorie scientifique demande trois étapes :

- Enoncer la théorie, spécifier et élaborer les concepts de base. Ces concepts doivent exprimer le cadre formel qui est à la base de la théorie.

- Exprimer ces concepts sous forme mathématique.
- Réaliser un ensemble expérimental qui permette de vérifier la théorie.

Voici comment la vision par ordinateur peut s'énoncer brièvement dans les termes de ce paradigme. La vision est un processus de traitement de l'information. Elle utilise des stratégies bien définies afin d'atteindre ses buts. L'entrée d'un système de vision est constituée par une séquence d'images. Le système lui-même apporte un certain nombre de connaissances qui interviennent à tous les niveaux. La sortie est une description de l'entrée en termes d'objets et de relations entre ces objets.

Deux types de stratégies (**Figure 1.1**) sont mis en jeu ascendant et descendant.

Les stratégies ascendantes tentent de construire à partir de l'information sensorielle une représentation la plus abstraite possible par exemple un ensemble de primitives géométriques 3D,

Les stratégies descendantes déduisent à partir de l'ensemble d'objets connus par le système une description compatible avec les primitives extraites de l'image. Il est alors possible de mettre en correspondance la représentation extraite de l'image avec les descriptions des objets afin de décrire les données sensorielles en termes de ces objets.

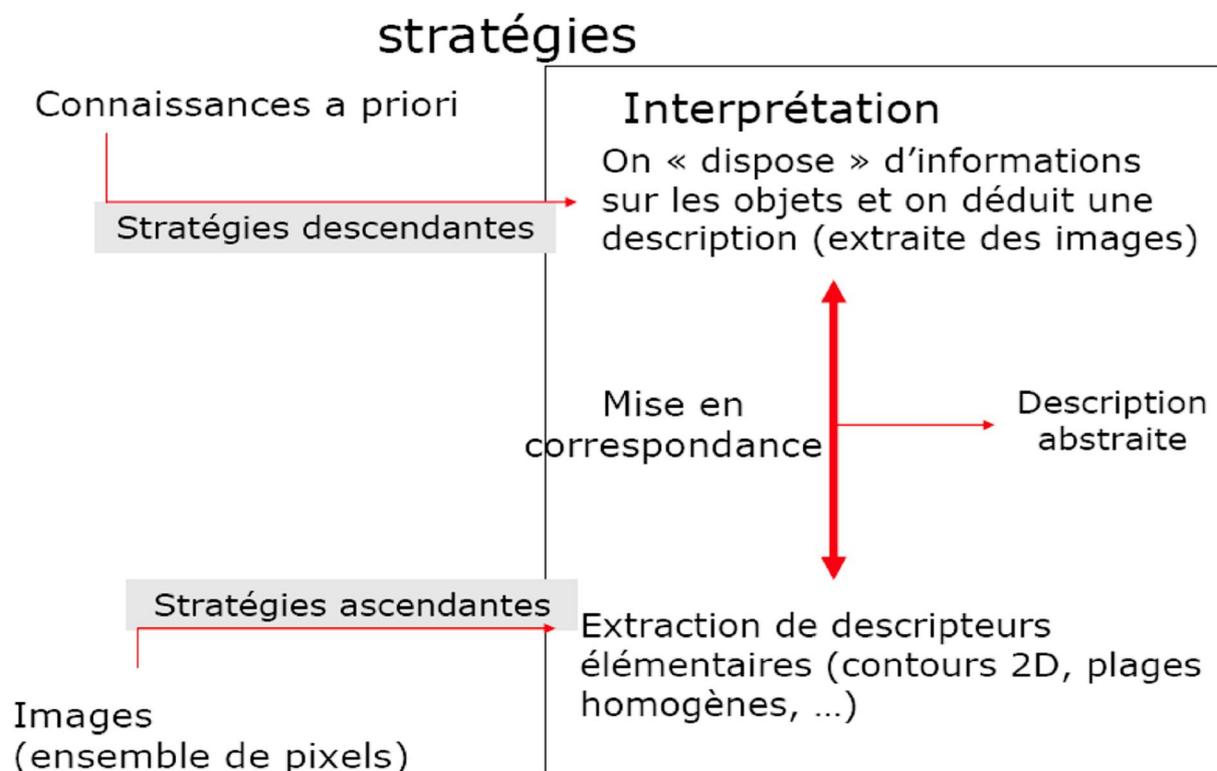


Figure 1.1 Stratégies adaptées par la vision par ordinateur

4. Le paradigme de David Marr :

Vers la fin des années 70 David MARR a proposé un modèle calculatoire pour le traitement et la représentation de l'information visuelle. Voici les principaux traits de ce paradigme :

- A partir d'une ou de plusieurs images un processus d'extraction de caractéristiques produit une description en termes d'attributs bidimensionnels. Ce niveau de représentation est appelé première ébauche (primal sketch).
- La première ébauche constitue l'entrée d'un certain nombre de processus plus ou moins indépendants qui calculent des propriétés tridimensionnelles locales relatives à la scène. Il s'agit d'une représentation, centrée sur l'observateur appelée ébauche 2.5D. Ces processus opèrent sur une séquence d'images (analyse du mouvement) ou sur une paire d'images stéréoscopiques ou sur une seule image. Dans ce dernier cas il s'agit de processus d'inférence qui utilisent des connaissances géométriques d'analyse des contours géométriques et statistiques (analyse des textures), photométriques (analyse des ombres) ou colorimétriques (analyse des reflets).
- l'ébauche 2.5D est mise en correspondance avec des connaissances 3D afin de construire une description de la scène en termes d'objets et de relations entre les objets. Il s'agit maintenant d'une représentation centrée sur la scène, la description ne dépend plus de la position de l'observateur.

Quelle que soit son origine, une image constitue une représentation d'un univers composé d'entités d'objets dans une scène d'intérieur, cellules, surfaces sismiques, organes du corps humain, etc.

Le but de toute méthode de segmentation est l'extraction d'attributs caractérisant ces entités. Les attributs étudiés correspondent à des points d'intérêt ou à des zones caractéristiques de l'image contours et régions.

La détection des contours (cf. chapitre 2) implique la recherche des discontinuités locales de la fonction des niveaux de gris de l'image.

La segmentation des contours consiste à approximer les contours par des représentations analytiques, telles que des droites ou des coniques. L'extraction de régions revient à déterminer des zones homogènes en niveaux de gris de l'image.

Par exemple dans le cas d'images réelles les contours correspondent aux frontières des objets et les régions à leurs surfaces. Ces deux approches contour et région, sont duales, en ce sens qu'une région définit une ligne par son contour et qu'une ligne fermée définit une région.

Elles amènent cependant à des algorithmes complètement différents et ne fournissant pas les mêmes résultats. Cette dualité est cependant peu exploitée dans la plupart des méthodes existantes.

5. Vision pour la navigation de robots mobiles

Chez l'humain, la vision est le sens qui transmet le plus d'informations au cerveau. Elle nous fournit une grande quantité de données en provenance de l'environnement et nous permet d'entreprendre une interaction intelligente avec les environnements dynamiques (éviter d'obstacles mobiles, rendez-vous avec autres agents mobiles, etc.).

De ce fait, il n'est pas surprenant de trouver une grande quantité de recherches sur le développement de capteurs qui essaient d'imiter le système visuel humain [Sie04].

De plus, les capteurs visuels utilisés par les robots intelligents doivent avoir les mêmes sensibilités et réponses à la lumière que notre système de vision.

Par exemple en robotique, au cours des deux dernières décennies, les innovations technologiques concernant la fabrication de caméras et l'évolution des ordinateurs ont permis d'intégrer des systèmes complexes de vision dans les systèmes embarqués, que ce soit sur des robots mobiles pour la navigation autonome ou sur des véhicules pour l'aide à la conduite.

La vision artificielle revêt une importance toute particulière car elle permet de fournir à la machine les capacités nécessaires pour réagir avec son environnement. Elle fournit les représentations à partir desquelles le robot prend des décisions.

Pour décrire l'état de l'art dans le domaine de la navigation de robots mobiles, nous faisons une distinction basée sur la structuration de l'environnement :

- Les environnements structurés peuvent être représentés par des primitives géométriques simples (détection de lignes droites, surfaces planes, couloirs, portes, etc.).
- Les environnements non structurés sont considérés pour des applications en milieu vraiment naturel (site planétaire, polaire, forestier, etc.); en milieu terrestre. Ce sont de riches sources d'information contextuelle, de couleur et de texture.
- Les environnements semi-structurés sont en essence des environnements naturels qui ont subi une modification partielle de l'homme, typiquement les sentiers ou chemins laissés par des passages fréquents de l'homme ou des animaux, par exemple dans le cadre d'activités agricoles.

Plus précisément, les contributions scientifiques de la navigation visuelle sont classifiées en deux catégories: la première, la plus prolifique pour robots d'intérieur (notre cas) et la seconde, en pleine croissance, pour robots d'extérieur.

5.1. Navigation visuelle en milieu intérieur

Dés les premiers projets en robotique mobile, les séquences d'images ont été proposées pour fournir des informations utiles à la navigation d'un robot [Gir79]. Elles sont généralement traitées par des méthodes de reconnaissance de formes pour détecter une cible connue par un modèle (apparence de la cible où modèle géométrique d'un objet) dans les images successives.

La plupart de ces approches utilisent des modèles structurés, le processus de navigation étant associé à l'une des modalités suivantes :

➤ **Utilisation de cartes connues de l'environnement:** le système dépend, pour se déplacer et se localiser, d'une carte (modèle) géométrique, probabiliste où topologique de son environnement. Elle est fournie par l'utilisateur [Hay02], mais ces modèles ne sont pas toujours faciles à obtenir. Les systèmes de vision permettent au robot de se localiser dans ces modèles, en détectant certains amers (obstacles naturels) de l'environnement.

➤ **Construction incrémentale d'une carte au fur et à mesure des déplacements :** Le robot construit son propre modèle de l'environnement en utilisant l'information provenant des différents capteurs (ultrasons, cameras, lasers, etc..) pendant une phase d'exploration [Thr98]. Dans un premier temps, le robot doit acquérir un modèle adapté pour sa propre localisation; ce sont souvent des cartes stochastiques éparées contenant les représentations paramétriques des amers détectés par le robot tandis qu'il se déplace dans l'environnement. Ces cartes sont construites de manière incrémentale grâce à des techniques d'estimation (filtrage de Kalman, filtrage particulaire..) proposées pour localiser le robot dans l'environnement et ce, pour fusionner les données acquises depuis la position courante. Ces méthodes, très étudiées depuis une quinzaine d'années, traitent donc du problème connu sous le mnémonique SLAM, Simultaneous Localization And Mapping.

➤ **Navigation dépourvue de carte :** Dans ce type de navigation, le robot ne se sert pas d'une représentation explicite de son environnement, mais plutôt de connaissances plus qualitatives où topologiques, construites à partir de la reconnaissance d'objets immergés dans la scène où de la détection, et du suivi temporel de cibles visuelles (extincteurs, bureaux, amers plans, corridors, etc.), [Mar04]. Citons aussi des approches bio-inspirées, comme par exemple une méthode fondée sur le flux optique, inspirée par le système visuel des abeilles [Riz 98].

En définitive, il s'avère que les approches géométriques sont bien adaptées aux environnements d'intérieur et des modèles mathématiques formels sont souvent proposés dans

la littérature. Ces modèles contiennent implicitement des actions pour éviter les obstacles, pour la détection d'amers, la construction ou l'actualisation de cartes, la reconnaissance d'objets et l'estimation de la position du robot.

5.2. Navigation visuelle en milieu extérieur

Pour les milieux naturels d'extérieur, la construction d'une carte ou la reconnaissance d'une forme est nettement plus compliquée (niveau de structuration faible: extraction plus complexe de primitives géométriques), à plus forte raison quand les scènes changent dynamiquement (à cause de la météo, de la saison, des conditions d'illumination, etc..) où quand d'autres agents dynamiques partagent le même environnement (piétons, autres robots, véhicules,..). Ce type de navigation peut être divisé en deux classes suivant le type de structuration, i.e., navigation en environnements structurés ou non structurés. Etant donné qu'il y a la fois des éléments naturels (sols ou terrains plats) et des éléments artificiels produits par l'homme (chemins, bâtisses,..).

6. Reconnaissance d'objets (l'aspect général)

La reconnaissance d'objets est un domaine très actif en vision par ordinateur. L'approche générale consiste à considérer que on dispose d'une banque de données où sont stockés des modèles d'objets et une vue du monde réel. On doit répondre à une question essentielle :

L'objet extrait (par les moyens précédents) existe-t-il dans la banque de données? Le problème dépend de la modélisation d'objets adoptée. Dans la majorité des cas cette modélisation est purement géométrique : un ensemble de caractéristiques appelées primitives.

L'opération de reconnaissance consiste en l'appariement des ces caractéristiques et celles décelées dans l'image.

Ce thème prend une part importante dans le domaine de recherche de la vision artificielle, et de nombreux systèmes, algorithmes et paradigmes ont été proposés durant ces dernières décennies [AGG96], [PER91]. Cependant, aucune solution universelle n'a pu être développée jusqu'à maintenant, seules des réponses partielles avec des réussites limitées ont été appliquées. D'ailleurs certains chercheurs avancent qu'il est impossible de concevoir un système de reconnaissance à large utilisation aussi performant qu'un système ciblé sur une situation spécifique.

Le problème majeur de la reconnaissance d'objets tient essentiellement dans son implication à tous les niveaux de traitement en vision et dans l'obligation de succès dans chacun d'entre eux. Ainsi parmi les traitements bas-niveau, nous trouvons les méthodes de

filtrage, ainsi que la détection et l'extraction de primitives. Au niveau intermédiaire les étapes de regroupement de primitives, de description et de représentation correspondent à ce que nous appelons la détection d'objets dans la scène. Finalement au plus haut niveau, la reconnaissance fait appel à des méthodes de classification, d'indexation et de mise en correspondance.

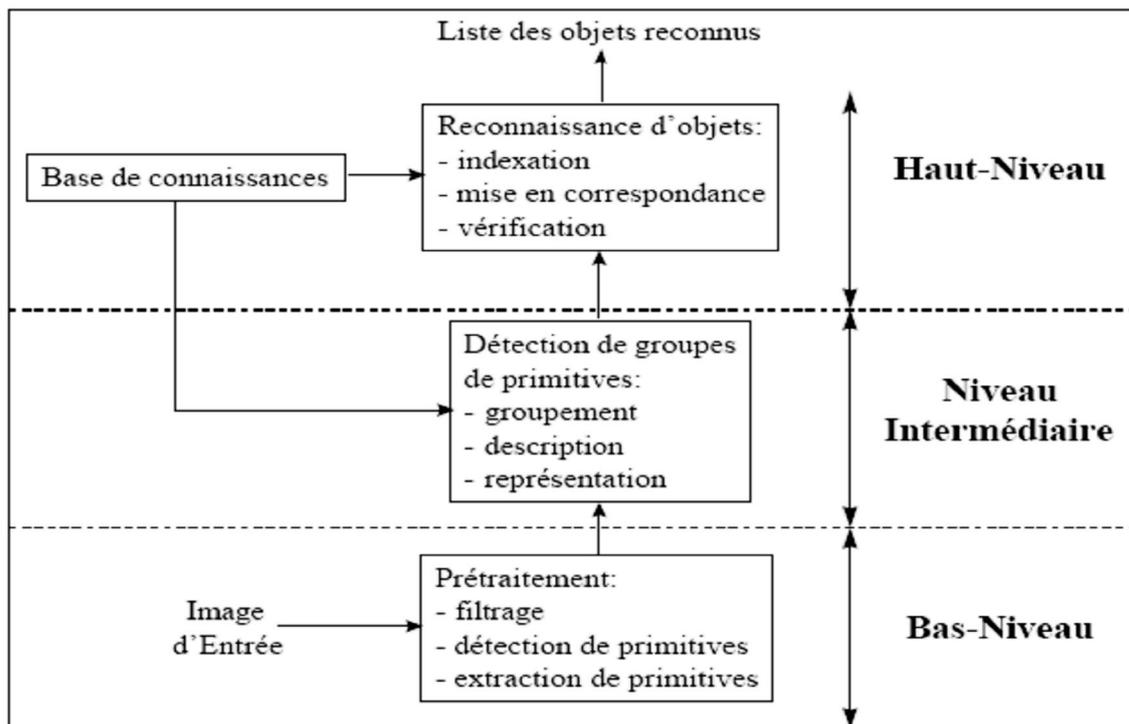


Figure 1.2 : Synoptique général de la reconnaissance d'objets

L'application de la reconnaissance d'objets tridimensionnels dirigée par les données est une tâche complexe et de nombreux facteurs participent à ces difficultés [POP94].

- Ne pas savoir quels sont les objets susceptibles d'être présents dans l'image.
- Ne pas connaître l'attitude d'un objet supposé être dans l'image.
- La possibilité qu'un objet soit partiellement occulté dans la scène.
- La possibilité que la scène soit très encombrée par des objets inconnus ou par des artefacts dus au procédé de traitement.
- Les limites de fiabilité et de précision des capteurs.

Le procédé de reconnaissance peut donc être assimilé à une mise en correspondance entre des composantes caractéristiques dans l'image et des caractéristiques similaires du modèle de l'objet. Les deux points les plus importants dans ce processus sont alors l'identification du

type de caractéristiques à utiliser et la détermination de la procédure permettant d'obtenir la correspondance entre les caractéristiques de l'image et du modèle.

Ainsi trois types d'appariements distincts sont généralement utilisés.

Le premier concerne la mise en correspondance de primitives 3D avec un modèle géométrique 3D. Cette catégorie implique une reconstruction tridimensionnelle des données image.

Le second type correspond à l'appariement entre primitives 2D qui implique alors la connaissance de modèles bidimensionnels ou l'utilisation d'images des objets à reconnaître (apparence des objets).

Finalement, nous distinguerons le cas de la mise en correspondance de primitives 2D avec un modèle géométrique 3D.

6.1. Choix des Primitives

Comme cela a déjà été cité précédemment, les primitives conditionnent non seulement le type d'objets qui pourra être reconnu mais également le type de représentations associé à chacun des objets détectés. Ainsi ces primitives sont directement destinées à décrire la forme et l'apparence des objets.

Deux catégories d'informations duales permettent d'aboutir à ces descriptions. La première consiste à se baser sur les contours des objets tandis que la seconde repose sur les régions. Alors qu'un contour est défini comme une variation brutale d'intensité lumineuse dans une image, une région est un ensemble de pixels homogènes au sens de critères tels que la texture, la couleur ou le niveau de gris. A partir de ces informations caractéristiques de l'image, il est alors nécessaire d'extraire des primitives permettant un codage simple, précis, rapide et caractéristique des objets à reconnaître.

➤ **Points caractéristiques** : Ces primitives sont susceptibles de représenter différents types d'informations dans l'image. Effectivement, ils peuvent caractériser des sommets, des coins, des intersections, des minima ou des maxima d'intensité lumineuse. Ainsi Basri utilise ces points caractéristiques afin d'établir des classes d'objets qui permettront une indexation des différents modèles prototypes [BAS92], [BAS93].

➤ **Segments** : Ils sont issus du regroupement des points de contour proches et similaires sous certains critères. Ces primitives forment une part très importante dans la caractérisation d'objets. Les segments sont ainsi souvent utilisés dans la reconnaissance d'objets polyédriques pouvant être représentés par des modèles filaires car ils sont beaucoup plus

robustes que les points. Cependant ils permettent également la description de formes courbes bien que cette description soit instable. Havalдар et al. proposent un système de reconnaissance d'objets génériques à partir d'une seule image de luminance basé sur l'utilisation de segments et de leurs caractéristiques [HAV94].

➤ **Courbes** : Ce terme désigne une large gamme de représentations qui incluent les courbes de Bézier, les B-splines, les cercles, les ellipses, les arcs, les rubans ou les coniques. Ces primitives permettent d'obtenir des descriptions plus stables au détriment d'une représentation et d'une extraction plus complexes.

➤ **Cônes et cylindres généralisés** : Ces primitives sont des surfaces créées en déplaçant une section de forme constante mais de taille variable le long d'un axe (**Figure 3.1**). Ainsi on trouve dans cette catégorie des formes géométriques telles que les pyramides ou les sphères.

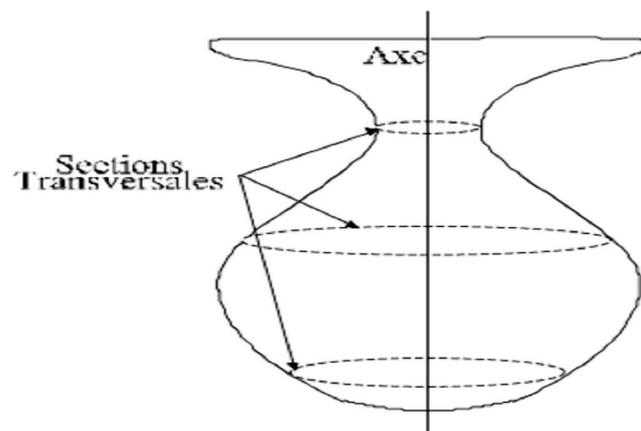


Figure 1.3 : Cônes et cylindres généralisés

➤ **Superquadriques** : Ces primitives volumétriques comprennent des formes de base comme les sphères, mais aussi toutes sortes de déformations de ces formes pour autant que soit préservée leur forme lisse et qu'aucune concavité n'apparaît. Ce sont en fait des équations d'une forme particulière dont les solutions définissent une surface fermée. Deux termes sont variables et définissent ainsi une gamme de formes qui inclut les cubes, pyramides, diamants, etc.. Six autres paramètres sont également ajoutés afin de préciser des notions telles que la taille. Ces primitives sont très peu utilisées en reconnaissance d'objets car elles nécessitent trop de paramètres à déterminer.

➤ **Géons** : Également appelées ions géométriques, ces composants élémentaires ont été introduits par Biederman [BIE87]. Ces primitives présentent un niveau d'abstraction supérieur aux précédentes et font appel à des techniques de regroupement afin d'être

constituées. Ainsi ces objets génériques représentés sous forme de parallélépipèdes, de cylindres, de pyramides, etc.. sont ensuite associés pour décrire des objets plus complexes.

Les primitives en vision sont très nombreuses du fait essentiellement de la diversité des objets existants. Ainsi parmi toutes ces possibilités, nous en avons sélectionné une afin de bâtir notre système. Elle repose sur les segments qui constituent notre primitive. Ce choix est guidé par la nature de milieu où le robot va évoluer, c'est-à-dire un milieu structuré qui composé essentiellement de segment de droites et de la simplicité d'extraction.

6.2. DEFINITIONS PRINCIPALES

6.2.1. Systèmes de Vision pour la Reconnaissance

Implicitement, le terme de vision renvoie à la notion d'image. L'image a été de tout temps un mode de représentation et de description de l'environnement pour l'homme que ce soit au travers de peintures, de gravures ou de photographies.

De plus physiologiquement, la vision est la capacité sensorielle qui permet de donner aux êtres vivants le plus d'informations sur leur environnement. Malheureusement, les explications fournies par la physiologie et la psychologie n'autorisent toujours pas une compréhension complète du phénomène de reconnaissance d'objet par images rétiniennes.

Ainsi en voulant donner à ses robots la capacité de voir, l'homme n'a pu imiter entièrement son propre capteur et a dû trouver de nouveaux moyens de perception visuelle.

Nous donnons dans la figure 1.4 le schéma synoptique d'une chaîne de vision par ordinateur

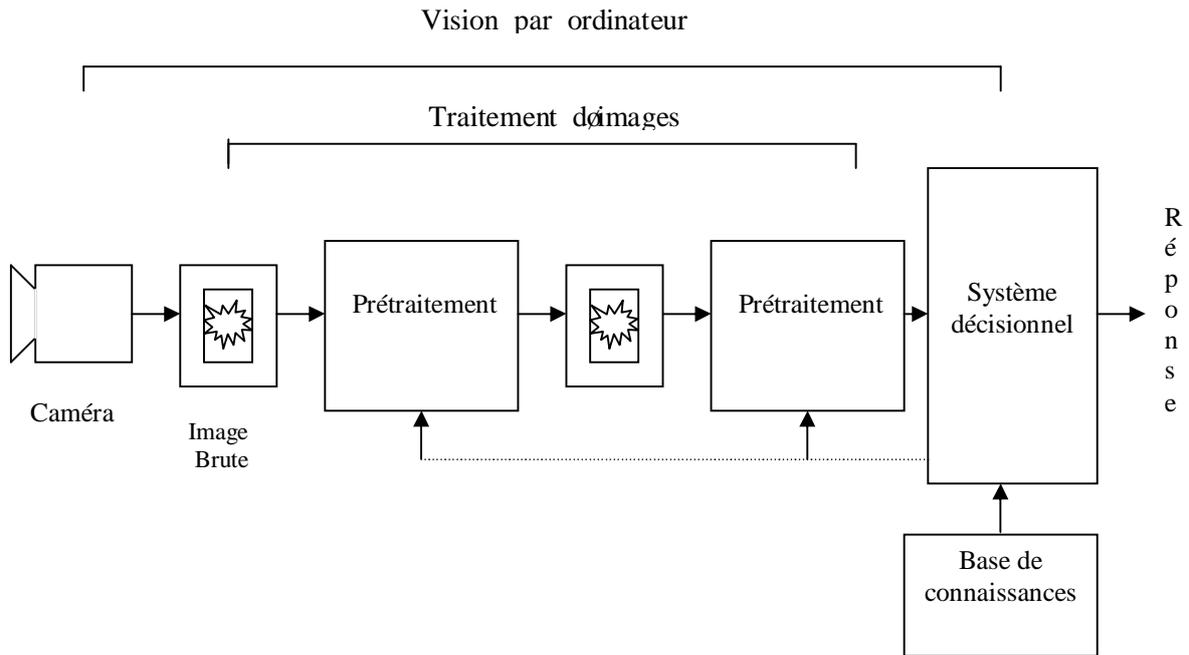


Figure 1.4 : Chaîne d'une vision artificielle.

La segmentation d'images constitue une étape clé dans l'unité de traitement d'images. Elle permet d'engendrer une description plus compacte de l'image (contours, régions) et plus exploitable que l'ensemble des pixels. La réussite de cette étape dépend fortement de la qualité de l'image à segmenter. Souvent un prétraitement s'impose pour améliorer la qualité de l'image brute et faciliter les étapes ultérieures.

L'étape de prétraitement doit lisser le bruit dans les zones homogènes en conservant les transitions formant les frontières entre les différentes régions de l'image.

6.2.2. Définition de l'Image

Une image désigne une structure de données de type matriciel contenant des pixels (picture elements) dont les valeurs discrètes représentent la quantification d'un signal de nature diverse. Le plus souvent, cette forme est bidimensionnelle et l'information est l'intensité lumineuse (couleur ou N.G). $I : [0, L-1] \times [0, C-1] \rightarrow [0, M]^p$ définit une image de L lignes et C colonnes dont l'information portée est définie dans un espace à p dimensions.

Si I est une image binaire, alors $(p, M) = (1, 1)$.

Si I est une image en niveaux de gris, alors $p = 1$ et le plus souvent $M = 255$.

Si I est une image couleur, alors $p = 3$ et le plus souvent $M = 255$.

6.2.3. Acquisition d'image

Nous nous limitons ici à une brève description de caméra à base de capteur CCD qui actuellement est très utilisée dans plusieurs domaines tels que la photographie grand public, la vision industrielle etc..

6.2.4. Capteur CCD

L'arrivée des capteurs CCD (Charge Coupled Device) en 1975 a été déterminante dans l'évolution de la vision bidimensionnelle. La rapidité d'acquisition, la robustesse et la miniaturisation sont autant d'avantages qui ont facilité leur intégration. Ces capteurs fournissent des informations de luminance correspondante à la lumière réfléchie par les objets de l'environnement et captée par la caméra. Les systèmes de vision sont très performants en termes de portée, précision et quantité d'informations exploitables.

6.2.5. Principe

Le capteur à CCD (charge-coupled device), ou capteur à transfert de charge est un composant basé sur une technologie de composants à semi-conducteurs. L'arrivée de photons sur le silicium va créer des charges électriques, qui vont s'accumuler dans des zones appelées "puits". Ces zones sont la résultante de différentes tensions électriques appliquées localement.

En jouant sur ces tensions, nous allons modifier la position des puits, et donc forcer les charges à se déplacer.

6.2.6. Caractéristiques

- Sensibilité : 1/100 Lux
- Sensibilité spectrale : Le visible et l'infrarouge
- Résolution : Actuellement 4096*4096 pixels
- Temps d'intégration : 1/100 s.

6.2.7. Avantages

- Solide, léger et compact.
- Aucune dérive géométrique de l'image.
- Aucun brûlage.
- Aucune rémanence.
- Durée de vie et fidélité très fortement accrues.
- Correction numérique des défauts éventuels envisageable.
- Aucune inclinaison de l'image sur les mouvements.

6.2.8. Inconvénients

- Moins de choix en gamme spectrale (utilisation de filtres).
- Sensibilité inférieure à certains capteurs à tubes.
- Résolution inférieure aux capteurs à tubes de haut de gamme (scanner).

7. Discussion

La vision artificielle, ou (vision par ordinateur), est une discipline qui consiste à convertir une image en données objets ou, plus explicitement, à identifier les objets contenus dans l'image par l'extraction et l'analyse de caractéristiques abstraites (features) à partir des pixels, suivant un processus de reconnaissance de forme similaire à celui opéré par l'humain.

Parmi les applications de la vision par ordinateur, on compte le contrôle de la qualité sur une chaîne de production, l'identification d'un individu par biométrie (analyse automatisée des empreintes digitales), le diagnostic médical et la classification de terrains à partir d'images satellites. Dans ce chapitre aussi nous avons présenté d'une manière générale le contexte de nos travaux, qui concernent les fonctionnalités visuelles pour la navigation de robots autonomes dans des environnements autant d'intérieur et que d'extérieur. Ces fonctionnalités visuelles seront plus spécifiquement le suivi d'objets dans des séquences d'images acquises depuis un robot en mouvement, et son application à la navigation d'un robot mobile dans les types d'environnement.

Chapitre 2

Analyse des outils de détection des contours

1. Introduction.

Le traitement d'images s'intègre dans le processus de la vision par ordinateur comme un outil et n'a donc pas de pouvoir décisionnel. Le traitement d'images se définit comme un ensemble de tâches destinées à extraire de l'image brut des informations qualitatives et quantitatives.

La segmentation d'image constitue une étape clé dans l'unité de traitement d'image dans un système de vision par ordinateur. Elle permet d'engendrer une description plus compacte de l'image (contours, régions) et plus exploitable que l'ensemble des pixels. Le processus de **segmentation** est le passage obligé pour la vision par ordinateur. En effet la littérature abonde sur le sujet et le nombre de travaux de recherche est important sur la segmentation des images numériques. Nous distinguons deux entités de segmentation. L'une est basée sur la détection de contours des objets c a d approche frontière (**discontinuité**). L'autre est basée sur notion de région qui fait référence à des groupements de points ayant des propriétés communes (niveau de gris, texture, **homogénéité**..).

Quelque soit l'origine de l'image, la réussite de cette étape de segmentation dépend fortement de la qualité de l'image et la méthode utilisée. Souvent un prétraitement s'impose pour améliorer la qualité de l'image brute et faciliter les étapes ultérieures. L'étape de prétraitement doit lisser le bruit dans les zones homogènes en conservant les transitions formant les frontières entre les différentes régions de l'image.

2. APPROCHE FRONTIERE :

La notion de frontière est associée à une variation de l'intensité lumineuse, comme les bords des objets, des variations d'illumination et d'absorption des rayons lumineux, í í ..

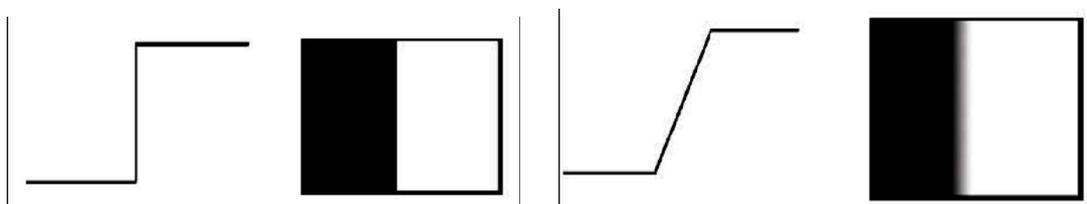
Dans un grand nombre de cas ces variations sont des informations importantes pour les opérations situées en aval de la segmentation. Elles constituent les frontières des régions correspondant à des bords des objets.

La recherche des contours dans une image numérique est l'un des problèmes les plus étudiés depuis l'origine des travaux sur l'imagerie numérique. Ceci est en grande partie dû à la nature très

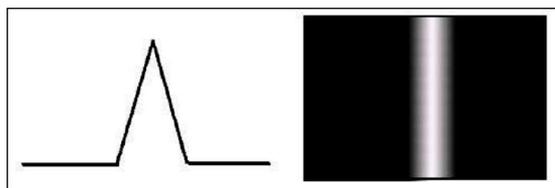
intuitive du contour qui apparaît très naturellement comme l'indice visuel idéal dans la plus grande partie des situations. Très schématiquement, les contours sont les lieux de variations significatives de l'information niveaux de gris (il y a très peu de travaux sur les contours dans les images couleurs ou multi spectrales). Dans cette approche, on suppose que l'image est une mosaïque de régions parfaitement homogènes. C'est à dire que les contours recherchés sont de type **créneaux**. De plus, la transition étant stricte, un contour doit être une chaîne de pixels d'épaisseur d'un pixel. Cette restriction sur la nature du contour a été imposée dans un premier temps pour des raisons de formalisation mathématique. Il est possible de construire des processus capables d'extraire d'autres types de contours comme par exemple des vallées ou des toits. Cependant, il n'existe pas à l'heure actuelle de processus complet et général qui pourrait extraire tous les types de contour ou a déterminé le type de transition afin d'appliquer la méthode la plus adéquate. On dénombre trois types pour l'approche frontière : méthodes dérivatives, surfacique et morphologique. Dans notre cas l'accent sera mis sur les méthodes dérivatives, spécialement sur le filtre de Deriche qui sera utilisé pour l'extraction des points contour dans l'image.

2.1. EXTRACTON DES CONTOURS :

Un contour peut être défini comme une "marche d'escalier" si le contour est net, comme une "Rampe" si le contour est plus flou ou comme un "toit" s'il s'agit d'une ligne sur un fond uniforme.



Marche : le contour est net (contour idéal). Rampe : le contour est plus flou.



Toit : il s'agit d'une ligne sur un fond uniforme.

Fig 2.1 : quelques modèles de contours.

L'extraction de contours joue un rôle primordial dans tout système de vision par ordinateur. Beaucoup d'efforts ont été faits pour extraire les contours d'une image et plusieurs techniques ont été proposées à ce jour. Les contours correspondent généralement à des changements brusques de propriétés physiques ou géométriques de l'image perçue et forment ainsi des attributs très importants pour l'analyse.

Dans la plupart des techniques développées à ce jour, plusieurs opérateurs locaux (directionnels ou pas) de dérivation du premier ordre et du second ordre sont utilisés. Ils sont alors suivis respectivement d'une recherche de maxima locaux et de passage par zéro. Toutefois, ces opérateurs ne donnent souvent pas, à eux seuls, de bons résultats sur une image réelle où les changements d'intensité sont rarement nets et abrupts ajout à ça le bruit. Une opération de seuillage et un suivi de contours sont en effet nécessaires pour éliminer le bruit et affiner les contours à une épaisseur d'un pixel.

2.1.1. METHODES SURFACIQUE:

L'image $A(x,y)$ est considéré comme une surface, alors il est possible de définir un modèle surfacique de transition sur un voisinage donné, un contour sera détecté en point $B(x,y)$ si y a une bonne corrélation entre le modèle et la fenêtre de l'image centrée en ce point. Citant l'opérateur de Heuckel considère le modèle comme une transition représenté par un modèle bidimensionnel. Et Harilick qui approche l'image en série de polynômes.

2.1.2. METHODE MORPHOLOGIQUE :

Dans cette approche, on trouve le gradient morphologique et ligne de partage des eaux où l'image est considérée comme une surface dans l'espace 3D. la ligne de partage des eaux est la crête qui forme la limite entre deux bassins versants, là nous citons juste le principe sur lequel est basé la méthode.

2.1.3. METHODES DERIVATIVE :

En général l'extraction des contours par les méthodes dérivatives nécessite quatre étapes voir (fig.2.2):

- 1) Lissage : suppression du bruit, conditionnement du signal.
- 2) Calcul de gradient magnitude et orientation.
- 3) Détection des maxima locaux avec seuillage par hystérésis.
- 4) Chaînage des points contours.

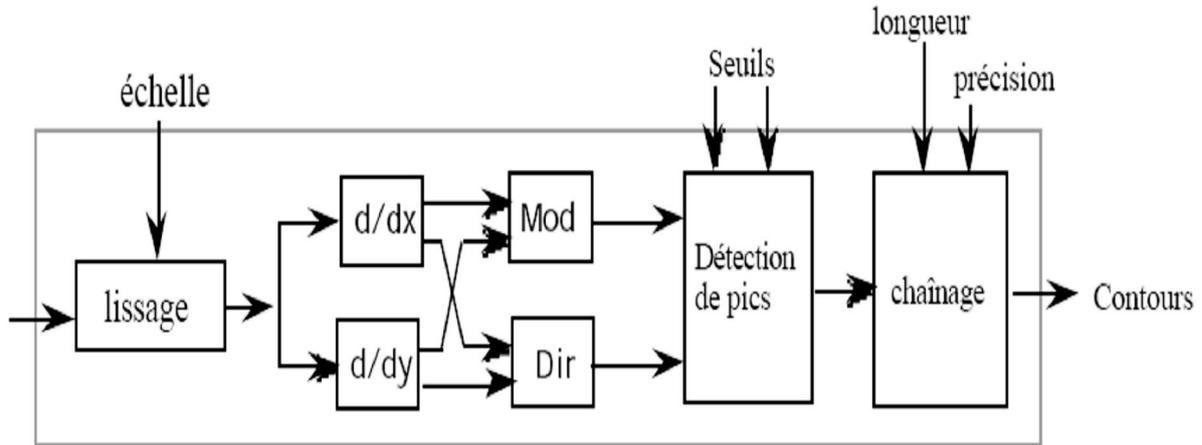


Fig.2.2 : les étapes d'extraction des contours

3. FILTRAGE ET LISAGE DE BRUIT :

3.1. FILTRAGE LINEAIRE D'UNE IMAGE

Rappelons rapidement que filtrer signifie convoluer une image $I(x,y)$ avec une fonction $f(x,y)$ qui s'appelle réponse impulsionnelle du filtre. Dans le cas continu l'image filtrée est donnée par : [HOR95],

$$I'(x, y) = f(x, y) * I(x, y) \tag{2.1}$$

$$I'(x; y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(u, v) I(x - u, y - v) du \cdot dv \tag{2.1.a}$$

$$I'(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x - u, y - v) I(u, v) du \cdot dv$$

Dans le cas discret les domaines de I et de f sont bornes. Le domaine de I est $(-N/2, N/2)$, et le domaine de f est $(-K/2, K/2)$ On a nécessairement $K \leq N$, tel N est la taille de l'image dans le cas où elle soit carre. Dans le cas discret la convolution s'écrit :

$$I'(x, y) = \sum_{u=-K/2}^{+K/2} \sum_{v=-K/2}^{+K/2} f(u, v) I(x - u, y - v) \tag{2.2}$$

K correspond à la taille de filtre

On notera que le filtrage linéaire consiste simplement a remplacer chaque niveau de gris par une combinaison linéaire des niveaux de gris des points voisins les coefficients de cette combinaison linéaire sont définis par la réponse impulsionnelle du filtre.

Note : plusieurs filtres existent et le choix n'est pas évident surtout si on n'a pas de connaissance a priori sur le type de bruit. et l'estimation de la nature de bruit [Ima97]. Les filtres linéaires les plus connus sont le filtre moyen et le filtre gaussien.

3.1.1. FILTRE MOYEN :

Le lissage par le filtre moyenner consiste a effectuer une moyenne sur les pixels voisines, en général ce filtre est donne sous la forme suivante :

$$M = \frac{1}{(2+n)^2} \begin{bmatrix} 1 & n & 1 \\ n & n^2 & n \\ 1 & n & 1 \end{bmatrix} \tag{2.3}$$

n représente le poids que on pu donner au pixel central

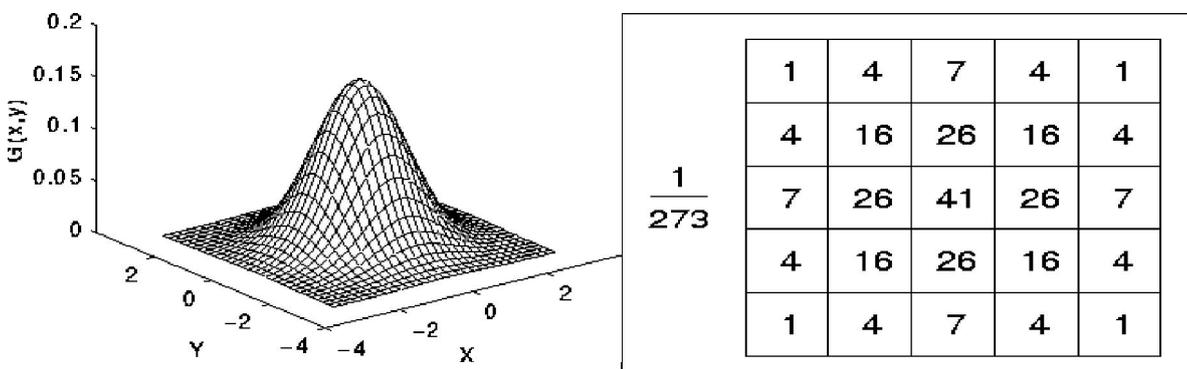
$\frac{1}{(2+n)^2}$: Facteur de normalisation.

3.1.2. Filtre Gaussien :

Le filtre gaussien est donné en 2D sous forme :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.4}$$

Pour filtrer une image par le filtre gaussien il faut faire une approximation de la fonction gaussienne ensuite faire une convolution entre celle-ci et l'image.(voir fig2.3)



La distribution de la fonction gaussienne

une approximation pour $\sigma=1$

Fig.2.3: filtre gaussien

Remarque : les filtre linéaire ont tendance a laisser les transitions donnant une impression de flou sur les bords des objets et a rendre délicat l'extraction et la localisation des contours.

3.2. Filtres non linéaires :

Plusieurs filtres existent, filtre médiane et le plus proche voisin. Nous citons sur le passage le filtre médiane.

3.2.1 Filtre médiane:

Le filtre médiane réalise un lissage un plus performant surtout dans le cas d'un bruit impulsionnel que le filtre moyenne.

Chaque pixel est traité en considérant ses voisins sur un voisinage donné. Le pixel lui même et ses voisins forme alors un ensemble dont on calcule la *valeur médiane*, le pixel sera alors remplacé par cette celle-ci.

4. Différents approche dérivative :

Les approches dérivatives se divisent en deux classes :

4.1.les approches gradients : détermination des extrema locaux dans la direction du gradient tel que filtres de Sobel, Prewitt, Canny, Deriche í ..

Le gradient d'une image est le vecteur $\nabla I(x, y) = \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)$ 2.5

Il est caractérisé par le module Mod et sa direction Dir, tel que :

$$Mod = \sqrt{\left(\left(\frac{\partial I(x, y)}{\partial x} \right)^2 + \left(\frac{\partial I(x, y)}{\partial y} \right)^2 \right)} \quad \text{ou} \quad Mod = \max \left\{ \left| \frac{\partial I(x, y)}{\partial x} \right|, \left| \frac{\partial I(x, y)}{\partial y} \right| \right\} \quad 2.6$$

$$Dir = \phi = \arctan \left(\frac{\frac{\partial I(x, y)}{\partial y}}{\frac{\partial I(x, y)}{\partial x}} \right) \quad 2.7$$

La direction du gradient maximise la dérivée directionnelle.

4.2. Les approches Laplaciennes :

Permettent la détermination des passages par zéro du Laplacien. Le Laplacien est utilisé à la place de la dérivée seconde [Coc95] dans le souci d'alléger les calculs, car on évite le calcul des dérivées directionnelles suivant la direction de gradient.

$$\text{Laplacien de } I(x,y) \text{ s'écrit : } \nabla^2 I(x,y) = \frac{\partial^2 I(x,y)}{\partial \cdot x^2} + \frac{\partial^2 I(x,y)}{\partial \cdot y^2} \quad 2.8$$

5. application des approches dérivative aux images numériques :

La dérivation d'un signal échantillonné peut être considérée comme équivalente à l'application d'un filtre dérivée gaussien au signal original suivie d'un échantillonnage [Cocquerez 95]. Notons que ce problème ne se pose pas dans notre cas car l'image est à l'origine numérique elle nous parvient d'une caméra CCD.

5.1. Operateurs Dérivatifs de premier ordre :

Dans le cas discret, les dérivées directionnelles suivant les directions horizontales et verticales au site (i, j) sont approchées par simple différences finies.

$$\frac{\partial I(x,y)}{\partial x} = \nabla_x I(x,y) \approx \nabla_i I(x,y) = I(i+1, j) - I(i, j) \quad 2.9$$

$$\frac{\partial I(x,y)}{\partial y} = \nabla_y I(x,y) \approx \nabla_j I(x,y) = I(i, j+1) - I(i, j) \quad 2.10$$

5.1.1. Operateur de Prewitt et Sobel :

Considérons le repère de l'image décrit par la figure 2.4:

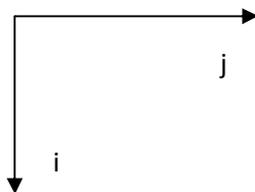


Fig.2.4: Orientation des repères dans l'image numérique.

Les dérivées directionnelles sont obtenues par convolution de l'image avec les filtres suivants :

$$h_j = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \quad h_i = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix}$$

si $c=1$ alors on a le filtre (masque) de Prewitt ,

si $c=2$ on a le masque de Sobel.

On remarque que ces deux filtres sont séparables comprenant un lissage suivant une direction et une dérivation suivant une autre.

$$h_j = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} = h_{Li} \cdot h_{Dj} = {}^t [1, c, 1] [1, c, -1] \text{ tel que}$$

h_{Li} : Lissage suivant la verticale, et h_{Dj} : dérivation suivant la direction horizontale.

5.1.2. Operateur MDIF :(Filtre Moyeneur Dérivateur)

Le MDIF est la combinaison d'un filtre moyeneur de noyau m et d'un dérivateur utilisant les masques directionnel de Prewitt. [Coquerez 95]

$$m = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad h_j = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad h_i = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$m_j = m \cdot h_j \quad \text{et} \quad m_i = m \cdot h_i$$

5.1.3. OPERATEUR NAGDIF :

Le NAGDIF est la combinaison d'un lissage non-linéaire de Nagao [Nag79], suivi d'une dérivation effectuée par l'opérateur suivant : [coc95]

$$\text{Contour existe si } \max \left\{ |I(i, j) - I(i - z_1, j - z_2)| \right\} > \text{seuil}$$

Z_1 et Z_2 sont des valeurs entières dans l'intervalle $[-1, 1]$.

5.2. OPERATEUR DEUXIEME ORDRE :

Les contours peuvent être localisés par les passages par zéro du Laplacien

$$\nabla^2 I(x,y) = \frac{\partial^2 I(x,y)}{\partial \cdot x^2} + \frac{\partial^2 I(x,y)}{\partial \cdot y^2} \quad 2.11$$

L'approximation discrète la plus simple calculée au voisinage 3*3 est le masque suivant :

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

5.2.1. Operateurs de Marr et Hildreth :

L'image $I(i, j)$ est laissée par filtre gaussien ,en suite on calcule la dérive seconde dans la direction du gradient, pour des raison de calcul et simplification on utilise le Laplacian ou lieu de la dérivée seconde. Puisque le filtre gaussien est linéaire, cela revient a appliquer directement l'opérateur Laplacian d'une gaussien.

Huertas et Médioni ont proposé en clôture un filtre, l'opérateur Laplacian d'une gaussien sous forme de deux filtres séparables : [Hue86]

$$\Delta G(x, y) = g_1(x)g_2(y) + g_2(x)g_1(y) \quad 2.12$$

$$g_1(x) = \frac{1}{\sqrt{2G_0}} \left(1 - \frac{x^2}{\sigma^2} \right) e^{-\frac{x^2}{2\sigma^2}} \quad \text{Dérivée seconde d'une gaussienne}$$

$$g_2(x) = \frac{1}{\sqrt{2G_0}} e^{-\frac{x^2}{2\sigma^2}} \quad \text{lissage}$$

Sous forme numérique. Pour $\sigma = \sqrt{2}$ et $1/G_0 = 4232$ qui un facteur de normalisation

On a : $g_1 = (-1 \ -6 \ -17 \ -17 \ 18 \ 46 \ 18 \ -17 \ -17 \ -6 \ -1)$.

$$g_2 = (0 \ 1 \ 5 \ 17 \ 36 \ 46 \ 36 \ 17 \ 5 \ 1 \ 0).$$

6. Approche par filtres optimaux :

Dans le cas réel, les images ne sont pas en bons qualités et des contours sont bruités. Cependant qu'il y a des inconvénients dans l'approche dérivative, on trouve que les opérateurs de dérivation du premier ordre exagèrent les effets du bruit. Les deuxièmes dérivés va exagérer le bruit deux fois plus. Il y a une autre approche qui propose des filtres optimaux pour détecter et localiser le contour.

6.1. Approche de Canny

L'approche de Canny est basée sur trois critères [Can86] :

- ✓ La bonne détection.
- ✓ Bonne localisation.
- ✓ Réponse unique.

Canny a proposé un modèle de contour pour l'application de ces trois(03) critères, il a considéré un contour de type marche (cas idéal) en mono dimensionnel $U(x)$ perturbé par un bruit gaussien $N(x)$ (fig.2.5)

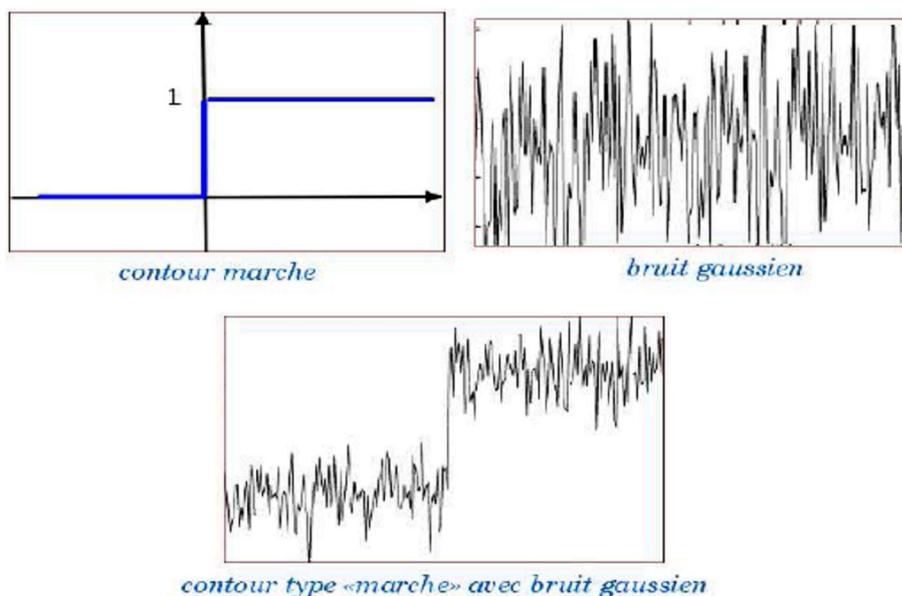


fig.2.5 : contour idéal perturbé par un bruit gaussien

$$A(x) = U_0 U(x) + N(x)$$

2.13

Tel que U_0 est une constante.

$$\text{et } U(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{autrement} \end{cases}$$

Le signal de sortie est donné par l'expression :

$$C(x) = A * h(x) = \int_{-\infty}^{+\infty} A(t)h(x-t)dt \tag{2.14}$$

La Bonne détection :

Une bonne détection revient à trouver un meilleur rapport signal sur bruit RSB a la sortie de filtre h(t).ce critère est formalise comme rapport de la réponse due au signal seul sur la racine carre de la puissance du bruit en sortie.

$$RSB = \frac{U_0 \int_0^{+\infty} h(x-t)dt}{\left\langle \left| \int_{-\infty}^{+\infty} N(t)h(x-t)dt \right|^2 \right\rangle^{1/2}} = \frac{U_0 \int_0^{+\infty} h(x-t)dt}{N_0 \left\langle \int_{-\infty}^{+\infty} h^2(t)dt \right\rangle^{1/2}} = \frac{U_0}{N_0} \Sigma \tag{2.15}$$

Tel que $\Sigma = \frac{\int_0^{+\infty} h(x-t)dt}{\left\langle \int_{-\infty}^{+\infty} h^2(t)dt \right\rangle^{1/2}}$

Bonne localisation :

C'est avoir le maximum de la réponse a la position x_0 la plus proche de la position réelle de la transition ($x=0$), ce critère est mesuré par l'inverse de la variance $E(x_0^2)$, x_0 étant la position calculée de la transition, qui correspond au maximum du C(x).

Vers la fin Canny a aboutit à la formule suivant :

$$\frac{U_0}{N_0} \nabla = \left\langle \frac{U_0^2 h'^2(0)}{N_0^2 \int_{-\infty}^{+\infty} h'^2(t)dt} \right\rangle^{1/2} = \frac{U_0}{N_0^2} \frac{|h'(0)|}{\left\langle \int_{-\infty}^{+\infty} h'^2(t)dt \right\rangle^{1/2}} \tag{2.16}$$

Le produit $\Sigma \nabla$ est un critère qui combine la bonne détection et la bonne localisation, il ne dépend pas de l'amplitude U_0 de l'échelon, ni de facteur d'échelle [coquerez95].

NON MULTIPLICITE DE REPONSES :

Ce critère correspond à la limitation du nombre de maxima locaux détectés en réponse à un seul contour. La distance moyenne entre deux maxima locaux, notée

$$x_{\max} = 2\pi \left(\frac{\int_{-\infty}^{+\infty} h'^2(t) dt}{\int_{-\infty}^{+\infty} h''^2(t) dt} \right)^{\frac{1}{2}} = k \quad 2.17$$

Trouver $h(t)$ qui maximise le produit $\sum \nabla$ sous la contrainte que le troisième critère soit fixé à une constante k a abouti à l'équation différentielle suivante :

$$2h''^2(t) - 2\lambda_1 h''(t) + 2\lambda_2 h''''(t) + \lambda_3 = 0 \quad 2.18$$

Qui admet comme solution générale :

$$h(t) = a_1 e^{\alpha \cdot t} \cos(\omega \cdot t) + a_2 e^{\alpha \cdot t} \sin(\omega \cdot t) + a_3 e^{-\alpha \cdot t} \cos(\omega \cdot t) + a_4 e^{-\alpha \cdot t} \sin(\omega \cdot t) \quad 2.19$$

Pour trouver le filtre $h(t)$ sous forme d'un filtre à réponse impulsionnelle finie (RIF) défini sur l'intervalle $[M, +M]$ et présentant une pente S à l'origine Canny a imposé les conditions aux limites suivantes $h(0)=0$, $h(M)=0$, $h'(0)=S$, $h'(M)=0$.

Ces 4 conditions permettent de trouver les coefficients a_1 à a_4 , $h(t)$ étant- k , impaire la solution est étendue aux t négatifs avec $h(t) = -h(-t)$.

Utilisant une optimisation numérique sous contrainte, Canny trouva que l'opérateur le plus performant correspond à $\sum \nabla = 1,12$.

Pour des raisons de mise en œuvre il approxima cet opérateur à la première dérivée d'une gaussienne qui présente une dégradation d'environ 20%.

6.2 Filtres de Shen-Castan

Le filtre de Shen et Castan est obtenu par optimisation d'un critère incluant la détection et la localisation. La solution obtenue est assez proche de celle de Canny :

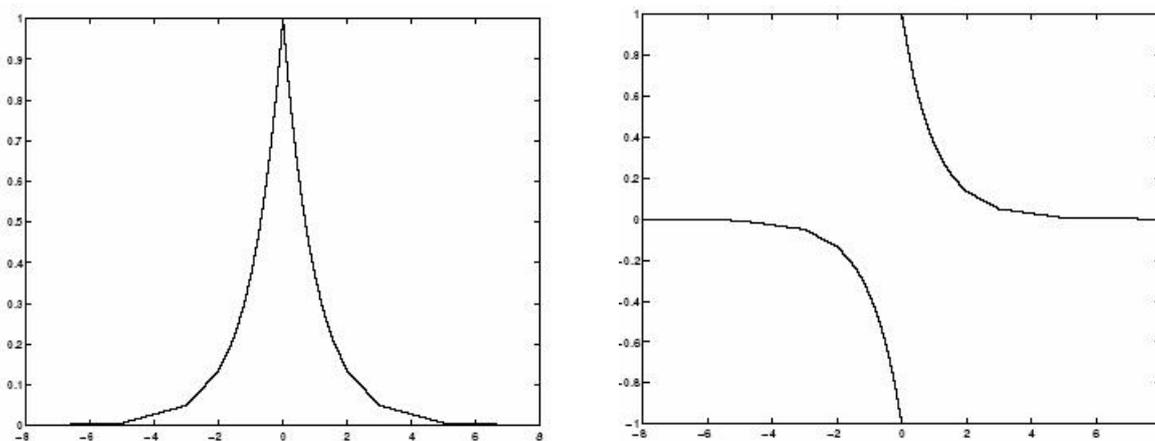
$$f(x) = C e^{(-\alpha |x|)} \quad 2.20$$

➤ C'est choisi de façon à normaliser le filtre : $C = \frac{\alpha}{2}$ 2.21

➤ le filtre de dérivation s'écrit : $f'(x) = \begin{cases} \frac{\alpha^2}{2} e^{-\alpha x} & \text{si } x > 0 \\ -\frac{\alpha^2}{2} e^{-\alpha x} & \text{si } x < 0 \end{cases}$ 2.22

pour $x=0$ la dérivée n'existe pas

Le paramètre α détermine la largeur du filtre, plus α est petit plus le lissage est important (perte en localisation). la figure 2.6 illustre le filtre de Shen-Castan [Shen85]



Réponse impulsionnelle du filtre.

Réponse impulsionnelle du filtre de dérivée.

Figure 2.6 : Filtre de Shen-Castan

6.2.1 Implémentation des opérateurs bidimensionnels de Shen-Castan.

6.2.1.1. Lissage.

$f(x)$ est séparable en $f_1(x)$ et $f_2(x)$ tel que $f(x) = f_1 * f_2(x)$ * : représente le produit de convolution

$$f_1(x) = \begin{cases} \alpha e^{-\alpha x} & \text{si } x > 0 \\ 0 & \text{si } x < 0 \end{cases} \quad 2.23.a$$

$$f_2(x) = \begin{cases} 0 & \text{si } x > 0 \\ \alpha e^{\alpha x} & \text{si } x < 0 \end{cases} \quad 2.23.b$$

Les équations aux différences sont données comme suite :

$$B(i)=[A*f_1(i)]*f_2(i) \quad 2.24$$

$$B_1(i) = e^{-\alpha x} [B_1(i-1) - A(i)] + A(i) \text{ pour } i=1 \dots m$$

$$B(i) = e^{-\alpha x} [B(i-1) - B_1(i)] + B_1(i) \text{ pour } i=m \dots 1$$
2.24.a

Soit une image discrète $A(i,j)$. L'image lissée $B(i,j)$ est donnée comme suite :

$$B(i, j) = [A * f(i)] * f(j) = [[A * f_1(i)] * f_2(i)] * f_1(j) * f_2(j) \quad 2.25$$

Dérivation :

Les équations aux différences sont :

$$B_1(i) = e^{-\alpha x} [B_1(i-1) - A(i)] + A(i) \quad \text{pour } i=1 \dots m \quad 2.26$$

$$B_2(i) = e^{-\alpha x} [B_2(i-1) - A(i)] + A(i) \quad \text{pour } i=m \dots 1$$

$$B(i) = B_1(i) - B_2(i)$$

Soit une image discrète $A(i,j)$. L'image dérivée $B(i,j)$ est donnée comme suite :

$$\text{Selon la direction } i : B_i(i, j) = [A * f_2(i) - A * f_1(i)] * f_2(j) * f_1(j) \quad 2.27.a$$

$$\text{Selon la direction } j : B_j(i, j) = [A * f_2(j) - A * f_1(j)] * f_2(i) * f_1(i) \quad 2.27.b$$

Remarque : les constantes de normalisations ne le sont pas les mêmes quand on passe de domaine continu au domaine discret.

6.3 Filtre de Deriche :

En utilisant la même démarche que celle de Canny [Der 87], Deriche a changé les conditions aux limites

$$h(0) = 0, \quad h(+\infty) = 0, \quad h'(0) = S, \quad h'(-\infty) = 0$$

Ce qui a lui permet d'aboutir à un filtre à réponse impulsionnelle infinie (RII).

$$h(t) = \frac{S}{w} e^{-\alpha|t|} \cdot \sin(wt) \quad 2.28$$

En faisant tend w vers zéro on aura l'équation suivante :

$$h(t) = S t e^{-\alpha|t|} \quad 2.28.a$$

Pour normalisée ce filtre, on peut ajoute une constante b tel que $C = k.S$, $h(t)$ devient alors :

$$h(t) = C t e^{-\alpha|t|} \quad 2.28.b$$

C est constante de normalisation calculée pour avoir une réponse égale à 1 pour un échelon unité $U(x)$, on obtient alors :

$$h(t) = \int_{-\infty}^0 C t e^{-\alpha t} = 1 \quad \Rightarrow C = -\alpha^2 \quad 2.28.c$$

6.3.1 Utilisation de filtre de Deriche :

Afin d'appliquer ce filtre aux images, et améliorer l'immunité au bruit un lissage préalable est effectué.

❖ LISSAGE :

Deriche propose pour cette opération l'utilisation de deux filtres monodimensionnels dans les directions X et Y. le filtre $f(t)$ choisi n'est que l'intégral de $h(t)$.

$$f(t) = B(\alpha|t|+1) e^{-\alpha|t|} \quad 29$$

B est une constante de normalisation calculée pour avoir une réponse indicielle égale à 1. On obtient

$$B = \frac{\alpha}{4}.$$

Extension en 2D :

L'expression de $f(x)$ en 2D est donne par :

$$f(x, y) = B^2(\alpha|x|+1) e^{-\alpha|x|}(\alpha|y|+1) e^{-\alpha|y|} \quad 30$$

Si $A(x,y)$ est l'image originale, l'image lissée aura l'expression suivante :

$$L(x,y) = A * f(x,y) \quad 31$$

6.3.2. Calcul du gradient :

Le calcul se fait a partir des dérivées selon x et y de produit de convolution de l'image par le filtre de lissage.

$$\text{On a donc : } \frac{\partial L(x, y)}{\partial x} = L_x(x, y) = \frac{\partial(A * f)}{\partial x}(x, y) = A * \frac{\partial f}{\partial x}(x, y) \quad 32$$

Les operateurs de dérivation suivant x et y sont donnés comme suite :

$$f_x(x, y) = \eta x e^{-\alpha|x|} \cdot (\alpha|y| + 1) e^{-\alpha|y|} \quad 2.33$$

$$f_y(x, y) = \eta y e^{-\alpha|y|} \cdot (\alpha|x| + 1) e^{-\alpha|x|}$$

η est un facteur de normalisation, il vaut $\frac{\alpha^3}{4}$

Après le développement des relations ci dessus on trouve :

$$f_x(x, y) = h(x) \cdot f(y) \quad \text{et} \quad f_y(x, y) = h(y) \cdot f(x). \quad 2.34$$

d'après ces équations la dérivée directionnelle selon x (respectivement selon y) est le résultat d'un lissage dans la direction y (respectivement direction x), suivi d'une dérivation selon x (respectivement selon y) .

6.3.3 Discrétisation de filtre de Dérivée :

Une mise en œuvre récursive de filtre dérivée $h(t)$ optimal de dérivation est obtenu en employant transformée en Z. [DER 90].

$$\text{on } h(t) = h(t)^+ + h(t)^- .$$

l'image résultat $B[i]$ est sous la forme suivant:

$$B_1(i) = C e^{-\alpha} A(n-1) + 2 e^{-2\alpha} B_1(i-1) - e^{-2\alpha} B_1(i-2) \quad \text{pour } i=1 \dots m \quad 2.35$$

$$B_2(n) = -C e^{-\alpha} A(i+1) + 2 e^{-\alpha} B_2(i+1) - e^{-2\alpha} B_2(i+2) \quad \text{pour } i=m \dots 1$$

$$B(i) = B_1(i) + B_2(i)$$

$$C = \frac{(1 - e^{-\alpha})^2}{e^{-\alpha}}$$

m : est la longueur de la séquence $A[i]$. C la constante de normalisation dans le cas discret.

6.3.4 Discrétisation de filtre de lissage :

On adopte la même démarche, nous arrivons à une implémentation de récursive :

$$f(t) = b(\alpha|t|+1)e^{-\alpha|t|} \quad 2.36$$

Les équations aux différences ont pour expression :

$$B_1(i) = bA(i) + b e^{-\alpha} (\alpha - 1)A(i-1) + 2 e^{-\alpha} B_1(i-1) - e^{-2\alpha} B_1(i-2) \quad 2.36.a$$

pour $i=1 \dots m$

$$B_2(i) = b e^{-\alpha} (\alpha - 1)A(i+1) - b e^{-2\alpha} A(i+2) + 2 e^{-\alpha} B_2(i+1) - e^{-2\alpha} B_2(i+2)$$

pour $i=m \dots 1$

$$B(n) = B_1(i) + B_2(i)$$

$$b = \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}}$$

7. Application aux images :

Soit $B_j(i,j)$ la dérivée directionnelle suivant la direction j est calculer en appliquant la relation suivante :

$B_j(i,j) = [A * f(i)] * h(j)$ on a un lissage suivant l'axe i suivi d'une dérivation suivant l'axe j .

* produit de convolution

De même pour $B_i(i,j)$

$B_i(i,j) = [A * f(j)] * h(i)$ lissage suivant l'axe j et dérivation suivant l'axe i .

7.1 Suppression des non maxima locaux :

Cette étape consiste à déterminer si le point $x(i,j)$ est un maxima en le comparant à ces voisins de l'un et de l'autre côté de contour.

Un point $x(i,j)$ est maximum local si il remplit les condition suivante :

$$G(i, j) \geq G(i + \cos\theta, j + \sin\theta) \text{ et } G(i, j) \geq G(i - \cos\theta, j - \sin\theta)$$

$G(i, j)$ Gradient pour un pixel sur la ligne i et la colonne j

θ Direction du Gradient

La tâche à effectuer consiste donc à estimer la valeur du gradient pour les points $(i + \cos\theta, j + \sin\theta)$ et $(i - \cos\theta, j - \sin\theta)$ et à la comparer avec le gradient pour le point (i,j) . Si les conditions définies précédemment ne sont pas remplies, alors ce non maxima est supprimé. Comme les points

$(i + \cos\theta, j + \sin\theta)$ et $(i - \cos\theta, j - \sin\theta)$ sont situés sur un cercle autour du point i , A défaut d'une complexité, on calcule à leurs places les gradients aux points A et B. (Voir fig2.7).

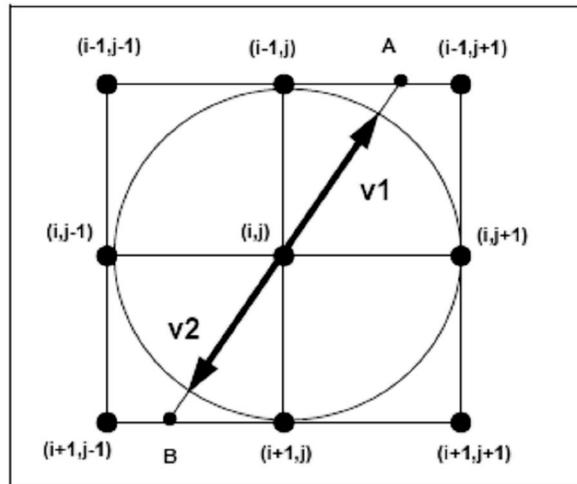


Fig.2.7 : Interpolation dans la direction du gradient

Afin que calcul de l'angle θ ne soit pas pénalisant. L'information orientation sera prise directement de rapport R des gradients directionnels $G_x(i,j)$ et $G_y(i,j)$.

7.1.1. Etape de l'algorithme :

Pour tout point de l'image calcule le module de gradient :

$$M(i, j) = \sqrt{G_x(i, j)^2 + G_y(i, j)^2} \tag{2.37.a}$$

Le rapport des gradients $R(i, j) = \frac{G_y(i, j)}{G_x(i, j)}$ 2.37.b

Si $G_x(i,j) > 0$ alors le point A(i,j) sera déclaré comme maxima local si $M(i,j) > U$ et $M(i,j) > Z$

Avec U et Z sont données par :

Si $R > 1$ $U = [(R-1)M(i,j+1) + M(i+1,j+1)]/R$ $Z = [(R-1)M(i,j-1) + M(i-1,j-1)]/R$

Si $1 > R > 0$ $U = (1-R)M(i+1,j) + RM(i+1,j+1)$ $Z = (1-R)M(i-1,j) + RM(i-1,j-1)$

Si $-1 < R < 0$ $U = [(R+1)M(i,j-1) - M(i+1,j-1)]/R$ $Z = [(R+1)M(i,j+1) - M(i-1,j+1)]/R$

Si $M_x(i,j) < 0$ et $M_x(j,i) = 0$ $U = M(i,j-1)$ $Z = M(i,j+1)$

Si $M_x(i,j) > 0$ et $M_x(j,i) = 0$ $U = M(i,j+1)$ $Z = M(i,j-1)$

7.2 Seuillage par hystérésis

Ce seuillage présenté par Canny est basé sur l'utilisation de deux (02) seuils. La procédure peut être décrite de deux façons différentes :

1. Seuiller toute l'image des maxima locaux par le seuil bas
2. Ne valider que les ensembles connexes de point présentant au moins un point dont la norme de gradient est supérieur au seuil haut.

Ou :

1. Seuiller toute l'image des maximas par le seuil haut.
2. Valider tous les pixels qui sont connexes aux points détectés et qui possèdent une norme du gradient supérieur au seuil bas.

7.3 Affinage des contours

L'hystérésis fournit en sortie une image contenant les contours sous forme de chaînes.

Puisqu'un contour doit être une chaîne de pixels d'épaisseur (1), cette restriction étant imposée pour des raisons de formalisation mathématique, il est nécessaire d'affiner les chaînes de pixels

Les chaînes de pixels détectées doivent subir un (amincissement) de l'épaisseur. Les règles d'affinage et l'effet de celui-ci est illustré dans la figure 2.8. On peut répartir ces règles en trois catégories :

ÉCelle qui consiste à éliminer des pixels non-souhaités d'un contour

ÉCelle qui ajoute un pixel

ÉCelle qui déplace un pixel

Si un pixel est isolé et n'a pas de voisin direct, ce point est éliminé dans l'image de contour. Pour les cas, où une ligne est interrompue (le pixel n'a qu'un seul voisin) on ajoute un pixel pour fermer le trou du contour.

La méthode qu'on applique aux pixels qui ont deux voisins est de déplacer le pixel pour fermer un trou ou d'éliminer celui-ci s'il est situé à proximité d'une diagonale.

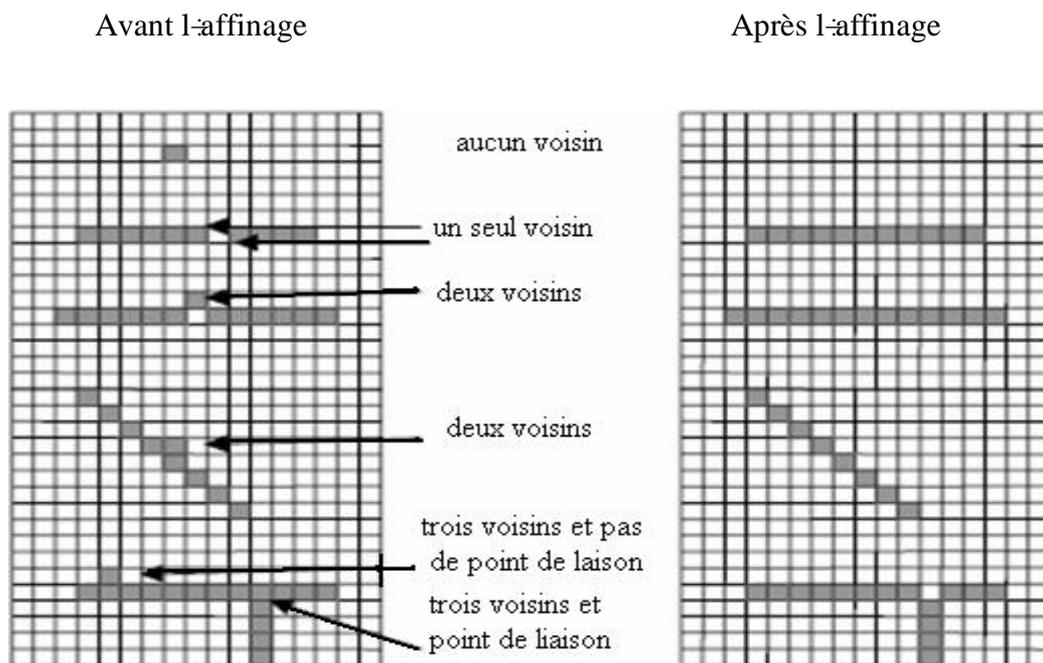


Fig.2.8 Principe d'affinage des contours.

Quant aux points qui ont trois voisins, ils sont éliminés des contours.

La fonction qui permet de faire l'affinage prend en entrée un objet « Image » avec les contours et fournit en sortie le même objet ayant que des contours avec une épaisseur de (1).

Pour construire une représentation des contours affinés sous forme de poly lignes le prochain pas consiste à réaliser un chaînage.

7.4 Chaînage des contours

Comme il est déjà ressorti dans le paragraphe 3.4.1 l'objectif du chaînage consiste d'une part à réduire et compresser les données des contours en appliquant un codage spécial, et d'autre part à faciliter le traitement ultérieur des contours.

La technique de codage est une opération usuelle qui suit le principe décrit dans la figure 2.9. Ainsi, il suffit d'enregistrer les coordonnées pixel de départ d'un contour, de suivre le contour et de coder la direction de la chaîne pour chaque pixel jusqu'à ce qu'on atteigne la fin de la chaîne. Pour un voisinage de 8 pixels, 8 directions sont possibles, ce qui signifie que 8 valeurs (ou 3 Bits) suffisent pour contenir cette information. Cela montre clairement qu'on profite d'une compression considérable des données. Au lieu de stocker 2 coordonnées (i,j) pour chaque pixel d'un contour on n'enregistre que la direction d'un pixel au pixel suivant.

Cette information ne consomme que 3 Bit par pixel au lieu des 8 Bits pour une coordonnée dans l'image.

La fonction permettant de faire le chaînage se sert de l'image contenant des contours affinés pour fournir en sortie des vecteurs en mémoire, chacun contenant la position de départ d'un contour et le codage des directions

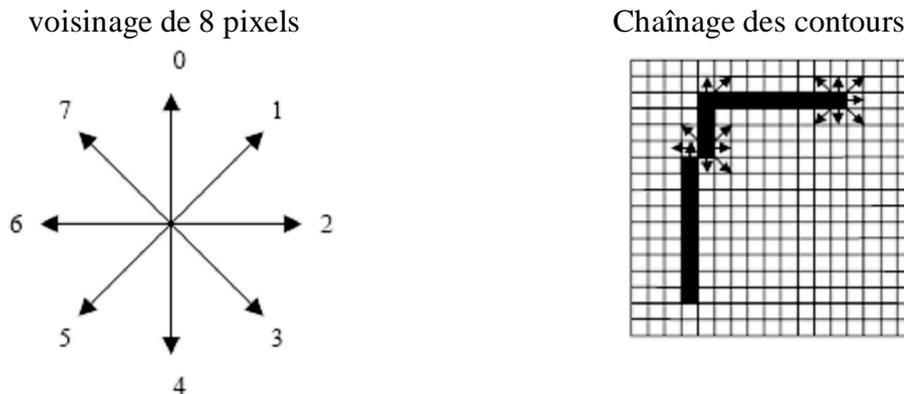


Fig.2.9. Principe de chaînage.

Codage des directions pour le contour commencent avec le point bas gauche:

Écriture décimale: (0 0 0 0 0 0 0 1 0 0 0 2 2 2 2 2 2 2 2)

Le chaînage accompli, on dispose d'une représentation des contours d'une épaisseur (1) sous forme de chaînes (vecteur).

Nous montrons quelques résultats de l'application de filtre de Deriche sur l'image bureau sur les figure fig2.10, fig2.11 et fig2.12

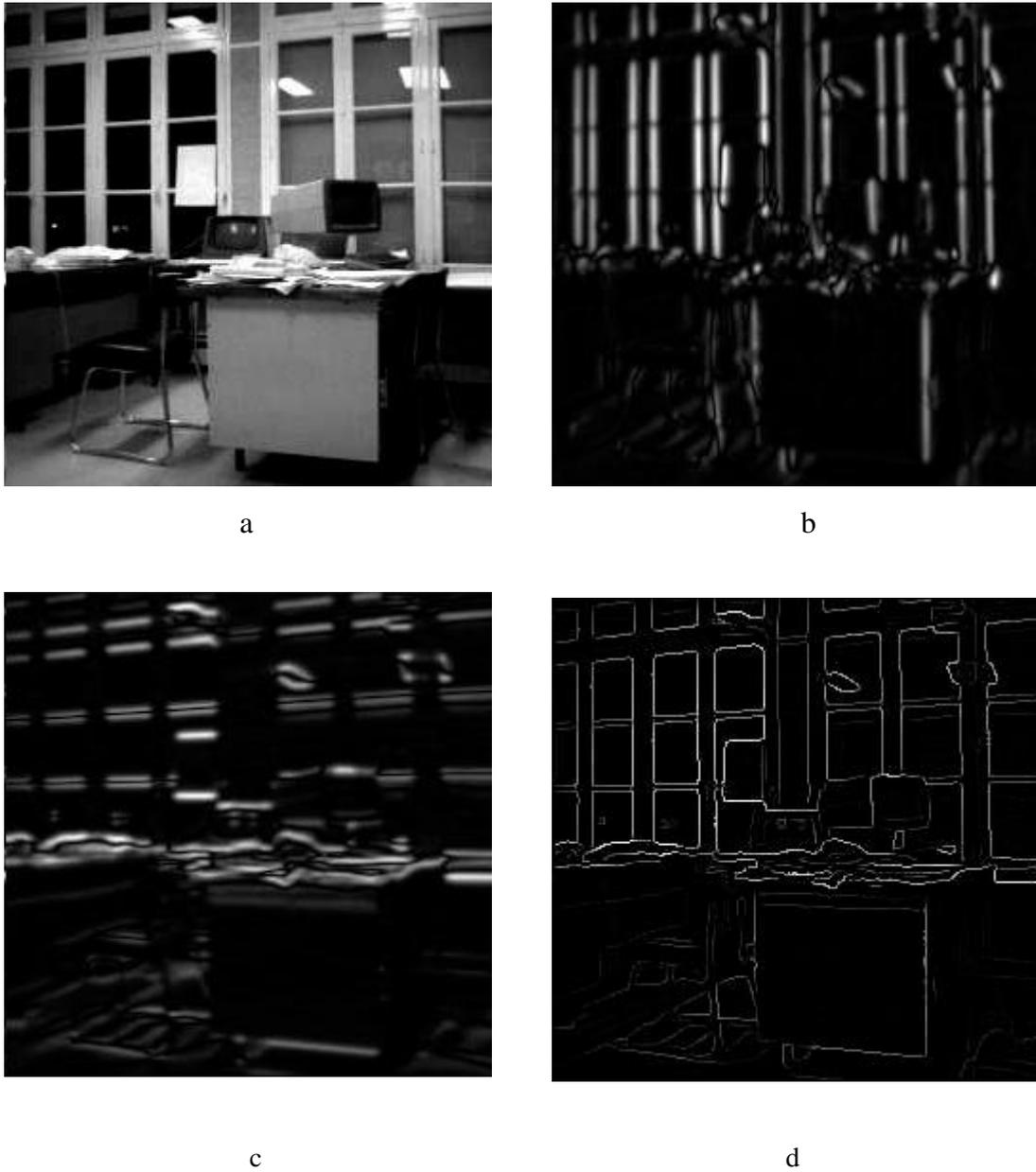


Fig.2.10. (a) : image originale, (b) : gradient en x (filtre de Deriche), (c) gradient en y (filtre de Deriche), (d) extrema locaux de la norme du gradient dans la direction du gradient.



Fig2.11. : image maxima locaux



Fig.2.12. : Contours obtenus après seuillage par hystéresis.

8. Discussion :

Dans ce chapitre nous avons abordé quelques principes de la segmentation de l'image par l'approche frontière. Ainsi on a présenté quelques filtres classiques, qui donnent de bons résultats dans le cas des images non bruitées, l'accent est mis sur le filtre de Deriche qui présente plusieurs avantages vu les principes sur lesquels a été faite sa conception, à savoir résistance au bruit, bonne localisation et limitation des reponses due au bruit. Ajoutant à tout ça l'avantage d'une implémentation complètement récursive, chose qui est très intéressante de point de vue vitesse d'exécution surtout dans des applications temps réel. Dans le chapitre Application nous allons voir les possibilités de parallélisme.

Chapitre 3

La Transformée de Hough et son application

1. Introduction

La navigation autonome des robots mobiles utilisant les images comme entrées à pousser les chercheurs à s'intéresser aux outils qui permettent d'extraire les informations nécessaires à la navigation, reconnaissance des objets, la surveillance.

La réussite de ces tâches dépend essentiellement de la compréhension des informations issues des capteurs visuels, qui doivent se présenter après traitement sous forme de données utilisables par les composants décisionnels de haut niveau. Ces données peuvent être des segments de droite, des courbes, des formes élémentaires ou des régions.

Dans notre cas on utilisera des segments de droite, plusieurs méthodes existent afin de les extraire et elles sont classées en deux catégories.

2. Classification des méthodes

La localisation d'une droite dans une image est un problème classique dans le domaine du traitement de l'image. En général, il y a deux contraintes posées sur une méthode dont le but est la détection d'une droite: le temps de calcul et la stabilité de la méthode, notamment dans le cas des images fortement bruitées. Plusieurs techniques ont été déjà mises en œuvre. On peut les classer en deux groupes principaux : les méthodes locales, et les méthodes globales.

2.1. Méthodes locales : Elles testent l'appartenance d'un pixel à une droite sur un voisinage local du pixel. On peut les diviser en deux sous groupes :

2.1.1. Détection de droite à partir d'une image de gradient

L'image d'entrée est convolutive avec un masque de convolution (par exemple l'opérateur de Sobel).

Ce dernier nous obtient une image de gradient dans laquelle les pixels sont traités de la façon suivante : Un pixel appartient à une droite si l'amplitude de son gradient est supérieure à un seuil et si

dans son voisinage il y a un autre pixel avec une amplitude et une direction de gradient similaire [Bur86].

2.1.2. Regroupement des segments élémentaires de droites

On construit d'abord des segments élémentaires en liant des pixels voisins dans une image de gradient. Ensuite, les segments voisins sont liés dans un seul élément de droite selon un certain critère de regroupement. Le processus se répète, jusqu'à ce que des segments se regroupent. Les méthodes locales sont moins coûteuses en temps que les méthodes globales.

Par contre, à cause du traitement des pixels sur un voisinage local, les méthodes locales sont sensibles au bruit.

2.2. Méthodes globales

Elles sont basées sur une approche globale. C'est à dire qu'on ne détecte pas une droite à partir d'un voisinage limité de chaque pixel, mais on considère tous les pixels dont l'amplitude du gradient dépasse un seuil. Grâce à cette approche, les méthodes globales donnent de meilleurs résultats dans le cas des images fortement bruitées. De plus, elles sont capables de détecter des pixels colinéaires, même si ceux-ci sont isolés dans leur voisinage.

L'inconvénient des méthodes globales est le temps de calcul important. Les images sont en général bruitées ce qui rend difficile la mise en œuvre des méthodes locales. Pour ces raisons, nous avons décidé d'orienter le travail sur les méthodes globales, notamment sur la transformation de Hough et la transformation de Radon.

2.2.1. La Transformée de Radon

Soient (x,y) les coordonnées cartésiennes d'un point d'un espace 2D, $f(x,y)$ une fonction continue. La transformée de Radon de $f(x,y)$, notée $R_f(\rho, \theta)$ est définie par la relation suivante :

$$R_f(\rho, \theta) = \iint f(x,y) \delta(\rho - x \cos \theta - y \sin \theta) dx dy \quad (3.1)$$

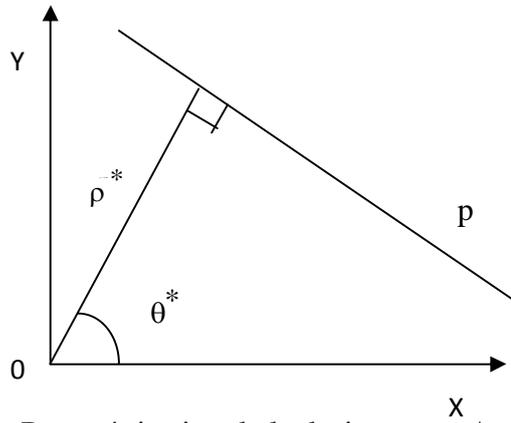


Figure 3.1: Paramétrisation de la droite p avec les paramètres ρ^* et θ^*

La transformée de Randon de la fonction $f(x,y)$ est alors :

$$R_f(\rho, \theta) = \iint f(x,y) \delta(\rho - x \cos \theta - y \sin \theta) dx dy \tag{3.2}$$

$$R_f(\rho, \theta) = \iint \delta(\rho^* - x \cos \theta^* - y \sin \theta^*) \delta(\rho - x \cos \theta - y \sin \theta) dx dy \tag{3.2a}$$

$$R_f(\rho, \theta) = 0; \rho \neq \rho^* \vee \theta \neq \theta^* \tag{3.2b}$$

$$R_f(\rho, \theta) = \int \delta(0); \rho = \rho^* \wedge \theta = \theta^* \tag{3.2c}$$

La Transformation de Randon d'une droite avec les paramètres ρ^*, θ^* est dans l'espace de Randon un Dirac sur $[\rho^*, \theta^*]$ (voir **figure 2.3**). De ce fait, nous pouvons estimer les paramètres d'une droite à partir de la Transformée de Radon d'une image.

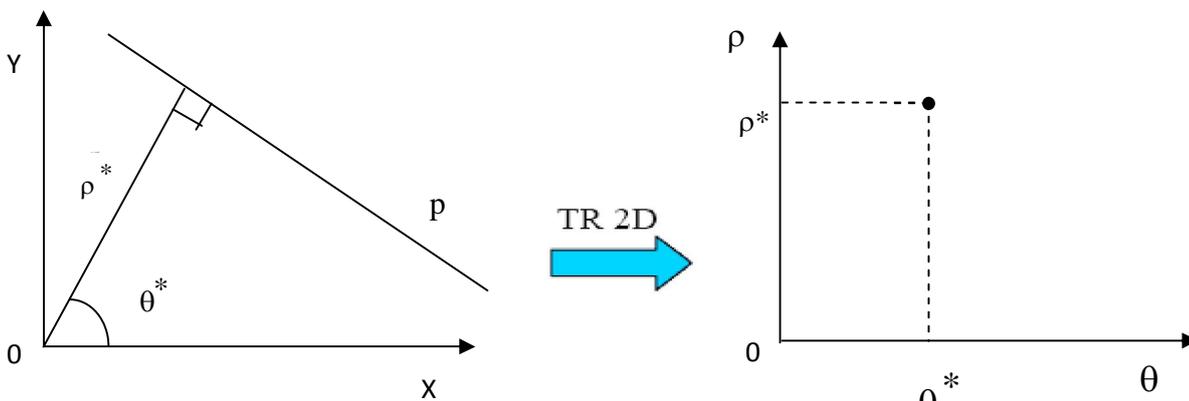


Figure 3.2 : La Transformée de Randon d'une droite

En quelque sorte le point $p(x, y)$ est la projection de la droite P sur une droite perpendiculaire à celle-ci. On omet de donner des détails sur cette transformée, on passe directement à la Transformée de Hough sur laquelle est basé la suite de notre travail.

2.2.2. La Transformée de Hough (TH)

La transformée de Hough a été développée par Paul Hough en 1962, et a été brevetée par IBM. Dans les dernières décennies, la transformée de Hough est devenue un outil standard puissant dans le domaine de la vision artificielle. Elle permet la détection de droites, de cercles ou d'ellipses de façon traditionnelle. Elle peut aussi être étendue à des cas de description d'objets plus complexes. La transformée de Hough permet d'avoir une bonne robustesse contre les discontinuités et les éléments manquants d'une structure. Le principe général de la transformée de Hough est d'établir une projection entre l'espace de l'image et un espace de paramètres représentatif de la forme recherchée.

2.2.2.1 Développement

Nous allons illustrer rapidement le principe de la TH sur l'exemple de la détection des droites. Étant donné un ensemble de contours d'objets représentés par un ensemble de points discrets M , nous cherchons à déterminer si un ou plusieurs sous-ensembles des points M_i appartiennent à des mêmes droites. En effet on peut lire l'équation d'une droite de deux manières. Soit par la lecture traditionnelle qui nous indique que $y=ax+b$, cela implique que tous les points (x,y) vérifiant cette équation seront associés au même couple (a,b) dans l'espace de paramètres. En l'occurrence, il nous faudra simplement deux points pour définir complètement notre droite. Soit on peut lire $b = -ax + y$, alors un couple (x,y) de l'image sera projeté en une droite dans l'espace de paramètres (a,b) . Ensuite pour étudier l'existence de droites. Il faut se pencher sur l'espace des paramètres. Dans cette méthode, chaque pixel appartenant à un contour vote pour une ligne dans l'espace paramétrique. Plus précisément, l'espace de représentation (a,b) est utilisé pour représenter l'espace des droites (c.f. **figure 3.3**). Les votes sont accumulés dans un tableau dit accumulateur où chaque case est associée à une ligne physique donnée. Finalement, les lignes caractérisées par le plus grand nombre de votes sont considérées. Cette condition se traduit par une comparaison avec un seuil fixé S qui peut être défini proportionnellement à la taille de l'image.

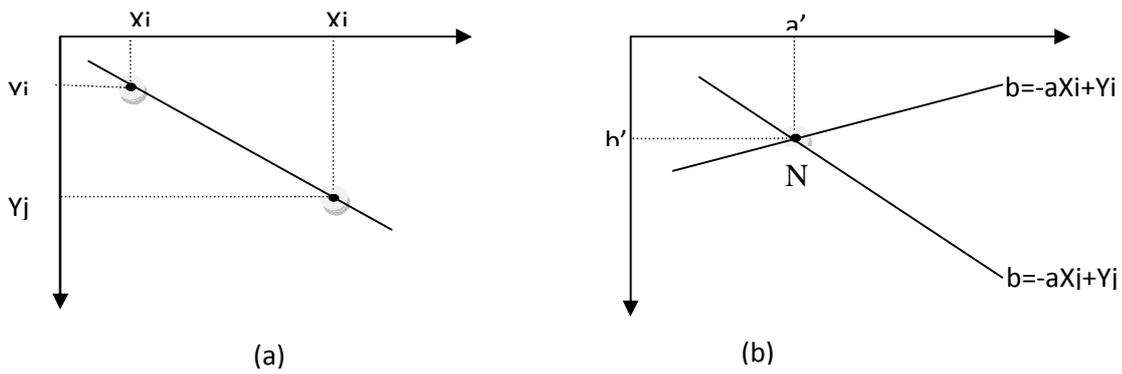


Figure 3.3 : Transformée de Hough (a): plan cartésien, (b): plan des paramètres (a,b).

Le traitement se effectue en une quantification du plan des paramètres en cellules accumultrices où (a_{min}, a_{max}) et (b_{min}, b_{max}) sont les valeurs limites des intervalles de la pente a et de l'ordonnée a l'origine des abscisses b .

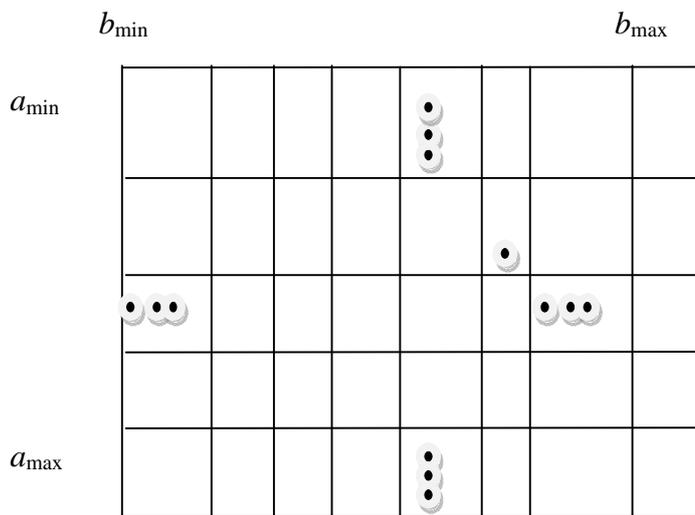


Figure 3.4 : Quantification du plan des paramètres (a,b)

L'inconvénient major de ce paramétrage réside dans l'incapacité à détecter les droites verticales (a tend vers l'infini). Pour remédier a ce problème, un paramétrage polaire (ρ, θ) est satisfaisant. Une droite est définie par l'équation suivante :

$$f(x, y) = \rho - x \cos \theta - y \sin \theta = 0 \tag{3.3}$$

Avec ρ la distance de la droite par rapport à l'origine du plan (x,y) ,

et θ l'angle entre cette droite et l'axe des abscisses x comme illustré sur la **figure3.5** suivante ci-dessous :

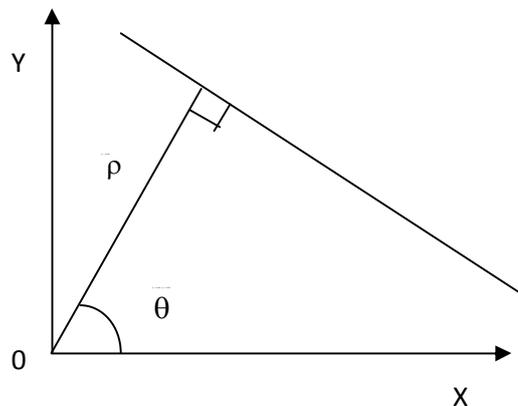


Figure 3.5 : Une droite avec un paramétrage polaire

Le tableau accumulateur est fait de manière identique à celui du premier cas. Il nous reste le choix du pas de quantification pour les deux paramètres qui satisferont les conditions :

1. Une précision de détection aussi bonne que possible.
2. Diminuer l'espace mémoire nécessaire au stockage des accumulateurs.
3. Accélération des calculs.

Chaque point M_i de coordonnées (x_i, y_i) se transforme en une sinuséale dans le plan des paramètres (ρ, θ) et a pour équation $\rho = x_i \cos\theta + y_i \sin\theta$. Une droite sera représentée par un ensemble de sinuséide qui se coupent en un seul point de coordonnée polaires (ρ_0, θ_0) comme on le voit sur la figure suivante :

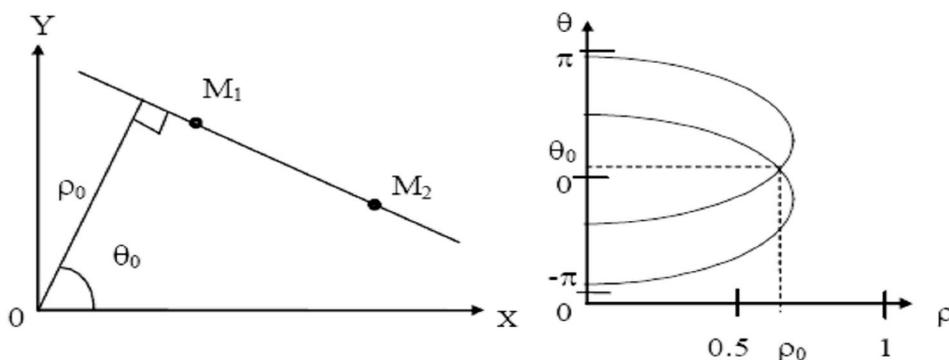


Figure3.6 : Transformée de Hough pour un paramétrage polaire

2.2.2.2 Dimension des paramètres ρ_0 et θ_0

a. **Dimension de θ** : Le champ de la dimension θ est compris dans l'intervalle $[0, 2\pi]$, avec cet intervalle, on peut exprimer toutes les droites. Si en plus on considère les droites appartenant à l'intervalle $[\pi, 2\pi]$, comme des droites à ρ négatif, alors on peut réduire l'intervalle de θ à la moitié.

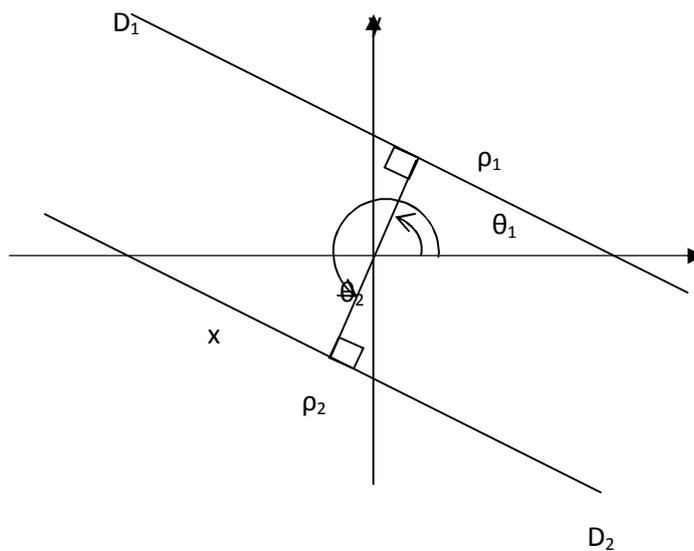


Figure 3.7 : Paramètres polaires de deux droites opposées

b. **Dimension de ρ** : Le champ de dimension de ρ est de $[0, N\sqrt{2}]$ dans le cas d'une image de dimension $N \times N$.

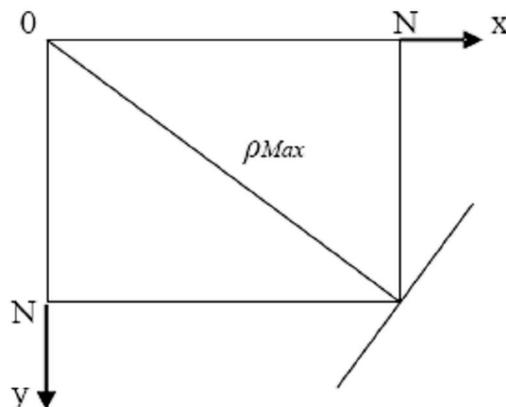


Figure 3.8 : Champ de la dimension de ρ

En plus, si nous déplaçons l'origine au centre de l'image, on aura un champ de dimension allant de $\rho_{\min} = -N/\sqrt{2}$ à $\rho_{\max} = N/\sqrt{2}$

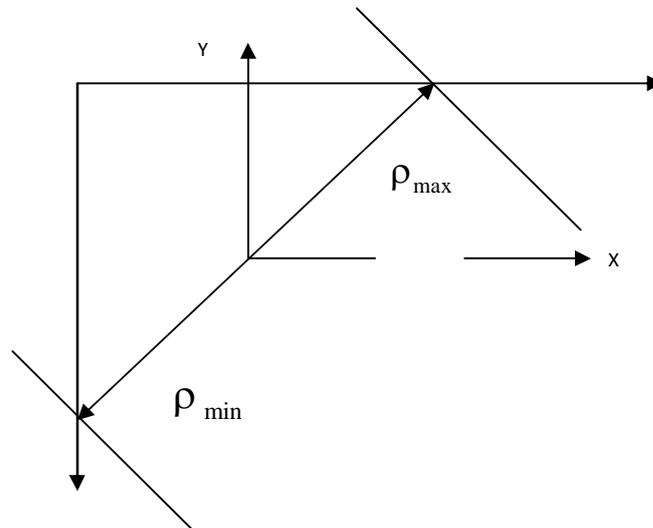


Figure 3.9 : Le champ de dimension de ρ après décalage de l'origine

2.2.2.3 Propriétés de la Transformée de Hough (cas d'une droite)

- Un point du plan cartésien correspond à une sinusoïde dans le plan des paramètres.
- Un point du plan des paramètres correspond à une droite dans le plan cartésien.
- Les points appartenant à la même droite dans le plan cartésien correspondent au point d'intersection des courbes du plan des paramètres.
- Les points appartenant à la même courbe dans le plan des paramètres correspondent en un point dans le plan cartésien.

3. Application de la Transformée de Hough

La Transformée de Hough est une méthode d'extraction dite globale, de ce fait elle opère sur toute l'image à la fois. Elle nous donne tous les points contour qui se trouvent sur la même droite. Par contre elle ne donne aucune information sur la manière dont ces points sont repartis. Ceci induit l'extraction de fausses droites.

Afin de remédier à ce problème, un autre traitement s'impose, d'abord, on élimine tous les segments de droites ayant un nombre de points inférieur à un seuil qui dépend du type d'image et de sa taille. Ensuite pour qu'une droite soit significative, elle doit satisfaire les contraintes suivantes :

- Sa longueur ne doit pas être inférieure à seuil donné (seuilH).
- La longueur des segments qui forment cette droite ne doit pas être inférieure à un seuil donné (seuilC), sinon ils seront éliminés.
- Les points qui forment ces segments de droite ne doivent pas être trop espacés, donc la distance séparant deux points ne doit pas être supérieure à un autre seuil donné (seuilD).

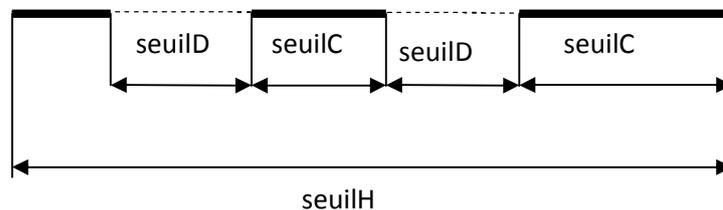


Figure 3.10 : Illustration des différents seuils sur une droite

Pour réaliser une bonne détection, il faut faire un compromis entre ces trois seuils suivant le type d'image.

3.1. Implémentation de la Transformée de Hough :

L'implémentation de la Transformée de Hough sous sa version de base, dite la TH standard s'effectue par la discrétisation de l'espace des paramètres. Pour chaque point contour, on calcule tous les couples (ρ_j, θ_j) qui vont vérifier l'équation $\rho = x_i \cos \theta + y_i \sin \theta$, ensuite on incrémente d'un point toutes les cases correspondantes dans le tableau accumulateur. Le nombre de points accumulés dans chaque case dans l'espace des paramètres correspond au nombre de points qui appartiennent à la même droite sur l'image.

3.2. Quantification du plan des paramètres

La quantification du plan des paramètres revient à quantifier l'angle θ dans l'intervalle $[0, \pi]$, et ρ dans l'intervalle $[-R, R]$ avec R la moitié de la diagonale de l'image. Si on prend n_ρ et n_θ le nombre de valeurs discrètes dans les intervalles de ρ et θ alors les pas de quantification seront :

$\rho_q = \frac{2R}{n_p}$ et $\theta_q = \frac{\pi}{n_q}$. Les valeurs de ρ et θ discrétisées s'écrivent comme suit : $\theta = t * \theta_q$

tel que $0 \leq t \leq n_\theta$,

$\rho = -R + r * \rho_q$ tel que $0 \leq r \leq n_\rho$

Le calcul s'effectue comme suit :

Pour chaque point contour de coordonnées (x, y) de l'image, on calcul pour chaque valeur discrète de θ les valeurs ρ en suite on les approxime aux valeurs discrètes les plus proches. Ensuite, nous incrémentons toutes les cases de tableau correspondantes.

Nous répétons cette opération pour tout point contour de l'image, la figure 3.11 montre l'organigramme de calcul.

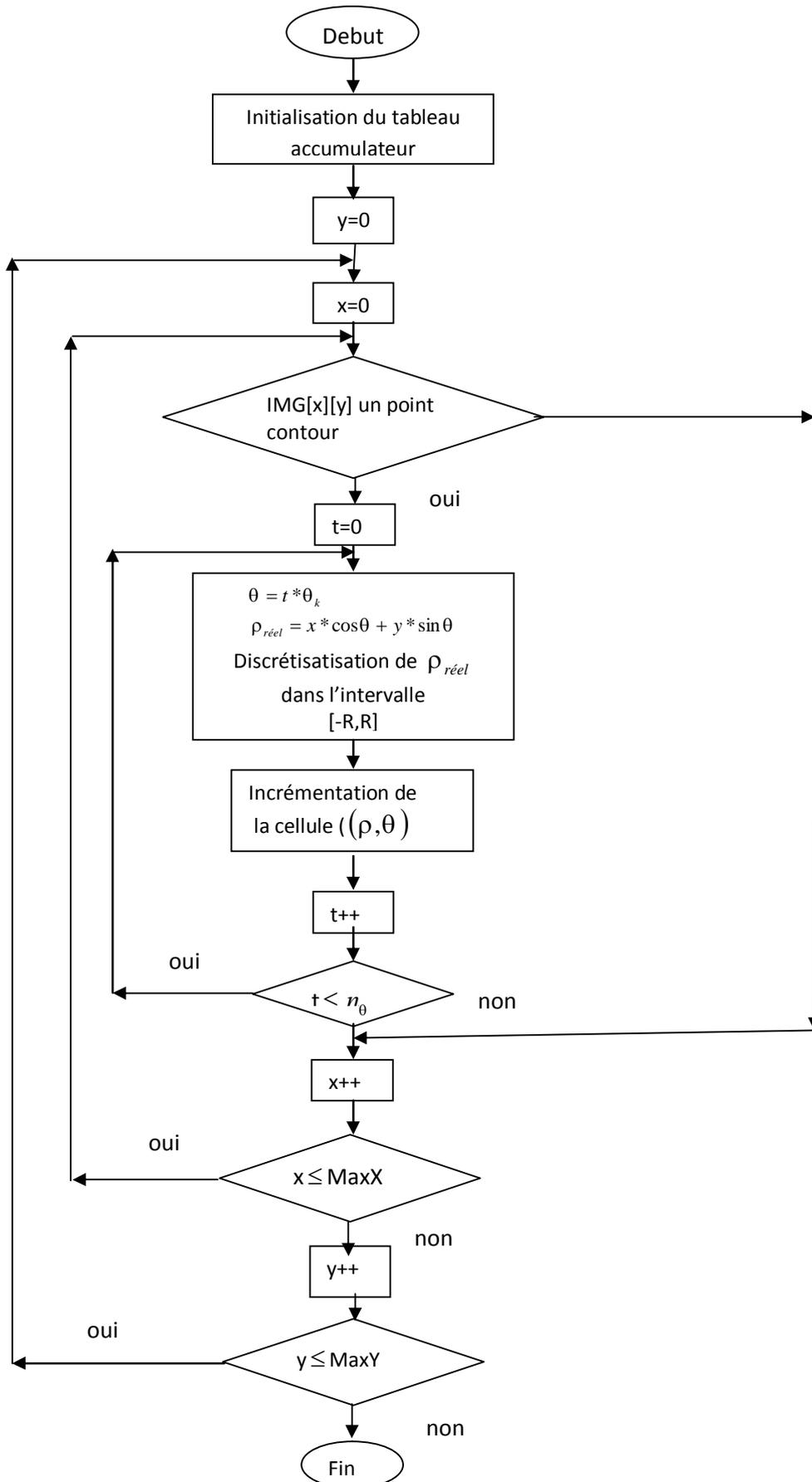


Figure 3.11 : Organigramme de la TH

4. Etat de l'art de la Transformée de Hough

Parmi les inconvénients de la Transformée de Hough, le temps de calcul qui est assez long du fait que pour chaque point contour, on doit calculer pour chaque valeur discrète θ^* , la valeur ρ , surtout pour les cas des applications temps réel. Ainsi plusieurs variantes ont été élaborées en essayant de réduire le nombre d'opérations nécessaires et d'exploiter la possibilité de parallélisme sans réduire pour autant la performance de la TH originale, c'est-à-dire la TH Standard. On va citer quelques variantes.

4.1. Transformée de Hough probabiliste

Cette Transformée a pour but de calculer la TH seulement pour une portion de pixels contours choisis de manière aléatoire à partir de la densité de probabilité uniforme définie sur l'image. Kiraty et al préconisent d'utiliser une portion de 10 à 20 % selon l'application spécifiée [Kir91].

4.2. Transformée de Hough aléatoire

La TH aléatoire [Kit88] est utilisée dans le cas de la détection d'une ligne droite. Au lieu de calculer pour chaque point la courbe qui sera enregistrée dans le plan des paramètres (tableau accumulateur), on prend une paire de pixels aléatoirement, ensuite on calcule le paramètre de cette droite et seulement ces derniers qui seront enregistrés. Les pixels de cette ligne sont ensuite enlevés et cet algorithme est répété jusqu'à ce qu'il n'y ait pas de ligne à détecter.

4.3. Transformée de Hough hiérarchique

La TH hiérarchique combine une structure pyramidale avec la TH. L'image est organisée en grille de sous-images. La TH est appliquée à chacune de ces dernières. Les résultats obtenus sont propagés vers le niveau haut. Les lignes détectées sont propagées au prochain niveau hiérarchique ainsi de suite jusqu'au plus haut niveau qui est l'image.

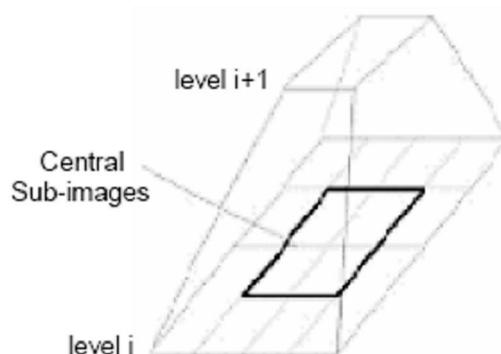


Figure 3.12 : Structure pyramidale de la TH hiérarchique

4.4. Transformée incrémentale : [Kos 91] [Tag01][Ach04][Bou06].

Cette variante a été conçue principalement afin de rendre le temps de calcul le plus court possible. Dans le cas d'une droite, le calcul de l'équation caractéristique définie par $f(x, y) = \rho - x \cos \theta - y \sin \theta = 0$ nécessite l'utilisation des formules trigonométriques et des multiplications. Pour cela la TH incrémentale utilise une autre expression de la TH basée sur des fonctions d'addition et de décalage.

4.5. Transformée de Hough Additive (Additive Hough Transform)

Parmi la plus récente variante de la TH, on citera la TH additive (AHT). Elle est conçue dans le but d'exécuter les calculs d'une manière parallèle tout en conservant la performance de la TH standard [Satz 08]. Le principe de cette méthode est basé sur une propriété de la TH standard énoncée comme suit :

Soit A, B, deux points dans une image. La TH d'un point A par rapport à l'origine O est égale à la somme de la TH de A par rapport à B plus la TH de B par rapport à l'origine O.

$$TH(A, O) = TH(A, B) + TH(B, O) \quad (3.4)$$

Tel que : TH(A, O) = La TH de A par rapport à l'origine O.

TH(A, B) = La TH de A par rapport à l'origine locale B.

TH(B, O) = La TH de l'origine locale B par rapport à l'origine O

La figure suivante illustre cette propriété intéressante de la TH :

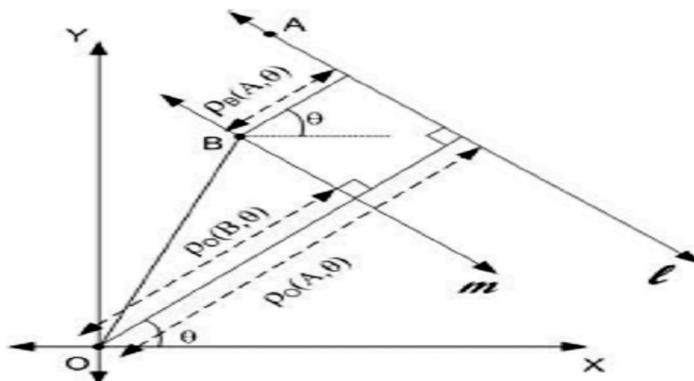


Figure 3.13 : illustration de la propriété de la TH

On remarque que sur la (**figure3.13**) ci-dessus, que l'angle θ ne change pas de valeur, et

$$\rho_O(A,\theta) = \rho_B(A,\theta) + \rho_O(B,\theta) \tag{3.5}$$

Soit : $\rho_O(A,\theta)$: TH du point A par rapport à l'origine pour un angle θ

$\rho_B(A,\theta)$: TH du point A par rapport au point B pour un angle θ

$\rho_O(B,\theta)$: TH de point B par rapport au point O pour un angle θ

5. Parallélisation de la Transformée de Hough

Afin de profiter de la propriété précédente de la TH, on subdivise l'image k^2 blocs et on calcul pour chacun d'eux la TH de ses points contours.

5.1. Calcul de la TH locale

Considérons l'exemple d'une image $8*8$ subdivisée en 4 blocs (voir **fig3.14** ci-dessous). On constate sur cette figure que le point P et P' ont les mêmes positions relatives, donc il suffit juste de calculer pour une seule et de propager le résultat pour les points des autres blocs ayant la même position relative. On peut aussi encore améliorer en calculant la TH de tous les points de blocs et propager le résultat pour les points contours des autres blocs.

s

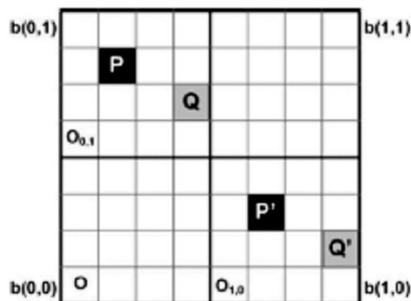


Figure 3.14 : Image $8*8$ subdivisée en 4 blocs

montrant des points qui ont les mêmes positions relatives.

5.2. Calcul de la TH globale

Avec les THs locales calculées, les THs globales peuvent être calculées en utilisant la propriété citée précédemment qui se résume par le fait que les THs des points origines des blocs sont calculées séparément une fois pour toute. De cette manière, on peut calculer la TH de k^2 à la fois si on réalise un système de telle façon qu'à chaque fois qu'on rencontre un pixel contour on cherche la valeur de la TH pixel dans la mémoire qui a les mêmes coordonnées relatives. Ensuite on fait la somme avec la

TH des pixels de référence. Cette opération est exécutée à la fois pour chaque bloc, ce qui permet de calculer la TH de k^2 points à la fois.

5.3. Avantage de Parallélisme :

Comme cité précédemment, l'image contour de taille $(m \times m)$ est subdivisée en k^2 blocs, pour que les THs globales des origines locales soient pré-calculés. Les THs locales des $(m/k)^2$ pixels sont aussi pré-calculées et mémorisées. Et aussi afin d'avoir la TH des k^2 points à la fois en utilisant juste des additions simples.

5.4. Accélération matérielle de l'ATH :

L'accélération matérielle pour différentes architectures de la TH est donnée comme suit :

$$T = (1+p) m^2 \quad (3.6)$$

m = nombre de pixels dans l'image.

p = rapport du nombre de pixels contours sur le nombre total.

: fait référence au nombre d'opérations nécessaires pour le calcul de chaque pixel contour.

On remarque que dans le cas où l'image ne contient aucun pixel contour, un nombre minimum $T = m^2$ est nécessaire pour représenter le temps nécessaire à la lecture de l'image.

Le temps de calcul total de cette méthode pour une implémentation hardware est donné dans [Sat 08] par : $T_{prop} = (m/k)^2$ (3.7)

K = nombre de divisions de l'axe horizontal et l'axe vertical (cas d'une image carrée).

m : nombre de pixels selon l'axe horizontal et l'axe vertical.

Remarque : Les THs locales des blocs et les THs globales des points d'origines des blocs sont pré-calculées d'où ils ne contribuent pas au temps total de calcul de la TH. De ce fait, il est clair qu'on aura un énorme gain au temps de calcul par rapport aux autres méthodes.

D'après [Sat 08], on aura un temps de calcul au moins 67 fois moins que les méthodes utilisant l'algorithme CORDIC (COordinate Rotation DIGital Computer) dans ce cas où on subdivise l'image en 8×8 blocs, avec un taux de 5% des pixels contours dans une image de taille 128×128 .

6. Discussion :

Dans ce chapitre, nous avons présenté une méthode largement utilisée pour l'extraction des formes géométriques analytiques comme les droites, cercles, ellipses et aussi pour les formes non analytiques [Ser94]. La transformée de Hough est une méthode de vote très robuste. Nous l'avons illustré d'une manière simple, ainsi que dans différentes variantes.

La transformée de Hough est bien adaptée aux problèmes ayant un nombre important de données par rapport au nombre des paramètres à estimer. Cette dernière dispose de ressources dédiées aux opérations arithmétiques et aussi dans le domaine de traitement de signal. Le circuit donne de très bons résultats en termes de performances temporelles. L'idée d'utiliser la TH est due à sa robustesse dans l'analyse de l'extraction des primitives d'objets et aussi à ses propriétés d'implémentation hardware.

Dans notre expérimentation, nous avons opté pour l'ATH qui présente une possibilité de parallélisme très intéressante qui nous amène à une réduction de calcul pour chaque bloc et qui se réduit juste à des additions. Une implémentation sur un circuit tel que FPGA nous donnera un temps de calcul fort intéressant pour notre application en robotique mobile en temps réel.

Par conséquent, dans le chapitre qui va suivre, nous présentons un système de reconnaissance de formes 2D moyennant un processus basé sur une approche hybride associant la TH. Cette étape est très importante pour notre expérimentation en vision robotique en temps réel car elle solutionne un des aspects de la problématique de la perception et de la compréhension automatiques de l'environnement.

Chapitre 4

Application à la vision robotique

1. Introduction

Les applications de traitement d'images en temps réel telles que le contrôle de qualité non destructif où la robotique fait appel à des algorithmes qui se décomposent selon trois niveaux :

1. Le bas niveau concerne le prétraitement et la segmentation d'images qui utilisent des techniques de filtrage linéaire ou non, à savoir l'élimination du bruit, et la détection des contours où de régions. Les données à traiter sont de type image en entrée et image en sortie.

2. Le niveau moyen s'intéresse à l'extraction des primitives à partir de l'image traitée au premier niveau et dans ce cas, la structure des informations est de type image en entrée et liste en sortie. On peut pour le cas des contours associer la Transformée de Randon, la Transformée de Hough et le codage de Freeman, etc.

3. Le niveau haut concerne la fusion des primitives (listes) et effectue l'interprétation de la scène observée.

Notre cas d'étude s'arrête aux deux premiers niveaux à savoir les prétraitements : le filtrage et l'extraction des contours par l'opérateur de Deriche. Le deuxième niveau : s'intéresse à l'extraction des droites en utilisant la Transformée de Hough.

Tout système temps réel de traitement de données en plus de la contrainte de robustesse au bruit et de fournir des bons résultats. Le tout est conditionné par le temps d'exécution qui dépend du type d'application. Beaucoup de recherches se sont réalisées afin de rendre les algorithmes moins coûteux d'un côté, et l'exploitation de toutes les possibilités de parallélisme existantes qui s'avèrera comme étant la bonne solution pour accélérer l'exécution de la TH, sous sa forme THA.

2. Application du filtre de Deriche

Le filtre de détection de contours proposé par Deriche a été largement utilisé ces dernières années dans la communauté image. La version logicielle la plus efficace en rapidité utilise une implémentation récursive. La possibilité de changer la résolution à l'aide d'un seul paramètre et la qualité de ce filtre ont motivé les architectes à produire une version matérielle temps réel 25 image par seconde utilisant des ASICs ou des FPGAs. Son étude fait apparaître différentes solutions d'organisation qui s'accompagnent d'optimisations à tous les niveaux : ligne, pixel, opérateur, mémoire [Lor 97]. Nous proposons ici une organisation architecturale que nous qualifions d'optimale

dans le sens où elle minimise le nombre, la taille des mémoires et le nombre des opérateurs nécessaires.

2.1. Organisation originale 2D

Le gradient de Deriche est obtenu par le calcul des gradients horizontaux (G_h) et Verticaux (G_v). L'organisation des calculs est illustrée dans la figure 4.1. Par exemple le gradient est calculé par la combinaison d'un filtre dérivateur vertical D_v et d'un filtre de lissage horizontal L_h . L'entrée E est une image d'intensité et la sortie G , l'image de la norme du gradient.

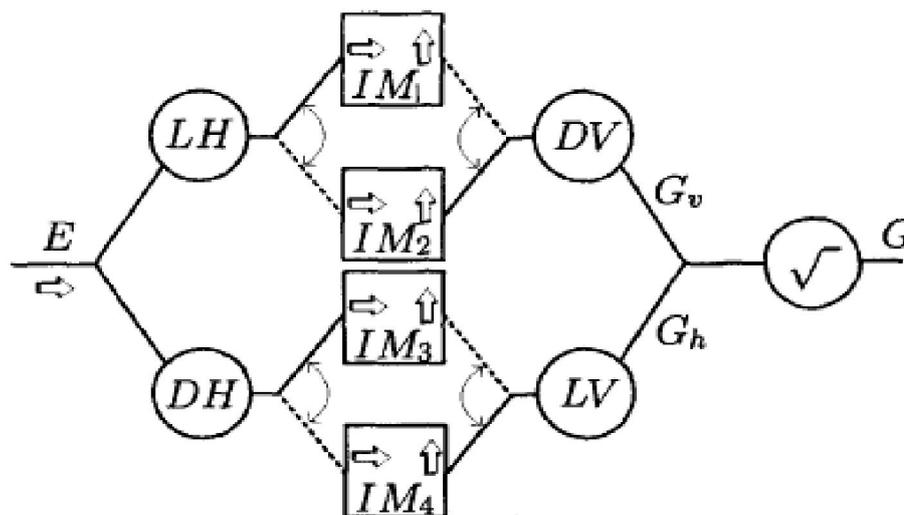


Figure 4.1 Organisation classique des calculs du filtre de Deriche

Chaque opérateur LH , LV , DH , DV est un filtre du deuxième ordre à réponse impulsionnelle infinie à droite et à gauche en haut et en bas (cf. équations 1 et 2). Les opérateurs verticaux sont rigoureusement identiques aux opérateurs horizontaux et travaillent sur des colonnes au lieu des lignes. À cause de la longueur infinie de la réponse impulsionnelle, le calcul en colonne demande la disponibilité de la dernière ligne de l'image traitée horizontalement, d'où la nécessité de stockage des images complètes de résultats horizontaux. Pendant le stockage horizontal des résultats de LH (respectivement DH) dans la mémoire d'image IM_1 (respectivement IM_3), les images des résultats précédents sont lues par colonnes IM_2 , IM_4 . Les mémoires sont donc écrites en lignes et lues en colonnes. IM_1 , IM_2 tout comme IM_3 et IM_4 qui fonctionnent en ping-pong et stockent une image sur deux. Le fait de se rendre compte de cette utilisation des mémoires dans une description de flot de donnée demande à considérer le traitement d'un flot d'image.

$$DH(z) = K_D \left(\frac{z}{(1-e^{-\alpha} z)^2} - \frac{z^{-1}}{(1-e^{-\alpha} z^{-1})^2} \right) \tag{4.1}$$

$$LH(z) = L_D \left(\frac{(\alpha + 1)e^{\alpha} z - e^{-2\alpha} z^2}{(1-e^{-\alpha} z)^2} + \frac{1 + (\alpha - 1)e^{\alpha} z^{-1}}{(1-e^{-\alpha} z^{-1})^2} \right) \tag{4.2}$$

$$K_D = (1 - e^{-\alpha})^2$$

2.2. Organisation parallèle d'un opérateur 1D

Pour simplifier, nous n'exprimons que le cas du dérivateur horizontal DH, la structure étant globalement similaire pour les autres opérateurs. La réalisation matérielle ou logicielle sous forme récursive n'est possible qu'en décomposant le filtre horizontal en deux sous-filtres (fig4.2), l'un infini à droite Dd, l'autre à gauche Dg.

$$DH(z) = K_D (Dd(z) - Dg(z^{-1})) \tag{4.2}$$

$$Dg = K_D \left(\frac{z^{-1}}{(1-e^{-\alpha} z^{-1})^2} \right) \tag{4.2.a}$$

$$Dd = K_D \left(\frac{z}{(1-e^{-\alpha} z)^2} \right) \tag{4.2.b}$$

$$K_D = (1 - e^{-\alpha})^2 \tag{4.2.c}$$

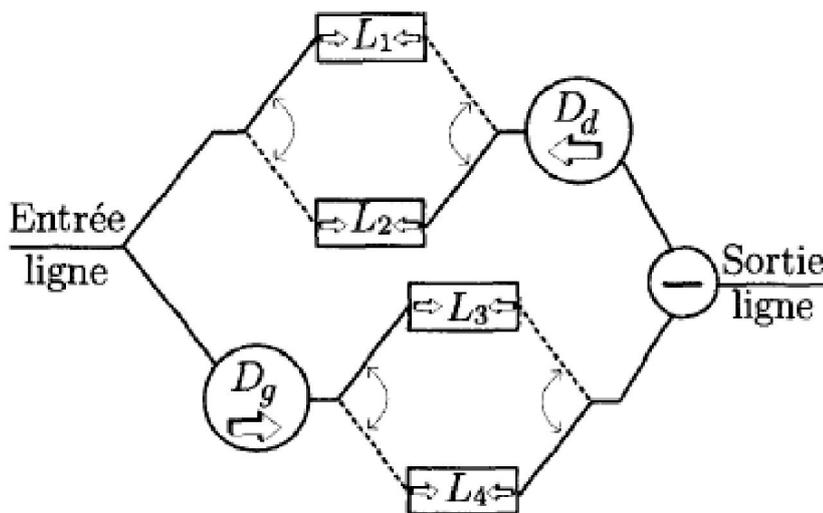


Figure 4.2 Forme parallèle de l'opérateur récursif DH

Puisque les pixels de l'image arrivent temporellement dans le sens gauche droite, le traitement D_d ne peut débuter qu'après l'arrivée du dernier pixel de la ligne. Les mémoires L_1 et L_2 permettent d'inverser l'ordre temporel des pixels pour l'opérateur D_d . Les mémoires L_3 et L_4 servent à synchroniser les flots des pixels avant la soustraction. Les mémoires sont donc écrites dans l'ordre croissant des pixels et lues dans l'ordre décroissant des pixels. L_1, L_2 tout comme L_3 et L_4 fonctionnent en ping-pong et stockent une ligne sur deux.

2.3. Complexité :

L'organisation 2D de la figure 4.1 utilise 4 mémoires d'images. Chaque opérateur 1D (figure 4.2) utilise 4 mémoires de ligne, 16 mémoires de ce type sont nécessaires à la réalisation 2D. Les équations aux différences ont été données dans le chapitre 2.

3. Optimisations algorithmiques

3.1. Forme cascade des opérateurs 1D

Pour réduire d'un facteur deux (2) le nombre des mémoires de ligne par opérateur tel que DH, il faut renoncer à la structure parallèle de la figure 4.2 et utiliser une structure cascade comme illustrée sur la figure 4.3.

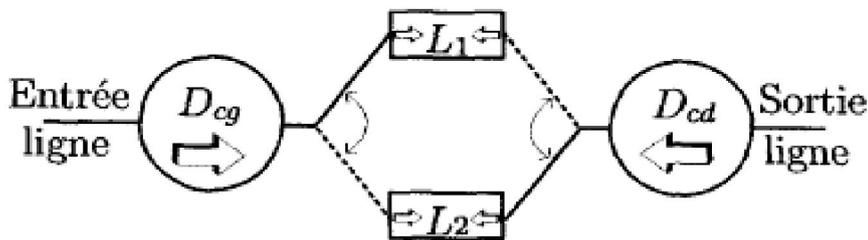


Figure 4.3 Forme cascade de l'opérateur récursif DH

La forme cascade nécessite exactement le même nombre de calculs que la forme parallèle initiale. Tout ceci nécessite la modification de l'expression du dérivateur qui deviendra D_c tel que :

$$D_c(z) = D_{cg}(z) \cdot D_{cd}(z) \tag{4.3}$$

Si on pose $\gamma = e^{-\alpha}$

$$D_{cg}(z) = \left(\frac{(1-\gamma)^2 z^{-1}}{(1-\gamma \cdot z^{-1})^2} \right) \tag{4.3a}$$

$$D_{cd}(z) = \left(\frac{(1-\gamma^2) \cdot (z^2 - 1)}{(1-\gamma \cdot z)^2} \right) \tag{4.3b}$$

3.2. Nouvelle expression du Lisseur

L'optimisation précédente peu aussi s'appliqué au filtre de lissage. Cependant l'équation (2) montre que les coefficients de numérateur sont moins simples que ceux du dérivateur. Afin de simplifier, plutôt de calculer le lisseur comme l'intégrale continue de réponse impulsionnelle de filtre dérivateur ainsi que l'a établi Deriche [Der 87], [lor97] a utilisé la méthode de Trapèze pour calculer l'intégrale numérique de l'équation du filtre lisseur. Le filtre obtenu est donné par l'équation suivante :

$$Lc(z) = k.Dc(z) \frac{1+z^{-1}}{1-Z^{-1}} \quad 4.4$$

Où k est une constante de normalisation choisie pour que l'amplification statique soit unitaire.

L'équation en z du lisseur en cascade se déduit de la manière suivante :

$$Lc(z) = Lcg(z).Lcd(z) \quad 4.5$$

$$Lcg(z) = \left(\frac{(1-\gamma)^2.(1+z^{-1})}{2(1-\gamma.z^{-1})^2} \right) \quad \text{et} \quad Lcd(z) = Lcg(z^{-1}) \quad 4.5a$$

En final l'expression du filtre Lisseur L_1 utilisé est donné comme suit :

$$L_1(z) = Ld(z).Lg(z) \quad \text{où} \quad Lg(z) = Ld(z^{-1}) \quad \text{tel que :} \quad 4.5b$$

$$Lg(z) = \left(\frac{(1-\gamma)^2}{(1-\gamma.z^{-1})^2} \right) \quad 4.5c$$

Remarque : Dans la dernière équation le terme $\frac{(1+z^{-1})}{2}$ a été supprimé, pour des raisons de simplification, il en résulte une légère perte de l'amélioration du signal sur le bruit S/B pour les petites valeurs de γ mais une augmentation en proportion de la localisation de contour.

3.3. Nouvelle Organisation 2D

Le calcul du gradient horizontal G_h (respectivement vertical G_v) par un lissage vertical suivi d'une dérivation horizontale (respectivement lissage horizontal suivi d'une dérivation verticale) est présenté ainsi :

$$G_h(z) = Lc_v(z).Dc_h(z) \quad \text{et} \quad G_v(z) = Lc_h(z).Dc_v(z)$$

Lorca et son équipe ont poussé en avance leurs travaux dans le souci de réduire l'espace mémoire, ils ont aboutit au résultat suivant :

$$G_h(z) = L_2(z).S_h(z) \quad 4.6a$$

$$G_v(z) = L_2(z) \cdot S_v(z) \quad 4.6b$$

$$\text{Tel que } L_2(z) = \left(\frac{(1+\gamma)}{(1-\gamma)} \cdot L_v(z) \cdot L_h(z) \right) \quad 4.6c$$

$$S_v(z) = \begin{bmatrix} +1 & +1 \\ -1 & -1 \end{bmatrix} \quad S_h(z) = \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix} \quad 4.6d$$

$$\text{Et } L_h(z) = L_v(z) = \left(\frac{(1+z^{-1})}{2} \cdot \frac{(1-\gamma)^2}{(1-\gamma \cdot z^{-1})^2} \cdot \frac{(1-\gamma)^2}{(1-\gamma \cdot z)^2} \right) \quad 4.6e$$

L'architecture originale de la figure 4.1 est remplacée par celle de la figure 4.4. On retrouve les deux filtres de lissage horizontaux et verticaux séparés par une mémoire unique d'image. La partie à droite du trait pointillé réalise la dérivée à l'aide d'une unique mémoire ligne et les deux opérateurs S_h et S_v utilisant seulement 4 additionneurs.

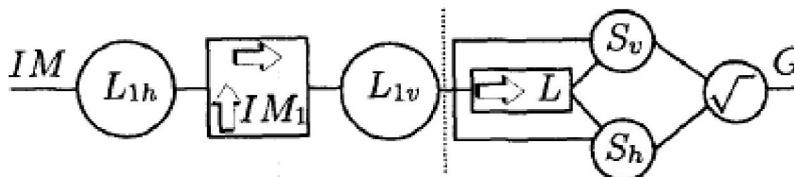


Figure 4.4 Nouvelle organisation 2D du filtre de Deriche.

Dans le souci de simplification [Lor 97] propose la structure de la figure 4.5. Le coefficient de normalisation $\frac{(1+\gamma)}{(1-\gamma)}$ a été supprimé, en conséquence les seuils de détection contours peuvent être ajustés comme cela a été mentionné dans la remarque précédente sur les conséquences de la suppression de terme $\frac{(1+z^{-1})}{2}$.

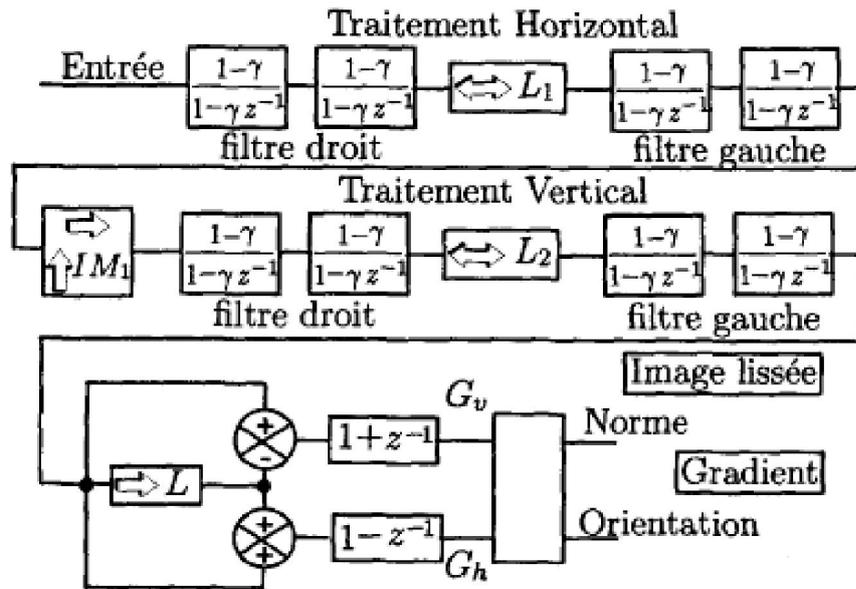


Figure 4.5 Détail de la nouvelle organisation 2D du filtre de Deriche.

3.4. Précision des Calculs

La connaissance de l'organisation architecturale précédente nous permet de dimensionner les opérateurs, les chemins de données et la largeur des mémoires. Les multiplicateurs sont les opérateurs les plus critiques en matière de rapidité et de dimensionnement des mémoires.

En effet l'emploi des coefficients tel que (γ inférieur à 1) engendre des résultats de multiplication comportant une partie décimale de longueur croissante au fil des itérations, les registres qui mémorisent la valeur précédente de la sortie étant de taille finie, il en résulte obligatoirement une erreur de troncature ou d'arrondi.

En se basant sur deux critères, l'un est le bruit du signal et l'autre est le fait que le calcul de gradient est suivi d'une suppression des maxima locaux. [Lor 97] a montré que la sortie L_2 doit être codé sur $1,5\varepsilon + 11,68$ bits tel que : $\gamma = 1 - 2^{-\varepsilon}$ pour la première condition et $3\varepsilon + 12$ bits pour la deuxième critère. On note que le second critère est plus contraignant que le premier. Ainsi pour des données en entrées codées sur 8 bits la sortie de L_2 doit être codé sur 21 bits pour une valeur de $\gamma = 0,875$ ($\alpha = 0,13$) et deux bits de moins si on prend une valeur de $\gamma = 0,75$ ($\alpha = 0,29$).

Conclusion : Nous avons présenté dans ce paragraphe les possibilités d'implémentation du filtre de Deriche de manière générale ainsi implicitement on a proposé une méthode pour laquelle on minimise les ressources matérielles en mémoire. Ce qui est un des objectifs assignés.

4. Calcul de gradient

Afin de simplifier le calcul de la magnitude du gradient, le gradient est apposé de la manière suivante :

$$G = \max\left(|G_h|, |G_v|, \frac{3}{4}(|G_h| + |G_v|)\right) \quad 4.7$$

Cette forme est facile à calculer et à mettre en œuvre, ce qui engendre un gain de temps énorme par rapport au calcul de la racine. Cette forme engendre dans les pires des cas une erreur de 5% de la valeur exacte du gradient calculé par la formule suivante $G = \sqrt{G_h^2 + G_v^2}$.

4.1. Détection des Maximas Locaux

Après avoir calculé l'image gradient et le rapport des gradients $R(i, j) = \frac{G_j(i, j)}{G_i(i, j)}$, on procède à la

suppression des non-maximas non locaux.

1^{er} cas $G_j(i, j) > 1$:

Si $G(i, j) > 0$ alors le point $A(i, j)$ sera alors déclaré comme maxima local si $G(i, j) > U$ et $G(i, j) > Z$. avec U et Z qui sont données par les relations suivantes :

$Si R \geq 1 \quad U = [(R-1)G(i, j+1) + G(i+1, j+1)]/R$	$Z = [(R-1)G(i, j-1) + G(i-1, j-1)]/R$
$Si 1 > R \geq 0 \quad U = (1-R)G(i+1, j) + RG(i+1, j+1)$	$Z = (1-R)G(i-1, j) + RG(i-1, j-1)$
$Si -1 < R < 0 \quad U = [(R+1)G(i, j-1) - G(i+1, j-1)]/R$	$Z = [(R+1)G(i, j+1) - G(i-1, j+1)]/R$
$Si M_f(i, j) < 0 \text{ et } M_i(j, i) = 0 \quad U = M(i, j-1)$	$Z = M(i, j+1)$
$Si M_f(i, j) > 0 \text{ et } M_i(j, i) = 0 \quad U = M(i, j+1)$	$Z = M(i, j-1)$

2^{ème} cas $G_j(i, j) < 1$:

La même procédure est appliquée comme pour le premier cas.

4.2. Seuillage par hystérésis

Le seuillage présenté par Canny a permis de faire un seuillage doux, il consiste à l'utilisation de deux seuils, un seuil bas (S_b) et un seuil haut S_h , cette procédure est expliquée dans le deuxième chapitre, à la fin de cette procédure, on aura une image de contour à qui on appliquera la TH pour la détection des droites.

5. Implémentation de La Transformée de Hough

Parmi les méthodes de détection des droites dans une image on distingue la transformée de Hough, qui a l'avantage de résiste au bruit et de remédier aux discontinuités des droites. Son inconvénient est le grand nombre de calculs, et la mémoire qu'elle demande pour son exécution. Afin de remédier à cet inconvénient plusieurs variante ont été proposées dans le but de réduire le temps de calcul.

5.1. Parallélisme

Pour beaucoup d'applications en temps réel, le parallélisme est vu comme le meilleur moyen de réduire le temps de calcul, en faisant la répartition des calculs sur plusieurs processeurs où élément de calcul dans la limite offerte par la possibilité de parallélisme.

L'utilisation de la TH implique le calcul des équations trigonométriques (sinus et cosinus). Afin de réduire le temps de calcul, [Kar 05] propose l'utilisation de l'algorithme CORDIC et l'orientation du gradient à implémenté sur une FPGA qui donne un temps de calcul très intéressant.

Afin d'appliquer la THA additive, on doit avoir stocké les valeurs de la TH de toutes les coordonnées relatives de tous les points qui se trouvent dans un bloc et les TH des points d'origines relatives des blocs. De cette manière tous les calculs trigonométriques et les multiplications deviennent des additions. Après l'application de la THA, on passe à l'extraction des segments.

5.2. Extraction des droites significatives :

Comme suscité dans le troisième chapitre afin d'éliminer les droites non significatives, on procède par trois étapes :

- On élimine tous les segments de droites ayant un nombre de points inférieur à un seuil qui dépend essentiellement de la taille de l'image.
- Sa longueur ne doit pas être inférieure à seuil donné (seuil H).
- La longueur des segments qui forment cette droite ne doit pas être inférieure à un seuil donné (seuil C), sinon ils seront éliminés.
- Les points qui forment ces segments de droite ne doivent pas être trop espacés, donc la distance séparant deux points ne doit pas être supérieure à un autre seuil donné (seuil D).

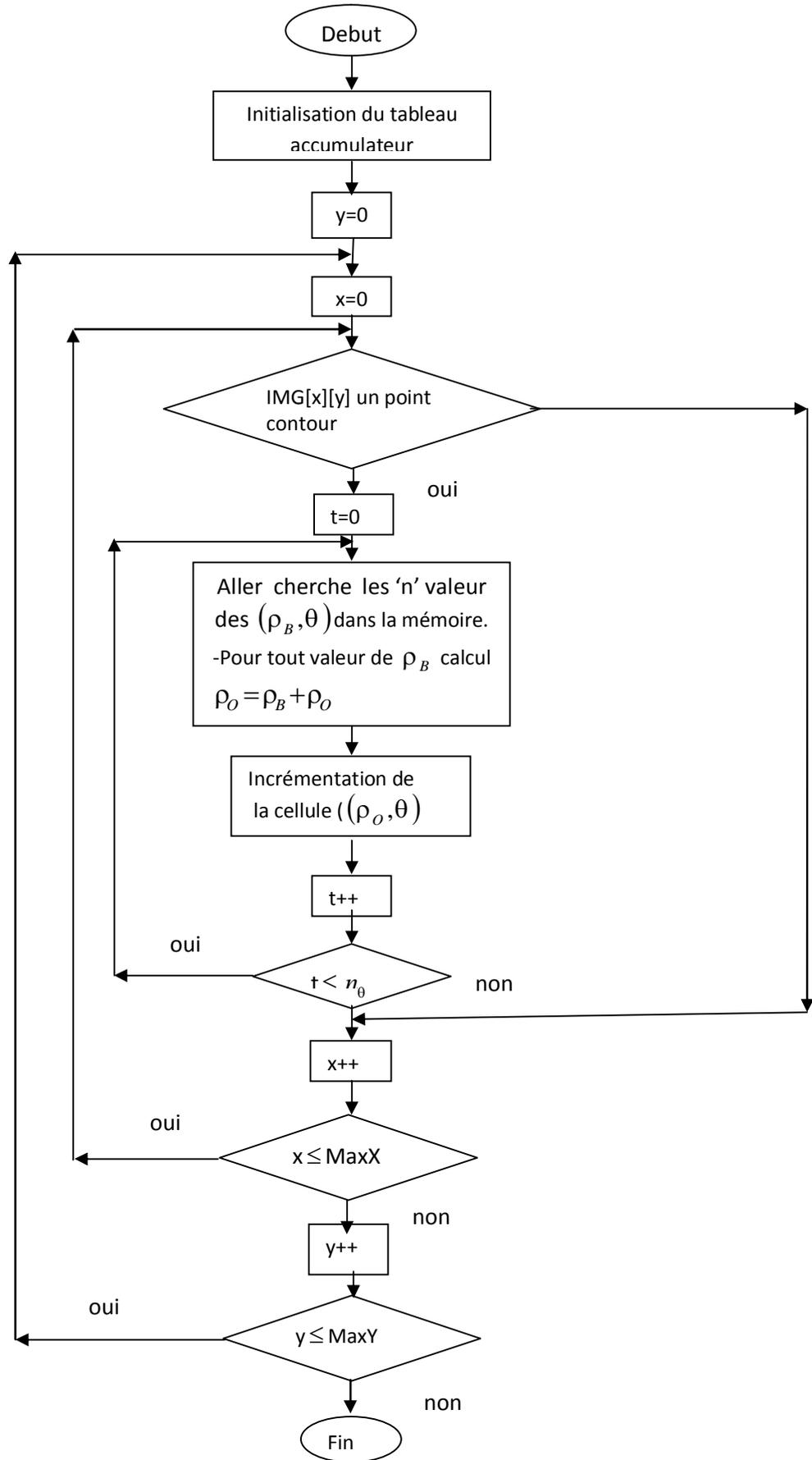


Figure 4.6 Organigramme de la TH pour un seul bloc

6. Tests et résultats

La Transformée de Hough telle qu'elle a été détaillée dans le chapitre précédent, permet d'extraire de l'image contours l'information pertinente sous forme de segments de droite ou de courbes, selon les besoins des applications envisagées [For 93]. En ce qui nous concerne, nous cherchons à détecter les segments de droite contenus dans l'image contours.

L'algorithme de détection que nous avons mis en œuvre procède en deux étapes, toujours dans l'optique d'optimiser le temps de calcul [Sat08].

➤ L'étape de l'application de la transformée de Hough (TH) avec un pas de 10° pour la dimension de θ et un pas de 1 pixel pour la dimension de ρ . Dans cette étape, nous avons la création du tableau accumulateur où chaque cellule de ce dernier indique le nombre de points d'une droite de caractéristiques polaires et les coordonnées de cette cellule dans ce même tableau.

➤ L'étape de recherche des droites significatives dans le tableau accumulateur. Ici, nous cherchons les sommets ou pics dans le tableau accumulateur puis nous fusionnons les pics adjacents.

Cette dernière étape est très importante car elle diminue considérablement le temps de traitement et donne à l'algorithme une fiabilité et efficacité meilleur.

Cet algorithme implémenté sur la station Bull DPX2000, a été testé sur 3 images prises dans 3 positions différentes dans une salle de laboratoire - milieu d'évolution - avec les paramètres suivants :

$$\rho_k = 1; \quad \theta_k = 10^\circ; \quad \text{seuil H} = 20; \quad \text{seuil D} = 5; \quad \text{seuil C} = 10; \quad \text{seuil } \rho = 2 * \rho_k.$$

trait_{Hough} est le temps de traitement de l'étape 1

trait_{det_pics} est le temps de traitement de l'étape 2

I_1, J_1, I_2, J_2 sont les coordonnées des points extrêmes du segment de droite détecté.

Les images des figures (4-7-a) e (4-8-a) sont des images réelles respectives de balises de classes II et I prises en salle de laboratoire. La transformée de Hough dans notre cas ne peut s'appliquer que sur des images contours binaires résultat d'une détection de contours par le filtre de Deriche (Fig.4-7-b, Fig.4-7-c, Fig.4-8-b et Fig.4-8-c).

Le programme de Deriche est élaboré avec $\alpha = 1.5$.

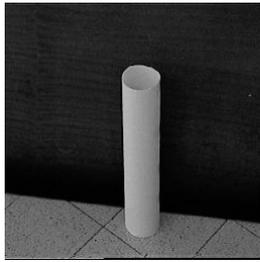


Figure 4.7-a- Image en niveaux de gris

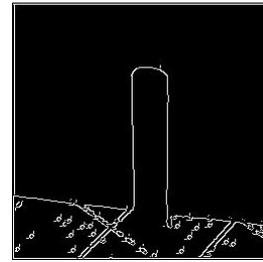


Figure 4.7-b- Image contours binaire par Deriche ($\sigma = 1.5$)

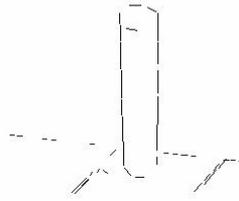


Figure 4.7-c- Image résultat_TH5

Balise de classe II

$tr_{Hough} = 4 \text{ s}$, $tr_{det_pics} = 1 \text{ s}$.

ρ	θ	Nombre de pixels	l_1	J_1	l_2	J_2
102	50	22	104	362	153	423
-111	10	23	14	98	93	110
75	50	23	109	356	162	442
82	60	27	97	346	156	454
98	50	40	87	328	105	360
107	135	20	107	91	125	68
85	125	23	115	84	124	71
60	115	31	118	79	123	70
97	120	44	105	75	137	22
117	130	117	94	89	154	20
23	90	21	105	349	102	385

Tab. 4-1. Coordonnées polaires des pics dans le tableau accumulateur de l'image résultat_TH5

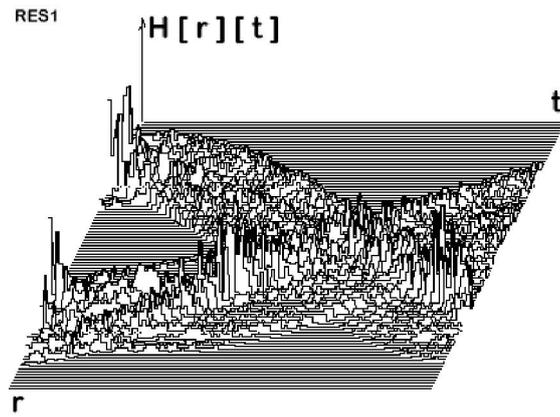


Figure 4.8. Tableau accumulateur de l'image résultat_TH5
Balise de classe II

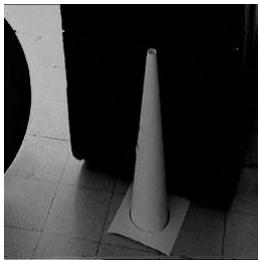


Figure 4.9-a- Image en niveaux de gris



Figure 4.9-b- Image résultat Deriche.

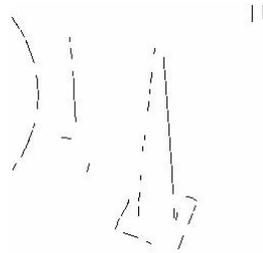


Figure 4.9-c- Image résultat_TH5
Balise de classe I

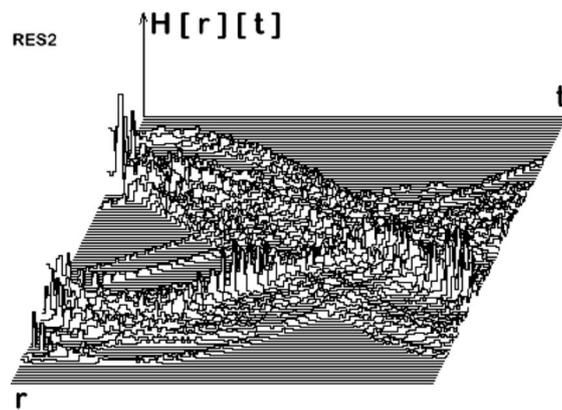


Figure 4.10. Tableau accumulateur de l'image résultat_TH5
Balise de classe I

$tr_{\text{Hough}} = 3 \text{ s}$, $tr_{\text{det_pics}} = 1 \text{ s}$.

ρ	θ	Nombre de pixels	l_1	J_1	l_2	J_2
-108	10	21	8	101	108	119
-121	15	23	161	113	195	117
98	55	45	114	387	132	424
120	135	20	106	93	144	81
75	125	22	129	88	138	76
102	130	27	117	85	135	59
80	120	27	118	82	132	63
90	120	31	127	78	148	64

Tab. 4-2. Coordonnées polaires des pics dans le tableau accumulateur
De l'image résultat_TH5

6.1. Interprétations des résultats

Nous remarquons que les programmes élaborés en langage c de Deriche et celui de la TH sont opérationnels et nous donnent de bons résultats. L'application de ces étapes du processus de traitement sur quelques images prises par notre robot mobile dans différentes positions à l'intérieur de la salle de laboratoire donne les résultats suivants:

Ces résultats sont calculés avec les paramètres suivants :

- $\rho_k = 1$.

- $\theta_k = 10^\circ$.
- Distance focale de la camera $\lambda = 25$.
- L'angle d'inclinaison de la caméra par rapport à l'horizontale
 $\alpha = \text{tg}^{-1}(1.8 / 11)$.
- Distance verticale entre la caméra et le sol $d = 600$ mm.
- Nombre de colonnes de l'écran $r_x = 438$.
- Nombre de lignes de l'écran $r_y = 216$.

Notre programme permet de valider, d'étudier les performances et de vérifier la rapidité et l'efficacité de l'application de la transformée de Hough en temps réel sur les différentes prises de vue de nos balises.

Les tests de robustesse de l'algorithme de la transformée de Hough seul sont observés et permettent assurément de l'intégrer dans notre logiciel de traitement d'images.

Les images résultats des figures (4-7-c) et (4-8-c) illustrant l'utilisation de la transformée de Hough avec un seuil $\theta = 5$ pixels montre que la segmentation a permis une reconnaissance acceptable. Cependant cette reconnaissance peut comme même être améliorée en variant le paramètre θ [Dje 97], [Dag 89].

6.2. Application de notre modèle Rdf

Notre modèle Rdf a été implanté et testé sous c et sur PC embarqué sur notre robot mobile (annexe A et B).

Il nécessite bien entendu la connexion des différents modules assurant la perception, la localisation et la commande des mouvements.

Nous présentons quelques photos prises pendant l'expérimentation avec les paramètres indiqués précédemment.

Les essais tests d'acquisition et de traitement ont été faits sur trois distances 1m, 2m, et 3m. Il est à rappeler que chaque prototype de balises de la base de données, existe sous différentes vues. La base de données fait l'objet d'un apprentissage et d'une généralisation [Jur91], [Pis91].

Pour l'apprentissage, le nombre des images prises est de :

3 prototypes x 2 distances x 15 images = 90 images JPEG.

Cette Base de données permet à notre robot mobile de procéder à la mise en correspondance pour pouvoir reconnaître les types de balises qui lui ont été assignés [Nov 90].



Figure 4.11 Image robot avec caméra embarquée le long du couloir pendant la préparation.



Fig.4-12. Image robot aménagé.



Figure 4.13 Image robot pendant l'acquisition et le traitement (Distance = 2m).



Figure 4.14 Image robot pendant l'acquisition et le traitement
(Distance = 0.8m).



Figure 4.15 Image robot à sa sortie de salle
Fin de tâche.

6.3. Présentation de l'organigramme de notre modèle Rdf :

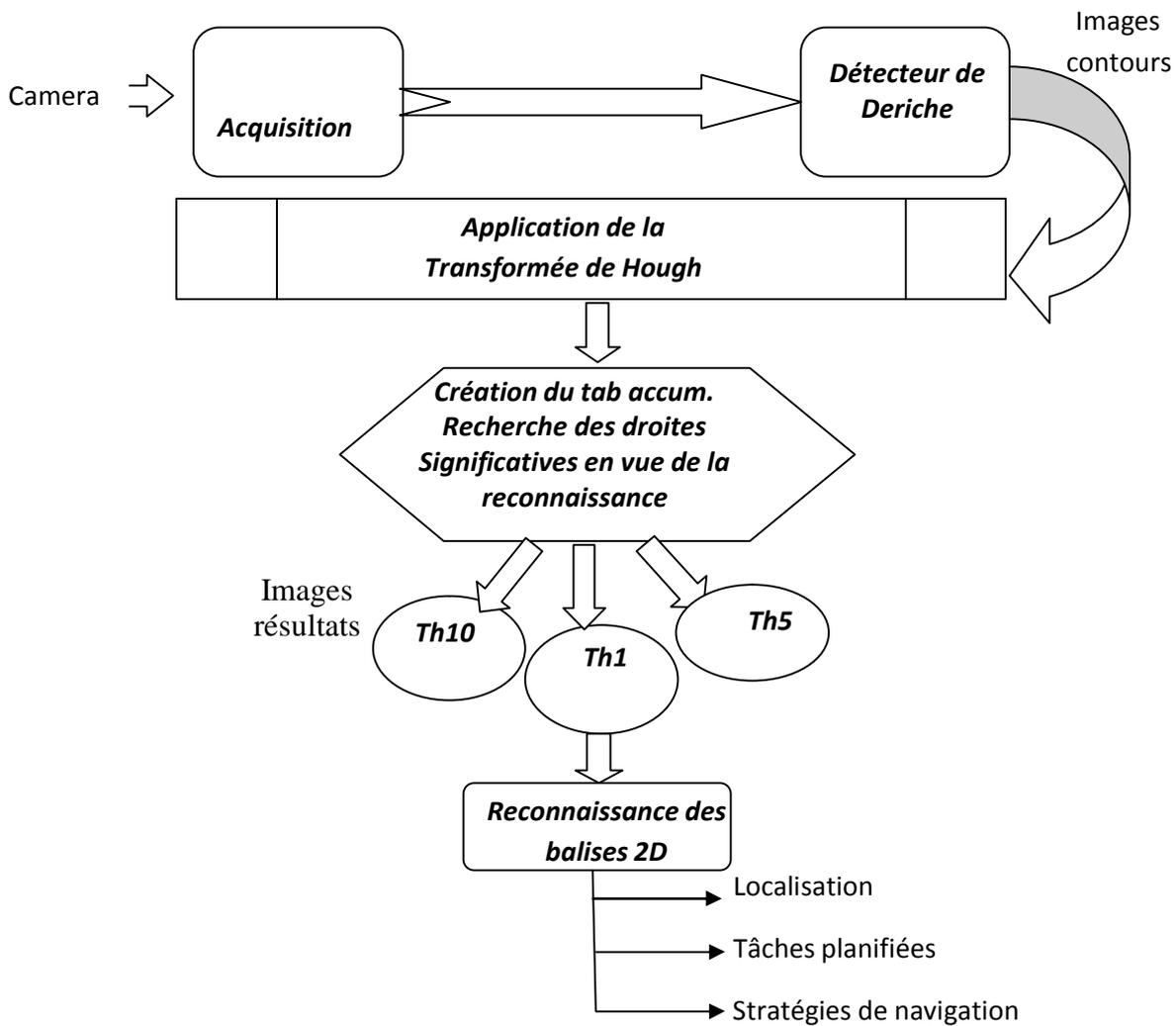


Figure 4.16 : l'organigramme de notre modèle Rdf

7. Résultats expérimentaux

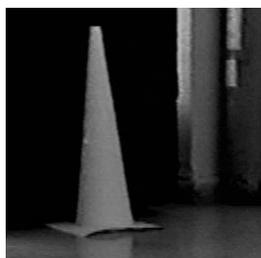


Figure 4.17 Image en niveaux de gris

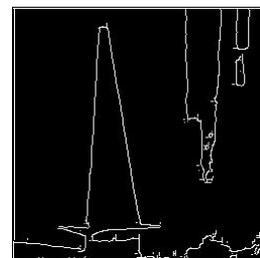


Figure 4.17.a Image résultats Der_bal-poly

d = 2 m



Figure 4.17-b- Image résultats Th10_bal-poly Utilisation de la transformée de Hough en vue de la reconnaissance Seuil =10



Figure 4.17-c- Image résultats Th5_bal-poly Utilisation de la transformée de Hough en vue de la reconnaissance Seuil =5

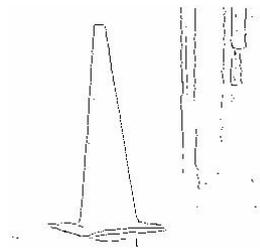


Figure 4.17-d- Image résultats Th1_bal-poly Utilisation de la transformée de Hough en vue de la reconnaissance Seuil =1

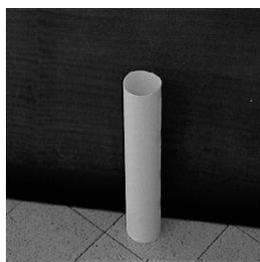


Figure 4.18 Image en niveaux de gris

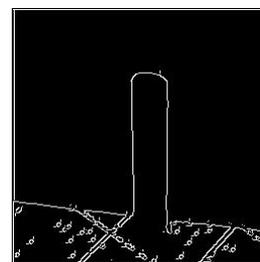


Figure 4.18.a Image résultats Der_bal-cyl

$d = 0.8 \text{ m}$

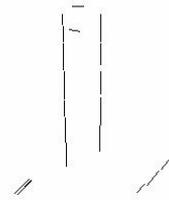


Figure 4.18-b- Image résultats Th10_bal-cyl

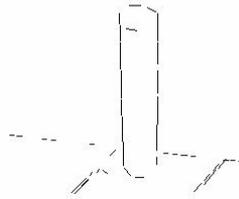


Figure 4.18-c- Image résultats Th5_bal-cyl

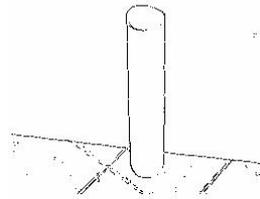


Figure 4.18-d- Image résultats Th1_bal-cyl

8. Discussion

Lors d'une application de vision comme la notre, on est souvent confronté au problème du détecteur de contours approprié parmi l'ensemble des détecteurs rencontrés dans la littérature. Ils sont caractérisés négativement par la non détection des vrais points de contours, l'étiquetage erroné de faux points, et enfin une mauvaise estimation dans la position des contours détectés.

Ces erreurs proviennent essentiellement de la présence du bruit dans l'image. Or le bruit est un élément de l'image difficile à caractériser et donc difficile à éliminer en préservant les contours. Il existe une bibliothèque constituée de cinq détecteurs informels classiques : Sobel, Prewitt, Roberts, Kirsh, et Compass-Gradient. Ajouté à ceux-là il y a aussi cinq détecteurs optimaux de type marche : Canny, le détecteur gradient de Deriche, le laplacien de Shen, le zéro-crossing, et le détecteur de crêtes de Ziou.

Tout détecteur appartenant à cette bibliothèque est considéré comme étant un triplet (algorithme, paramètres, précondition). La précondition est représentée par le contexte dans lequel le couple (algorithme, paramètres) s'exécute correctement. C'est le cas de notre application.

Notre choix s'est porté sur le détecteur de Deriche pour ses performances pratiques.

Ce dernier est un opérateur de détection de contour optimal basé sur le critère de Canny implémenté récursivement. Il apparaît que le détecteur de Deriche semble être mieux adapté à la mise en évidence de contours peu nets et de contours bruités. D'ailleurs il fournit des contours mieux localisés, et il engendre ainsi une plus faible erreur de localisation et d'omission. Ce qui parachève le but recherché.

Pour l'utilisation rationnelle de la TH, il est impératif d'avoir pour notre système de vision des images contours binaires. Le programme Deriche élaboré à cet effet assure correctement l'application. Pendant l'expérimentation, il est à souligner qu'avant de faire entrer notre robot mobile dans la salle de laboratoire, nous lui modélisons le couloir par ses deux arêtes inférieures pour que ce dernier puisse se localiser.

En se référant à sa position, le robot doit aussi générer une trajectoire lui permettant de se déplacer de sa position vers une position destination - le milieu des deux arêtes détectées qui forment le couloir - en fixant le bout du couloir. La détection des deux arêtes du couloir se fait par la recherche de la plus grande valeur accumulée dans les cellules du tableau accumulateur. L'arête gauche se distingue de l'arête droite par la valeur de sa variable θ dans l'intervalle $[0, \pi]$.

L'arête gauche a un θ compris entre 0° et 90° et l'arête droite a un θ compris entre 90° et 180° . Force est de constater que la mauvaise détection des arêtes de l'image de la scène observée influe négativement sur la bonne génération de la trajectoire car le robot mobile suppose que c'est un espace libre. Il peut donc heurter le mur de la salle de laboratoire ou un éventuel obstacle.

Ce qui nous fait remarquer que cette approche donne de bons résultats si nous avons une bonne acquisition de l'image de la scène observée et si le milieu d'évolution est bien éclairé. Il y va de soi que la mauvaise acquisition de l'image de la scène observée et le mauvais éclairage de l'environnement influent négativement sur la bonne détection des segments de droite de l'image contours de la scène observée.

D'après les images résultats des figures (4-18-b, 4-18-c et 4-18-d), on remarque que le programme de la TH est fonctionnel et que la reconnaissance augmente au fur à mesure que le pas de quantification diminue.

Avec $\Delta = 10^\circ$ nous prenons en compte tous les segments de droites dont le pas est supérieur à 10 pixels.

Le taux de précision est faible et on le voit bien sur la figure (4-18-b).

Avec un $\theta = 5^\circ$, la reconnaissance de notre balise de forme polyédrique augmente sensiblement mais demeure encore insuffisante. Par contre en diminuant encore le pas de quantification jusqu'à la valeur de $\theta = 1^\circ$, nous observons effectivement d'après les images résultats des figures (4-16-c et 4-18-c) que la reconnaissance de nos balises toute forme confondue est bonne. Le taux de précision est estimé à 99.0 %.

On observe aussi dans l'étape de détection de contours par le filtre de Deriche, une réduction du bruit assez importante qui a atténué les erreurs de reconnaissance.

Les images résultats nous le confirment. On peut aisément affirmer que les tests de traitement ont été réalisés sur les trois distances conformément à la base de données et que les temps de traitement sont vraiment réduits.

A titre d'exemple le temps de traitement de la balise cylindrique sur le PC embarqué a avoisiné 3 secondes (cf. figure 4-18-d). Ce qui démontre encore la rapidité et la fiabilité de notre algorithme.

Les figures ci-dessous (Fig. 4-19-a, 4-19-b, 4-20-a, et 4-20-b) montrent l'efficacité et la robustesse de notre modèle pour le processus Rdf. En présence de bruits d'origine diverses (amer, reflets dus à l'éclairage) notre robot mobile s'est accoutumé à l'environnement. Il a pu cependant reconnaître nos balises et par la même différencier entre elles et les amers dans la scène (pilier, reflet, etc.). Ce qui prouve que l'algorithme est suffisamment robuste qui est en fait un des buts assignés.

Exemple de l'application de la TH sur des images réelles

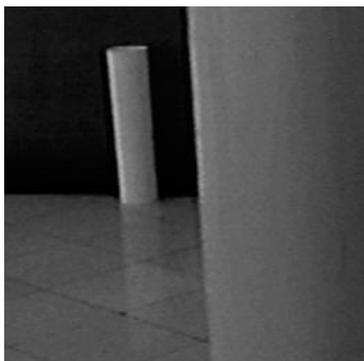


Figure 4.19-a- Image originale



Figure 4.20-a- Image originale

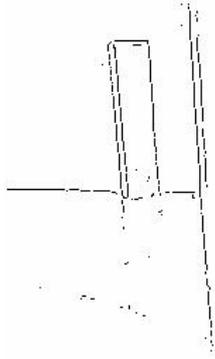


Figure 4.19-b- Image résultat



Figure 4.20-b- Image résultat



Figure 4.21 a-Image Bureau



Figure 4.21 b- TH de l'image Bureau



Figure 4.22 a- Image couloir du Labo LAMPA

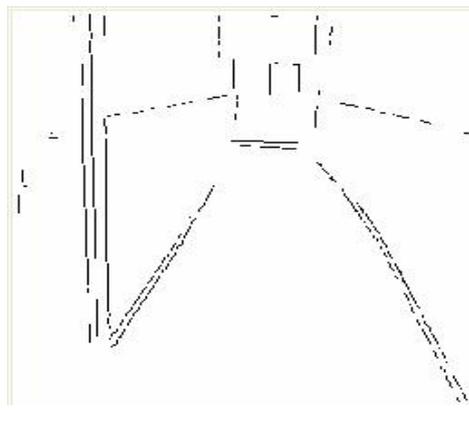


Figure 4.22 b- TH de l'image couloir

Conclusion et Perspectives

Il existe deux approches pour la résolution des problèmes de la reconnaissance des formes : Une approche statistique et une approche structurelle. La reconnaissance statistique, dite aussi globale, opère sur une représentation non segmentée de la forme. Elle est basée sur l'étude statistique des mesures effectuées sur la forme à reconnaître.

La représentation structurelle peut sembler plus naturelle en reconnaissance des formes. En effet, elle tente de trouver dans une forme les éléments signifiants et leurs relations. Cette approche consiste à représenter la forme à reconnaître par une suite ordonnée de composants élémentaires. Une telle représentation peut se faire selon deux méthodes :

1. L'approche syntaxique : les formes sont codées par des mots utilisant un alphabet dont chaque terme représente un élément de la forme à décrire.
2. L'approche numérique: Elle utilise des techniques combinatoires comme les arbres d'interprétation et les isomorphismes des graphes.

La simulation en Rdf est la technique la plus efficace pour l'analyse et la conception des systèmes complexes. Cette dernière imite un système réel et permet de l'étudier sans nécessiter sa présence ou sa disponibilité. Elle s'applique à divers domaines tels que les systèmes industriels robotisés, les systèmes de production et les systèmes naturels.

En effet, notre travail décrit dans ce mémoire s'inscrit dans cette problématique et porte sur l'utilisation de filtre de Deriche et sur la Transformée de Hough (TH) en robotique mobile.

Conclusion générale

Il nous a permis par la recherche de l'état de l'art d'avoir une idée générale sur une application non des moindres qui est la robotique mobile.

Cette recherche nous a permise aussi de faire une synthèse des différentes méthodes utilisées en robotique et des différents capteurs utilisés dans les robots.

L'utilisation de certains outils pour le processus de Rdf proposé comme le filtre de Deriche nous a permis de comprendre la mise en œuvre de la détection de contours de manière optimale à condition de choisir le bon paramètre afin de garder une meilleure localisation et un bon rapport PNSR (signal/bruit). En effet, les méthodes de détection de contours permettent d'obtenir très rapidement (de l'ordre de quelques millisecondes) un ensemble de contours qui serviront de base à des algorithmes de fermeture de contours. En raison de leur caractère local, ces méthodes doivent diminuer le bruit en appliquant un filtre passe-bas comme la gaussienne ou le filtre de Deriche.

Cependant l'utilisation de ces filtres pose le problème du réglage des paramètres car si l'on filtre trop, on risque de rater des bouts de contours importants alors que si on ne filtre pas assez, l'on risque de se retrouver avec une quantité de contours non significatifs. C'est pour cela que nous avons opté pour le choix de Deriche qui solutionne cet aspect en rapport à ses performances (bruit) mais aussi par rapport au temps d'exécution du fait qu'il est récursif et demande un nombre fixe d'itérations quelque soit l'échelle du filtre.

Après l'application des de filtre et les traitements ultérieurs annexes à savoir la suppression des non maximas locaux et du seuillage, nous avons aussi proposé une organisation qui permet l'exécution d'une manière parallèle. Nous avons remarqué par l'expérimentation que la détection des balises augmentait au fur à mesure que le pas de l'angle diminuait, alors que d'un autre coté le temps de calcul augmentait d'où la nécessité de faire un compromis.

Par suite arrive l'application de la Transformée de Hough dont on a déjà cité les différentes variantes dans l'optique d'améliorer le temps de calcul. et de garder les mêmes performances que la TH standard.

Conclusion générale

Cette technique associée à notre processus de reconnaissance de formes nous a permis d'avoir tous les concepts de base liés au champ de dimension de θ et de ρ , ainsi que son implémentation hardware sur un circuit FPGA.

Pour une meilleure illustration, nous avons implémenté notre algorithme sur un circuit FPGA de Xilinx, plus précisément la famille Virtex2, qui contient un module de génération des paramètres (ρ, θ) ainsi que le module nécessaire pour le processus de vote. Cette dernière dispose de ressources dédiées aux opérations arithmétiques et aussi dans le domaine de traitement de signal. Le circuit donne de très bons résultats en termes de performances temporelles.

L'idée d'utiliser la TH est due à sa robustesse dans l'analyse de l'extraction des primitives d'objets et aussi à ses propriétés d'implémentation hardware.

La Transformée de Hough Additive (THA) permet d'optimiser en temps et en espace le calcul global de la TH, ce qui justifie notre choix de l'étudier dans notre projet.

On peut relater et dire que notre modeste contribution peut être améliorée surtout au niveau de la reconnaissance. Ce dernier point est très utile pour permettre au robot de se localiser dans son environnement peu connu et d'analyser les objets qui se trouvent autour de lui.

En effet un des problèmes de la reconnaissance d'objets est le fait de la nécessité de succès dans toutes les étapes allant du traitement de bas niveau jusqu'au traitement de haut niveau. Dans le cas d'une application réelle, on doit étudier le type de matériel à utiliser et analyser toutes les possibilités d'implantation possibles. Parmi d'autres outils de la Rdf, on retrouve les réseaux de neurones, les systèmes hybrides basés sur les algorithmes génétiques que nous projetons d'utiliser dans les travaux à venir. On peut résumer les perspectives de notre travail comme suit :

- L'analyse de la précision des résultats. En particulier, nous pensons qu'il est important d'avoir des mesures quantitatives sur la qualité de la reconstruction et de s'intéresser au problème du choix des points pris en compte pour l'auto-calibration. Sachant que le problème discret de l'auto-

Conclusion générale

calibration du mouvement consiste à retrouver la translation et la rotation de la caméra entre deux prises de vues (on suppose que la scène est statique).

- Réduction du temps de calcul de la méthode de segmentation, afin que celle-ci puisse être raisonnablement intégrée dans le processus de reconnaissance de formes.
- Traitement d'images 3D moyennant la vision stéréoscopique en robotique mobile.
- Vision dynamique 3D.

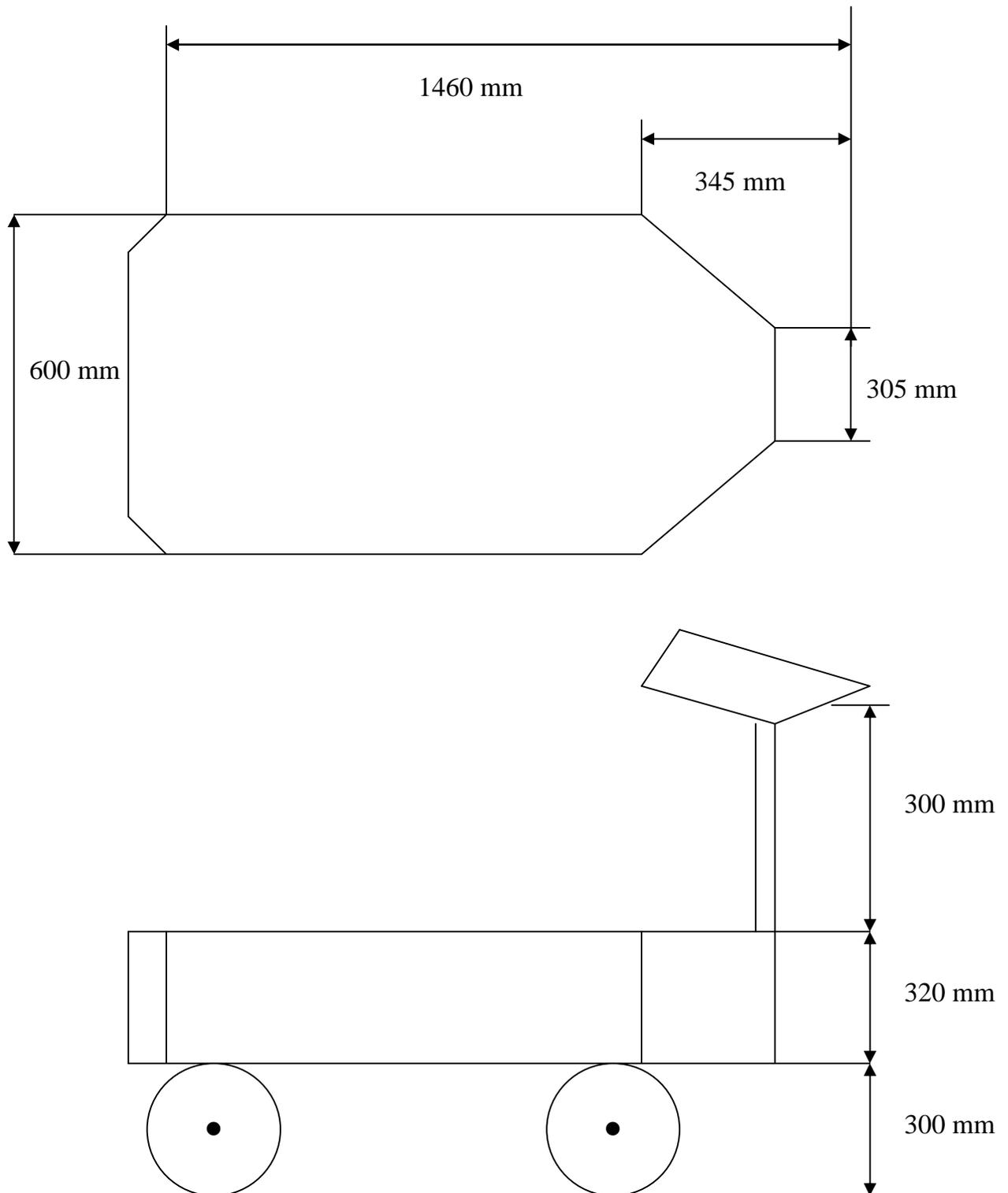
ANNEXE A

ROBOT MOBILE



CESA avec camera CCD

STRUCTURE DU ROBOT MOBILE (CESA)



ANNEXE B

SYSTEME DE VISION

I- CAPTEUR D'IMAGES

I-1. Caractéristiques du capteur

1. Haute résolution (500 pixel / ligne).
2. Grande sensibilité.
3. Durée de vie quasiment illimitée et fiabilité élevée.
4. Résistance aux chocs et aux vibrations.
5. Résistance aux bruits microphonique et à l'influence électromagnétique.
6. Pas de distorsion géométrique.
7. Localisation précise d'un point dans une image.
8. Compacité et faible poids.

I-2. Caractéristiques de la caméra

1. Contrôle du gain automatique ou programmable.
2. Sortie vidéo composite (RS 170) avec correction gamma pour monitoring TV.
3. Sortie vidéo composite linéaire (gamma =1), pour le traitement d'images.
4. Pas de pixel défectueux en sortie.
5. Câble unique de raccordement au système utilisateur (vidéo, synchronisation, alimentation).

I-3. Spécifications physiques

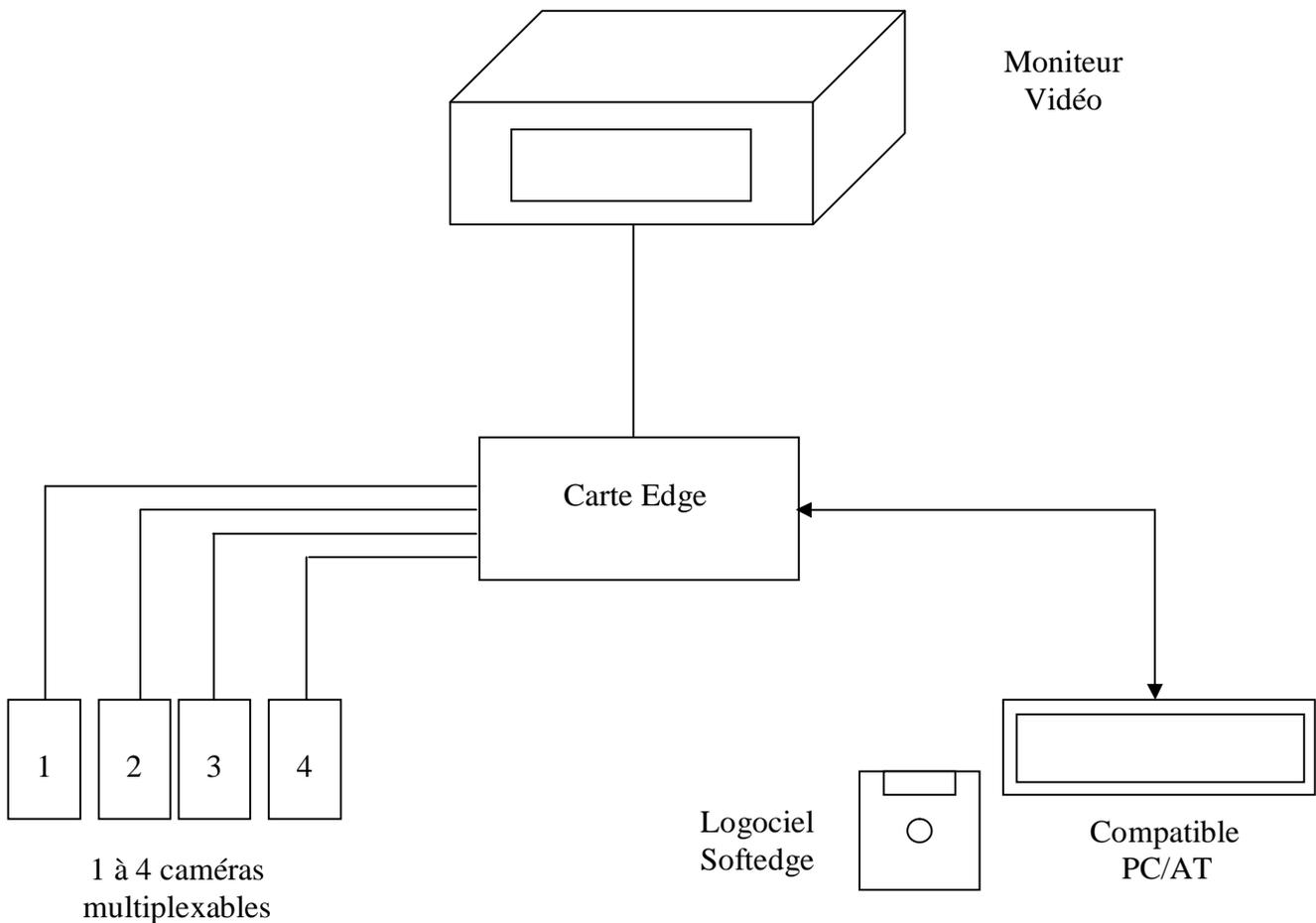
1. Température de fonctionnement : -10° C à 50° C.
2. Alimentation requise : 12V \pm 10%.
3. Consommation : 420 mA en DC in = 12V.
4. Poids : 700 g.
5. Boîtier en aluminium.

II- LA CARTE ELECTRONIQUE

I-1. LA CARTE EDGE 90

Les principales fonctions de la carte Edge sont :

- a. De numériser le signal vidéo délivré par la caméra et de réaliser des transformations sur les niveaux de gris.
- b. D'engendrer à partir des images en niveaux de gris, soit des images binaires résultant d'un seuillage programmable, soit des "images des contours" résultant de l'extraction de points de contraste par calcul du maximum local de gradient.
- c. De stocker ces images dans une mémoire double accès disponible sur le bus du calculateur.



A-1. Système de vision par ordinateur construit autour de Vision Edge 90.

Les caractéristiques techniques sont les suivants :

1. Carte électronique au standard IBM PC / AT.
2. Multiplexage de 1 à 4 caméras (standard CCIR ou RS 170).
3. Conversion analogique-digital sur 64-128-256 niveaux de gris du signal vidéo de la caméra sélectionnée (fréquence de conversion = 10 MHz).
4. Conversion de 64 à 16 niveaux de gris par table de conversion réglable en entrée (Look-up-table).
5. Génération d'une image binaire par seuillage programmable sur l'image en niveaux de gris.
6. Génération d'une " image des contours" par calcul du maximum local de gradient.
7. Visualisation des images engendrées (256, 128, 64, 16 niveaux de gris, binaire, contour).
8. Transfert des images binaires seuillées ou de points de contraste dans une mémoire de sortie double accès interfacée en bus du calculateur.

II-2. LE SOFTWARE SOFTEDGE

Softedge comporte trois couches :

La première couche Edge1 qui interface la carte correspond à un ensemble de fonctions écrites en assembleur 8086. Cette couche, proche du matériel gère la communication avec le micro-contrôleur intégré à la carte.

La deuxième couche Edge2 correspond à un ensemble de fonctions écrites en langage C qui réalisent l'interface entre le programmeur et la couche Edge.

Elle permet entre autres de :

- 1) Livrer l'état de la carte à tout moment.
- 2) Choisir le mode de fonctionnement (binaire, contour sur binaire, contour sur niveaux de gris).
- 3) Choisir le mode de visualisation (256, 128, 64, 16NG, binaire, contour).
- 4) Sélectionner la source vidéo : caméra 1 à 4 ou mire.
- 5) Sélectionner une table de conversion utilisateur : 8 à 15.
- 6) Charger une table de conversion utilisateur : 8 à 15.
- 7) Définir le seuil de binarisation : 0 à 63.
- 8) Acquérir l'image.
- 9) Stocker ou lire une image binaire ou de point de contraste dans la mémoire Edgre.

- 10) Lire ou écrire un pixel dans la mémoire.
- 11) Lire ou écrire un pixel dans la mémoire Edge.

La troisième couche MENU utilise les fonctions de Edge et permet à l'utilisateur grâce à un ensemble de menus interactifs de :

1. Tester les fonctionnalités de la carte à la réception du produit.
2. Choisir les différentes fonctions de la carte qui pourront être utilisées dans une application.

BIBLIOGRAPHIE

- [AGG96] J.K. Aggarwal, J. Ghosh, D. Nair, I. Taha, "A Comparative Study of Three Paradigms for Object Recognition - Bayesian Statistics, Neural Networks and Expert Systems," *Advances in Image Understanding*, K. Bowyer and N. Ahuja, Eds, IEEE Computer Society Press, 1996, pp. 241-262.
- [Ach 04] Achour K., Djekoune O. "Incremental Hough transform: an improved algorithm for digital device implementation." *Real Time Imaging*, Elsevier, 2004.
- [Ati 92] Atiqzaman M. "Multiresolution Hough transforms: an efficient method of detecting pattern in images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*; 14(11):1090-65, 1992.
- [BAS92] R. Basri, "Recognition by Prototypes," *MIT Artificial Intelligence Laboratory, A.I. Memo No. 1391*, Dec. 1992.
- [BAS93] R. Basri, "Recognition by Prototypes," in *Proc. IEEE Conf. Comput. Vis. Patt. Recog.*, 1993, pp. 161-167.
- [BIE87] I. Biederman, "Recognition by Components : A Theory of Human Image Understanding," *Psychological Review*, vol .94, no. 2, 1987, pp. 115-147.
- [Bou06] A. Boutarfa. « Reconnaissance de formes 3D par approche neuronale associant la Transformée de Hough en robotique mobile, Application a productique » thèse doctorat, Département d'électronique Faculte des science de l'ingénieur Université de Batna Algerie. 2006
- [Bur86] J.B. Burns, A.R. Hanson, E.W. Riseman, *Extracting straight lines*, IEEE Trans. Pattern Anal. Mach. Intell, p. 425-455, (1986).
- [Can86] A computational approach to edge detection *Trans. Pattern Anal. Machine Intell* Vol .PAMI-8,6 IEEE . (November 1986)
- [Coc95] J.P. Cocquerez , Sylvie Philipp. « Analyse d'image : filtrage et segmentation » *Traitem ent du Signal* . Edition MASSON
- [Che 05] Cheng I., Boulanger P. "Feature Extraction on 3D TexMesh Using Scale-space Analysis and Perceptual Evaluation." *IEEE Transactions on Circuits and Systems for Video Technology*, Special Issue, October 2005.
- [Der90] R .Deriche, «Fast Algorithms for Low-Level Vision», *IEEE Transactions on Pattern Anal. and Machine Intell.*, vol .PAMI-12, no .1, pp . 78-87, Jan.90
- [Der 87] R. Deriche, "Optimal edge detection using recursive filtering." In *Proc. First Int. Conference of Computer Vision*. London. June 8-12, 1987.

- [Der 87b] Deriche R., Cocquerez JP. "Extraction de composantes connexes basée sur une détection optimale des contours". Actes du congrès MARI-87, pages 1-9, Tome 2, Paris 1987.
- [Dje 97] O. Djekoun, "Localisation et guidage d'un robot mobile". Thèse de magister, réf 07/97_M/EL, USTHB, Alger, 1997.
- [Dag 89] A. Dagba. "Localisation d'un objet polyédrique à partir d'une seule image perspective de luminance". Thèse de Doctorat d'Université, Université Blaise Pascal de Clermont-Ferrand, n°.167, mai 1989.
- [Eng 03] Grandchamp E., Marthon P. "Driving Segmentation and Recognition Phases using multiscale characterization". IGARSS 2003.
- [Fla 01] Flandin G., Chaumette F. "Fusion d'informations visuelles pour la localisation d'objets complexes". In ORASIS 2001, Congrès francophone de vision,
- [For 93] Forsberg. Johan, Larson. Ulf, Ahman. Per and Wernersson. Ake. "The Hough transform inside the feedback loop of a mobile robot". IEEE, International Conference on Robotics and Automation, Vol. 1, pp.791-798, Atlanta, Georgia, May 2-6, 1993.
- [Jur 91] Jurie. F, Chapuis. R, Gallice. J, Alison. "Estimation de la position d'un mobile dans un environnement paramétrable". 8^{ème} congrès de la reconnaissance de formes et de l'intelligence artificielle, Lyon Villeurbanne, Vol. 2, pp. 637-647, 25-29 novembre 1991.
- [Jol 06] E.K. Jolly, M. Fleury "Multi-Sector algorithm for hardware acceleration of the Hough Transform" *Image and Vision Computing* 24 (2006) pages: 970-976
- [HAV94] P. Havaldar, G. Medioni, F. Stein, "Extraction of Groups for Recognition", in *Proc. Euro. Conf. Comput. Vision*, 1994.
- [Hay02] Jean Bernard Hayet, Frederic Lerasle & Michel Devy. A visual landmark framework for indoor mobile robot navigation. In IEEE International Conference on Robotics and Automation, volume 4, pages 3942-3947. Washington (USA), May 11-15 2002.
- [Hor95] R. Houraud, O. Monga "vision par ordinateur : outils fondamentaux" 2ième édition. Edition Hermès 1995
- [Hue86] A. Huertes and G. Medioni, Detection of intensity changes with subpixel Accuracy using Laplacian-gaussian Masks, IEEE Trans.on PAMI, Vol. 8, n 5, pages 651-661 (1986)

- [Ima97] Imad Issa "Filtre d'ordre adaptatifs pondérés : Application au traitement d'images", thèse Doctorat, Université de Savoie octobre 1997.
- [Gir79] G. Giralt, R. Sobek & R. Chatila. A Multi-Level Planning and Navigation System for a Mobile Robot ; A First Approach to Hilare. In 6th International Joint Conference on Artificial Intelligence, volume 1, pages 335-337, 1979.
- [Kar 05] M. Karabernou, .F. Terranti, "Real-time FPGA implementation of Hough Transform using gradient and CORDIC algorithm", *Image and Vision Computing* Vol. 23, pp. 1009-1017, 2005.
- [Kir 91] N. Kiryati, Y. Eldar and A. M. Bruckstein, "A Probabilistic Hough Transform", *Pattern Recognition*, vol. 24, no. 4, pp. 303-316, 1991.
- [Kit 88] Kitter J., Illing J. "A survey of the Hough Transform". In *computing Vision, Graphics and Image Processing* 44: (1), pp. 87-116, 1988.
- [Kos 91] Koshimizu H., Numada M. "FIHT2 algorithm: a fast incremental Hough Transform". *IEICE Transactions*; E74 (10), 1991.
- [Lor 97] F.G. Lorca, L. Kessal, D. Demigny. "Efficient ASIC and FPGA Implementation of IIR Filters for Real Time Edge detection". *Proceedings of International Conference on Image processing*, Vol. 2 . no. 26, pp. 406-409, 1997.
- [Mar04] Antonio Marin-Hernandez. *Vision dynamique pour la navigation d'un robot mobile*. Thèse de doctorat, Institut National Polytechnique, LAAS, Toulouse, France, 9 Février 2004.
- [Pis91] Pissard-Gibolet. R et Rives. P. "Asservissement visuel appliqué à un robot mobile". *Etat de l'art et modélisation cinématique*. INRIA, rapport de recherche n°.1577, décembre 1991.
- [Nag79] M. Nagao. T. Matsuyama, Edge preserving Smoothing. *CVGIP* vol 9 ,pages 394-407, 1979
- [Oso 05] Osorio G., Boulanger P., Prieto F. "An Experimental Comparison of a Hierarchical Range Image Segmentation Algorithm". *18th Canadian Conference on Artificial Intelligence Graphics Interface in Victoria BC*, pp. 571-578, May 8-11, 2005.
- [PER91] C.G. Perrott, L.G.C. Hamey, "Object Recognition, A Survey of the Literature", *Macquarie Computing Reports no. 91-0065C*, 1991.
- [POP94] A.R. Pope, "Model-Based Object Recognition - A Survey of Recent Research", *Technical Report 94-04*, 1994.

- [Nov 90] Novikoff. P, Chellali. R, Milgram. M. "Navigation of a mobile robot with obstacle and landmark recognition", I.C.R.V, 1990.
- [Sat 08] R.K. Satzoda, S. Suchitra, and T. Srikanthan, IEEE, Parallelizing the Hough Transform Computation, IEEE Signal Processing Letters, Vol. 15, pp. 297-300, 2008.
- [Tag01] Tagzout S., Achour K., Djekoune O. "Hough transform for FPGA implementation". Elsevier Journal, Signal Processing; 81(6):1295-1301, 2001.
- [Riz98] A. Rizzi, G. Bianco & R. Cassinis. A Bee-Inspired Visual Homing Using Color Images. Robotics and Autonomous Systems, vol. 25, no. 3-4, pages 159-164, November 1998.
- [Sie04] Roland Siegwart & Illah R. Nourbakhsh. Introduction to autonomous mobile robots. The MIT Press, 2004.
- [Tzv91] Tzvi Ben., Sandler M. "Analogue Implementation of the Hough Transform". In. IEEE Proceeding-G, Vol. 138, No. 4, August 1991.
- [Thr98] S. Thrun. Learning Metric-Topological Map for Indoor Mobile Robot Navigation. Artificial Intelligence, vol. 99, no. 1, pages 21-71, February 1998.