



**HAL**  
open science

## Contribution au guidage des robots mobiles par vision

Mohamed Elarbi Boudihr

► **To cite this version:**

Mohamed Elarbi Boudihr. Contribution au guidage des robots mobiles par vision. Sciences de l'ingénieur [physics]. Institut National Polytechnique de Lorraine, 1992. Français. NNT: 1992INPL086N . tel-01751089

**HAL Id: tel-01751089**

**<https://hal.univ-lorraine.fr/tel-01751089v1>**

Submitted on 29 Mar 2018

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

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



## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

[M]1992 ELARBI BOUDIHIR, M.

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE  
ECOLE NATIONALE SUPERIEURE D'ELECTRICITE ET DE MECANIQUE

---

**THESE**

Présentée à l'I.N.P.L  
pour l'obtention du

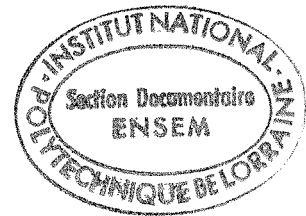
**DOCTORAT**

DE L'I.N.P.L

Section : Génie Electrique  
Option : Automatique

présentée par

Mohamed. ELARBI BOUDIHIR  
Ingénieur I.N.E.L.E.C



---

Sujet :

**CONTRIBUTION AU GUIDAGE DE ROBOT  
MOBILE PAR VISION.**

---

*Soutenue publiquement le 14 octobre 1992 devant la commission d'examen:*

*Rapporteurs* : P.L. WENDEL  
J.G. POSTAIRE  
*Examineurs* : J. CICCOTELLI  
K. TOMBRE  
M. DUFAUT  
R. HUSSON

## AVANT - PROPOS

*Le travail présenté dans ce mémoire a été effectué au Centre de Recherche en Automatique de Nancy ( CNRS - UA 821 ) au sein de l'équipe Vision - Robotique de l' E.N.S.E.M <sup>(1)</sup> .*

*Je tiens à exprimer ma profonde reconnaissance à Monsieur R. HUSSON, Professeur à l' E.N.S.E.M et M. DUFAUT, Maître de Conférence à l'E.N.S.E.M qui m'ont accueilli dans le laboratoire Vision - Robot. Qu'ils soient assurés de ma profonde gratitude pour le soutien qu'ils m'ont apporté dans la réalisation et la rédaction de cette thèse.*

*J'exprime ma sincère reconnaissance à Monsieur P.L. WENDEL, Professeur d'Electronique à l'ENSPS <sup>(2)</sup>, à Monsieur J.G. POSTAIRE, Professeur d'Automatique à l'U.S.T.L <sup>(3)</sup>, à Monsieur K.TOMBRE Chargé de Recherches à l'I.N.R.I.A, et à Monsieur J. CICCOTELLI, Chef du Service Etudes et Projets Industriels du C.T.B.A <sup>(4)</sup>, qui ont accepté de s'intéresser à ce travail et de participer au jury.*

*Que tous mes collègues du laboratoire et de l'E.N.S.E.M soient ici vivement remerciés pour leur accueil chaleureux et leur contribution à ce travail.*

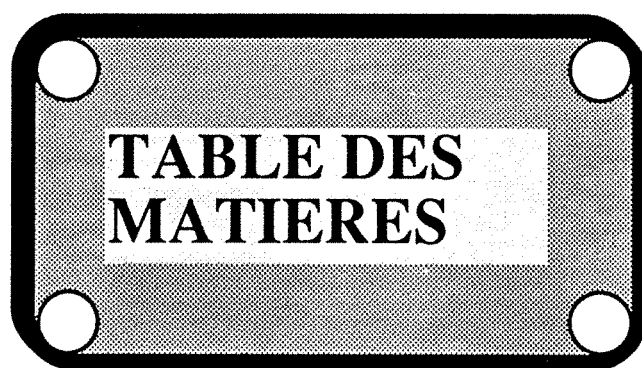
<sup>(1)</sup> E.N.S.E.M : Ecole Nationale Supérieure d'Electricité et de Mécanique

<sup>(2)</sup> E.N.S.P.S : Ecole Nationale Supérieure de Physique de Strasbourg

<sup>(3)</sup> U.S.T.L : Université des Sciences et technologies de Lille

<sup>(4)</sup> C.T.B.A : Centre Technique du Bois et de l'Ameublement





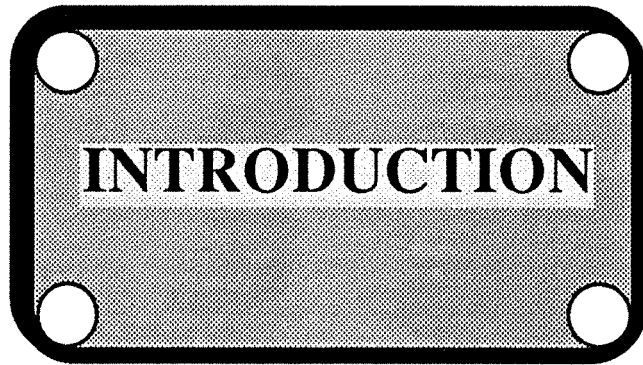
<b>INTRODUCTION.....</b>	<b>7</b>
<b>CHAPITRE I - GENERALITES.....</b>	<b>11</b>
I.1 - La vision et la navigation.....	12
I.1.1 - La vision bidimensionnelle.....	13
I.1.1.1 - Niveau acquisition .....	15
I.1.1.2 - Niveau pré-traitement.....	15
I.1.1.3 - Niveau analyse d'image.....	17
I.1.1.4 - Niveau interprétation.....	18
I.2 - Le robot mobile.....	18
I.2.1 - Conception des niveaux de traitement.....	19
I.2.2 - Description des tâches.....	21
I.2.2.1 - La tâche.....	22
I.2.2.2 - Les capteurs utilisés.....	22
<b>CHAPITRE II - ARCHITECTURE LOGICIELLE DU SYSTEME DE VISION.....</b>	<b>26</b>
II.1 - Introduction.....	27
II.2 - Aperçu sur la vision en robotique mobile.....	28
II.3 - Architecture logicielle du système de vision.....	31
II.3.1 - Fonction des algorithmes.....	32
II.3.2 - Coordination entre les différentes tâches .....	37
<b>CHAPITRE III - EXPLOITATION DE LA SCENE.....</b>	<b>42</b>
III.1 - Introduction.....	43
III.2 - Phase initiale.....	45
III.2.1 - Accentuation et extraction des éléments dominants de la scène .....	47
III.2.1.1 - Pré-traitement et formation de l'image des orientations.....	48

III.2.1.2 - Dualité amplitude-orientation du gradient.....	50
III.2.1.3 - Choix du masque pour le calcul du gradient.....	50
III.2.1.4 - Choix du type de gradient.....	51
III.2.1.5 - Quantification et segmentation de l'orientation du gradient.....	54
III.2.1.6 - Discrimination des éléments linéaires.....	56
III.2.1.7 - Reconstruction de l'image orientation .....	73
III.2.2 - Localisation et paramétrisation des éléments détectés.....	74
III.2.2.1 - Délimitation du champs de recherches.....	75
III.2.2.2 - Application de la transformée de Hough.....	76
III.2.2.3 - Résultats expérimentaux.....	86
III.3- Phase de suivi continu.....	88
III.3.1 - Placement des fenêtres initiales.....	88
III.3.2 - traitement en mode de suivi continu.....	89
III.4 - Conclusion.....	97

## **CHAPITRE IV - CONTRIBUTION DE LA CARTOGRAPHIE A LA NAVIGATION**

<b>AUTONOME.....</b>	<b>99</b>
IV.1 - Introduction.....	100
IV.2 - Modélisation de la caméra.....	101
IV.2.1 - Transformation perspective en coordonnées homogènes.....	104
IV.2.2 - Les paramètres de la configuration Robot-caméra.....	106
IV.3 - Détermination des bords de route sur l'image.....	107
IV.3.1 - Représentation de la carte.....	107
IV.3.2 - Gestion de la carte virtuelle .....	112
IV.3.3 - Prédiction de la scène à partir de la carte.....	115
IV.3.4 - Placement des fenêtres initiales pour le mode de suivi continu.....	121
IV.4 - Génération de trajectoire à partir de la carte .....	121

IV.5 - Conclusion.....	127
<b>CHAPITRE V - ROLE DE LA BASE DE CONNAISSANCE VISUELLE DANS</b>	
<b>LE SYSTEME DE VISION.....</b>	<b>130</b>
V.1 - Introduction.....	131
V.2 - Contraintes géométriques sur la structure de la route .....	131
V.3 - Sélection des bords de route.....	133
V.4 - Modèle de la scène.....	134
V.5 - Détermination des paramètres physiques .....	140
V.5.1 - Détermination de $\theta$ .....	142
V.5.2 - Détermination de $\Delta$ .....	148
V.5.3 - Détermination de $\Lambda$ .....	152
V.6 - Conclusion.....	155
<b>CHAPITRE VI - APPLICATION: TRAITEMENT D'UNE SEQUENCE D'IMAGES</b>	
<b>DECRIVANT LE PARCOURS DU ROBOT.....</b>	<b>158</b>
VI.1 - Calibration de la caméra.....	159
VI.2 - Détermination des paramètres physiques.....	160
VI.3-Traitement d'une séquence d'images ( suivi de route).....	165
<b>CONCLUSION GENERALE.....</b>	<b>177.</b>
<b>ANNEXE .....</b>	<b>181</b>
<b>BIBLIOGRAPHIE.....</b>	<b>185</b>



Actuellement le robot mobile constitue un support unique et efficace pour développer les techniques de perception et de navigation dans des environnements réels de structure complexe.

Pour tenter d'approcher le monde réel complexe par une représentation mathématique idéalisée, les robots mobiles utilisent des capteurs, le traitement d'image, leur interprétation géométrique, et leur mouvement dans le temps. Parmi les différents types de capteurs, le capteur visuel peut fournir des éléments d'information très utiles à la description de l'environnement et contribuer d'une manière efficace à la navigation autonome. C'est ce qui a permis le développement extraordinaire des recherches en vision pour la robotique ces dix dernières années.

Nous présentons dans ce mémoire l'étude et l'organisation d'un système de navigation en milieu naturel. Ce système est composé de différents modules assurant chacun une tâche spécifique programmée par un module superviseur.

Dans le **premier chapitre**, nous exposons l'état de l'art dans les domaines de la vision et de la navigation, ainsi que des différents niveaux de traitement qui caractérisent la vision bidimensionnelle, suivi d'une description du robot mobile utilisé et de ses capteurs .

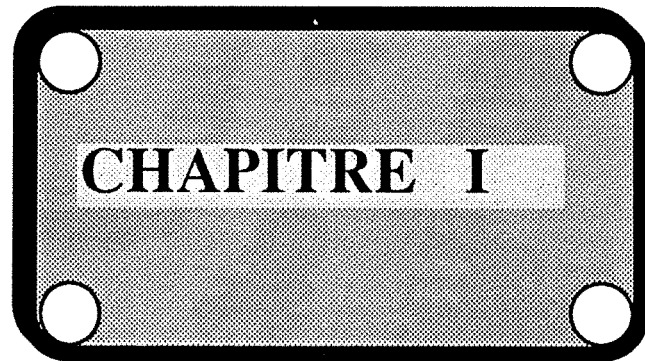
Le **second chapitre** donne l'architecture logicielle du système de vision et décrit la fonction de chaque module qui le compose. Ensuite, nous exposons la façon dont les tâches sont synchronisées au niveau du module de supervision.

Dans le **chapitre III**, le problème de détection des bords de route à partir des scènes naturelles est abordé. Le traitement utilisé est établi au niveau du module exploitation de la scène. Nous séparons à ce niveau donc le mode opérationnel de ce module en deux phases distinctes: phase initiale qui correspond au traitement qui suit la première acquisition, et la phase de suivi continu qui concerne les acquisitions ultérieures dont le traitement est réduit à des fenêtres pour obtenir un gain de temps d'exécution. Ensuite, l'application de la transformée de Hough est établit pour les deux modes d'opérations et, enfin, des résultats expérimentaux sont présentés et discutés.

**Le quatrième chapitre** traite de l'aspect contribution de la carte de l'environnement au système de vision. En premier lieu, la caméra est modélisée et la matrice de transformation des points objet aux points image est déterminée. Ensuite, cette transformation est utilisée pour construire une scène à partir des données de la carte. Les bords de route sont ainsi localisés sur cette scène et les fenêtres initiales pour déclencher le mode de suivi continu sont déterminées. En plus, en se basant sur les données de la carte, une trajectoire est générée sous forme d'une séquence des coordonnées de points de passage du robot en évitant les obstacles .

Dans le **chapitre V**, nous établissons la base de connaissance visuelle qui permet de formuler un raisonnement sur ce qui a été détecté et ce qui a été prédit. A ce niveau, les bords de route sont sélectionnés et le modèle de la scène établi. Ensuite, à partir du modèle de la caméra et de la transformation perspective des points objet aux points image, nous évaluons les paramètres physiques qui servent au guidage du robot. Ces paramètres sont la position du robot par rapport à l'axe de la route, l'orientation du robot par rapport à l'axe de la route, et la largeur de la route.

Enfin, les résultats expérimentaux de l'application de ce traitement sur des scènes correspondant à un trajet dans un site pilote sont exposés.





***GENERALITES.***

## I.1 - LA VISION ET LA NAVIGATION

La navigation est une opération de base dans le domaine de la robotique mobile qui doit être assurée par tout système en interaction avec des objets physiques de son environnement. Dans les systèmes industriels existant, une telle navigation est typiquement le suivi d'un chemin pré-déterminé dans un repère absolu ou relatif. Les méthodes basées sur ce principe sont efficaces en ce qui concerne les capacités de calcul et simples à analyser et programmer. Par contre, les solutions pré-programmées sont de nature rigide vis-à-vis des variations de la tâche et intolérantes aux changements de l'environnement. Une opération de suivi de joint de soudure pré-défini est mise en échec si les pièces à souder sont à quelques millimètres de la position prévue. Ce constat a conforté le développement de systèmes capables de réaliser une navigation active en utilisant une structure de bouclage. Dans ce contexte, la navigation visuelle a pris une importance particulière, dûe principalement à sa capacité, dans les organismes biologiques, à fournir une quantité très importante d'informations sur les différentes caractéristiques de l'environnement.

La tendance actuelle des concepteurs est de réaliser des systèmes artificiels ayant des capacités similaires à celles de la vision des organismes biologiques comme l'obtention de l'information sur la position, le mouvement, la forme et l'identification des objets. Ceci s'est avéré difficile car extraire l'information visuelle à partir d'images est une tâche très délicate. Ainsi, le cortex visuel d'un animal, qui manifeste un comportement simple contient des millions de neurones dont chacun exécute des opérations qui ne peuvent être simulées sur une machine, que par des milliers d'étapes à la seconde. En outre, aucun système existant ne peut assurer une navigation parfaite dans un environnement réel : car dans la plupart des cas la solution mathématique au problème général n'a pu être obtenue qu'avec certaines simplifications qui ne sont pas rigoureusement valables dans le monde réel. D'autre part, plusieurs applications ont été réalisées avec succès pour un type bien défini d'opération, ce qui a été un handicap pour l'étude des principes généraux de l'application de la vision.

Une façon d'exploiter au mieux les capacités de navigation d'un robot mobile est de lui adjoindre un système de vision. Celui-ci peut fournir deux types d'informations:

l'information de l'image et l'information spatiale. La première peut être stockée sous forme d'image plane. Elle est essentielle dans la distinction et l'identification des objets visibles. L'information spatiale est, par contre, liée à l'espace stéréoscopique, comme la direction, la distance et le mouvement des objets visibles. Elle joue un rôle dominant dans le contrôle du mouvement.

### **I.1.1 - La vision bidimensionnelle**

Le passage d'une scène réelle à son interprétation implique tout d'abord l'acquisition d'une information visuelle, puis l'application aux données qui en résultent d'un certain nombre d'algorithmes pour extraire l'information utile. L'ensemble de ces opérations est généralement décomposé en quatre niveaux selon un ordre chronologique d'application: acquisition, prétraitement, analyse et enfin interprétation ( voir figure I.1.1 ). La plupart des tâches de vision artificielle, pour la robotique du moins, sont décomposables de cette manière.

- Le niveau acquisition regroupe toutes les opérations liées à la saisie et à la quantification des images.

- Le niveau prétraitement regroupe les opérations qui effectuent l'amélioration d'une image détériorée ou la mise en évidence dans une image de composantes particulières utiles aux traitements ultérieurs.

- Le niveau analyse regroupe les opérations d'extraction de l'information utile d'une image prétraitée, puis le codage, l'organisation et le rangement de cette information sous forme symbolique.

- Le niveau interprétation regroupe les opérations dont l'objectif est (en se basant sur la représentation symbolique précédente) la compréhension de la scène en terme de concepts propres à chaque application.

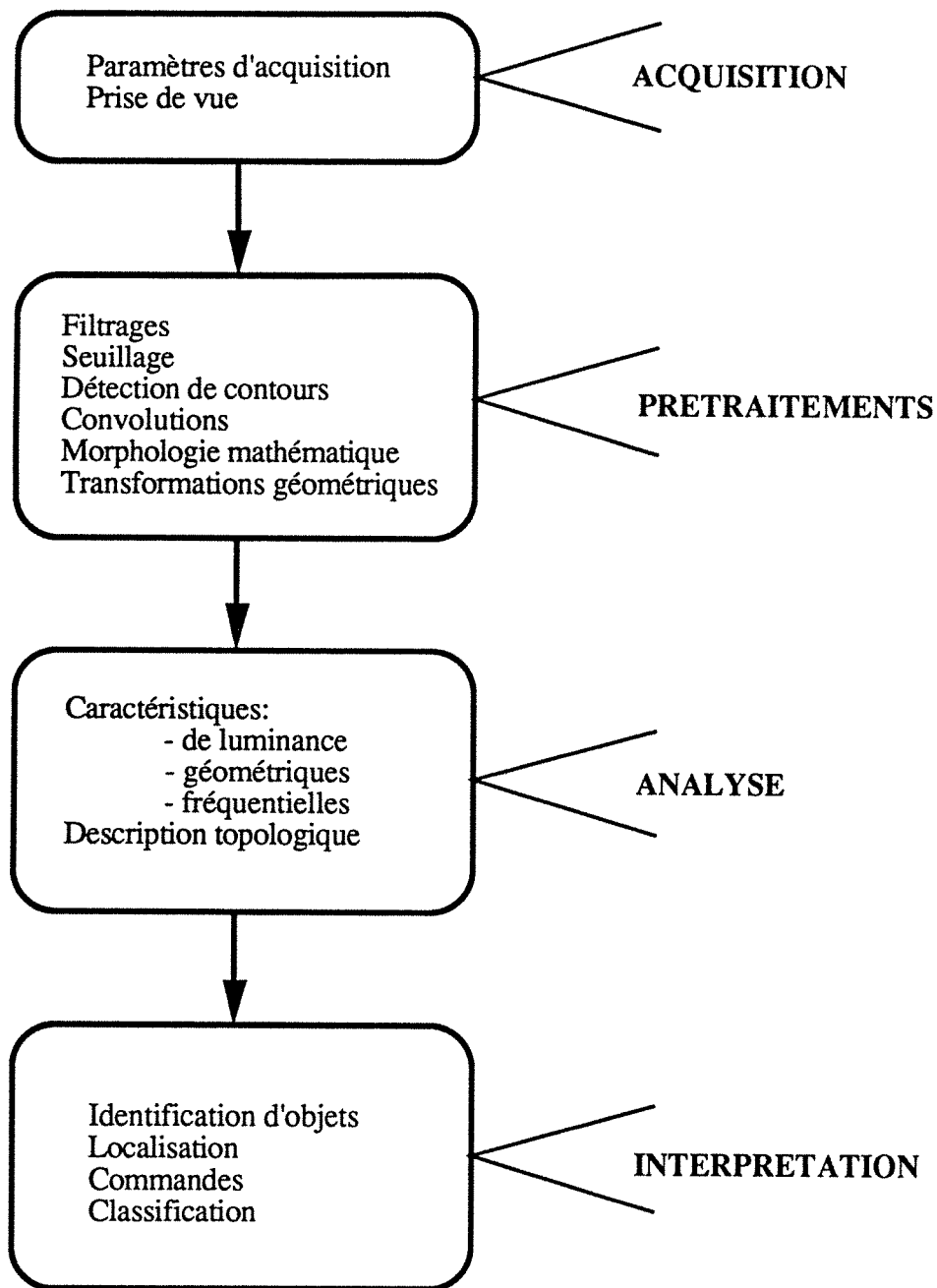


Figure I.1.1 : Les différents niveaux de traitement en vision.

#### **I.1.1.1 - Niveau acquisition**

Une bonne connaissance de l'environnement réel perceptible par la caméra est un facteur essentiel. Il est évident que l'utilisation rationnelle d'une algorithmique qui s'appuie sur une information de type visuel, nécessite la compréhension de la structure de l'environnement. En plus, la qualité et la stabilité des prises de vue déterminent la robustesse des traitements qui suivent, et peuvent même induire une simplification au niveau de l'interprétation sur une classe de scènes. L'environnement extérieur ne peut être constamment conditionné, donc l'effort doit être concentré sur le développement des algorithmes de traitement capables de s'adapter aux imprévus et à la variabilité de la scène.

Les aspects technologiques des chaînes d'acquisition sont aujourd'hui parfaitement maîtrisés, et sont sans cesse améliorées en performance et dotées de puissance plus grande.

L'opération de calibration permet dans le plus simple des cas d'établir une correspondance entre un pixel et une unité de distance physique. Des calibrations plus complexes fournissent le modèle géométrique du capteur dans un repère quelconque. L'intérêt de cette procédure est d'effectuer des opérations d'analyse et d'interprétation bidimensionnelle en s'affranchissant de la distance de la caméra à la scène.

#### **I.1.1.2 - Niveau pré-traitement**

Ce niveau d'opérations est souvent désigné dans la littérature sous les noms de prétraitement ou de vision de bas niveau. Les algorithmes des procédures et leurs implémentations matérielles ont fait l'objet de nombreux développements et recherches. On donne dans le paragraphe suivant une description brève des opérations les plus courantes.

\* Un seuillage transforme une image multi-niveaux en image binaire en affectant la valeur 1 à tous les pixels dont le niveau de gris se trouve dans une certaine plage de valeurs, et 0 au reste. Exemples de seuillage: à seuil simple, à double seuil, à seuils locaux, à seuil déterminé par analyse d'histogramme.

\* Un filtrage est basé sur la théorie du traitement du signal pour éliminer partiellement le bruit d'une image. Il remplace des niveaux de gris de chaque pixels par des valeurs moins bruitées. On trouve dans la littérature une multitude de filtrages: filtres passe-haut, filtres passe-bas, lissage gaussien, filtres médians, filtres obtenus à partir de la transformée de Fourier.

\* Une convolution est la corrélation d'une image avec un masque M de taille typique égale à 3x3, 5x5 ou 7x7. Les applications des convolutions sont nombreuses: calcul des vecteurs gradients et Laplaciens d'intensité lumineuse.

0	-1	0
-1	4	-1
0	-1	0

Laplacien

1	1	1
1	2	1
1	1	1

Lissage

1	0	-1
2	0	-2
1	0	-1

Gradient en x

-1	-2	-1
0	0	0
1	2	1

Gradient en y

\* Une transformation géométrique est une similitude plane appliquée de manière discrète à une image. Les pixels manquants sont, suivant leur position, interpolés ou mis à zéro. Une transformation géométrique a la propriété de conserver les formes, elle est donc intéressante pour normaliser une image afin d'effectuer la comparaison d'un motif avec un modèle de référence.

\* Les opérateurs de morphologie mathématique ( binaires ou multiniveaux) forment une classe particulière de prétraitement. Basés sur une approche mathématique dérivée de la théorie des ensembles, ces opérateurs permettent de faire l'érosion, la dilatation, l'extraction d'une image squelette, et des transformées diverses.

\* Le sous échantillonnage d'images permet d'augmenter la résolution pour obtenir de meilleures segmentations. Les niveaux des pixels sont interpolés comme combinaison linéaire de leur voisins. Ceci permet des détections de contours plus fiables.

\* Les opérations de détection de contours constituent le sous-ensemble le plus intéressant du niveau prétraitement. Ceci est dû au fait qu'elles préparent directement à l'analyse. Les détecteurs de contours binaires sont basés sur les propriétés topologiques au voisinage des pixels. Un pixel de valeur 1 est point de contour si et seulement si il possède un voisin de valeur 0. Des voisinage 4-connexes ou 8-connexes peuvent être utilisés. Les détecteurs de contours multiniveaux sont eux basés sur des opérateurs locaux de différentiation des intensités lumineuses au sein d'une image. Ces opérateurs sont souvent de type gradient ou laplacien. Généralement, l'image obtenue par application d'un détecteur de contours est appelée image des contours. Les détecteurs de contours constituent dans notre cas une étape initiale dans l'extraction des bords de route à partir d'une scène. A ce titre, une discussion plus approfondie sera proposée au chapitre III .

### **I.1.1.3 - Niveau analyse d'image**

L'objectif d'une opération du niveau analyse est de compiler l'information utile en une ou plusieurs structures de données symboliques, avec l'idée de les utiliser pour prendre une décision au niveau interprétation. Parmi les principaux types d'opérations on peut citer : l'extraction des caractéristiques globales de luminance dans une image et l'extraction des caractéristiques colorimétriques.

La représentation des lignes d'une image sous forme symbolique est un outil de construction d'un modèle topologique de la scène. La description topologique d'une image en termes d'entités de base de type ligne, point et région fournit une représentation de l'information pour l'interprétation basée sur des méthodes géométriques. Ces entités sont décrites par leur position et leur structure géométrique. La diversité des approches dans ce domaine est très grande.

#### I.1.1.4 - Niveau interprétation

En se basant sur l'information issue des structures de données fournies par l'analyse d'image, le niveau interprétation identifie et localise les éléments recherchés et prend des décisions. Il faut noter que la plupart des objets réels ont une structure tridimensionnelle. Ainsi pour les identifier, il faut tenir compte de toutes les variables qui peuvent affecter l'intensité d'un pixel appartenant à ces objets. Parmi ces variables on trouve: la position de prise de vue qui se traduit par la distance et l'angle des surfaces de l'objet par rapport à la source d'éclairage, la composition spectrale de la source, les propriétés de réflectance des surfaces de l'objet, les caractéristiques spectrales de la caméra, et enfin le contenu de la scène.

Les processus de décision basés sur l'information contenue dans la scène sont déterminés à ce niveau par des modèles et des méthodes mathématiques. Enfin, ce niveau forme l'aboutissement d'un processus de vision artificielle.

## I.2 - LE ROBOT MOBILE

Un robot mobile est composé d'un véhicule, d'un porteur articulé et d'un effecteur. Dans ce paragraphe nous nous intéressons au véhicule. Celui-ci doit se déplacer d'une position spécifiée à une autre sans assistance externe. Ceci exige une autonomie de mobilité qui entraîne toujours une complexité technologique qui s'ajoute aux problèmes rencontrés dans le cas des robots manipulateurs.

L'autonomie des robots mobiles se base sur la connaissance de l'environnement où ces derniers progressent. Ainsi, l'environnement par ses modifications et ses évolutions permet d'adapter le comportement du robot mobile. En outre, la complexité de l'environnement entraîne généralement la nécessité d'avoir plus de souplesse dans les mouvements du robot et un mécanisme compliqué pour son guidage. En examinant de près la constitution d'un robot, on constate son équivalence à une boucle d'asservissement.



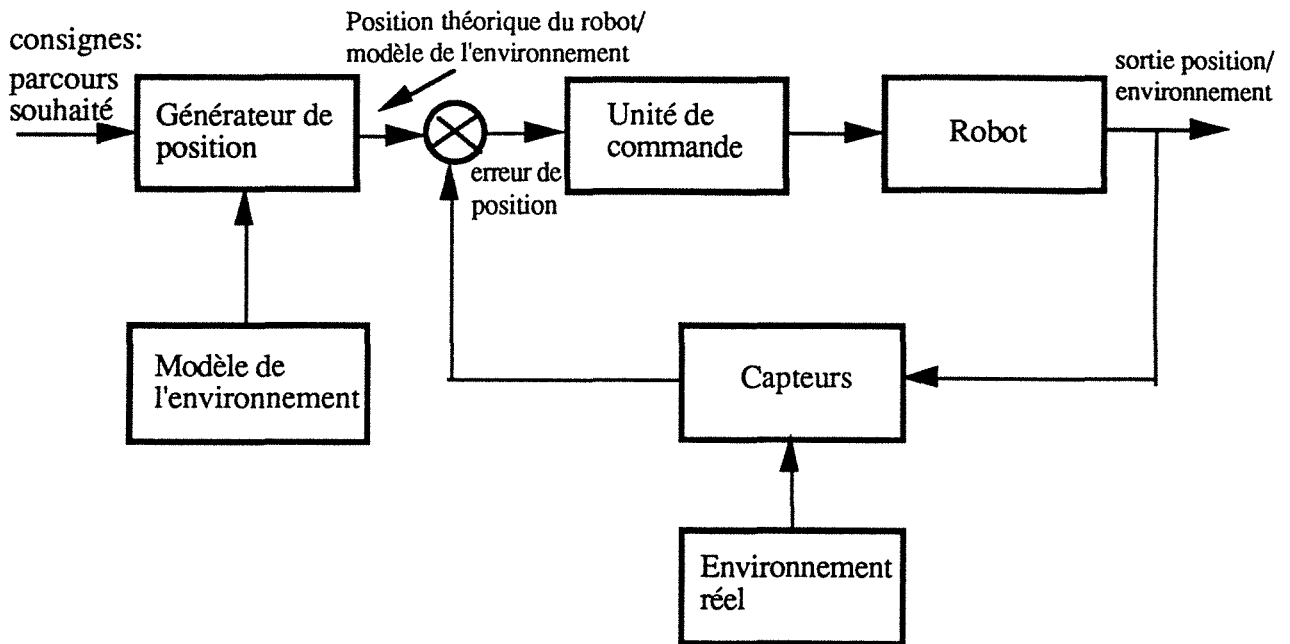


Figure I.2 : Boucle d'asservissement du robot mobile.

La consigne est représentée par les tâches spécifiées par l'opérateur ( trajectoire et parcours ). La structure mécanique motorisée représente le système mécanique à commander. L'unité de contrôle constitue le comparateur , et enfin la chaîne de retour est constituée de capteurs qui permettent d'évaluer la position du robot par rapport à l'environnement réel.

### I.2.1 - Conception des niveaux de traitements

On identifie six niveaux de traitement dans l'architecture logicielle du robot: supervision, planification, modélisation, perception et interprétation , pilotage et commande du robot.

On donne une description brève de chaque niveau:

\* **Supervision** : à ce niveau, la surveillance globale du système est effectuée et les différentes tâches sont spécifiées par l'opérateur.

\* **Planification** : pour exécuter les tâches spécifiées, ce niveau planifie la séquence d'actions à entreprendre pour définir et gérer le traitement approprié .

\* **Modélisation** : pour assurer l'autonomie, le robot doit avoir une compréhension de son univers par l'acquisition et, la manipulation du modèle de son environnement. Ce modèle est basé sur des hypothèses et une représentation paramétrique décrivant géométriquement et qualitativement le milieu extérieur. Il peut être utilisé pour la reconnaissance et la vérification de l'information fournie par la carte de l'environnement.

Niveaux de traitement	Fonctions
Supervision	<ul style="list-style-type: none"> <li>* Interface homme/machine</li> <li>* Spécification des tâches</li> <li>* Surveillance globale du comportement du robot</li> </ul>
Planification	<ul style="list-style-type: none"> <li>* Planification des actions (logiciel )</li> <li>* Simulation</li> <li>* Rétablissement en cas d'échec</li> </ul>
Modélisation	<ul style="list-style-type: none"> <li>* Modélisation de l'environnement</li> <li>* Comparaison des données acquises avec celles de la carte</li> </ul>
Perception et interprétation	<ul style="list-style-type: none"> <li>* Acquisition des données ( vision, odométrie, magnétométrie )</li> <li>* Fusion de données et interprétation</li> <li>* Etude de la compatibilité des données et représentation commune de l'environnement</li> </ul>
Pilotage	<ul style="list-style-type: none"> <li>* Gestion et programmation des activités</li> </ul>
Commande du robot	<ul style="list-style-type: none"> <li>* Commande des actionneurs</li> <li>* Commande du système de locomotion.</li> </ul>

\* **Perception et interprétation** : à ce niveau, l'acquisition des données par les capteurs et leur interprétation est effectuée. L'existence de plusieurs types de capteurs exige une représentation commune de l'information et l'utilisation de systèmes de référence compatibles afin de pouvoir intégrer les données.

\* **Pilotage** : ce niveau est responsable de la cohérence des différentes activités du comportement robot. En fait, il exécute les actions planifiées en les adaptant aux variations des conditions de l'environnement détectées par les capteurs.

\* **Commande du robot** : ce niveau réalise de la commande physique des différents actionneurs et du système de locomotion du robot. Il constitue une interface avec la partie physique du robot afin de libérer les niveaux supérieurs des détails de bas-niveau.

Le niveau de planification utilise les propriétés des tâches spécifiées par l'opérateur ( mission ), les informations sur l'environnement local à partir de la carte, et les caractéristiques et les capacités du robot pour planifier une séquence d'ordres pour accomplir la mission. Cette séquence constitue une décomposition de la mission. Ainsi chaque partie forme un plan local et spécifie le comportement désiré plutôt que l'action exacte du robot qui dépend de l'information détaillée sur l'environnement acquise ultérieurement.

Le niveau de perception assure ainsi l'acquisition de l'information à travers les différents capteurs et construit par fusion de données une description locale de l'environnement. Elle est utilisée par le module de pilotage pour enfin déterminer les commandes de mobilité appropriées. On remarque en identifiant ces champs d'activités, que la communication interne ne se fait pas uniquement entre niveaux adjacents.

### **I.2.2 - Description des tâches**

La conception d'un système autonome doit commencer par la définition de ces différentes tâches qui peuvent être décrites en terme d'environnement et d'objectifs. Ceci détermine donc les conditions qui assurent le succès ou l'échec d'une tâche. En outre, une telle description contient l'information requise pour déterminer les différents éléments de commande, de traitement et de détection, nécessaires à l'accomplissement de la tâche.

### **I.2.2.1 - La tâche**

Ce robot est conçu pour progresser et assurer la surveillance dans un parc urbain de la ville de Nancy. L'environnement de ce parc est très dynamique en raison de la présence du public. La carte du parc a été numérisée et stockée ( voir chapitre IV ) ce qui évite le robot de construire sa propre carte, et améliore les performances du robot. Pour l'exécution d'une mission, les points de départ et de destination sont spécifiés sur la carte, le chemin qui permet d'assurer le parcours avec le minimum de risque est déterminé par le générateur de trajectoire. Tout événement survenant entre temps devra en principe être totalement pris en charge par le système.

### **I.2.2.2 - Les capteurs utilisés:**

Le véhicule mobile doit avoir des capteurs pour percevoir son environnement, un système de locomotion pour se déplacer et des moyens de traitement pour mettre en oeuvre les commandes appropriées aux actions désirées. En plus, pour assurer l'autonomie, le véhicule doit posséder une source d'énergie suffisante pour supporter le traitement, alimenter les capteurs et assurer les différentes activités. Sa structure mécanique doit protéger ses mécanismes contre les conditions défavorables de l'environnement ( chocs, vibration, température élevée, humidité, poussière, etc...).

On a conçu trois modalités de détection pour ce robot :

**\* Le magnétomètre :**

Cet outil sert à mesurer le champ magnétique terrestre servant de référence angulaire à l'orientation du robot. Son inconvénient majeur est sa grande sensibilité aux champs magnétiques ambiants : toute masse métallique a pour effet de modifier le champ magnétique local. Des méthodes de compensation permettent de limiter les influences externes notamment celles dues au robot lui-même.

**\* L'odométrie :**

Le principe de cette technique utilise les informations sur la rotation des roues pour déterminer la position du robot. Dans notre cas cette information est fournie par deux codeurs optiques attachés aux axes des roues motrices. La précision de la localisation est limitée en raison des erreurs créées par:

- Les défauts de planéité du sol : l'odométrie sur un terrain accidenté est difficilement exploitable.
- Le glissement : des roues lors du démarrage des moteurs ou d'un virage.
- L'accumulation des erreurs due à l'intégration numérique des déplacements élémentaires.

Néanmoins, l'odométrie délivre une information instantanée sur la position et possède une précision raisonnable . En plus, c'est une technique peu coûteuse.

**\* La caméra :**

La caméra utilisée est une caméra CCD qui fournit une image de 512x512 pixels ayant une résolution de 8 bits par pixel. Le traitement de l'image acquise par celle-ci permet de déterminer l'orientation du robot par rapport au chemin et d'assurer la tâche de

suivi de route. Les avantages et inconvénients de cette approche feront l'objet des chapitres qui suivent.



**CHAPITRE II**

**ARCHITECTURE LOGICIELLE  
DU SYSTEME DE VISION.**



## II.1 - INTRODUCTION

Ces dernières années, l'utilisation de la vision a été limitée à la reconnaissance, la localisation et l'inspection des éléments stationnaires. Actuellement, l'équipement de traitement d'images a atteint un degré d'évolution où la vision peut être utilisée pour la conduite en temps réel de systèmes aussi compliqués que les robots mobiles. Dans les installations industrielles modernes la mobilité de ces robots s'est avérée nécessaire pour plusieurs fonctions notamment le transport des objets. Les premiers efforts pour mettre au point la navigation des robots mobiles se sont orientés vers des méthodes de filoguidage et d'optoguidage qui restent toujours d'actualité. Mais ces méthodes exigent une modification considérable du milieu de travail et créent des problèmes énormes si les chemins doivent être changés rapidement. Ces raisons ont conduit à orienter les recherches vers des techniques de navigation utilisant des informations provenant de l'environnement naturel du robot. En particulier, on s'efforce de réduire au maximum les connaissances " à priori " sur cet environnement, au profit de connaissances acquises en temps réel. ceci, aussi bien pour la navigation à l'intérieur d'un atelier que dans un milieu naturel ouvert.

Les recherches actuelles sur les procédures de navigation en milieu extérieur concernent deux thèmes principaux :

- 1) Reconnaissance de la route , détermination du chemin à suivre et planification.
- 2) Détection et évitement des obstacles qui peuvent éventuellement apparaître sur le chemin .

Si la connaissance à priori de l'environnement est minimisée, le problème de navigation devient largement tributaire de la capacité des moyens de détection. L'utilisation d'un capteur visuel en collaboration avec d'autres types de capteurs est devenue à nos jours une idée courante. C'est dans cette optique que nous présentons

dans ce chapitre le principe, et l'architecture logicielle du système de vision utilisé pour assurer le guidage de notre robot mobile dans les allées d'un parc urbain.

## II.2 - APERÇU SUR LA VISION EN ROBOTIQUE MOBILE

Les robots mobiles forment un sous-ensemble de la robotique, dépendant au plus haut point des techniques de vision artificielle. On peut citer comme types d'applications: les robots de convoyage (déjà très actifs dans les milieux industriels mais souvent filoguidés), les robots de nettoyage, les robots de surveillance, les robots d'intervention en milieu hostile. Les applications militaires s'orientent aussi beaucoup vers les robots mobiles notamment pour les missions de déminage.

Des prototypes de robots mobiles capables de perception visuelle existent dans certains laboratoires. En grande partie, ils sont conçus comme des supports de recherche parmi lesquels on cite: Shaky du SRI, Allen du Massachusetts Institute of Technology (MIT), Terregator et Neptune de l'Université de Carnegie-Mellon, ROME0 (RObot Mobile Expérimental numéro 0) du groupe de recherches RMA (Robots Mobiles Autonomes) du LRP, et HILARE du L.A.A.S.

Shaky du SRI était le premier robot mobile équipé d'un système de vision ayant des capacités très limitées. Il a été conçu comme prototype pour étudier les problèmes de vision sur robot mobile. Ces dernières années les travaux à SRI International se sont concentrés sur la reconnaissance et la modélisation des objets, et la modélisation de l'incertitude dans différentes représentations.

Le robot JPL [THO-77] utilise l'information visuelle pour construire des modèles de terrain polygonaux afin de déterminer le chemin optimal. Malheureusement, le projet s'est arrêté avant que ce système ait été achevé. Le Cart de Stanford applique l'opérateur Moravec [MOR-83] sur une série de neuf images stéréo prises chaque 50 cm par une

caméra en mouvement. Cet opérateur permet d'extraire les éléments pertinents et leur position tri-dimensionnelle à l'aide d'un algorithme (stéréo). Commandé à distance par un ordinateur, il parcourt 20m en cinq heures, avec une précision de 20%. La sélection des éléments essentiels à la navigation dans des milieux encombrés était bonne, le seul problème est que son modèle de l'environnement n'était pas assez riche et compréhensif.

Crowley [CRO-85] a présenté un système de navigation pour un robot mobile équipé d'un capteur rotationnel ultrasonique. Ce système est basé sur un modèle dynamique de l'environnement local appelé modèle local composé. Ce dernier intègre les informations issues du capteur ultrasonique, du capteur de contact, et du modèle globale acquit lors du mouvement du robot dans son environnement. La position estimée du robot est corrigée par la différence entre la position observée à l'aide du capteur et les symboles correspondants du modèle local composé. Ce système est conçu pour une navigation efficace dans un milieu interne connu tel que le milieu industriel.

Tsugawa et al [TSU-79] ont présenté un véhicule autonome se déplaçant jusqu'à 30 km/h en se basant sur la stéréovision, mais sa perception de l'environnement était très limitée ce qui a réduit son utilisation à des milieux très simples. Inigo [INI-84] a présenté un système de vision pour robot mobile destiné à l'exploitation en milieu industriel. Ce système utilise la détection de contours, la perspective inverse, et la recherche des lignes droites par la transformée de Hough. Ces outils permettent de localiser le chemin sur l'image, de construire un modèle de celui-ci, et d'utiliser un algorithme stéréo, appelé corrélation de scène, basé sur le mouvement, pour détecter les obstacles. Le système de vision Fido [TUR-88] emploie la stéréovision pour localiser les obstacles par corrélation hiérarchique de points sélectionnés par un opérateur approprié. Bien que son modèle de l'environnement soit constitué seulement des points en 3-D du chemin qu'il suit, il a réussi à naviguer d'une manière autonome dans des environnements assez complexes.

La conception et la commande des robots mobiles au laboratoire L.A.A.S de Toulouse a commencé depuis 1977. Parmi les travaux réalisés on note ceux de Saint Vincent [VIN-86] en perception multi-capteurs pour le robot HILARE. ce système est doté d'une paire de caméra stéréo, d'un télémètre laser et d'un module de vision dynamique pour le recalage et le suivi de chemin. D'autre part, Noreils et Chatila ont

développé pour le robot HILARE une architecture de commande basé sur la décomposition hiérarchique des fonctions [NOR-89].

En Allemagne un projet de robot mobile vise à réaliser le guidage d'un véhicule autonome à grande vitesse [DIC-85]. Le système se base sur un algorithme de vision très rapide pour effectuer le suivi de route en temps réel.

Le projet ALV développé aux U.S.A est destiné à la présentation des derniers acquis et à faire progresser la connaissance dans les domaines du traitement d'image, de l'intelligence artificielle, des architectures avancées et de la navigation autonome. Ce projet est mené par plusieurs groupes de recherches notamment ceux de l'Université de Maryland [SHA-88], [DAV-86], et [MOI-88], l'Université de Carnegie-Mellon [TUR-88], [THO-88], et l'Université de Martin Marietta [TUR-87]. Les recherches entamées actuellement en vision portent sur : la modélisation des objets, la stéréovision, la texture, l'analyse et la détection du mouvement et la reconnaissance des objets. Waxman et al [WAX-87] ont présenté un système modulaire de navigation visuelle qui incorpore des modules de géométrie et de traitement d'images pour extraire les éléments dominants d'une scène. Hebert et Kanade ont analysé des scènes extérieures en utilisant les données fournies par un télémètre laser. Ils ont aussi proposé des méthodes pour le pré-traitement des données de profondeur du champ de vision, l'extraction des caractéristiques tri-dimensionnelles, l'interprétation de la scène, la construction d'une carte, et la reconnaissance des objets. Une description de ce projet et des recherches relatives à la vision sont présentées dans [THO-88] et [HEB-89].

Le classement établi dans les paragraphes précédents pour les différentes réalisations dans le domaine de la robotique mobile, a permis de mettre en relief l'approche de chacune d'elles. En effet certaines réalisations nécessitent une exploitation de plusieurs techniques.

Le processus d'interprétation visuelle d'une scène pour un robot mobile destiné à évoluer dans un univers réel même restreint, reste en général extrêmement complexe. L'objectif de ce processus est d'aboutir à une modélisation, à un instant donné de cet

univers, et ce en fusionnant progressivement l'information d'un modèle antérieur avec celle fournie par les capteurs . Cette modélisation doit être suffisamment robuste pour rendre possible la génération automatique des trajectoires, l'élaboration d'heuristiques de navigation et le calcul assez précis de la position du robot dans le repère de l'environnement. Les solutions envisagées sont souvent basées sur la fusion multi-capteurs. Ainsi les procédures décisionnelles employées doivent être suffisamment puissantes pour avoir la capacité de gérer les différentes représentations, et assurer une cohérence entre les différents niveaux de commande.

### II.3 - ARCHITECTURE LOGICIELLE DU SYSTEME DE VISION

En général, quand la vision est utilisée comme partie d'un système informatique, l'architecture de ce dernier influence la conception des différentes composantes du système de vision. Néanmoins, les caractéristiques suivantes restent toujours désirables quelque soit celui-ci :

- **Opération en temps réel** : ceci impose une contrainte sur le temps de réponse du système et donc impose une architecture matérielle adaptée. On remarque que la totalité des architectures matérielles conçues pour les robots mobiles ont évolué avec le développement des microprocesseurs. En effet, depuis l'apparition de ces derniers, les robots mobiles sont devenu autonomes du point de vue ressources de calcul. Le développement des architectures matérielles est essentiellement dû à la modularité puisqu'elle permet au système d'avoir plusieurs composants écrits en différents langages et exécutés sur un matériel spécialisé.

- **Généralité et adaptabilité** : le système doit s'adapter aux variations de l'environnement, et être opérationnel sur tout type de terrain. Parmi les variations de l'environnement , on note la variation du climat ( variation d'éclairage ), la position du soleil par rapport au champ de vision de la caméra, la présence éventuelle des objets de tailles différentes sur le chemin, etc... .

- **Robustesse** : le système doit maintenir ses performances tout en utilisant des informations très bruitées dans les conditions les plus sévères de l'environnement: perte de contraste, passage d'une zone trop éclairée à une zone trop sombre, disparition des bords de route, etc... .

En essayant de tenir compte des exigences ci-dessus, une architecture logicielle du système de vision a été proposée comme indiqué sur la figure II.3.1.1. Celle-ci fait partie intégrale du système de commande du robot mobile, et peut affecter et être affectée par les performances globales du système.

Dans cette architecture on trouve plusieurs modules qui mettent en oeuvre différents algorithmes dont les fonctions sont :

- \* Planification.
- \* Supervision.
- \* Exploitation de la scène.
- \* Interprétation géométrique.
- \* Base de connaissance.
- \* Cartographie.
- \* Prédiction de scène
- \* Génération de trajectoire.
- \* Pilotage

Une description brève de chaque algorithme est donné dans le paragraphe suivant.

### II.3.1 - FONCTION DES ALGORITHMES

**\* Planification :**

Ce niveau constitue une interface entre l'opérateur et le système de vision. En fait, actuellement la fonction de planification est très simple, elle se résume à la spécification des objectifs sous forme d'une séquence de commandes au niveau supervision. Un niveau de planification plus évolué serait capable d'assurer la simulation, la détection des erreurs éventuelles du système, le diagnostic, et la reprise dans le cas des échecs ou des situations imprévues.

**\* Supervision :**

Cet algorithme prend en charge l'initialisation du système de vision et maintient le flux des commandes et des informations à travers les différents modules du système pour établir un comportement requis, par exemple : le suivi de route. L'initialisation concerne en grande partie le côté logiciel. Ainsi durant cette phase l'espace mémoire est configuré pour permettre de stocker l'image acquise, et les images qui résultent du traitement intermédiaire dont le nombre varie en fonction de la complexité de la scène. Du côté matériel, l'initialisation agit sur l'orientation de la caméra qui, généralement, doit pointer dans la direction de la route, ainsi que sur celle du robot.

A ce niveau aucun calcul n'est fait. La fonction principale est la coordination des activités pendant l'exécution d'une tâche. On note que l'information issue de certains modules ne sera pas modifiée à ce niveau mais uniquement transférée éventuellement avec d'autres types d'informations acquises auparavant vers d'autres modules. Ceci simplifie considérablement le fait d'ajouter ou de supprimer une fonction au système de vision car la modification qui en résulte est entièrement localisée au niveau supervision.

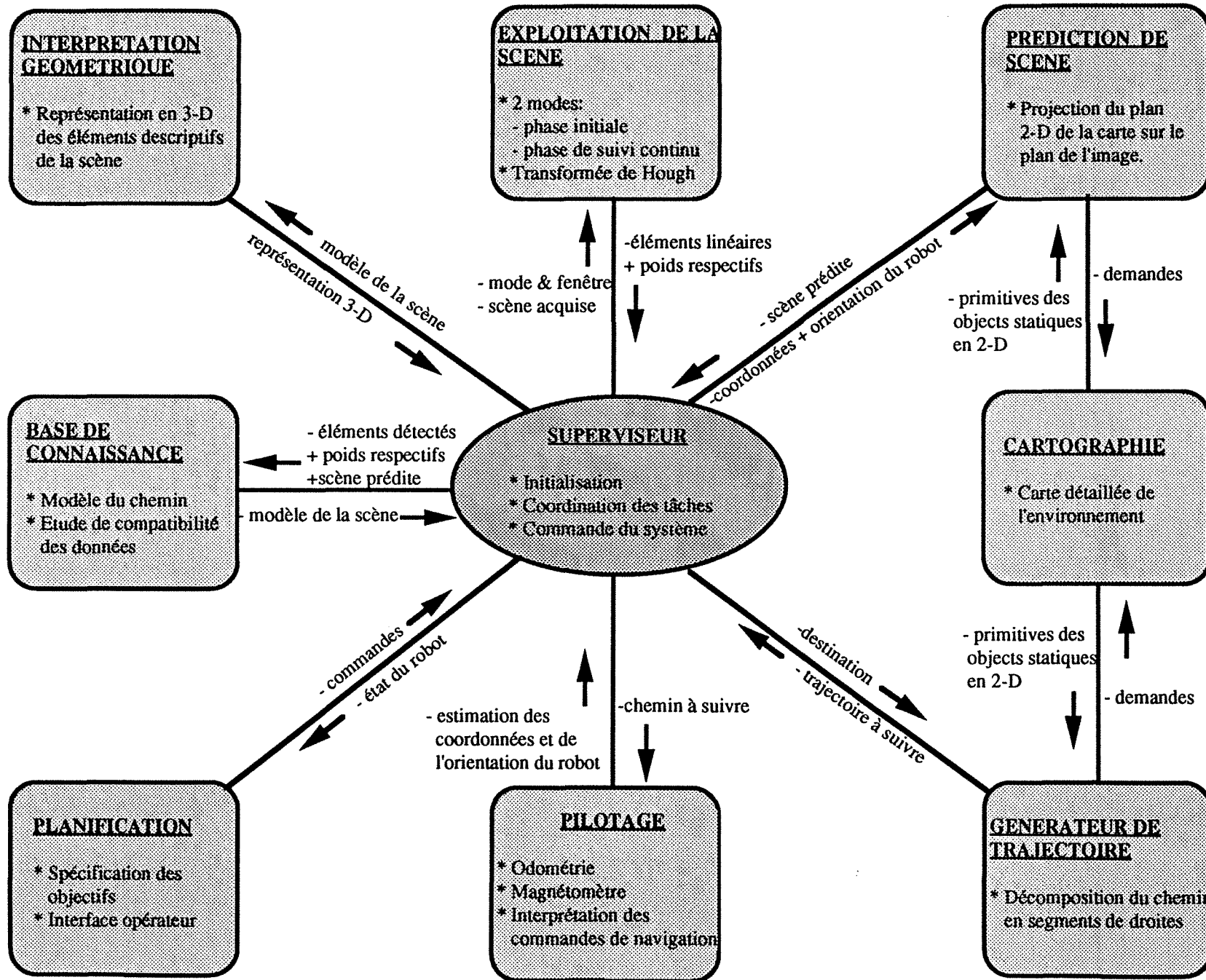


Figure II.3.1.1 : Architecture du système de vision.



*\* Exploitation de la scène :*

Ce niveau fonctionne suivant deux modes différents : phase initiale et phase de suivi continu . La décomposition du travail en deux modes est dû aux exigences de la navigation autonome durant la phase d'initialisation. Le traitement utilisé dans la phase de suivi continu est conçu pour permettre de réduire au maximum le temps requis à l'extraction des éléments pertinents de la scène, ainsi la vitesse du robot se trouve augmentée.

*- Phase initiale :*

L'image entière est traitée dans ce mode pour extraire les informations nécessaires à la localisation du robot par rapport à la route.

*- Phase de suivi continu :*

Le traitement se fait ici seulement sur des fenêtres où il y a une forte probabilité de trouver un élément linéaire correspondant à un bord de route. Ceci réduit considérablement le temps d'exécution.

Dans les deux phases, la détermination des paramètres de l'élément détecté se fait par l'application de la transformée de Hough. Cette dernière permet de localiser sur une image binaire les lignes droites ayant une orientation donnée. Ainsi à chaque image binaire est associée une description paramétrique du segment de droite dont l'orientation est dominante . Enfin, la sélection des bords de route parmi les éléments linéaires détectés est effectuée à partir d'une base de connaissance, par comparaison avec le modèle de la scène et l'étude de compatibilité des données. ( voir chapitre V ).

**\* Interprétation géométrique :**

Ce module produit à partir du modèle de la scène une description tridimensionnelle des objets du modèle. Il réalise une projection des bords de route sur le plan du terrain afin d'évaluer le parcours et de contribuer au calcul de la trajectoire et au recalage du robot.

**\* Base de connaissance :**

L'objectif de la base de connaissance visuelle est de construire le modèle de la scène en utilisant les caractéristiques géométriques déduites des éléments linéaires fournis par l'exploitation de la scène. C'est à ce niveau que les défaillances de l'exploitation de la scène liées à sa variabilité et à d'autres sources d'erreurs sont détectées et corrigées en faisant appel à la géométrie et à des connaissances à priori. En plus, les contraintes temporelles peuvent être utilisées à ce niveau pour lever l'ambiguïté entre des images successives et prédire des modifications éventuelles.

**\* Cartographie :**

Son rôle est de stocker une carte détaillée de l'environnement. Pour simplifier l'accès et la manipulation des différents objets cartographiés, la carte est présentée sous forme d'une séquence d'images contenant l'information sur la position et la hauteur des objets. Les méthodes d'accès et d'utilisation sont traitées dans le chapitre IV .

**\* Prédiction de scène :**

Pour exploiter les données de la base de cartographie, cet algorithme génère une carte virtuelle suivant la configuration robot-caméra et la positionne sur la carte réelle

suivant les coordonnées spécifiées par le système de localisation. La carte virtuelle forme donc le champ de vision du robot. Il projette ensuite les pixels formant les primitives des objets statiques appartenant à la carte virtuelle sur le plan de l'image. Par conséquent, cette projection construit une scène qui devrait se rapprocher de l'image acquise réellement sur le terrain. La méthode de construction est détaillée dans le chapitre IV.

**\* Génération de trajectoire :**

Une fois la destination spécifiée par le niveau supervision , ce module accède à la carte de l'environnement et étudie les informations servant à définir un chemin menant à cette destination et la présence d'obstacles statiques. En fonction de cette étude, le chemin est décomposé en plusieurs segments de droites tels que la trajectoire obtenue présente pour le robot un minimum de manoeuvres à effectuer.

**\* Pilotage :**

Ce module est responsable de la transmission des données fournies par les différents capteurs vers le superviseur. En plus, il interprète les commandes de navigation en signaux capables d'asservir directement le système de locomotion du robot.

### II.3.2 - COORDINATION ENTRE LES DIFFERENTES TACHES

Pour que le robot manifeste le comportement requis, il faut que les capacités logicielles et matérielles du système soient exploitées dans un ordre bien défini. Dans la figure II.3.1.1, on a indiqué le flux des données et des commandes entre les différentes parties pour assurer la tâche principale de la navigation autonome: le suivi de route.

Afin de ne pas induire le système de vision en erreur lors du début de la navigation, le robot doit être placé à peu près au milieu de la route et orienté dans la direction de celle-ci. Quand l'opérateur ordonne le départ, le superviseur entame l'initialisation logicielle et matérielle du système et informe le niveau de planification de l'état du robot pour que celui-ci lui transmette la première commande de la séquence .

Généralement la première commande pour entamer la navigation par vision est : trouver la route. Ainsi le superviseur sélectionne le mode : phase initiale, il déclenche ensuite l'acquisition d'une image de l'environnement, et informe le niveau exploitation de la scène. Ce dernier traite l'image entière pour extraire les éléments linéaires en se basant sur l'orientation du gradient du type Sobel.

A l'aide de la transformée de Hough, à chaque élément linéaire est associée une description paramétrique et un poids indiquant le degré de linéarité de l'élément. Jusqu'ici le robot reste toujours immobile dans une position initiale spécifiée au système par l'opérateur. Ayant les coordonnées de la position initiale du robot ainsi que son orientation, le superviseur initialise l'algorithme de prédiction de scène. Ce dernier accède à la carte détaillée de l'environnement pour en retirer les informations délimitées par la carte virtuelle qui est déterminée par la configuration robot-caméra. Les données de la carte seront alors projetées sur le plan de l'image pour construire une scène .

Une fois la scène prédite, les éléments linéaires détectés et leurs poids respectifs sont communiqués à la base de connaissance. A ce niveau, un raisonnement est établi sur la compatibilité entre les poids des éléments linéaires fournis par la transformée de Hough et les caractéristiques des bords de route à partir de la scène construite. Enfin, les bords de route sont sélectionnés en se basant sur un choix maximisant le taux de compatibilité entre les éléments détectés et ceux prédits.

Les bords de route étant déterminés, le niveau base de connaissance construit le modèle de la scène et le transmet au superviseur. Ce modèle, décrivant la forme du chemin observé, est un des produits essentiels du système de vision après le traitement de chaque image. Il contient un enregistrement de la position et de l'orientation du robot par rapport à la route à l'instant de l'acquisition. Une fois le superviseur informé de l'établissement du modèle de la scène, il demande une interprétation géométrique de celle-ci. Le module d'interprétation géométrique converti le modèle de la scène en une représentation tri-dimensionnelle des objets de la scène. La représentation se fait par projection inverse des bords de route sur le plan du terrain. Mais ce problème de projection inverse admet une infinité de solutions, ce qui a nécessité de faire des hypothèses et d'imposer des contraintes comme la planéité et la continuité de la route (voir chapitre IV) . Cette description géométrique en 3-D permet, en fait, de faire des recalages du robot à l'instant de l'acquisition. Le recalage se fait par fusion de l'information obtenue à partir de la scène prédite, la représentation 3-D et les données fournies par le module de pilotage ( odomètre, magnétomètre).

Le robot étant localisé par rapport à la route, le superviseur spécifie au générateur de trajectoire la destination à atteindre afin que celui-ci accède à la cartographie et examine la région à traverser. Ainsi, en fonction de l'aspect de cette région, le générateur de trajectoire décompose celle-ci en reliant le robot à sa destination par un ensemble de segments de droite. Le superviseur sélectionne alors le premier segment et le transmet au module de pilotage qui l'interprète en commandes de mouvement. Ces commandes agissent sur le système de locomotion, et le robot entame enfin la navigation. D'une manière périodique, le module de pilotage informe le superviseur de la distance parcourue et de l'orientation actuelle du robot. Dès que le premier segment est achevé, le superviseur commande une nouvelle acquisition d'image et passe en mode : phase de suivi continu.

Dans le mode: phase de suivi continu, le superviseur commence par spécifier la position et l'orientation actuelles du robot au module prédiction de scène. Celui-ci construit la scène et positionne deux fenêtres en bas de l'image à travers lesquelles les bords de route entrent dans la scène. En conséquence, le traitement établi au niveau

exploitation de la scène sera limité aux fenêtres. Pour pouvoir discerner les bords de route l'ensemble des fenêtres doit être placé de telle façon que le maximum des bords passe à travers ces fenêtres. Par conséquent, le positionnement de chaque fenêtre dépend des résultats de traitement de la fenêtre précédente. Une fois que la position de la nouvelle fenêtre atteint un certain pourcentage de la surface de l'image, le traitement s'arrête et la transformée de Hough est appliquée. L'espace dans lequel est appliquée la transformée de Hough est considérablement réduit car l'orientation des bords de route dans la fenêtre est approximativement connue. Ainsi, le système reprend la même procédure, sans toutefois refaire la prédiction de scène qui a été déjà faite, jusqu'à l'atteinte de la destination ou à la rencontre d'une situation imprévue ou une instruction d'arrêt.



**CHAPITRE III**

*EXPLOITATION DE LA SCENE.*



### III.1 - INTRODUCTION

Le problème principal que devra résoudre un système de vision en robotique mobile est d'assurer la perception correcte de l'environnement pour permettre une navigation efficace. La robustesse de la perception dépend essentiellement de la robustesse de l'algorithme d'extraction des éléments pertinents de la scène. La complexité des systèmes de navigation est généralement fonction de la complexité des chemins à suivre. Leur tâches typiques sont le suivi de route, la détection et l'évitement d'obstacles, la détection des balises, la construction de cartes décrivant les formes des différents chemins à suivre, et enfin la navigation dans des milieux complètement inconnus. En général, le système de vision doit interpréter les informations fournies par la caméra et éventuellement par d'autres capteurs de différents types pour produire des descriptions symboliques des caractéristiques du monde entourant le robot mobile. La navigation des robots autonomes a suscité un intérêt considérable ces dix dernières années [CRO-85],[DAV-86],[HAR-87],[MAT-90].

Actuellement, quelques systèmes utilisent des modules de reconnaissance de formes pour établir une compréhension complète de la scène [DHO-89], [CRI-91]. En pratique, dans beaucoup de cas, cette compréhension est basée sur la détermination des bords de la route puisqu'ils prennent une importance primordiale dans la navigation des véhicules autonomes.

D. Brzakovic et L. Hong ont développé un algorithme pour détecter les bords de route dans une séquence d'images [BRZ-89]. Leur approche peut se résumer à la modélisation de la route par des fonctions splines et à la détection des bords de celle-ci par un filtre récursif linéaire. Celui-ci établit une recherche de point du bord autour d'une courbe de référence déterminée par le traitement de la première image. L'algorithme traite une séquence d'images sans connaissance a priori du contenu de celles-ci. Le groupe de recherches de l'université de Carnegie-Mellon [THO-88], [CRI-91] utilise le traitement de la couleur pour détecter les bords. Cette approche est basée sur une classification des pixels d'une image couleur en deux catégories suivant qu'ils appartiennent ou pas à la route. Le problème majeur de cette technique est que les paramètres couleur changent

d'une manière considérable en fonction des conditions extérieures. Le système de l'université de Maryland [DAV-86], [WAX-87], [MOI-88] est basé sur la segmentation de la couleur et l'extraction de figures linéaires dans l'image. Ces analyses fournissent deux représentations indépendantes de l'information contenue dans l'image. L'extraction des éléments linéaires est obtenue à partir du groupement des pixels ayant une orientation du gradient similaire. L'application de cette technique est limitée au cas des scènes sans intersections, arbres, et autres objets qui se manifestent par un caractère dominant sur la scène.

Le présent travail s'inspire de cette dernière technique, et contribue à la généralisation de la détection des bords dans des scènes variables contenant une multitude d'objets .

En fait, pour que le robot détermine sa position absolue dans son univers durant la phase d'initialisation, le système de vision doit rechercher les éléments utiles à son repérage . La localisation s'effectue donc de manière relative par rapport à ces éléments. L'analyse dans ce cas concerne l'image entière, et nécessite des traitements importants en volume et en temps .

Une fois ces éléments localisés, le système de vision peut prédire leur position avec un certain degré de précision dans la séquence d'images suivante. Ce processus nous permet de réduire le temps d'exécution en réduisant l'analyse à une portion du champ visuel jugée suffisante au maintien de la navigation. Cette méthode de traitement est particulièrement efficace quand les ressources de calculs sont limitées, ce qui est souvent le cas. Ces deux méthodes d'analyse déterminent deux modes de fonctionnement du système de vision : la phase initiale et le suivi continu.

Actuellement , la plupart des algorithmes qui ont été mis en oeuvre utilisent un traitement en niveau de gris et la définition d'un modèle mathématique décrivant le chemin à suivre.

En général, le problème de détection des bords de route reste difficile à cause de la complexité des scènes, de leur variabilité, et du temps d'exécution requis.

### III.2 - PHASE INITIALE

Une fois le robot placé sur un chemin du parc, le système ne possède aucune information sur la situation du véhicule par rapport au chemin pour pouvoir débiter sa tâche de suivi de route. Après la première acquisition, l'image est traitée complètement pour construire un modèle de la partie visible de la route. Dans ce travail, on présente un algorithme, décrit par la figure III.2.1, pour la détection des bords de la route à partir de la première image acquise en phase initiale. Cette détection permet en fait d'estimer la position et l'orientation relatives actuelles du robot par rapport à la route et d'initialiser la phase de navigation autonome. Le traitement utilisé ici repose sur la vérification de deux hypothèses concernant les bords de route:

- Un bord de route peut être approché par une droite ou des segments de droites.
- Parmi l'ensemble des droites extraites de l'image, il existe toujours une ou plusieurs droites décrivant un bord de route.

Les différentes étapes de cet algorithme sont traitées en détail dans les paragraphes qui suivent.

Durant la phase initiale, le robot doit rester immobile et placé de telle façon que la caméra perçoive des éléments de bords de route. Le traitement durant cette phase fournit au système de guidage deux types d'informations:

- L'erreur d'orientation de l'axe du robot par rapport à l'axe de la route.
- L'erreur de position de l'axe du robot par rapport à l'axe de la route.

Ces données permettent de corriger la position du robot afin d'asservir sa trajectoire.

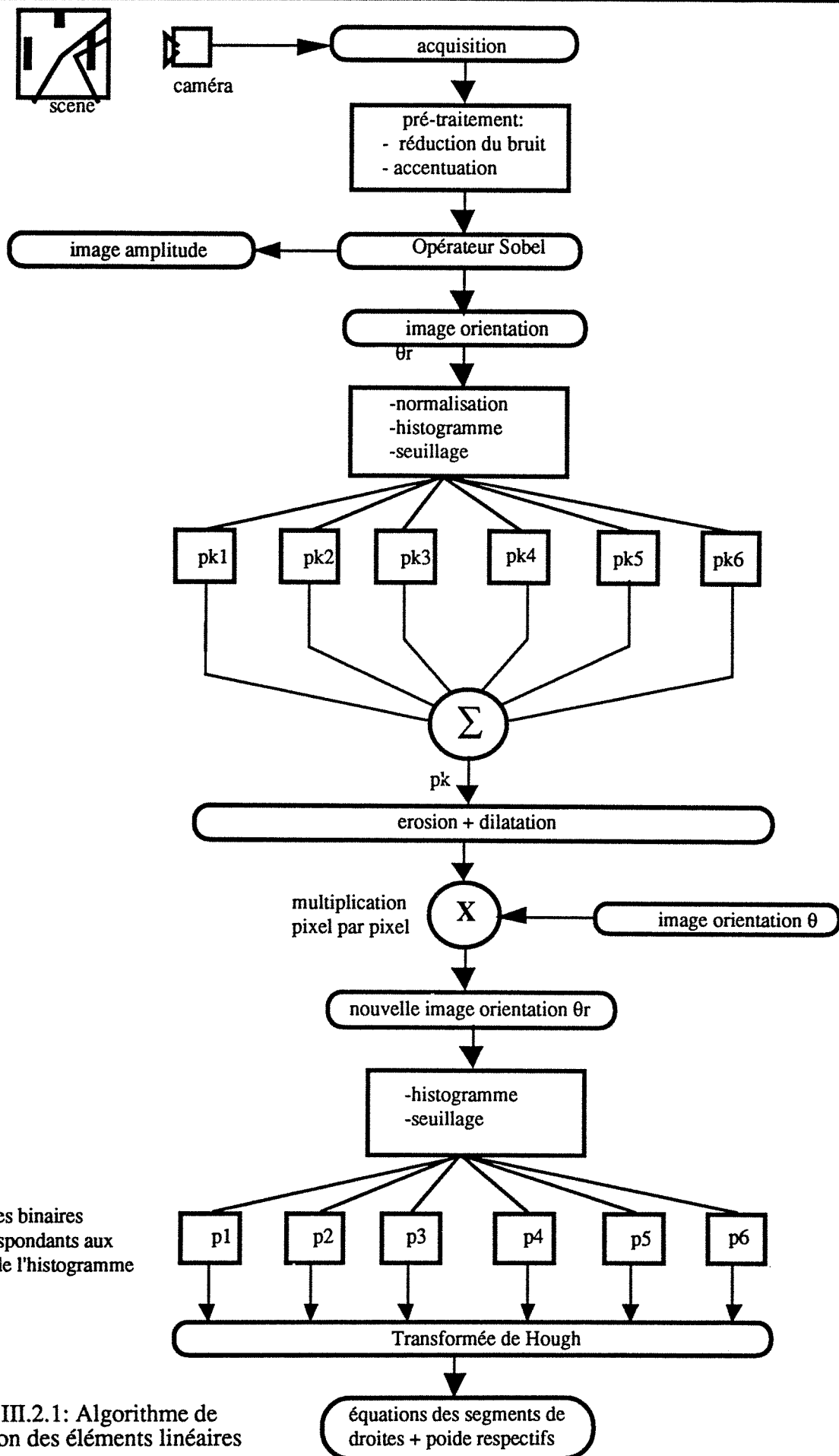


Figure III.2.1: Algorithme de détection des éléments linéaires

Cette méthode est décrite en deux parties. La première partie traite de l'extraction des caractéristiques linéaires de la scène à partir de l'orientation du gradient. La deuxième montre l'application de la transformée de Hough aux images contenant les éléments détectés et le processus d'affectation des poids.

### III.2.1 - ACCENTUATION ET EXTRACTION DES ELEMENTS DOMINANTS DE LA SCENE

Dans la majorité des scènes prises dans le parc urbain, les bords des chemins sont plus ou moins linéaires. En fait, une scène contient généralement plusieurs éléments qui possèdent un certain degré de linéarité comme les troncs d'arbres, les grillage, les bancs, les murets, les ombres, etc... . Parmi toutes ces formes il existe quelques paramètres qui diffèrent considérablement comme l'orientation, la longueur, la position dans la scène et enfin l'épaisseur de l'élément. En utilisant ces différents paramètres on peut extraire les éléments dont les caractéristiques se rapprochent à celles des bords de route et qui sont nécessaires à la navigation.

Les images analysées présentent un bruit aléatoire qu'il est impératif de réduire. Un lissage par un masque de convolution de taille 5x5, permet de réduire le niveau du bruit en préservant une netteté acceptable de l'image . Des dimensions de voisinage plus importantes dégraderaient les détails contrastés sur l'image, comme des lignes ou des contours.

Une des caractéristiques principales des bords de route dans une scène est qu'ils sont localement linéaires et possèdent approximativement la même orientation ou une orientation symétrique. En fait, dans la même scène il peut y avoir plusieurs autres éléments qui ont les mêmes propriétés, ce qui rend le problème de détection des bords très complexe. D'une manière générale pour réduire les possibilités d'erreur, on focalise la recherche dans des zones de forte probabilité de rencontrer un bord. Ces zones correspondent, en général, aux cotés gauche et droit de la moitié inférieure de la scène. La

réussite de cette méthode repose sur le degré de visibilité des bords dans cette partie de la scène.

### III.2.1.1 - Pré-traitement et formation de l'image des orientations

Un pas essentiel dans la transformation de l'information visuelle en primitives intermédiaires, utiles au processus d'interprétation, est l'extraction des lignes ou contours à partir d'une scène. Ceux-ci constituent généralement une variation locale significative de l'intensité lumineuse dans une image. Dans la plupart des cas la représentation d'un élément linéaire est basée sur le groupement d'un ensemble de segments de droites dont l'orientation relative de chacun est soumise à une contrainte. Ainsi, les courbes peuvent être raisonnablement approchées par une séquence de segments de droites.

Dans les scènes naturelles qu'on a rencontré, les bords de route forment souvent une discontinuité locale ou un changement rapide d'un paramètre comme celui de la luminance ou de la texture. Théoriquement, ces variations peuvent être détectées par un opérateur local dont les dimensions spatiales sont nettement inférieures aux dimensions de l'image.

En 2-D un bord idéal serait une discontinuité brusque des valeurs du paramètre descriptif dans une direction perpendiculaire à l'orientation spatiale du bord. Etant donné qu'un bord idéal est tellement rare dans les scènes naturelles, on adoptera la définition de Pratt et Fisher [PRAT], qui estiment que l'amplitude de la variation du paramètre descriptif à travers le bord est normalement distribué sur une surface. Ainsi, en plus de sa longueur, un bord possède une largeur à mesurer perpendiculairement à son orientation. Par contre [BUR-86] et [KAH-90] supposent qu'un élément linéaire ayant une largeur est un couple de bords parallèles ayant des contrastes opposés.

Dans les scènes naturelles qu'on a rencontré, les bords de route forment souvent des variations d'intensité lumineuse irrégulières et très larges. En outre, puisque les bords sont le siège de phénomènes de haute fréquence spatiale, le bruit accompagnant les images se trouve accentué par les opérateurs de détection de contours. Un des problèmes

difficile à maîtriser est que ces opérateurs ne font aucune distinction entre les bords engendrés par des objets, des ombres, des arbres, des changements d'orientation ou de réflectance d'une surface, etc... . Plusieurs approches ont été proposées, notamment la plus connue celle de Marr-Hildreth [MARR] qui ont employé un opérateur de passage par zéro défini par le Laplacien d'une Gaussienne. Néanmoins, puisque les phénomènes de hautes et basses fréquences ont une réponse optimale aux opérateurs de tailles différentes, la taille appropriée de l'opérateur doit être déterminée pour chaque région de l'image. Canny [CAN-83] a montré que le filtre optimal 1-D selon son critère est une combinaison de 4 exponentielles qui peuvent être approximées par la dérivée première d'une gaussienne. En 2-D, il propose d'utiliser une combinaison de ces filtres. Deriche [DER-87], [DER-90] a proposé un opérateur de détection de contour basé sur le critère de Canny, qu'il a implémenté récursivement, travaillant ainsi sur une fenêtre de taille infinie. Son modèle de contour est le même que celui de Canny. Castan et al [CAS-89] ont proposé un filtre linéaire optimal qui réalise une localisation meilleure que celle du détecteur par passage par zéro de Marr-Hildreth. Enfin, Ziou, dans son travail de thèse [ZIOU], a traité le problème de détection de contours dans des images à niveaux de gris, en visant la mise en oeuvre d'une sélection automatique de détecteurs et leurs paramètres les plus appropriés pour mettre en évidence un contour donné.

Plusieurs autres techniques ont tenté de supprimer l'information redondante et moins importante, ce qui est une tâche délicate dans la majorité des domaines d'application de la vision. Les bords de faible contraste, à cause de leur faible rapport signal/bruit, restent souvent un problème très difficile à surmonter.

Les problèmes rencontrés avec les opérateurs locaux de détection de contours sont les suivants:

- la taille de l'opérateur par rapport à celle du phénomène à détecter.
- la discordance entre l'information visuelle et les modèles supposés.
- le phénomène de recouvrement dû à la quantification.

Malgré l'ampleur des recherches effectuées, l'extraction des éléments linéaires à partir d'une scène naturelle reste un problème difficile dans plusieurs domaines. Dans ce chapitre on s'est fixé deux objectifs:

1- le développement d'un algorithme pour l'extraction des éléments linéaires à partir d'une scène naturelle.

2- la construction d'une représentation intermédiaire basée sur la transformée de Hough de ces éléments pour une interprétation de haut-niveau par la base de connaissance.

### III.2.1.2 - Dualité amplitude-orientation du gradient

Dans la majorité des algorithmes d'extraction de lignes ou bords, l'amplitude du gradient (variation d'intensité) est considérée comme une variable importante qui traduit l'existence du bord. Ainsi, elle joue un rôle dominant et central dans sa détection. L'orientation du gradient est définie par la direction de la variation maximale d'intensité sur une surface centrée en un pixel. D'une manière équivalente, c'est la direction locale d'une surface d'intensité. Burns a remarqué que sur une surface d'intensité, l'orientation locale du gradient varie d'une manière très faible par rapport à l'amplitude de celui-ci [BUR-86]. Les bords de route dans les scènes qu'on a traité, sont en fait, une transition entre deux surfaces d'intensité et de texture différentes: Une surface qui reflète la structure de la route, et une autre au-delà du bord de la route. Par conséquent, on a vérifié la remarque précédente, et on a développé un algorithme basé sur l'exploitation de l'orientation du gradient de l'image.

### III.2.1.3 - Choix du masque pour le calcul du gradient

Il existe une multitude de masques qui peuvent être utilisés pour calculer le gradient. Ils sont généralement classés suivant le degré de leur résolution. Une étude sur la sensibilité et la symétrie de rotation établie sur quatre types de masques de tailles: 1x2, 1x3, 2x2, et 3x3, a montré que seulement le masque de taille 1x2 a donné une réponse non-symétrique pour deux contours diagonaux et symétriques par rapport à l'axe vertical [BUR-86].

Il est important de remarquer que les masques de taille importante lissent l'image et réduisent les détails ( voir même, les suppriment complètement ), alors que ceux de taille très faible tendent à les amplifier. Par conséquent, il a été montré que les masques de



tailles 2x2 produisent des erreurs d'orientation assez importantes qui peuvent dépasser 11°, tandis qu'un masque de taille 3x3 produit une erreur maximale de 7° [KIT-89]. Ceci est dû principalement au fait que les dérivés  $G_x$  et  $G_y$  du gradient sont seulement des approximations de ses composantes réelles.

En tenant compte des remarques ci-dessus et en se basant sur les propriétés générales des scènes naturelles on a choisi un masque de taille 3x3 (voir figure III.2.1.3). Car en réalité les bords de route rencontrés dans ces scènes ne sont généralement pas des lignes d'épaisseur un pixel, mais des surfaces linéaires qui forment une transition entre deux régions de structures plus ou moins différentes.

e	d	c
f	a	b
g	h	i

Figure III.2.1.3.1 : Masque de taille 3x3.

#### III.2.1.4 - Choix du type de gradient

Malgré le développement des opérateurs de détection de bord complexes, des opérateurs simples tels que ceux de Sobel, Prewitt et Roberts sont bien connus et très souvent utilisés. Ceci est dû principalement à leur simplicité de calcul et à leur importance particulière dans l'application de la vision artificielle pour la robotique. L'étude du comportement de ces opérateurs a été largement abordée dans le domaine de la vision afin de déterminer leurs propriétés pour les utiliser correctement dans différentes applications. Cette étude se résume à une approche de comparaison pour évaluer l'efficacité de chaque opérateur dans différents types de scènes. Il existe généralement deux types d'évaluation : analytique et empirique. Dans l'étude analytique un modèle du bord est utilisé et la réponse de l'opérateur à ce bord est analysée mathématiquement en fonction des

paramètres du bord et de l'opérateur. D'autre part, dans l'étude empirique, l'opérateur est appliqué à plusieurs images de type naturel ou synthétique et ses performances sont évaluées par des critères statistiques. Un de ceux-ci, peut être la fidélité de l'opérateur à des bords dont la position est connue sur l'image. En général, il n'y a pas une grande distinction entre l'étude analytique et empirique puisque cette dernière peut utiliser un modèle mathématique du bord pour générer une image synthétique. En plus, l'étude analytique peut aussi se servir de la simulation numérique pour obtenir ses résultats. Enfin, il existe quelques études qui combinent les deux approches [BUR-86].

Rappelons que le calcul du gradient consiste à associer à chaque pixel  $I(i,j)$  de l'image originale une approximation  $G(i,j)$  du vecteur gradient, obtenue par des différences finies. Le vecteur  $G(i,j)$  possède deux composantes  $G_x$  et  $G_y$ , respectivement selon les directions horizontales et verticales, qui sont obtenues par convolution de l'image avec des masques  $M_x$  et  $M_y$  de dimension  $3 \times 3$ . Les éléments des masques ainsi que leur tailles sont caractéristiques de l'approximation obtenue.

On peut déterminer donc le module et l'orientation du vecteur gradient par :

$$\| G(i,j) \| = \sqrt{G_x^2(i,j) + G_y^2(i,j)}$$

$$\theta(i,j) = \arctan \frac{G_y(i,j)}{G_x(i,j)}$$

Pour déterminer les éléments des masques  $M_x$  et  $M_y$ , il faut choisir un type de gradient à utiliser. D'après [PRATT], les masques de Prewitt et de Sobel s'avèrent les meilleurs lorsqu'on se propose d'estimer l'orientation du gradient par un arctangente (voir Annexe 1). En plus, on remarque que l'opérateur de Sobel possède une performance qui est légèrement meilleure que celle de l'opérateur de Prewitt en ce qui concerne l'immunité au bruit. On note aussi que Kitchen [KIT-89] a montré que parmi ces opérateurs celui de Sobel produit la meilleure performance avec une erreur approximative d'orientation de  $4^\circ$ . Cette supériorité de performance de l'opérateur de Sobel est principalement due aux poids forts attribués aux pixels les plus proches du centre de voisinage. On a donc finalement choisi l'opérateur de Sobel.

La configuration de  $M_x$  et  $M_y$  est la suivante :

$$M_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad M_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Ces masques possèdent la propriété d'être séparables car on peut les obtenir par :

$$M_x = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}^T = M_1 M_2^T$$

$$M_y = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}^T = -M_2 M_1^T$$

Ainsi les composantes  $G_x$  et  $G_y$  s'obtiennent par :

$$G_x = I * M_x = - M_1^T I M_2$$

$$G_y = I * M_y = - M_2^T I M_1$$

où  $I$  est la matrice 3x3 du pixel  $I(i,j)$  et ses 8 voisins :

$$I = \begin{pmatrix} I(i+1,j-1) & I(i+1,j) & I(i+1,j+1) \\ I(i,j-1) & I(i,j) & I(i,j+1) \\ I(i-1,j-1) & I(i-1,j) & I(i-1,j+1) \end{pmatrix}$$

Cette forme est équivalente à:

$$G_x(i,j) = [ I(i+1,j-1) + 2 I(i,j-1) + I(i-1,j-1) ] - [ I(i+1,j+1) + 2 I(i,j+1) + I(i-1,j+1) ]$$

$$G_y(i,j) = [ I(i-1,j-1) + 2 I(i-1,j) + I(i-1,j+1) ] - [ I(i+1,j-1) + 2 I(i+1,j) + I(i+1,j+1) ]$$

On note que celle-ci est favorable à une exécution en temps réel car elle permet de réduire le nombre d'opérations et de donner à l'opérateur de convolution une structure pipe-line.

Dans le cas général les valeurs prises par les pixels de l'image d'orientation du gradient dépendent de la façon dont celle-ci a été quantifiée. Le paragraphe suivant illustre l'effet de la segmentation de la direction du gradient.

### III.2.1.5 - Quantification et segmentation de l'orientation du gradient

Une fois que les orientations locales du gradient sont estimées, elles sont regroupées pour être classées dans une partition qui permet d'exprimer une quantité angulaire. Ce problème peut être considéré comme celui de la segmentation de l'orientation du gradient. Figure III.2.1.5.1 montre un exemple de huit régions, ainsi il existe huit valeurs possibles d'orientation.

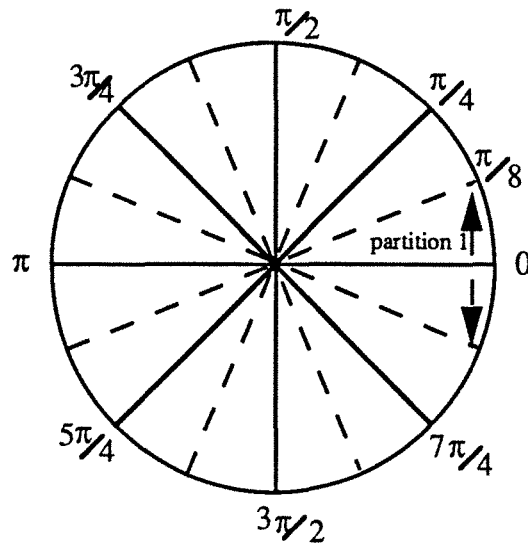


Figure III.2.1.5.1 : Quantification de la direction du gradient.

L'orientation du gradient en un pixel prendra la valeur la plus proche parmi les valeurs possibles. L'intervalle  $[0^\circ, 360^\circ]$  des orientations du gradient peut être quantifié d'une façon arbitraire en régions de taille semblables, par exemple 8 régions de  $45^\circ$  (figure III.2.1.5.1) ou 16 régions de  $22,5^\circ$  etc... . Un élément linéaire sur la scène doit en principe engendrer sur l'image d'orientation une région dont les valeurs des pixels sont plus ou moins similaires. Par conséquent, l'extraction de cet élément peut simplement être faite par une segmentation appropriée.

Malheureusement, la quantification crée un problème qui peut rendre l'extraction complètement impossible. Si le pas de quantification est important, un effet de fusionnement se produit : Deux primitives de directions différentes se trouvent classées dans une même partition, par conséquent, elles seront confondues et impossible à dissocier. Par contre, si le pas de quantification est très petit, il y aura une perte importante de ressources et un effet de fragmentation se produit : Un élément linéaire dont la distribution des orientations est divisée entre deux partitions, sera fragmenté en deux éléments distincts. Le mécanisme de groupement des directions similaires devient alors complexe et sensible. Plusieurs algorithmes ont été proposés pour résoudre le dilemme de fragmentation-fusionnement [BUR-86], [KAH-90].

On a remarqué qu'il était plus facile de reconstruire par des techniques appropriées un élément dont l'orientation est fragmentée à cause d'un pas trop faible que

de dissocier plusieurs éléments dont les orientations sont fusionnées. La dissociation des éléments fusionnés par l'orientation du gradient nécessite un temps de calcul considérable et une connaissance très précise de l'environnement. D'autre part, la reconstruction d'un élément fragmenté repose sur la connaissance approximative de l'environnement et l'application de certaines opérations de traitement d'image. On a donc préféré quantifier l'image de direction du gradient sur 256 niveaux (8 bits). L'illustration est faite par une séquence de trois scènes naturelles représentées par les figures III.2.1.5.2.a, III.2.1.5.3.a, et III.2.1.5.4.a. L'orientation de l'opérateur de Sobel de chaque scène est présenté respectivement sur les figures III.2.1.5.2.b, III.2.1.5.3.b, et III.2.1.5.4.b.

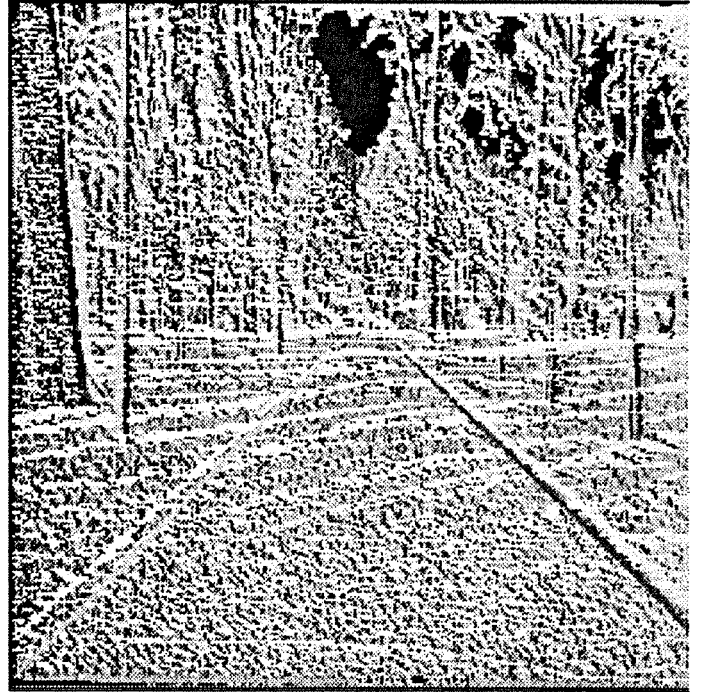
### III.2.1.6 - Discrimination des éléments linéaires

En général l'image orientation  $\theta(i,j)$  est très difficile à exploiter pour obtenir les éléments linéaires de l'image initiale. Car cette image inclue un nombre considérable d'éléments linéaires plus ou moins fragmentés. Une solution consiste à reconstruire à partir de cette image une nouvelle image d'orientation dans laquelle apparaissent uniquement les éléments linéaires. Ces éléments forment des régions d'intensités approximativement similaires, et la méthode la plus classique pour les extraire serait une segmentation. Cependant, on a remarqué que les techniques de croissance de région ne sont pas applicables car une erreur occasionelle de sur-groupement peut causer des résultats catastrophiques. Les petites variations d'orientation d'un pixel à un autre peuvent ainsi causer l'apparition de régions formées d'orientations complètement différentes. En plus, les changements d'orientation des bords peuvent produire des orientations de gradient intermédiaires au lieu d'une discontinuité d'orientation. On a donc préféré segmenter l'image par analyse de l'histogramme de l'image d'orientation du gradient. Les seuils seront définis à posteriori par une recherche des pics de l'histogramme qui permet de déterminer le nombre de seuillage à effectuer. A chaque seuil correspond une image binaire. Puisqu'on s'intéresse aux objets linéaires possédant approximativement la même orientation sur la scène, les seuils de segmentation doivent encadrer un pic de l'histogramme qui caractérise un grand nombre de pixels ayant la même orientation. Donc théoriquement le nombre de seuillage à effectuer correspondra exactement au nombre de pic de l'histogramme. Les pics qui apparaissent sur l'histogramme montrent l'existence d'une multitude d'éléments linéaires dans la scène,

leur nombre étant imprévisible d'une image à l'autre. Cette contrainte nous oblige à limiter le nombre de pics à détecter.



a : Scene 1



b: Image orientation  $\theta$  de la scène 1.

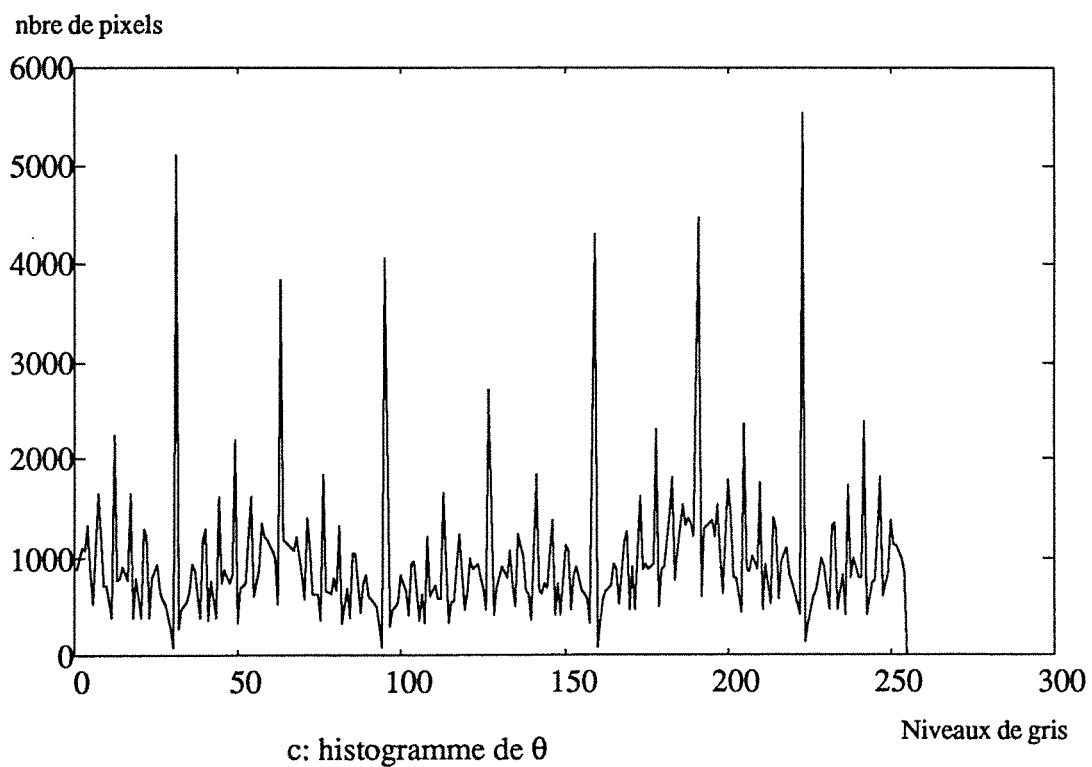
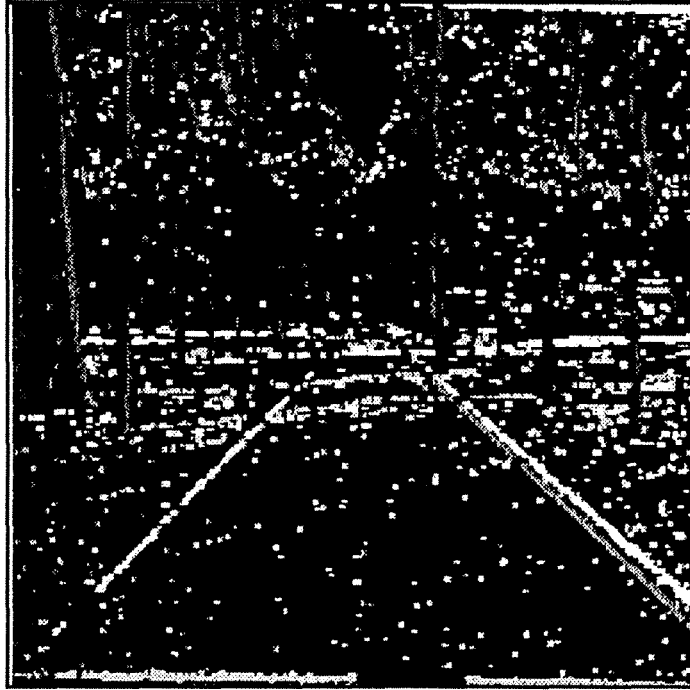


Figure III.2.1.5.2





d: Image orientation reconstruite  $\theta_n$

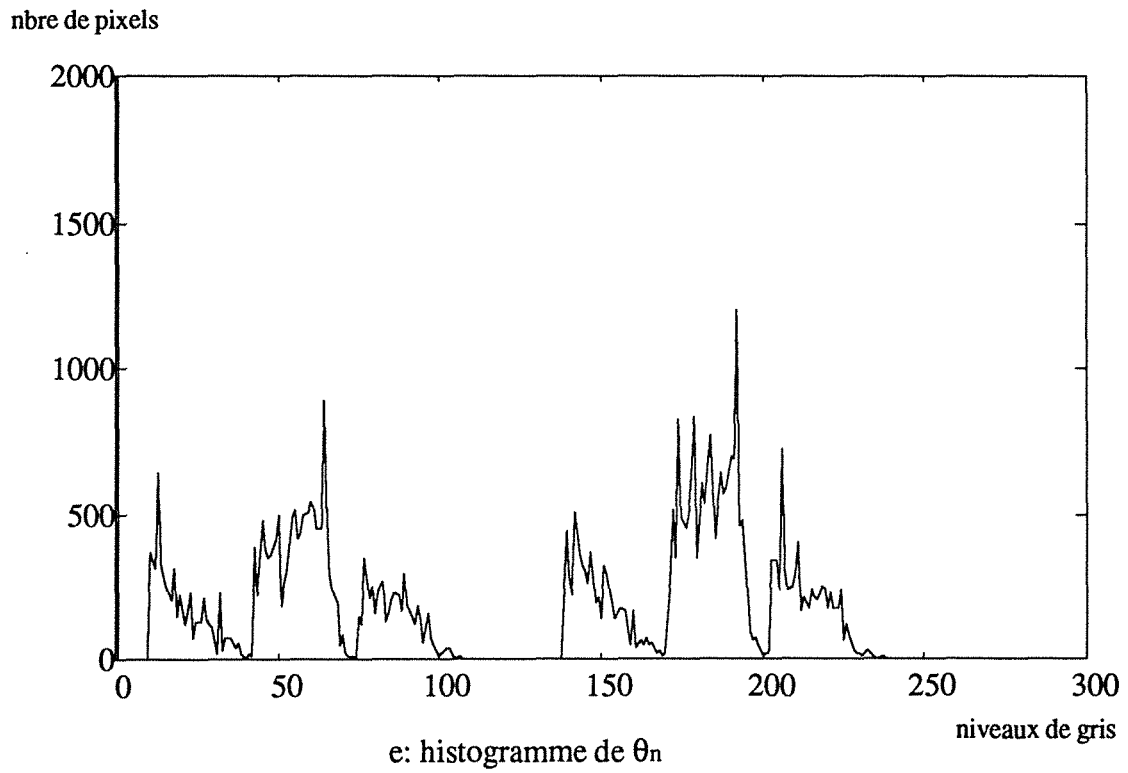
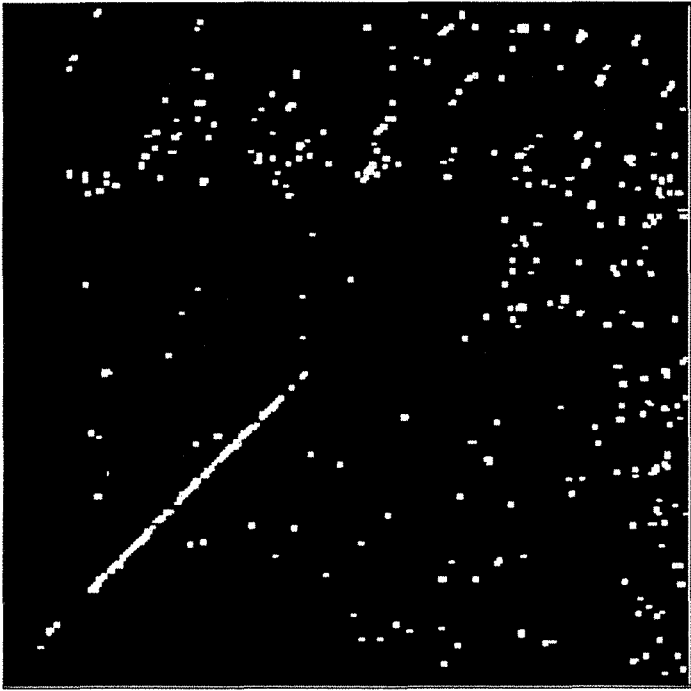
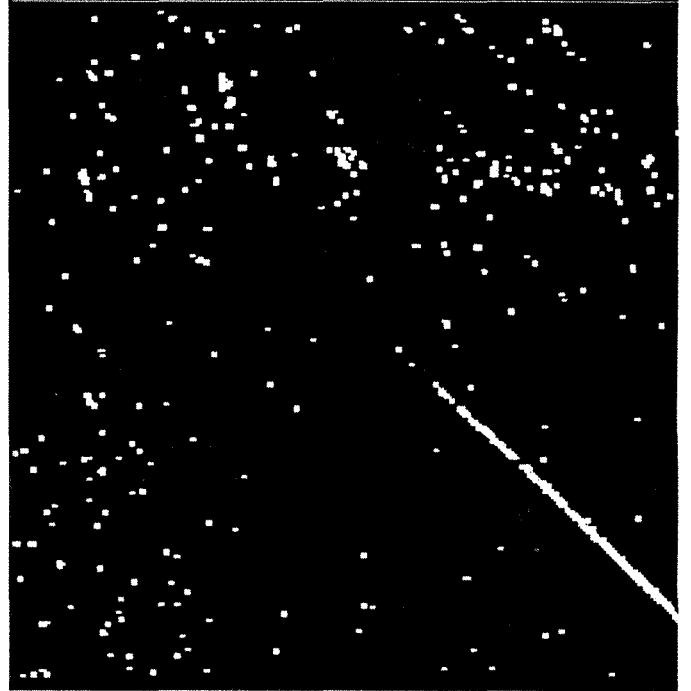


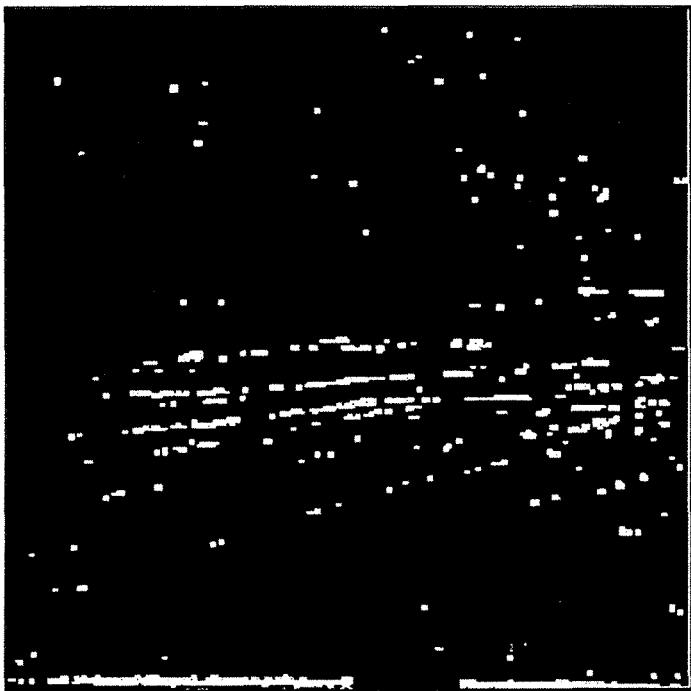
Figure III.2.1.5.2



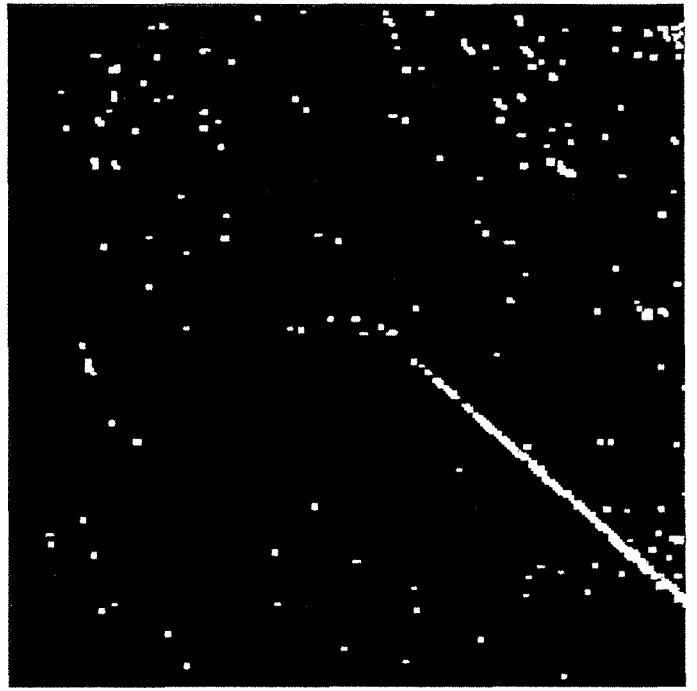
f: pk1



g: pk 2

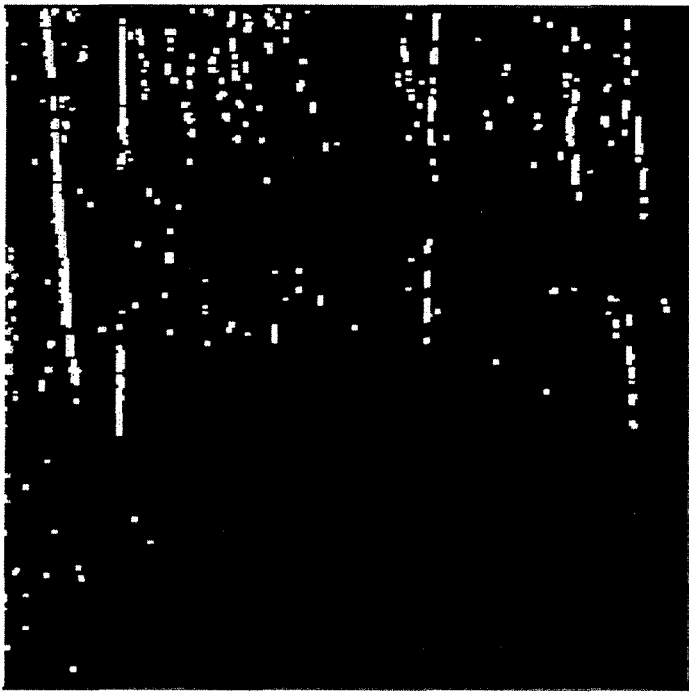


h: pk3

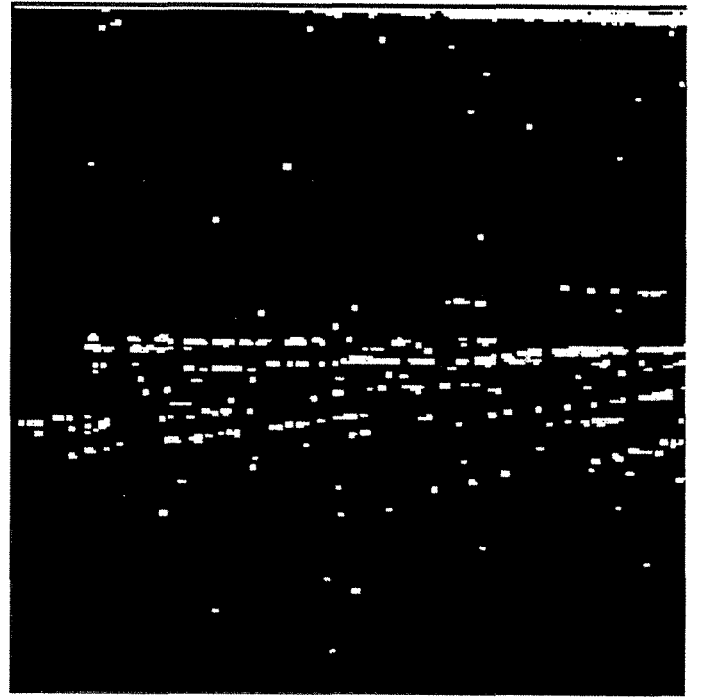


i: pk4

Figure III.2.1.5.2



j: pk5

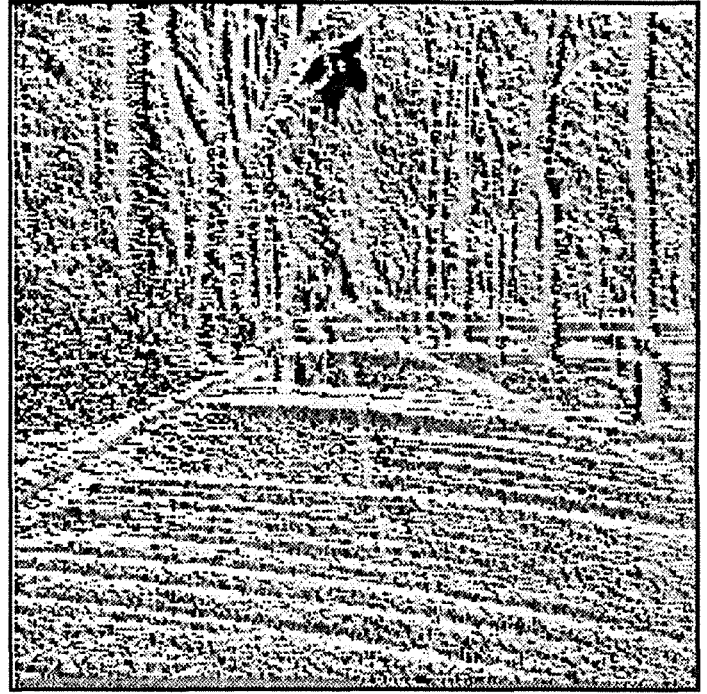


k: pk6

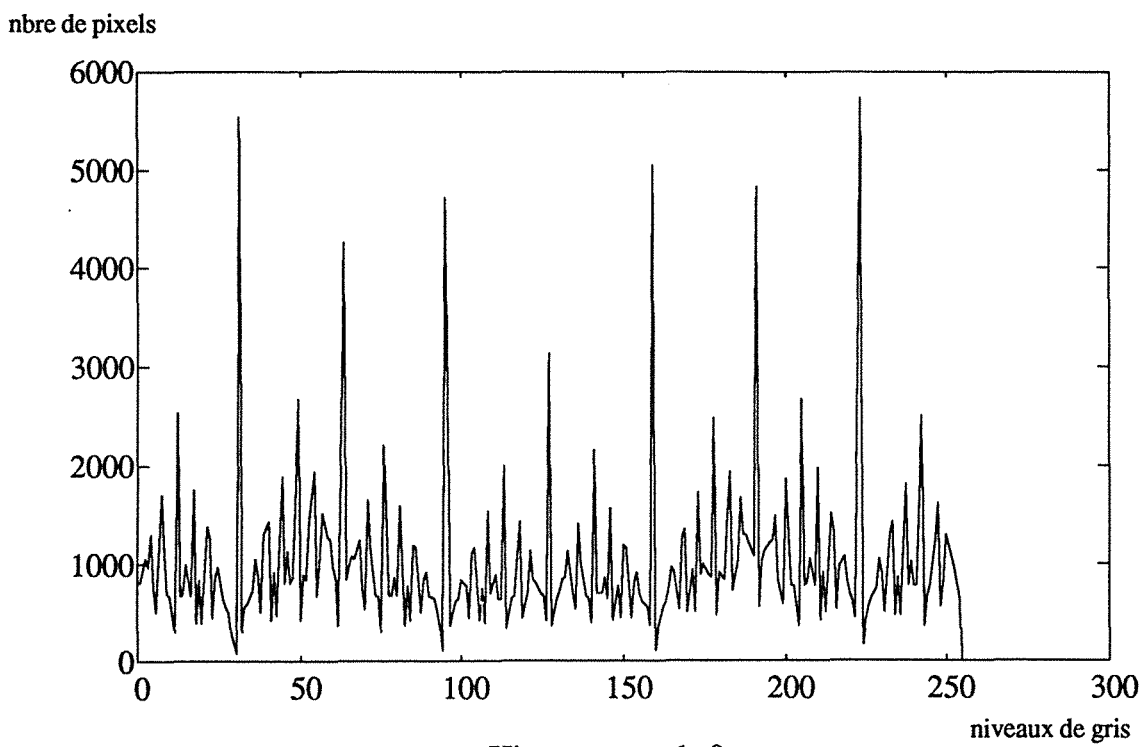
Figure III.2.1.5.2



a : Scene 2



b: Image orientation  $\theta$  de la scène 2



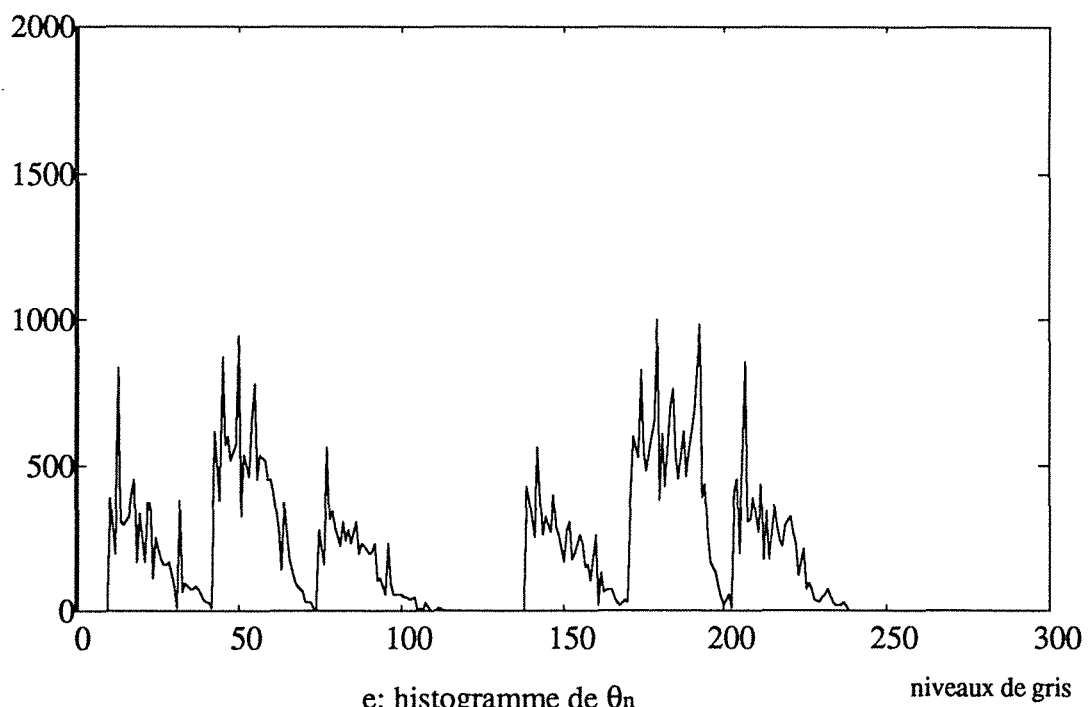
c: Histogramme de  $\theta$

Figure III.2.1.5.3



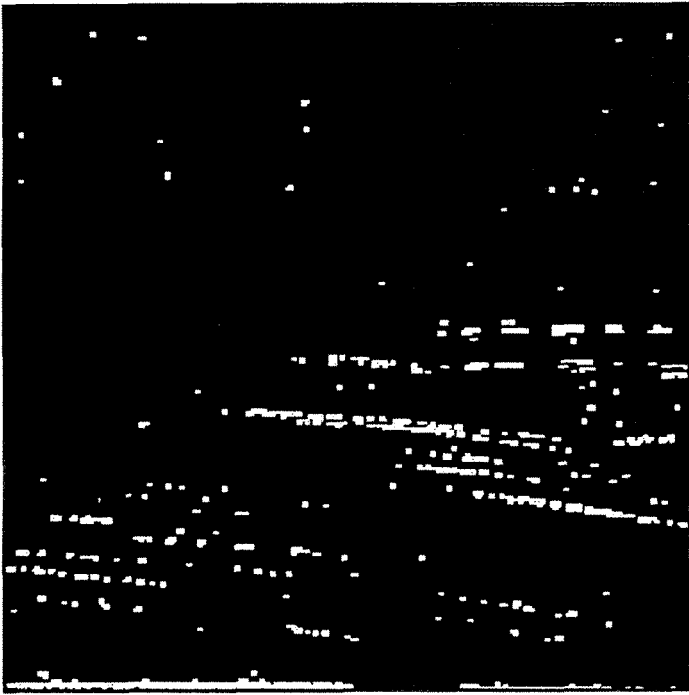
d: image orientation reconstruite  $\theta_n$

nbre de pixels

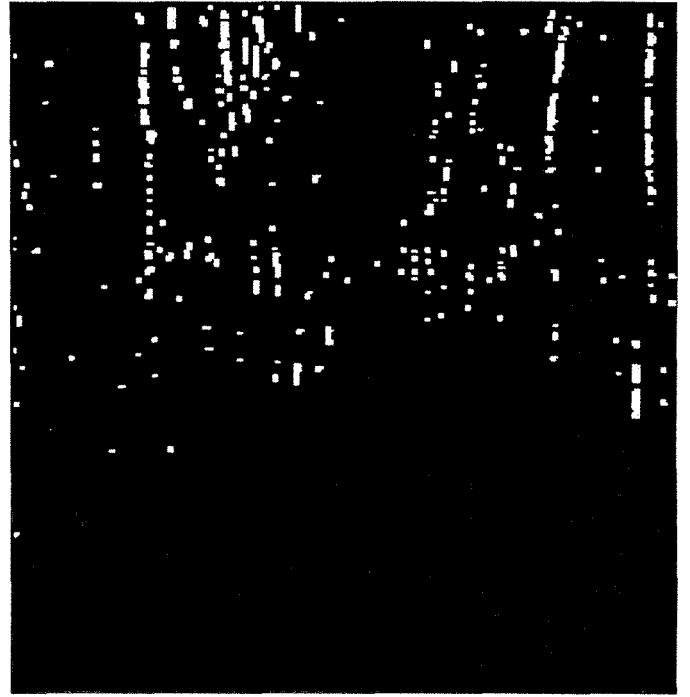


e: histogramme de  $\theta_n$

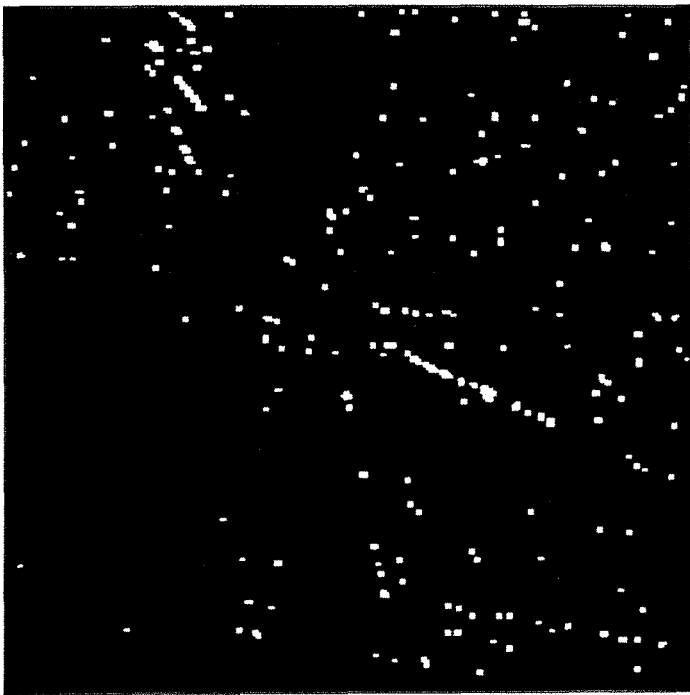
Figure III.2.1.5.3



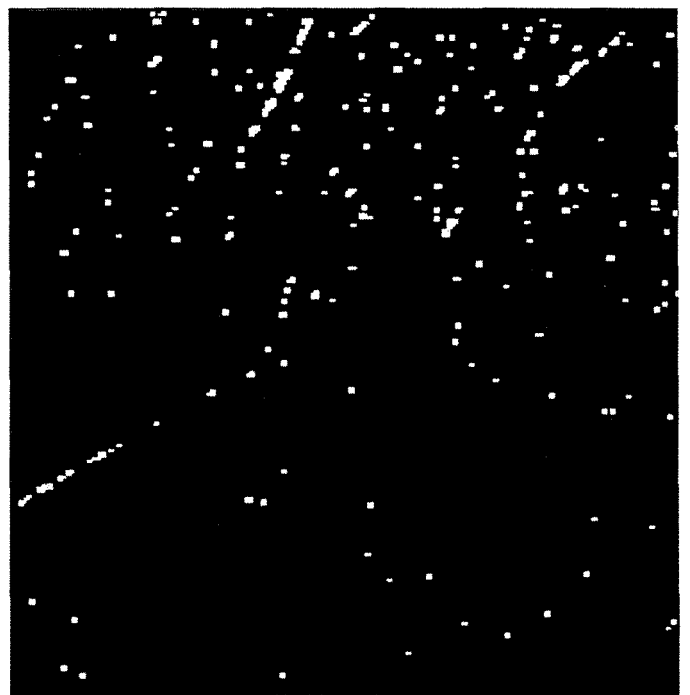
f : pk2



g: pk3

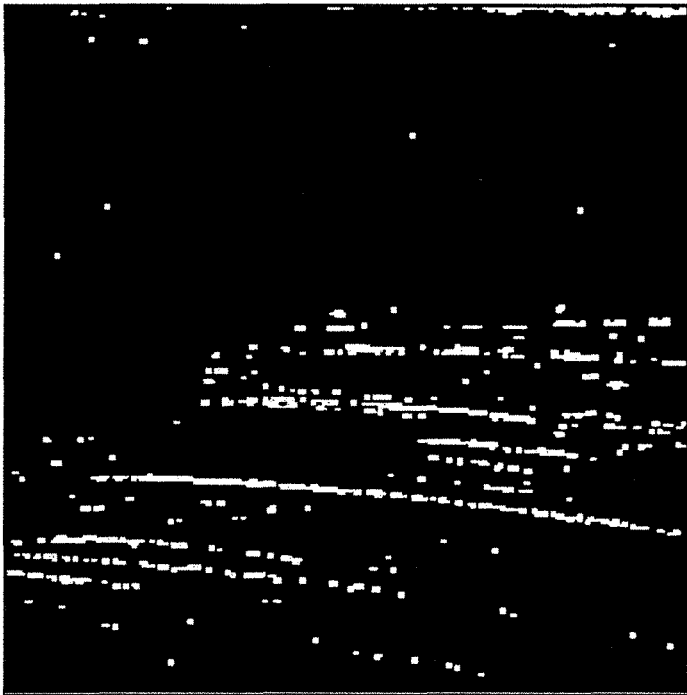


h : pk4

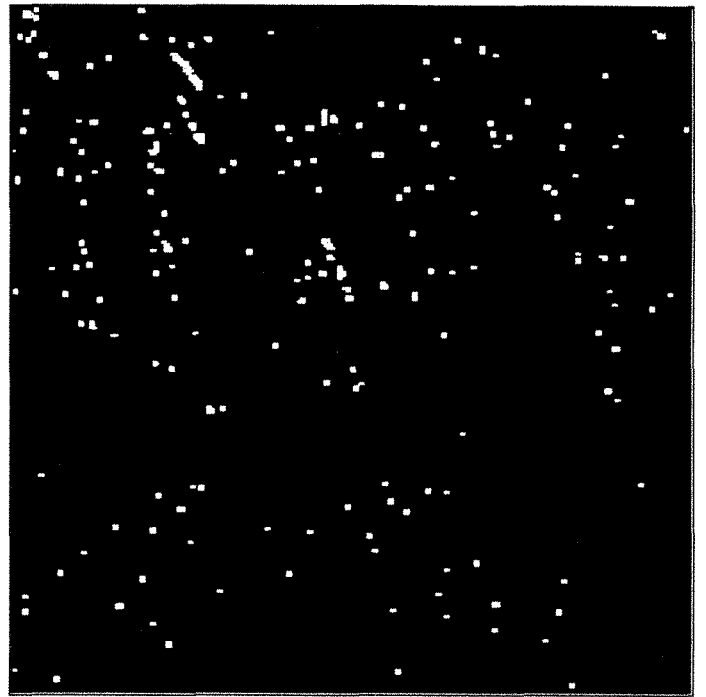


i : pk5

Figure III.2.1.5.3



j: pk5

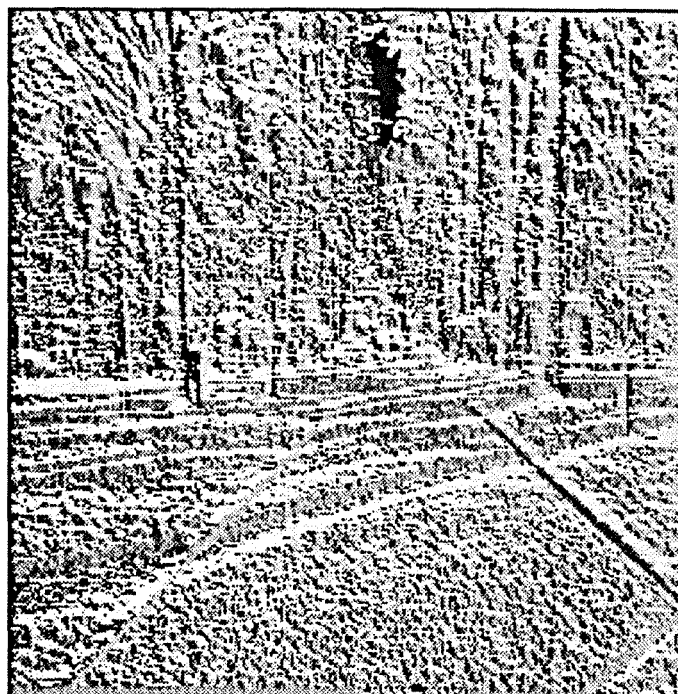


k: pk6

Figure III.2.1.5.3



a : Scene 3



b: Image orientation  $\theta$  de la scène 3

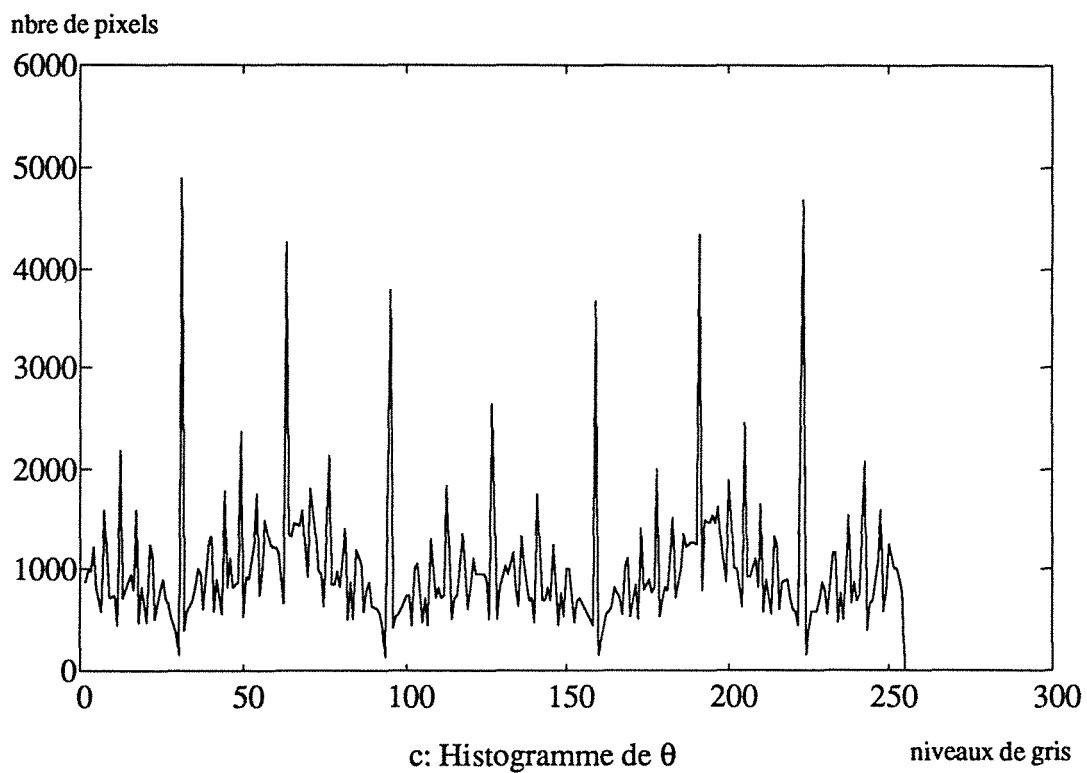
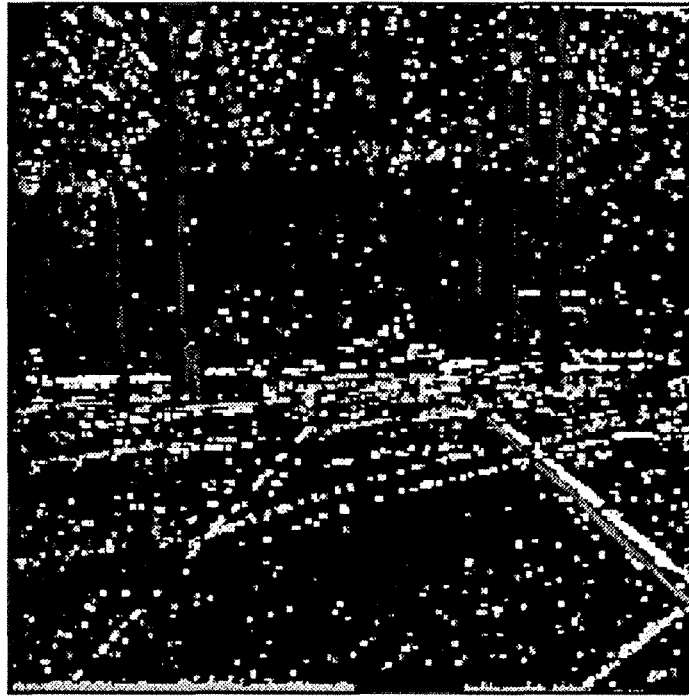


Figure III.2.1.5.4





d: image orientation reconstruite  $\theta_n$

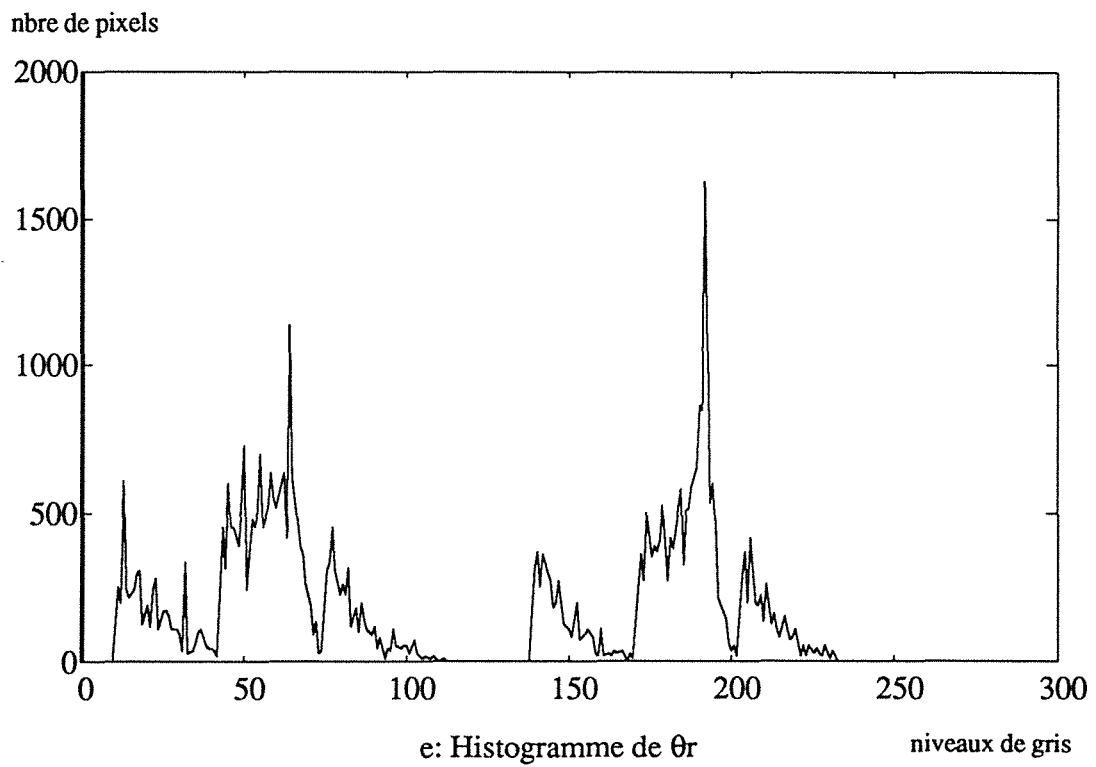
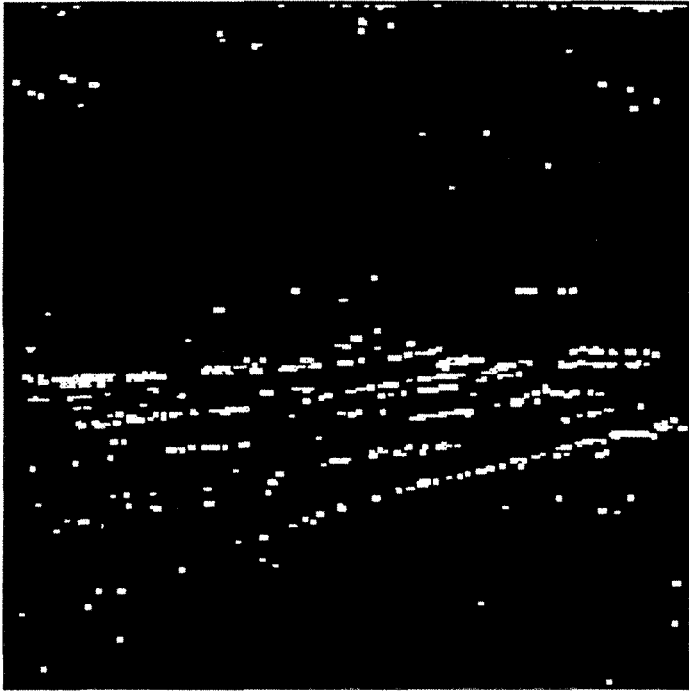
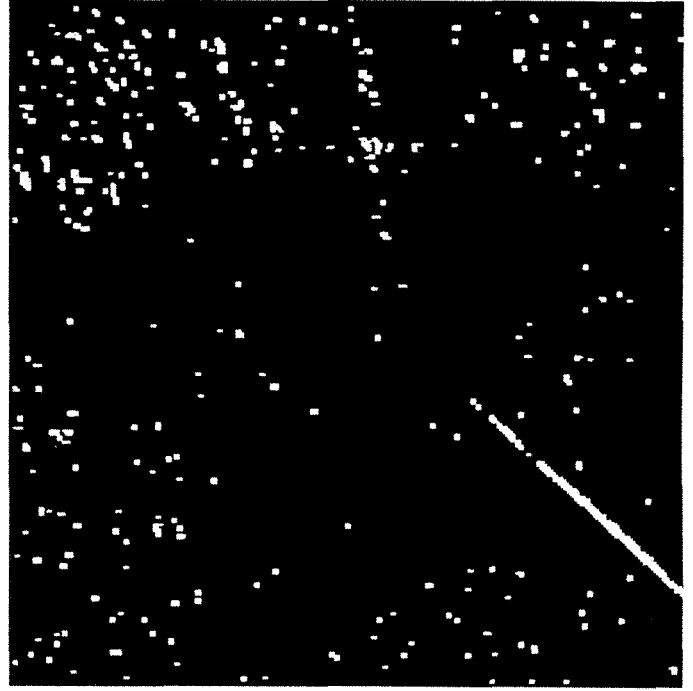


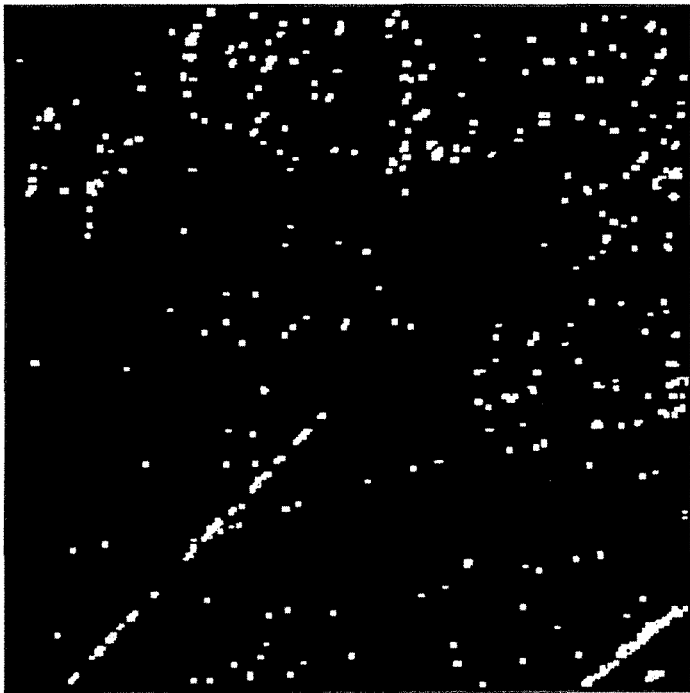
Figure III.2.1.5.4



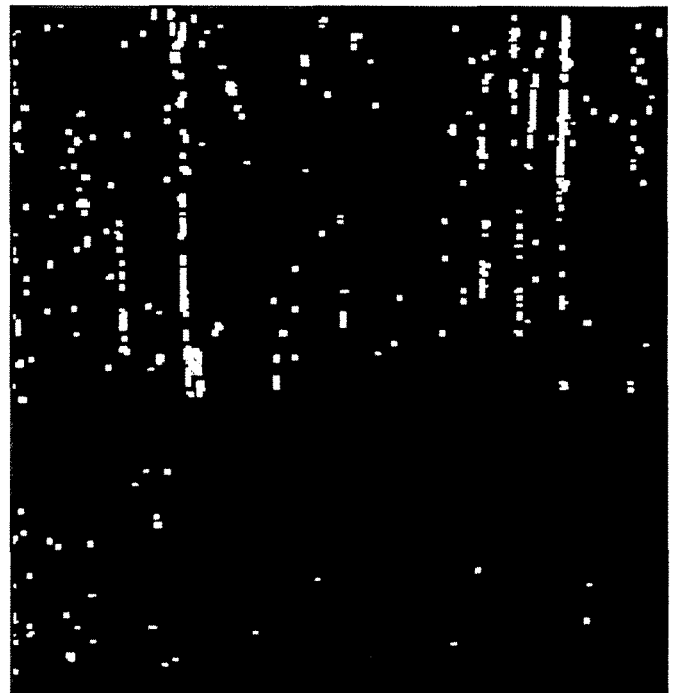
f : pk1



g : pk2

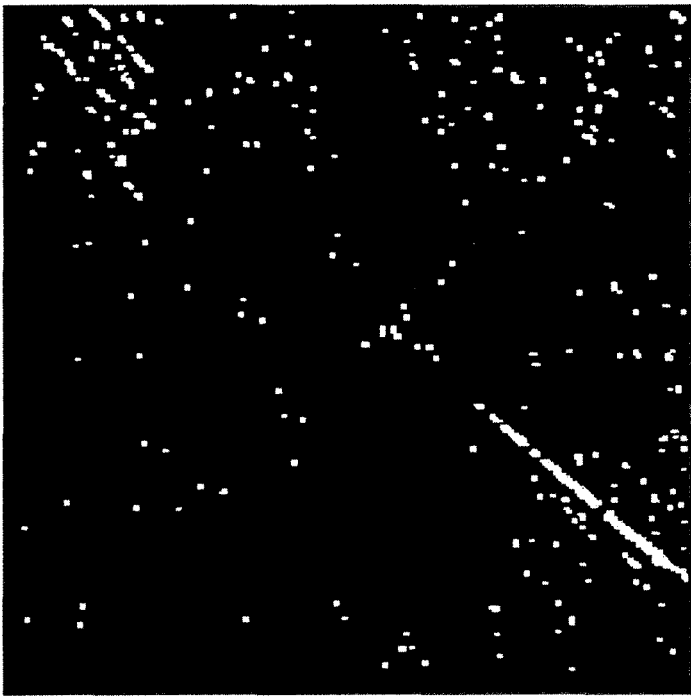


h : pk3

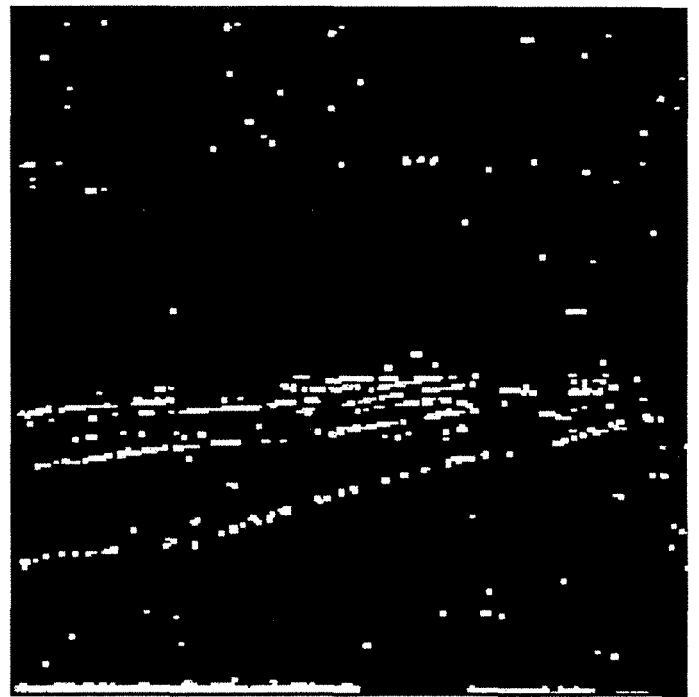


i : pk4

Figure III.2.1.5.4



j : pk5



k : pk6

Figure III.2.1.5.4

Une limite peut être obtenue en faisant la constatation suivante:

Naturellement dans toute scène, la partie inférieure de l'image correspond aux objets proches tandis que la partie supérieure correspond aux objets lointains. Cet effet de perspective engendre un certain degré de compression des dimensions réelles des objets qui se situent en haut de l'image. En conséquence, toute déviation d'une route apparait dans la partie haute de la scène sur une portion relativement faible de l'image, alors que les bords, les plus importants, pour la navigation du robot mobile, se situent au bas de la scène. Notre étude expérimentale a montré que dans la majorité des cas les images de chemins rencontrés comportent au plus deux déviations consécutives visibles. En considérant un chemin idéal ayant deux déviations, les bords de route après la deuxième déviation sont à peine visibles et ne présentent pratiquement aucun intérêt pour le guidage. Ceci a permis de limiter notre étude à un chemin qui possède au plus deux déviations. Ce nombre implique la nécessité d'extraire au plus six segments de droite qui représentent les différents bords du chemin (voir figure III.2.1.6.1).

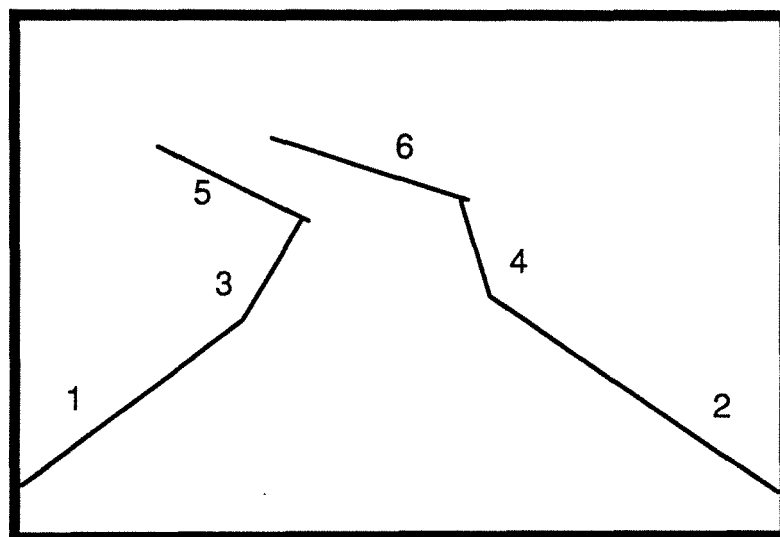


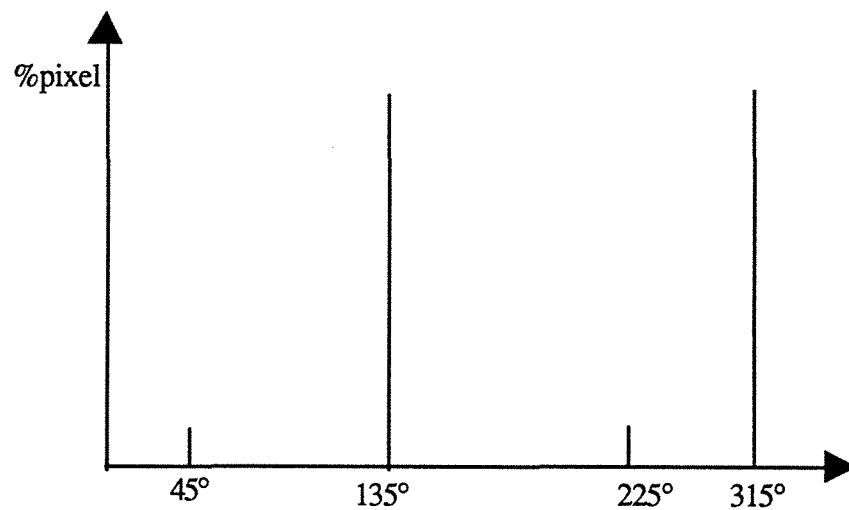
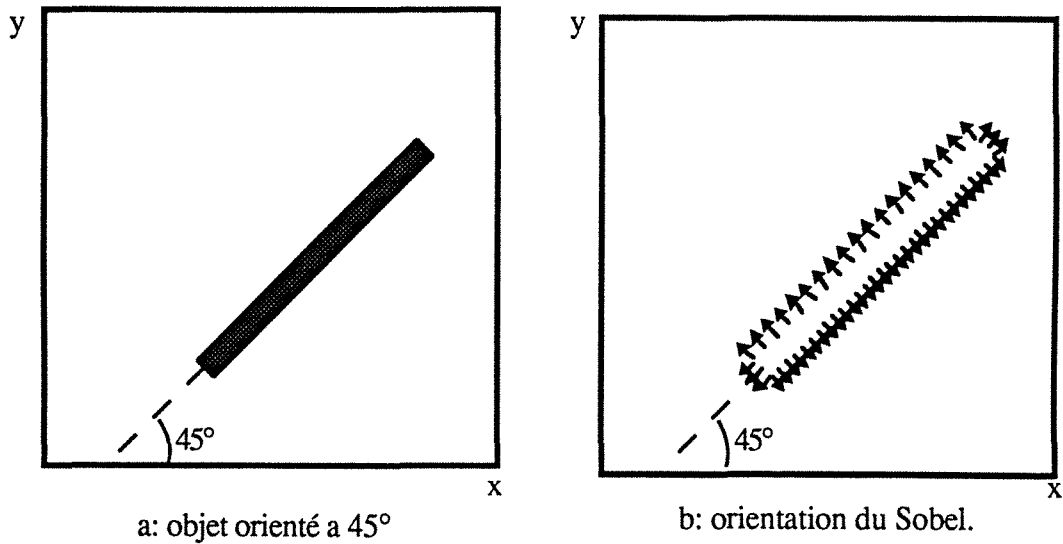
figure III.2.1.6.1: Approximation de bord de route par des segments de droites.

Dans le cadre de cette hypothèse, la linéarité des éléments de la scène engendre dans l'histogramme de l'image orientation  $\theta(i,j)$  un ensemble de pics parmi lesquels figurent au plus six pics correspondant aux bords du chemin .

La quantification de la direction est faite sur 256 niveaux, donc  $2\pi$  est représenté par le niveau 255. Par conséquent , la résolution est de  $\frac{2\pi}{256} = 0,0245 \text{ rds} = 1,4063^\circ$ . Les figures III.2.1.5.2.c, III.2.1.5.3.c, et III.2.1.5.4.c montrent les histogrammes des orientations du gradient pour chacune des scènes de la séquence. On remarque que l'histogramme de l'image de direction du sobel est presque symétrique par rapport à l'axe dont le niveau est 128 correspondant à  $180^\circ$ . Ceci est dû à deux raisons :

- 1) le vecteur gradient est perpendiculaire à l'élément à détecter et possède une orientation  $\theta$  par rapport à l'axe horizontal.
- 2) l'élément à détecter possède très souvent une surface qui donne deux vecteurs gradient dont les orientations diffèrent de  $\pi$ .

Pour illustrer ceci, considérant une image comportant un objet linéaire orienté à  $\frac{\pi}{4}$  par rapport à l'axe des abscisses  $x$  ( voir figure III.2.1.6.2.a ). L'application de l'opérateur de Sobel donne une image phase représentée par la figure III.2.1.6.2.b . L'histogramme de cette image ( figure III.2.1.6.2.c ) montre l'existence de deux pics dominants à  $135^\circ$  et  $315^\circ$  correspondant à deux directions perpendiculaires à l'orientation de l'objet. Par conséquent l'orientation de l'objet peut être obtenue par une addition ou une soustraction de  $90^\circ$  à partir des orientations dominantes.



c: Histogramme de l'orientation du gradient.

Figure III.2.1.6.2.

Puisque notre objectif est de détecter les bords de route éventuellement formés par des surfaces de largeur relativement très faibles par rapport à d'autres objets naturels, on peut détecter l'objet uniquement par un pic dominant. Ceci peut être fait en limitant l'étude de l'histogramme uniquement sur l'intervalle  $[0^\circ, 180^\circ]$  ce qui correspond à l'intervalle des niveaux  $[0,128]$ .

Puisqu'on a limité les angles d'orientation à l'intervalle  $[0^\circ, 180^\circ]$ , et le nombre de segments à six, l'intervalle d'orientation pour extraire les pics sera  $\frac{180^\circ}{6} = 30^\circ$  ou  $\frac{128}{6} \approx 21$  en niveau de gris.

On élabore un algorithme de recherche de pic avec la contrainte:

$$N(\text{pic}_i) - N(\text{pic}_{i+1}) \geq 21 \quad i=1,2,3,4,5.$$

$N(\text{pic}_i)$  : abscisse du pic  $i$ .

Dans l'histogramme, la discrimination des éléments linéaires passe par la formation d'images binaires obtenues à partir d'opérations de seuillage multiple de l'image orientation. Cela permet de localiser dans chaque image binaire les éléments de la scène qui possèdent à peu près la même orientation.

Ainsi on fixe l'intervalle de seuillage à une valeur égale à la marge d'orientation  $30^\circ$  et on prend la valeur  $N(\text{pic}_i)$  comme centre de cet intervalle.

En conséquence, les niveaux de seuillage seront pris comme :

$$S_{i1} = N(\text{pic}_i) - 10 \quad i=1,2,3,4,5,6.$$

$$S_{i2} = N(\text{pic}_i) + 10$$

Chaque pic  $i$  conduit à la génération par seuillage d'une image binaire  $pk_i$ . Le nombre de ces images binaires varie entre un minimum d'une image et un maximum de 6 images suivant la scène.

### III.2.1.7 - Reconstruction de l'image orientation

Chacune des images  $pk_i$  contient les éléments dont l'orientation est incluse dans l'intervalle  $[S_{i1}, S_{i2}]$ .

Une addition logique de ces images binaires telle que :

$$pk = \bigcup_{i=1,2,3,4,5,6} pk_i$$

permet de regrouper dans une seule image binaire  $pk$  uniquement les informations utiles de la scène. Dans certains cas les images binaires  $pk_i$  sont le siège de bruit qui se manifeste par des points isolés situés au bas et surtout sur les bords de l'image. Pour les éliminer, on procède à une opération morphologique d'ouverture sur l'image  $pk$  ce qui diminue le phénomène de fragmentation et le temps d'exécution consacré à la recherche des bords .

A partir de cette image fusionnée, on peut transformer l'image orientation originale en une nouvelle image qui accentue les aspects linéaires et atténue fortement les autres. La nouvelle image  $\theta_r(i,j)$  est obtenue par:

$$\theta_r(i,j) = pk \times \theta(i,j)$$

$\times$  : multiplication pixel par pixel.

Les figures III.2.1.5.2.d, III.2.1.5.3.d, et III.2.1.5.4.d montrent les orientations reconstruites du gradient pour la séquence des trois scènes.

L'élimination des aspects non-rectilignes de  $\theta(i,j)$  cause l'apparition de pics beaucoup plus accentués dans l'histogramme de l'image  $\theta_r(i,j)$  comme le montre les histogrammes des figures III.2.1.5.2.e, III.2.1.5.3.e, et III.2.1.5.4.e .



Pour faciliter le traitement ultérieur, il est préférable qu'une image binaire inclue autant que possible un seul élément linéaire dominant. Les images binaires résultent d'une segmentation de l'image orientation reconstruite  $\theta_r$ . Cela nous permet de reprendre le même algorithme de recherche de pics, mais avec une marge d'orientation plus faible pour mieux discerner l'élément. La contrainte devient alors:

$$N(\text{pic}_i) - N(\text{pic}_{i+1}) \geq 10 \quad 1 \leq i \leq 5$$

ce qui correspond à  $15^\circ$ , et les niveaux de seuillage sont donnés par:

$$S_{i1} = N(\text{pic}_i) - 5$$

$$S_{i2} = N(\text{pic}_i) + 5 \quad 1 \leq i \leq 6.$$

Le résultat de ce deuxième seuillage conduit à la formation d'au plus six images binaires dont les orientations sont relativement bien filtrées. Ces images binaires sont montrées par les figures III.2.1.5.2.f, III.2.1.5.3.f, et III.2.1.5.4.f .

On a toujours une perte d'information à partir de l'image orientation du gradient qui est principalement due aux opérations morphologiques appliquée aux images binaires avant et après la reconstruction. D'autre part, si 6 pics sont détectés l'intervalle d'orientation de  $180^\circ$  est divisé en 6 intervalles de  $30^\circ$  et aucune autre perte n'est produite. Par contre, si le nombre de pics est inférieure à 6 la perte d'information augmente en fonction du nombre d'images manquant. Ainsi , la quantité d'information perdue à partir de l'image orientation originale du gradient est modulée par le nombre de pics détecté.

### III.2.2 - LOCALISATION ET PARAMETRISATION DES ELEMENTS DETECTES

### III.2.2.1 - Délimitation du champs de recherches:

Pour diminuer le temps d'exécution et assurer la robustesse de l'algorithme de détection des primitives linéaires , on réduit le champ de recherche aux zones de forte probabilité de rencontrer des bords de route. L'examen attentif de scènes typiques montre que les bords de la route dans la partie supérieure de l'image ( zone 3 de la figure III.2.2.1.1 ) sont très faiblement contrastés ou n'apparaissent pas du tout. Par contre l'apparition des bords dans dans la zone 1 de l'image (figure III.2.2.1.1) dépend de la distance focale de l'optique utilisée, de l'orientation en élévation (voir figure III.2.2.1.2), et de l'altitude de la caméra. En conséquence, la zone 2 reste la zone de forte probabilité de trouver des bords de route sans perturbations. La recherche des segments de droites représentant ces bords se limite ainsi aux zones 1 et 2 . L'espace de recherche s'en trouve réduit d'environ 20% car dans notre cas, les zones 3 et 1 représentent chacune 1/5 de l'image .

zone 3	20%
zone 2	60%
zone 1	20%

Figure III.2.2.1.1 : Découpage de l'image en trois zones.

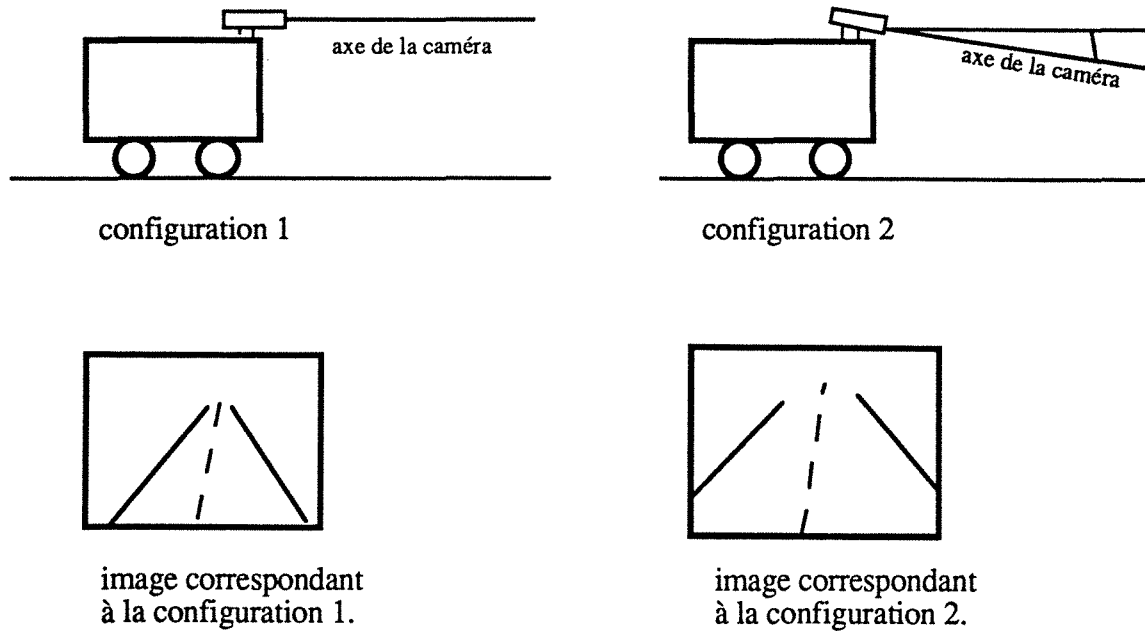


Figure III.2.2.1.2 : Configuration de la caméra et image correspondante.

### III.2.2.2- Application de la transformée de Hough

L'information pertinente des objets en 2-D est souvent contenues dans leur formes, d'où l'intérêt d'un traitement par extraction de contours. Les objets recherchés seront définis par le modèle de ce contour. Puis, la comparaison de ce modèle avec les lignes de l'image peut être effectuée par l'emploi de la transformée de Hough. Il s'agit d'une méthode associant à une courbe ( contour de l'objet ), un point d'un espace de paramètres caractéristiques de cette forme. Cette méthode a été introduite par HOUGH en 1960 par la proposition d'un algorithme adapté à la recherche des droites. Ce n'est qu'en 1972, que cette méthode a été généralisée par Duda et Hart, à la recherche de toute courbe connue paramétriquement (figure III.2.2.2.1).

La détection et la localisation d'un contour donné, se font par l'examen simultané de l'ensemble des points de l'espace des paramètres et de l'espace image. Pour chaque couple ( un élément du premier espace avec un élément du deuxième ), le point de l'espace des paramètres, ou accumulateur, est incrémenté. En fin d'analyse, la présence dans l'image d'un contour identique au modèle, est signalée par un pic de l'accumulateur.

Dans tous les cas, un maximum de l'accumulateur désigne les paramètres de la forme recherché, ayant reçus la plus grande contribution des pixels de l'image.

Forme	paramètres	Dimensions	Equations
Droite non parallèle à l'axe y	a , b	2, 2	$y = ax + b$
Droite non parallèle à l'axe x	a , b	2, 2	$x = ay + b$
Droite quelconque	$\rho , \theta$	2, 2	$x \cos \theta + y \sin \theta - \rho = 0$
Cercle	$x_c, y_c, \rho$	2, 3	$(x-x_c)^2 + (y-y_c)^2 - \rho^2 = 0$
Plan	a, b, c, d	3,4	$ax+by+cz+d=0$

Figure III.2.2.2.1

Habituellement, dans la recherche des segments on utilise une forme de la transformée de Hough, basée sur une définition en coordonnées polaires. Ce calcul a principalement deux avantages:

- l'espace des paramètres  $(\rho, \theta)$  est borné,
- une rotation de l'image correspond à une translation dans l'espace de Hough.

Mais le critère d'adéquation qui est ici :

$$x \cdot \cos \theta + y \cdot \sin \theta = \rho$$

nécessite de nombreux calculs :

- 2 fonctions trigonométriques.
- 2 multiplications.
- 1 addition.

La recherche des droites d'équation:

$$y = a \cdot x + b$$

nécessite seulement une addition et une multiplication.

La transformée de Hough permet enfin de détecter, à partir des données statistiques, la présence dans une image de courbes connues paramétriquement [KIR-91] et [CHE-89]. En particulier cette technique peut être appliquée à la recherche de segment. Dans notre cas on a choisi les segments de droites décrites par une représentation cartésienne:

$$y = ax+b$$

a : pente de la droite,

b : ordonnée à l'origine.

A chaque point  $M_0 (x_0, y_0)$  de l'image étudiée, est associée la famille des droites passant par ce point. Ce faisceau de droites sera représenté par une courbe dans l'espace des paramètres (a , b), soit l'espace de Hough. Sur cet espace, les points alignés de l'image étudiée définissent des courbes concourantes. ( figure III.2.2.2.2).

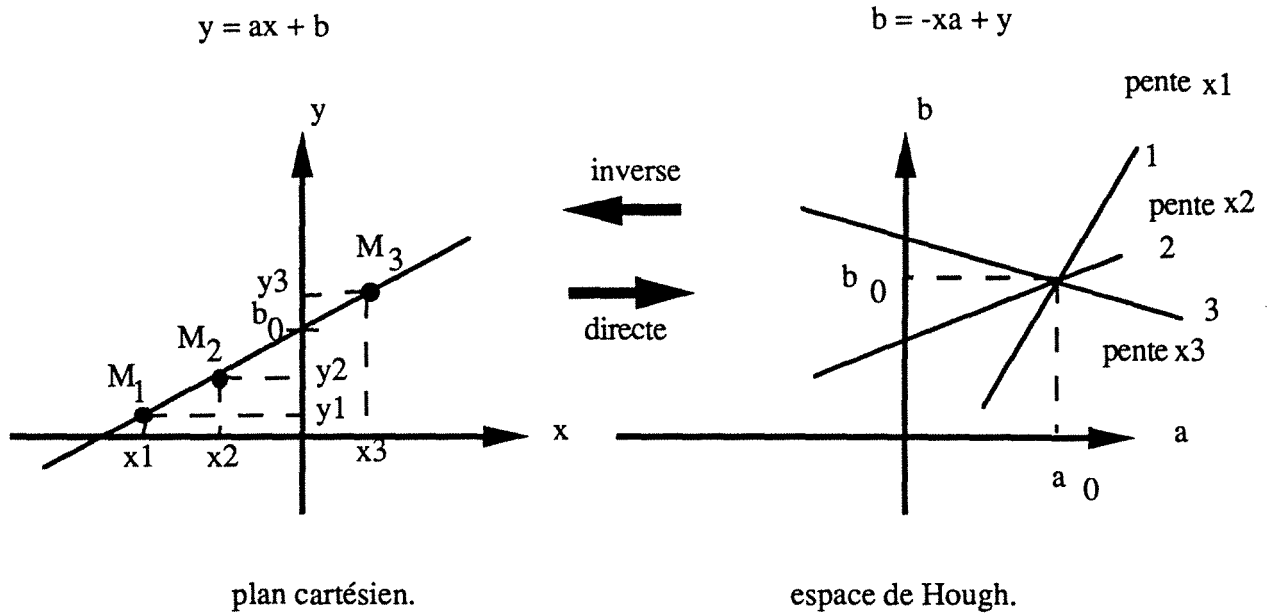


Figure III.2.2.2.2: Transformées de points alignés.

Pour permettre un traitement numérique, l'espace de Hough est échantillonné en un maillage de cases. Pour chaque point  $M_0$  de l'image analysée, on incrémente les cases correspondant à la transformée directe.

Les valeurs de la pente  $a$  ont été limitées dans l'espace de Hough par l'intervalle  $[-8, +8]$  (figure III.2.2.2.3), ceci correspond dans l'image à un intervalle d'orientation des bords de route de  $[-82,87^\circ, +82,87^\circ]$ . Cet intervalle suffit largement à englober les orientations des bords de routes dans les différentes scènes. L'espace de Hough est alors discrétisé en cases de taille  $da=0,1$ . Ce choix sur la taille des cases de l'espace de Hough permet d'avoir un compromis entre le temps d'exécution, l'espace mémoire, et la grandeur des erreurs. Cela veut dire qu'on cherche un segment dont la tangente est un multiple de  $da$ :

$$a = k da \quad -80 < k < 80.$$

Alors que l'ordonnée à l'origine  $b$  est discrétisée en case de taille  $db=20$  pixels sur un intervalle de  $[-800, +800]$ , cela implique que la recherche de l'ordonnée à l'origine d'une droite est faite tous les 20 pixels sur une image de  $512 \times 512$  pixels.

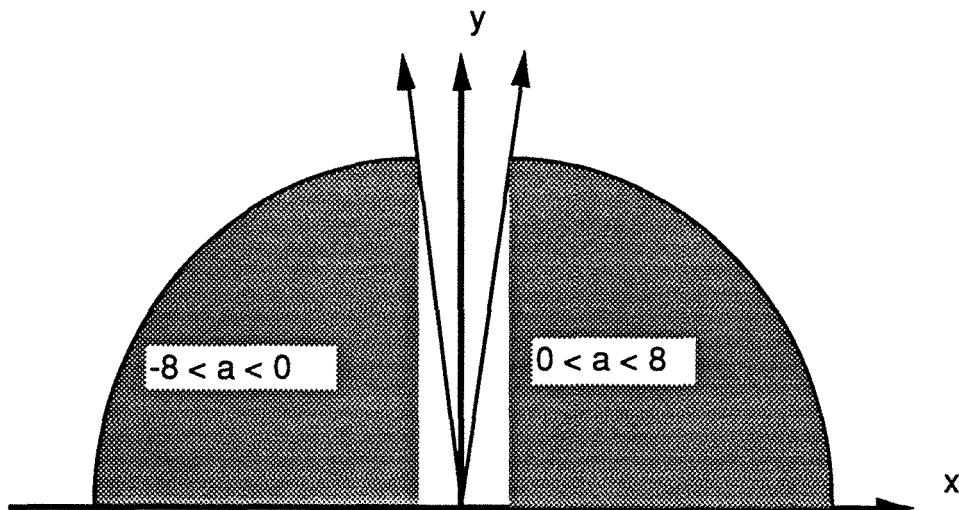


Figure III.2.2.2.3: Partition suivant la pente  $a$ .

On applique la transformée de Hough à chacune des images binaires (maximum 6 images) correspondant aux pics de l'histogramme de l'image d'orientation (voir figure III.2.1). D'une manière générale chaque image reflète les éléments de la scène qui possèdent approximativement les mêmes orientations. Le problème de la détection des segments est alors ramené à celui de la recherche des maxima de l'histogramme. La valeur d'un maximum et sa position indiquent respectivement le nombre de points alignés, qui ont contribué à la formation du segment, et les paramètres associés. En conséquence, le résultat de la transformée de Hough d'une image binaire fournit directement les paramètres  $a$  et  $b$  de la droite passant par le maximum de points de l'élément linéaire. Son orientation est donc définie par  $\varphi = \tan^{-1}(a)$ .

Le temps d'exécution de cette phase est très important. En raison des facteurs suivants:

- l'effet de bords de l'image induit un très grand nombre de points concentrés sur les extrémités .

- la linéarité des certains objets situés en haut de l'image, comme les troncs d'arbre, grillage ou horizon.

Pour réduire le temps d'exécution, on concentre la recherche par la transformée de Hough sur une image réduite (zone hachurée) comme l'indique la figure III.2.2.2.4.

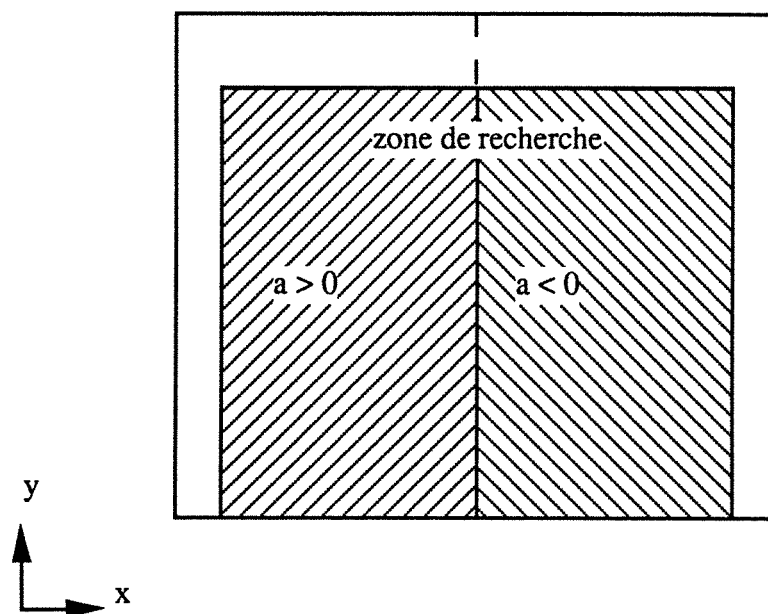


Figure III.2.2.2.4. Localisation des bords par leur signe de pente.

Durant la phase de mise en route, le robot est supposé orienté sur le chemin de telle façon que les bords soient visibles dans l'image, et tels que les bords droit et gauche du chemin apparaissent avec des pentes respectivement  $a < 0$  et  $a > 0$  au bas de la scène. Ainsi, en exécutant l'algorithme de détection en connaissant à priori le signe et la zone de l'élément à détecter (figure III.2.2.2.4), la localisation se trouve facilitée et le risque d'erreur réduit .

La discrétisation de l'espace de Hough induit des erreurs sur la pente  $a$  (voir figure III.2.2.2.5) et sur l'ordonnée à l'origine  $b$  . Puisque :



$$\varphi = \arctang a ,$$

$$d\varphi = \frac{da}{(1+a^2)}$$

donc:

\* l'erreur commise sur la pente a est comprise dans l'intervalle:

$$\left[ 0 , \frac{0,1}{(1+a^2)} \right[ .$$

\* l'erreur commise sur l'ordonnée b se situe sur l'intervalle:

$$[ 0 , 20 [ .$$

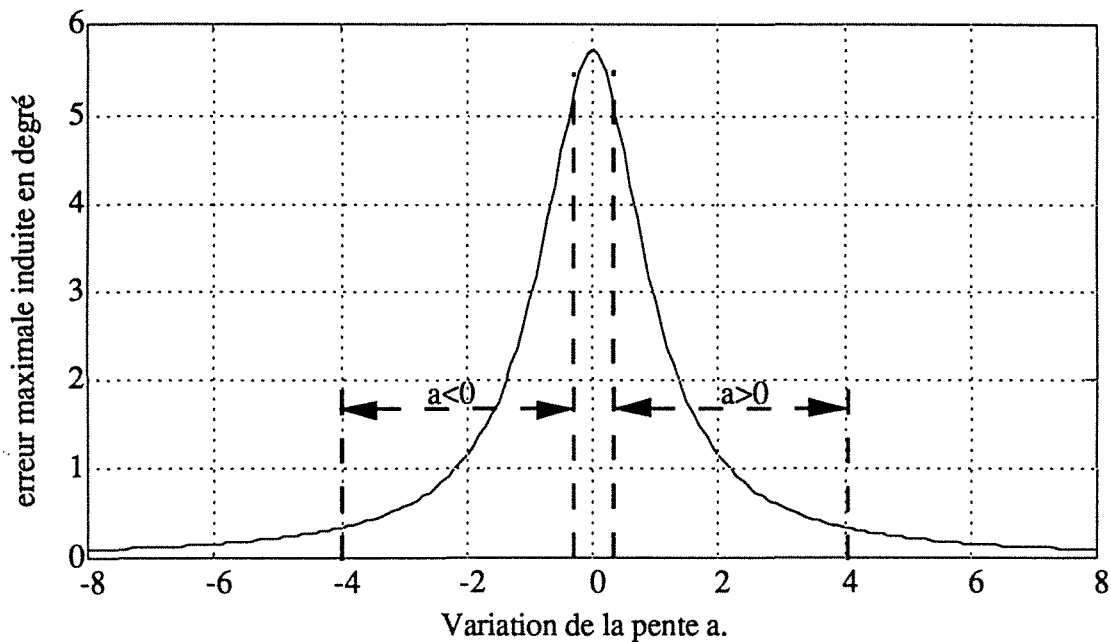


Figure III.2.2.2.5.

L'erreur maximale sur la pente a est de  $5,8^\circ$  quand  $a=0$ , Mais la pente a une forte probabilité d'être incluse dans les intervalles  $[-4 ; -0,2]$  et  $[0,2 ; 4]$  pour la plupart des bords de route, donc l'erreur commise est par conséquent incluse dans l'intervalle  $[0,3^\circ ; 5,3^\circ]$ .

En fait, l'erreur commise sur la localisation d'un segment de droite est la combinaison des deux erreurs commises sur la pente  $a$  et l'ordonnée à l'origine  $b$ . Ainsi puisque  $a$  et  $b$  sont fournis par un pic de l'espace de Hough, on peut conclure que leur valeur exacte  $a_{ex}$  sera limitée par les valeurs suivantes:

$$a - \frac{0,05}{(1+a^2)} \leq a_{ex} \leq a + \frac{0,05}{(1+a^2)}$$

et

$$b - 10 \leq b_{ex} \leq b + 10$$

Pour illustrer cette analyse considérons par exemple que la droite d'équation  $y = ax + b$  ait été détectée. La figure III.2.2.2.6 nous permet de déduire que les points qui ont contribué à la formation du pic de Hough correspondant à l'équation  $y = ax + b$  se situent dans un espace limité par les droites  $y = a'x + b'$  et  $y = a''x + b''$  avec :

$$a' = a + \frac{0,05}{(1+a^2)}$$

$$a'' = a - \frac{0,05}{(1+a^2)}$$

et

$$b' = b + 10$$

$$b'' = b - 10$$

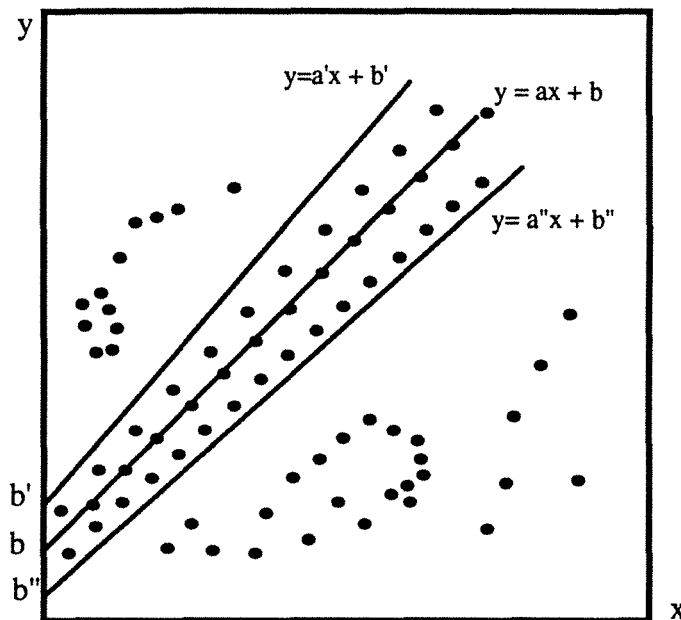


Figure III.2.2.2.6: détection de la droite d'équation  
 $y = ax + b$ .

En conséquence, on peut affirmer que le nombre de points appartenant à la droite détectée est au plus égale au nombre de points fournis par le pic de l'espace de Hough. Finalement, on constate que la sélection d'un bord de route à partir du pic de l'espace de Hough correspondant devient peu fiable.

Le traitement par la transformée de Hough fournit au plus 6 équations de la forme:

$$\text{bord droit} : \quad y = a_{Di} x + b_{Di} \quad 1 \leq i \leq 6$$

et 6 autres de la forme

$$\text{bord gauche} : \quad y = a_{Gi} x + b_{Gi} \quad 1 \leq i \leq 6$$

Une nouvelle approche nous a conduit à calculer pour chaque droite détectée trois grandeurs différentes :

- le nombre de points qui a contribué à la formation du pic de Hough qu'on note: `pic_hough`.

- le nombre de points qui appartiennent à la droite détectée qu'on note: pts\_droite.
- le nombre total de points de l'image binaire à partir de laquelle la droite a été détectée qu'on note: nbre\_tot.

Sur la figure III.2.2.2.6 :

$$\begin{aligned} \text{pic\_hough} &= 40 \\ \text{pts\_droite} &= 13 \\ \text{nbre\_tot} &= 77 \end{aligned}$$

On remarque que pic\_hough correspond au nuage de points autour de la droite détectée, de la même façon nbre\_tot correspond à la somme de tous les nuages de points de l'image binaire correspondant à une orientation donnée du gradient. A partir de ces trois grandeurs on définit les deux rapports suivants :

- %droite =  $100 \times \frac{\text{pts\_droite}}{\text{pic\_hough}}$ , qui exprime le pourcentage des points formant le pic de Hough qui appartiennent réellement à la droite.

- %orient =  $100 \times \frac{\text{pic\_hough}}{\text{nbre\_tot}}$ , qui exprime le pourcentage des points formant un nuage autour de la droite par rapport à l'ensemble des points de l'image.

La détection est la meilleure quand le maximum des nuages de points de l'image se trouvent autour de la droite détectée, et le maximum des points de ces nuages appartiennent à la droite. La première condition est satisfaite quand %orient est maximum, la deuxième quand %droite est maximum. En conséquence, la sélection des bords de route se base sur le maximum des deux pourcentages %orient et %droite. Pour prendre en compte ces deux grandeurs, on définit une variable,  $\omega$ , qui exprime le poids attribué à chaque segment détecté :

$$\omega = \%orient + \%droite.$$

Dans l'exemple de la figure III.2.2.2.6:

$$\%droite = 100 \times \frac{13}{40} = 32,5\%.$$

$$\%orient = 100 \times \frac{40}{77} = 51,95\%.$$

$$\omega = 84,45\%.$$

### III.2.2.3 - Résultats expérimentaux

Cette méthode a été testée hors-ligne sur station SUN 3-60 sous UNIX avec l'aide du logiciel de simulation VISILOG.

Les tableaux III.2.2.3.1 et III.2.2.3.2 suivant montrent les résultats numériques du traitement de deux scènes.

Scène 1:

pen	a	ordonné b	pic hough	pts droite	nbre tot	$\omega$ (%)
-0,5		425	125	13	4127	13,4288
-0,5		400	126	14	3523	14,6876
-1,4		1130	101	16	2937	19,2805
-0,6		415	153	25	4324	19,8783
-0,6		415	153	25	4324	19,8783
0,6		0	348	42	3967	20,8413
-0,5		330	145	26	3967	21,5862
0,6		0	354	49	4324	22,0287
0,7		105	117	24	2937	24,4965
0,6		-20	361	74	3523	30,7456
0,6		170	77	25	2544	35,4900
-1		280	524	111	2544	41,7807
0,7		-55	692	174	4127	41,9121

tableau III.2.2.3.1

Scène 2:

pen <sup>t</sup> e a	ordonné b	pic hough	pts droite	nbre tot	$\omega$ (%)
-0,5	400	126	14	3576	14,6346
-0,5	425	109	13	3524	15,0197
-1,4	1130	101	16	2936	19,2816
-0,6	415	153	25	4175	20,0004
0,7	0	364	36	3524	20,2193
0,6	0	318	40	3851	20,8362
-0,5	330	144	26	3851	21,7948
0,6	0	331	46	4175	21,8254
0,7	105	117	24	2936	24,4978
0,6	170	69	17	2065	27,9800
0,6	-20	377	81	3576	32,0279
-1	275	406	61	2065	34,6856

tableau III.2.2.3.2.

Le traitement d'une image de taille 512x512 représentant une scène typique de parc urbain demande en moyenne cinq minutes de temps machine. A elle seule, la transformée de Hough occupe plus de 60% de ce temps. En pratique, ce dernier peut être ramené à quelques dizaines de secondes par l'emploi de cartes spécialisées de traitement d'image réalisant les opérations de calcul du gradient et de la transformée de Hough en temps réel. En ce qui concerne les performances, la méthode détecte 90% des éléments linéaires dans une scène, y compris les bords de route. Les perturbations engendrées dans la linéarité des objets concernent en majorité les scènes à très faible contraste de luminance, ou comportant des ombrages qui masquent en grande partie les éléments recherchés. La majorité de ces problèmes sont résolus en établissant une coopération entre la discrimination et un modèle d'univers du robot établi à partir d'une base de connaissance (voir chapitre V).

### **III.3 - PHASE DE SUIVI DE CHEMIN EN CONTINU**

Généralement, il existe deux façons d'améliorer le temps d'exécution d'un algorithme : l'optimisation du code source afin de réduire le temps de traitement de chaque pixel, et la réduction du nombre de pixels à traiter par l'algorithme. La deuxième voie est beaucoup mieux adaptée à notre cas, car un traitement sélectif peut réduire d'une manière extraordinaire le temps d'exécution.

Le critère de sélectivité adopté utilise la relation qui existe entre l'orientation du gradient au niveau pixel, et l'orientation du segment de droite à détecter. Ainsi, les pixels ayant des orientations exclues de la plage des orientations recherchées du bord de route ne seront pas pris en compte par l'algorithme. Par conséquent, le temps de traitement se trouve amélioré par un facteur non-négligeable. Puisque la réduction du temps de traitement est notre objectif, on a également limité le traitement dans ce mode à des fenêtres. Celles-ci doivent, en principe, inclure une portion du bord de route. Le meilleur moyen pour y parvenir est de placer les fenêtres d'une façon séquentielle.

#### **III.3.1 - Placement des fenêtres initiales**

La position d'une fenêtre dépend fortement des résultats du traitement de la fenêtre précédente. Par conséquent, le positionnement des fenêtres initiales sur la scène nouvellement acquise, est critique, car leur emplacement correct dépend de la présence des bords de route .

L'importance de l'existence des bords de route dans les fenêtres initiales les oblige à avoir une taille convenable. Une taille trop importante des fenêtres tend à inclure d'autres éléments linéaires qui peuvent fausser la détection du bord de route. Par contre, une taille trop petite risque de ne pas refléter la linéarité du bord de route. On a donc adopté une taille de 50x50 pixels ( voir figure III.3.1.1). Le positionnement des fenêtres initiales se trouve facilité par l'utilisation de l'information fournie par le prédicteur de scène. En effet, dans ce mode, le module de prédiction de scène est utilisé avant celui de

l'exploitation de la scène. Donc en choisissant le mode " suivi continu", le superviseur est déjà en possession de la position des fenêtres initiales. Celle-ci est spécifiée par les coordonnées  $(x_{fg}, y_{fg})$  du point gauche du coté supérieur de chaque fenêtre ( voir figure III.3.1.1).

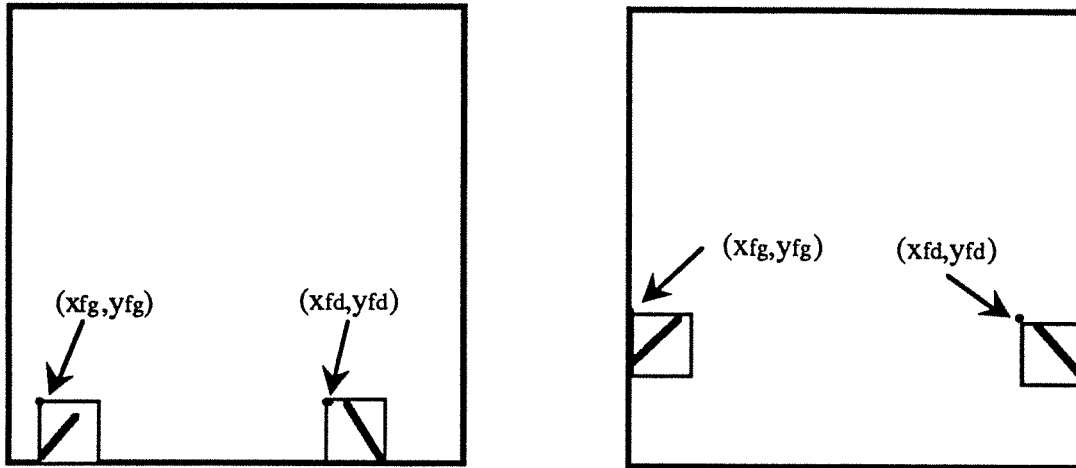


Figure III.3.1.1: positionnement des fenêtres suivant l'apparition des bords de route.

### III.3.2 - traitement en mode de suivi continu

Pour extraire les bords de route à l'intérieur des fenêtres, le traitement diffère légèrement de celui appliqué dans le mode "phase initiale". Cette différence réside dans l'exploitation des connaissances à priori, et de certaines propriétés locales propres aux fenêtres. On constate que le rapport signal/bruit est inversement proportionnel à la longueur de l'élément linéaire recherché. Donc l'opération de réduction de bruit dans ce mode n'améliore pas la détection. Une fenêtre correctement placée est sensée inclure au moins un bord de route comme élément linéaire. La présence dans la fenêtre d'autres éléments comme des objets, des nappes d'eau, ou des défauts de la structure de la route ne peut nuire à la détection à condition qu'elle ne masque pas complètement le bord. D'autre part, les ombres ne causent absolument pas de fausse détection au niveau fenêtre, puisque l'orientation du bord est globalement connue à partir du traitement de la fenêtre précédente.



On a remarqué sur le traitement des fenêtres des différentes scènes que les bords de route peuvent être classés en deux catégories:

a) les bords qui constituent une région de structure différente de celles des régions qu'ils séparent ( voir figure III.3.2.3.a ).

b) les bords qui forment une transition entre deux régions de structures différentes (voir figure III.3.2.3.b )

Pour plus de précision dans la détermination du bord de route dans le deuxième cas, on a choisi de le représenter par la transition entre la surface de la route et celle de l'autre région.

La connaissance à priori de l'orientation du bord de la route ainsi que sa présence dans la fenêtre rend inutile la détermination de la direction du gradient avec une grande précision . Ainsi, cette hypothèse nous permet d'exploiter directement l'orientation du gradient sans faire de reconstruction. L'opérateur qui nous a paru le plus convenable est le gradient boussole à 5 niveaux ( figure III.3.2.1 ).

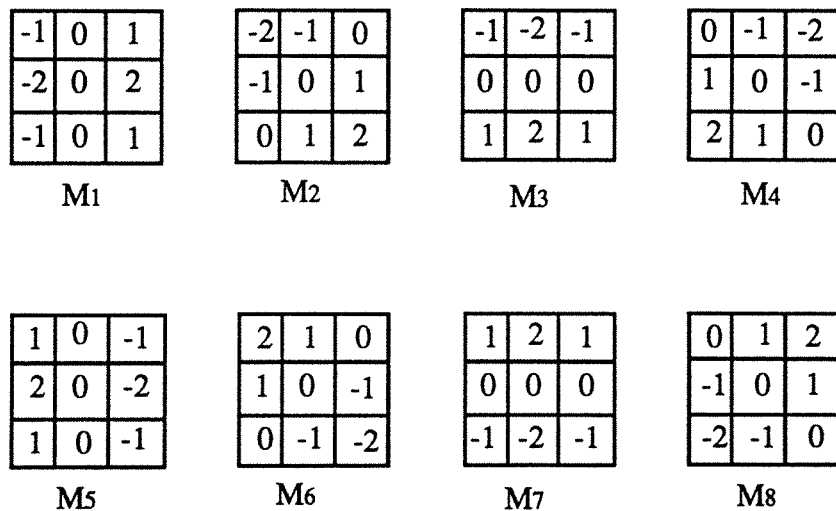


Figure III.3.2.1 : Masques de convolution du gradient boussole.

Un ensemble de 8 masques est ainsi défini, par rotation du masque générique par pas de 45° autour de son point central. Si on désigne par  $G_k$  la convolution de  $M_k$  avec la fenêtre  $f_i$ , le gradient résultant est donnée par :

$$G(i,j) = \text{MAX} \{ G_k(i,j) \} \quad \text{pour } k = 1, 2, 3, \dots, 8$$

et sa direction par :

$$O(i,j) = d \text{ tel que } \text{MAX}\{ G_k(i,j) \} = G_d(i,j) \quad \text{pour } k = 1, 2, 3, \dots, 8$$

L'orientation du gradient est donc quantifiée suivant les huit directions principales du masque de convolution. Comme pour le traitement en phase initiale, les directions du gradient considérées sont incluses dans l'intervalle  $[ 0^\circ, 180^\circ ]$ . On applique donc un seuillage de l'image orientation du gradient pour extraire les cinq directions  $0^\circ, 45^\circ, 90^\circ, 135^\circ$  et  $180^\circ$  ( voir figure III.3.2.2 ).

On obtient donc cinq images binaires  $pf_1, pf_2, pf_3, pf_4$  et  $pf_5$  qui mettent en évidence les éléments orientés suivant ces cinq directions. La figure III.3.2.3.a montre la fenêtre et les images binaires résultantes pour un bord de route de catégorie a ( formant une région ). L'image binaire  $pf_2$  permet de déduire la direction dominante correspondant au bord de route.

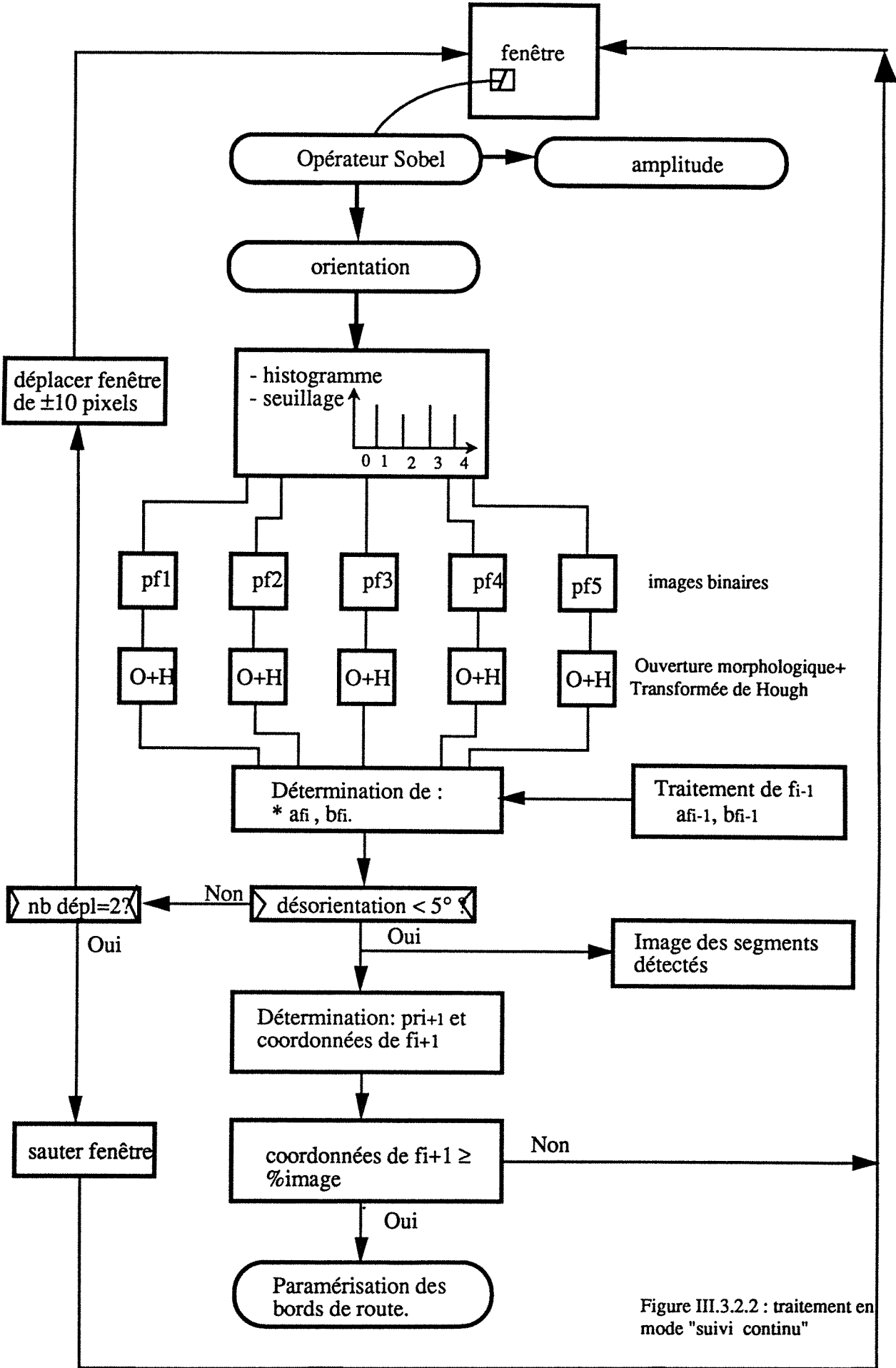
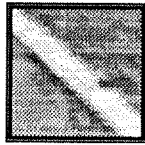


Figure III.3.2.2 : traitement en mode "suivi continu"



bord de route de catégorie a.



pf1

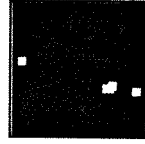


pf2

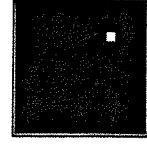


pf3

a



pf4



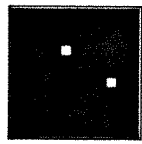
pf5



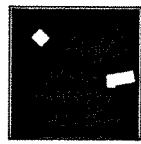
bord de route de catégorie b.



pf1



pf2

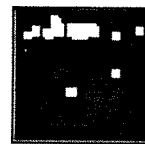


pf3

b



pf4



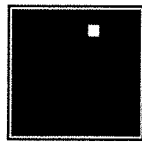
pf5



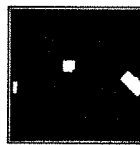
bord de route de catégorie b + ombre d'arbre.



pf1

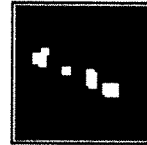


pf2



pf3

c



pf4



pf5

Figure III.3.2.3

La figure III.3.2.3.b montre le traitement d'une fenêtre comportant un bord de catégorie b ( formant une transition ), et  $pf_4$  reflétant son orientation. Enfin, la figure III.3.2.3.c montre le traitement pour un bord de même catégorie, mais dont une partie est sous l'ombre d'un arbre. On remarque que sa direction est donnée par  $pf_4$  tandis que  $pf_5$  montre la direction de l'ombre. La discrimination de la direction du bord de la route se fait par la connaissance de la direction du bord détecté dans la fenêtre précédente.

Pour positionner la fenêtre suivante on utilise le résultats du traitement de la fenêtre en cours d'étude. En effet, supposons que la transformée de Hough produise pour une fenêtre i une équation  $y_{fi} = a_{fi} x + b_{fi}$ . Pour assurer un suivi robuste et maximiser l'apparition du bord de la route dans la fenêtre i+1, la moitié du bord détecté dans la fenêtre i est incluse dans celle-ci. On détermine donc le point milieu de ce segment de droite et on le prend comme référence de cette fenêtre qu'on note  $pr_i(x_{ri}, y_{ri})$  ( voir figure III.3.2.4 ) .

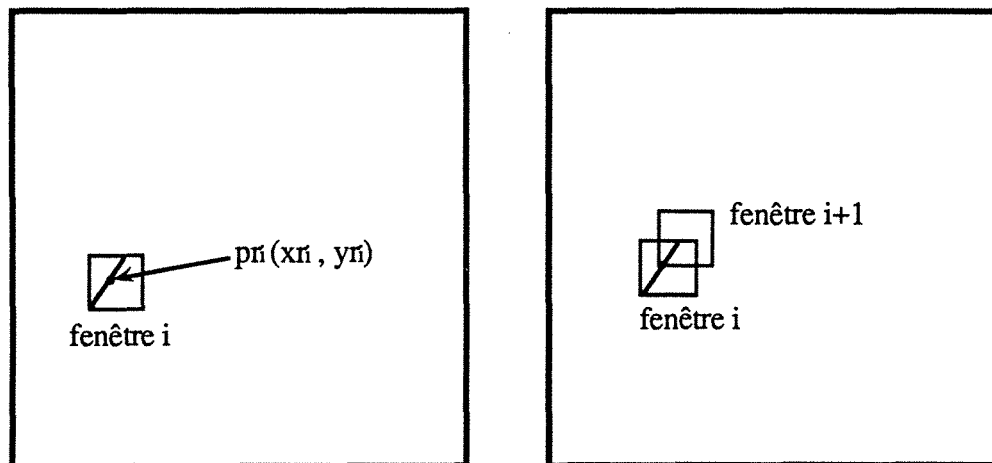


Figure III.3.2.4 : placement séquentiel des fenêtres.

La validité des bords détectés dans les fenêtres positionnées est vérifiée directement par la contrainte naturelle de continuité des bords de route. Par conséquent, si

dans la fenêtre  $i+1$  le bord détecté est décrit par l'équation :  $y_{fi+1} = a_{fi+1} x + b_{fi+1}$  alors le point de référence  $p_i (x_i, y_i)$  de la fenêtre  $i$  doit être solution de cette équation.

Donc

$$y_i = a_{fi+1} \cdot x_i + b_{fi+1}$$

Ceci indique que  $p_i$  doit toujours être le point de départ du segment détecté dans la fenêtre  $i+1$ .

La transformée de Hough est alors appliquée aux cinq images  $p_{fi}$  de la fenêtre, (après une ouverture morphologique pour éliminer les points isolés) pour obtenir l'équation du segment de droite passant par le maximum de points. Le choix des paramètres  $a_{fi}$  et  $b_{fi}$  de l'équation du bord de route dans la fenêtre  $i$ , parmi les quatre possibilités, est fait par comparaison avec les paramètres  $a_{fi-1}$  et  $b_{fi-1}$  ( voir figure III.3.2.2 ). Pour maximiser le champ de recherche des bords de route, il faut essayer d'avoir une longueur maximale de bord incluse dans la fenêtre et permettre un changement d'orientation suffisante pour détecter les différents profils de bords de route existant dans le milieu naturel ( virage, montée, ...,etc). On autorise donc une variation d'orientation maximale de  $5^\circ$  autour l'orientation détectée dans la fenêtre précédente (voir figure III.3.2..5 ).

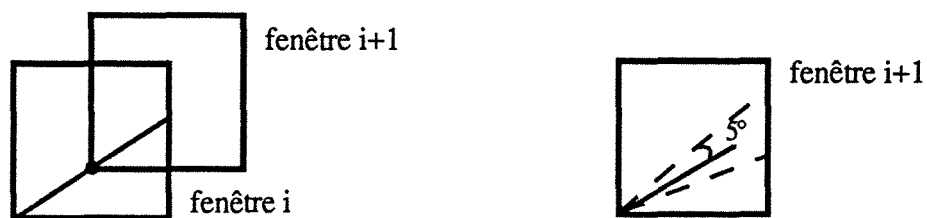


Figure III.3.2.5 : déviation maximale acceptée d'une fenêtre à l'autre.

Si la différence entre l'orientation actuelle et l'orientation précédente dépasse  $5^\circ$  , il y a deux possibilités :

1- Il existe dans la fenêtre un élément dont la linéarité domine celle du bord de route. Dans ce cas, la fenêtre est déplacée deux fois de 10 pixels vers la droite puis vers la gauche et la recherche est reprise pour chaque cas ( figure III.3.2.6.a ). Ce déplacement permet de diminuer l'importance de la linéarité de cet élément en réduisant la place qu'il occupe dans la fenêtre. Ceci permet en général d'accentuer la prédominance du bord.

2- Le bord est inexistant, soit naturellement ( discontinuité ), soit qu'un objet le masque complètement . Dans ce cas les deux déplacements ne permettent pas d'aboutir à l'orientation désirée. La seule solution est de sauter cette fenêtre et de supposer que la détection a réussi pour pouvoir placer la fenêtre suivante. La position de celle-ci sera le milieu du bord supposé correctement détecté ( figure III.3.2.6.b ).

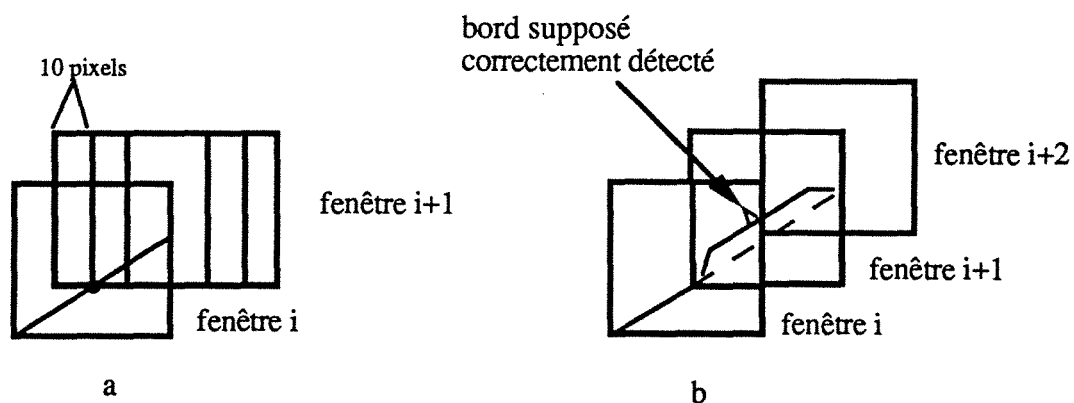


Figure III.3.2.6 : cas de discontinuité du bord de la route.

Quand les paramètres sont sélectionnés, l'image binaire correspondante est utilisée pour construire une image finale qui inclut toutes les images binaires qui ont fourni l'orientation désirée du bord. A la fin du traitement, cette image globale décrit les bords de route et est utilisée pour la détermination du modèle de la scène ( voir figure III.3.2.7 ).

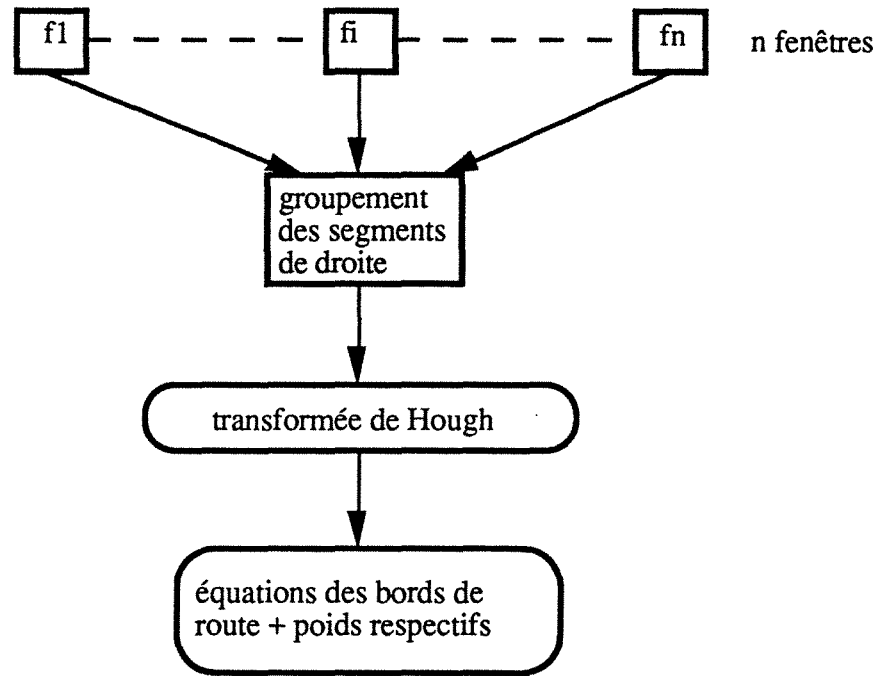


Figure III.3.2.7 : paramétrisation des bords de la route.

La figure III.3.2.8 montre le placement séquentiel des fenêtres en mode de suivi continu pour une route simple (a) et une route ayant une discontinuité du bord gauche (b).

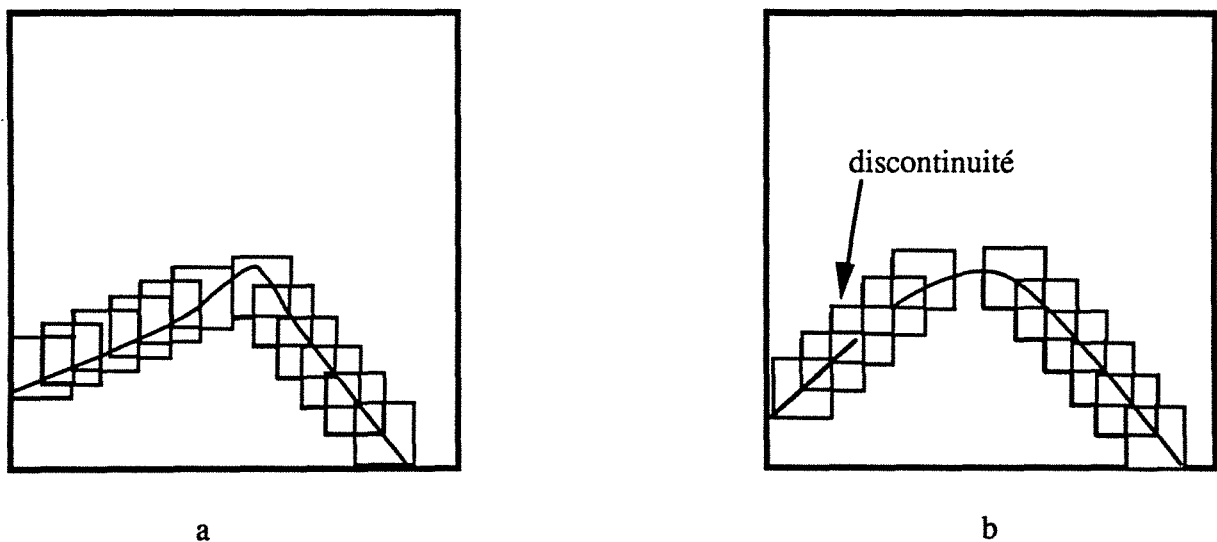
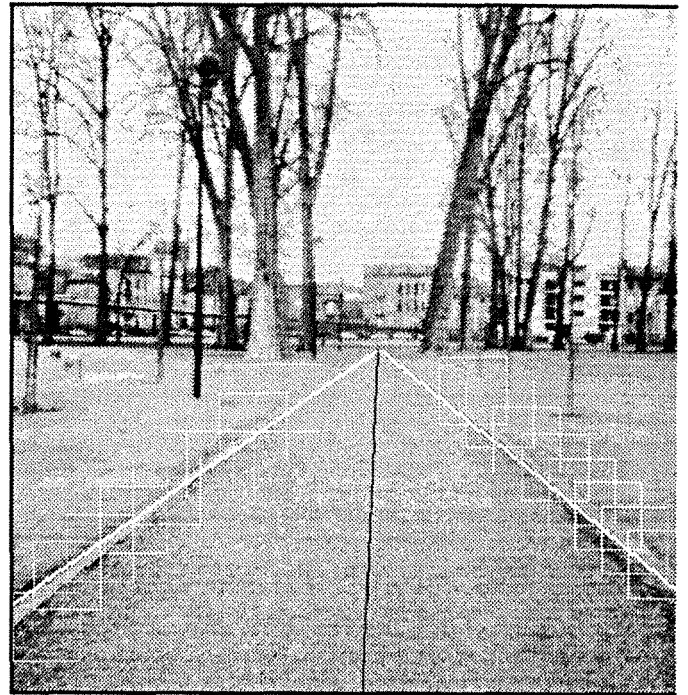
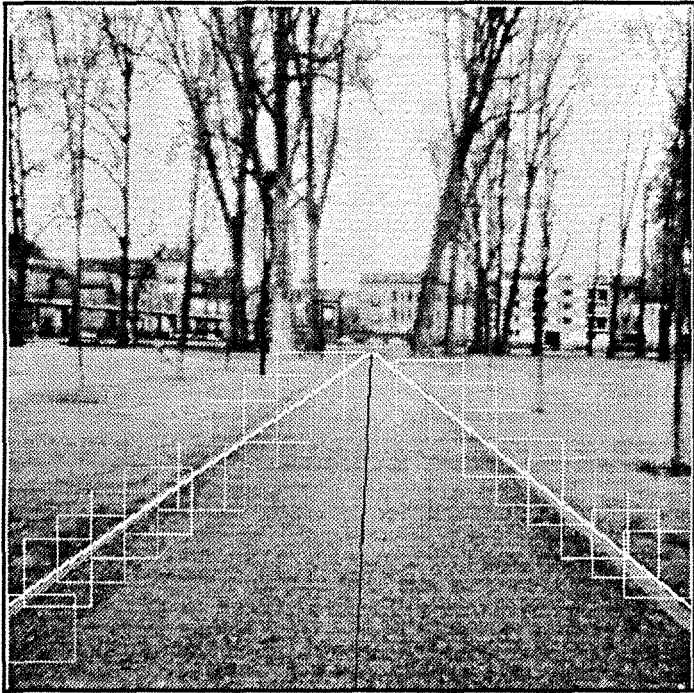
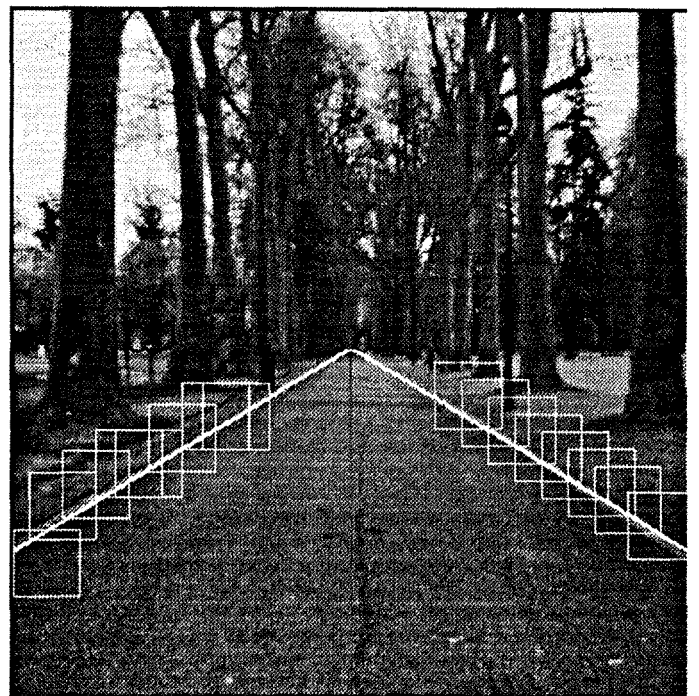
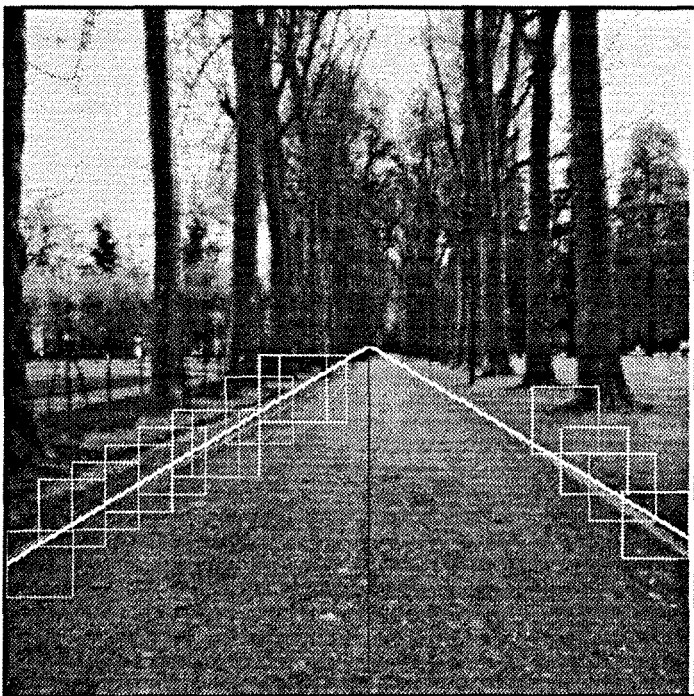


Figure III.3.2.8 : traitement en phase de suivi continu.

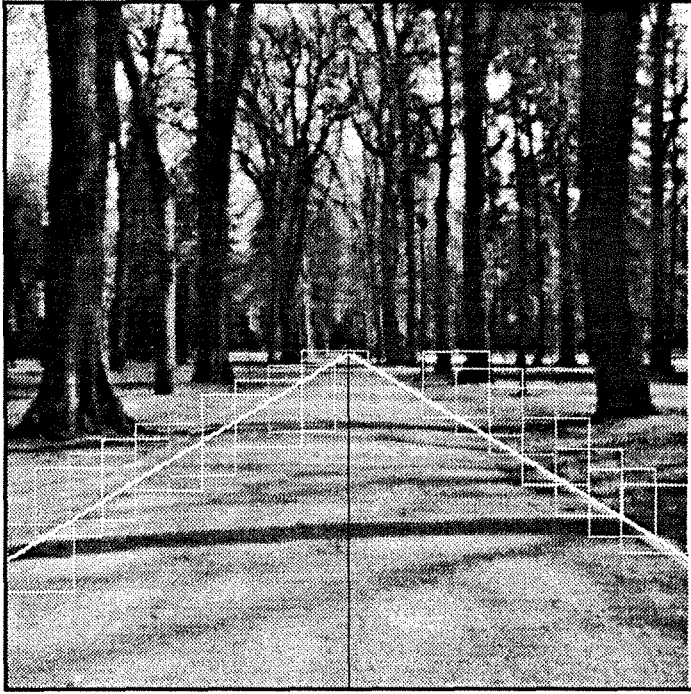




Traitement en mode de suivi continu d'une séquence de deux scène



Traitement en mode de suivi continu d'une séquence de deux scène



Traitement en mode de suivi continu d'une scène comportant des ombres d'arbres

Le positionnement des fenêtres s'achève si l'un des deux cas se présente :

1°) Deux sauts successifs de fenêtre indiquent la disparition du bord de route.

2°) La position de la fenêtre a atteint un certain pourcentage de l'image.

Dans notre cas ce pourcentage est 60%. En fait, ce pourcentage varie en fonction de la déviation verticale de l'axe de la caméra et de son altitude  $h$ . Ainsi, il est d'autant plus important que l'axe de la caméra est plus orienté vers le bas.

Une fois le traitement en phase de suivi continu est achevé, les équations des bords de routes ainsi que leur poids seront communiqués à la base de connaissance via le superviseur afin d'établir le modèle de la scène.

### **III.4- CONCLUSION**

L'initialisation est une tâche difficile et essentielle pour entamer la navigation autonome. Malgré le temps de traitement de plusieurs minutes nécessaire à l'accomplissement de celle-ci, les résultats obtenus montrent les possibilités intéressantes qu'offre la vision artificielle dans le guidage de robot mobile. Il faut noter que ce temps d'exécution un peu long n'est requis qu'une fois au début de la navigation. La phase de suivi continu garde presque les mêmes possibilités de détection que la phase initiale, mais elle est plus focalisée vers des zones de forte probabilité d'existence des bords de route. Mais, sa réussite dépend de la réussite de la phase initiale en collaboration avec la cartographie.



**CHAPITRE IV**

**CONTRIBUTION DE LA  
CARTOGRAPHIE A LA  
NAVIGATION AUTONOME.**

## IV.1- INTRODUCTION

La navigation autonome basée sur l'utilisation de l'information visuelle constitue un domaine d'intérêt très important pour la conduite des robots mobiles dans divers milieux. Les travaux menés actuellement ont pour objectif une navigation robuste et plus rapide. La robustesse est limitée par l'incertitude associée à l'information fournie par la caméra et le traitement qui l'accompagne. D'autre part la vitesse de la navigation est considérablement réduite par la quantité des informations à traiter par unité de temps. Une connaissance à priori de l'environnement permet d'être moins dépendant des données acquises par la caméra durant l'exécution d'un plan de navigation. En plus la perception et le traitement associé ne seront nécessaires que pour vérifier périodiquement si le robot suit correctement le chemin planifié. En conséquence, avec une moins grande contrainte sur la perception, la vitesse de navigation sera plus importante, et de la même manière, avec une connaissance plus précise du milieu, la navigation sera plus robuste.

Une carte détaillée de l'environnement peut fournir au système de vision la position et l'apparence de la route et des objets qui l'entourent. Cet apport facilite énormément le traitement des informations acquises. L'utilisation d'une carte détaillée du milieu ne peut aboutir à un système complètement autonome, que si le mobile est doté d'un système de localisation très précis dans un milieu intégralement statique. Néanmoins, les prédictions faites à partir d'une carte, associées à la contribution d'un système de localisation assez précis, constituent un outil très puissant pour planifier le type de traitement approprié à une situation, et définir des contraintes sur l'interprétation de la scène.

Martial Hebert a présenté un algorithme pour la construction des cartes décrivant la route à partir d'une scène acquise par un télémètre laser ERIM [HEB-89]. Cette carte est locale et est utilisée pour traverser uniquement une portion de l'univers déjà explorée. Sous forme de fichier elle ne fournit que l'information sur la topographie et la structure des routes. Le système de vision de l'université de Maryland a été équipé d'une telle carte [DIC-87].

Dans ce chapitre, l'aspect prédiction de scène à partir des différents paramètres est présenté dans la première partie. La deuxième traite de la construction de la carte ainsi que de son utilisation pour interpréter les primitives extraites à partir d'une scène.

## IV.2 - MODELISATION DE LA CAMERA

Un modèle simple de la caméra appelé "pinhole" est montré sur la figure IV.2.1. La lentille est placée à une distance  $f$ , appelée distance focale, du plan de l'image. Puisque les rayons lumineux provenant des différents points de l'objet doivent passer par le centre de la lentille ou centre de projection pour atteindre le plan image, la transformation des points de l'objet en points de l'image est appelée transformation perspective.

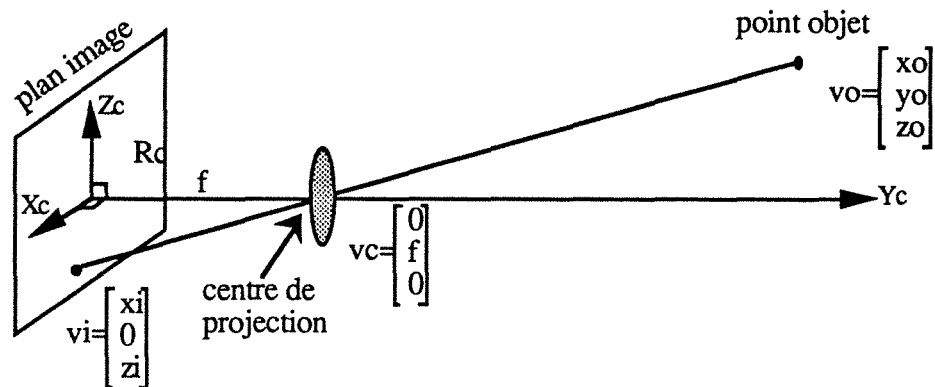


Figure IV.2.1 : Modèle élémentaire de la caméra.

En outre, puisque les points en trois dimensions (3-D) sont projetés sur le plan à deux dimensions (2-D) de l'image, la transformation est appelée perspective-projective. Un point objet sera transformé en un seul point image; en conséquence, étant donné les coordonnées d'un point objet et une description de la lentille, on peut déterminer les coordonnées du point correspondant sur le plan de l'image. Néanmoins, les points image ne sont inversement projetés que le long d'une ligne droite. Ainsi, une autre information comme la distance de la lentille à l'objet doit être fournie pour déterminer le point objet

unique correspondant.

La transformation perspective reliant les points image aux points objet est déterminée comme suit : on considère le modèle établi sur la figure IV.2.1 où l'origine du système de coordonnées  $R_c$  de la caméra est centré sur le plan de l'image et la lentille placée à une distance  $f$  sur l'axe  $y_c$  perpendiculaire à ce plan: Les points sur la droite liant le point image au centre de projection sont décrits par le vecteur:

$$v = v_c + \alpha ( v_i - v_c ) = \alpha v_i + ( 1 - \alpha ) v_o$$

où  $\alpha$  est un paramètre proportionnel à la distance entre  $v_c$  et  $v$ . Puisque l'origine du repère correspond au centre du plan image on a :

$$v_c = \begin{pmatrix} 0 \\ f \\ 0 \end{pmatrix} \quad \text{et} \quad v_i = \begin{pmatrix} x_i \\ 0 \\ z_i \end{pmatrix}$$

Puisque le point objet  $v_o$  appartient à cette droite, il doit satisfaire l'équation :

$$v_o = v_c + \alpha_o ( v_i - v_c ) \Rightarrow \begin{pmatrix} x_o \\ y_o \\ z_o \end{pmatrix} = \begin{pmatrix} 0 \\ f \\ 0 \end{pmatrix} + \alpha_o \begin{pmatrix} x_i \\ -f \\ z_i \end{pmatrix}$$

il en résulte: 
$$x_i = \frac{f \cdot x_o}{f - y_o} , \quad y_i = 0 , \quad z_i = \frac{f \cdot z_o}{f - y_o}$$

Ceci montre qu'on peut déterminer les points image connaissant les coordonnées des points de l'objet. D'autre part, connaissant les coordonnées des points image et la distance focale  $f$  , on peut déterminer la droite contenant le point objet par l'intersection des plans décrits par les équations suivantes:



$$\frac{x_o}{x_i} = \frac{z_o}{z_i} = \frac{f-y_o}{f}$$

En utilisant le modèle indiqué sur la figure IV.2.1, l'objet apparaît inversé sur le plan de l'image . Pour éviter cela, il est souvent convenable d'utiliser le modèle de projection indiqué sur la figure suivante:

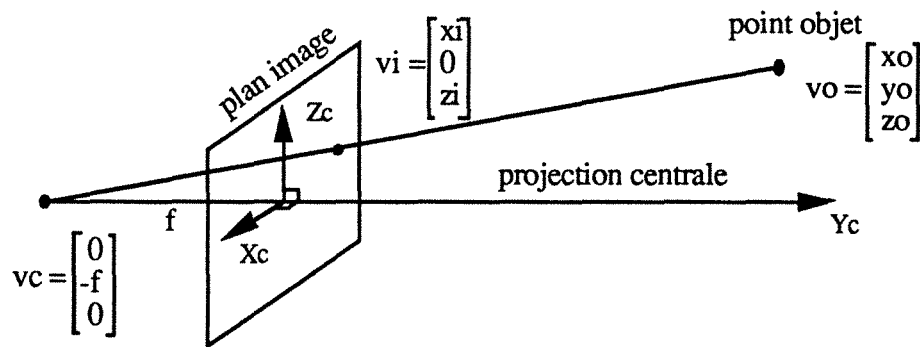


Figure IV.2.2 : Modèle de projection centrale.

Les points de la droite liant le centre de lentille et le point image doivent satisfaire :

$$v = v_c + \alpha (v_i - v_c)$$

puisque  $v_o$  est sur cette droite on a :

$$v_o = v_c + \alpha_o (v_i - v_c) \Rightarrow \begin{pmatrix} x_o \\ y_o \\ z_o \end{pmatrix} = \begin{pmatrix} 0 \\ -f \\ 0 \end{pmatrix} + \alpha_o \begin{pmatrix} x_i \\ f \\ z_i \end{pmatrix}$$

d'où :  $x_i = \frac{x_o \cdot f}{f + y_o}$  ,  $y_i = 0$  ,  $z_i = \frac{z_o \cdot f}{f + y_o}$

Cette transformation perspective directe est très simple à utiliser sauf qu'elle est non-linéaire. Pour la rendre linéaire on exprime les vecteurs en coordonnées homogènes. Les coordonnées homogènes  $\hat{a}$  d'un vecteur  $a = (x, y, z)$  sont  $\hat{a} = (\omega x, \omega y, \omega z, \omega)$  où  $\omega$  est une constante arbitraire. Evidemment, connaissant les coordonnées homogènes,  $\hat{a}$ , d'un point, on peut obtenir ses coordonnées non homogènes en divisant simplement par  $\omega$  toutes les composantes et en sélectionnant les trois premières. On note aussi que les vecteurs non homogènes  $\hat{a}$  et  $k\hat{a}$  correspondent physiquement au même point  $\hat{a}$ .

#### IV.2.1-Transformation perspective en coordonnées homogènes:

L'utilité des coordonnées homogènes est qu'elles rendent la transformation perspective linéaire. Si on considère un point objet de l'espace, exprimé dans le repère de la caméra  $R_c$  :

$$v_o = \begin{pmatrix} \omega \cdot x_o \\ \omega \cdot y_o \\ \omega \cdot z_o \\ \omega \end{pmatrix}.$$

alors la transformation linéaire définie par la matrice suivante :

$$T = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 1 & 0 & f \end{pmatrix}$$

le transforme en point image  $\hat{v}_i = T \cdot \hat{v}_o$  dont les coordonnées sont données par :

$$\begin{pmatrix} x_i \\ y_i \\ z_i \\ \omega_i \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 1 & 0 & f \end{pmatrix} \begin{pmatrix} x_o \\ y_o \\ z_o \\ 1 \end{pmatrix}.$$

soit ;

$$\hat{v}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \omega_i \end{pmatrix} = \begin{pmatrix} f \cdot x_o \\ f \cdot y_o \\ f \cdot z_o \\ (y_o + f) \end{pmatrix}.$$

Les coordonnées physiques sur le plan de l'image sont obtenues en divisant les trois premières composantes par le facteur d'échelle  $\omega_i$  :

$$v_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} \frac{f \cdot x_o}{(y_o + f)} \\ \frac{f \cdot y_o}{(y_o + f)} \\ \frac{f \cdot z_o}{(y_o + f)} \end{pmatrix}.$$

Les composantes  $x_i$  et  $z_i$  correspondent exactement aux expressions obtenues précédemment. La deuxième composante  $y_i$ , qui est supposée nulle, ne peut être prise pour une coordonnée physique. Par contre, elle doit être considérée comme un paramètre qui est proportionnel à la distance le long de l'axe de projection du point objet au centre de la lentille. Le calcul de ce paramètre est souvent inutile, sa détermination permet à la transformation T d'être une matrice carrée et inversible.

IV.2.2- Les paramètres de la configuration Robot-caméra.

Un robot mobile utilisant un système de vision pour assurer sa navigation autonome doit être doté d'une caméra comme le montre la Figure IV.2.2.1.

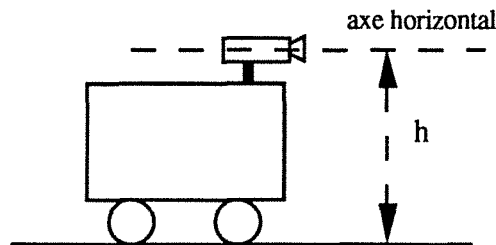


Figure IV.2.2.1 : caméra montée sur le robot.

La hauteur  $h$  de la caméra est supposée fixe pour une configuration donnée. En réalité, la variation de la hauteur  $h$  contribue aussi au changement de l'apparence de la scène sur l'image. Deux autres paramètres importants sont l'angle de déviation horizontale  $\Theta$ , l'angle de déviation verticale  $\Phi$  (voir figure IV.2.2.2).

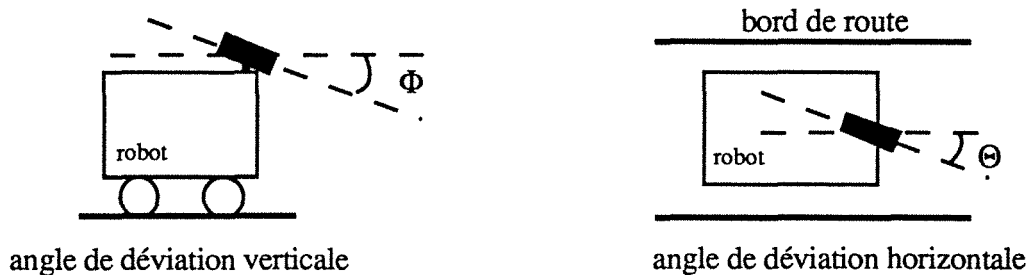


Figure IV.2.2.2 : paramètres considérés.

Les paramètres  $\Theta$  et  $\Phi$  sont fixés par programme. En effet la caméra, peut être orientée dans une direction donnée, par rotations autour de l'axe horizontal et vertical de la plate-forme. Les valeurs  $\Theta$  et  $\Phi$  sont alors traduites en signaux de commande des moteurs de la plate-forme qui assure les rotations. Ce processus est indispensable pour la

recherche des objets dans l'environnement du robot, ils doivent cependant être mesurés par des capteurs angulaires pour le recalage ou la navigation.

### IV.3 - DETERMINATION DES BORDS DE ROUTE SUR L'IMAGE

Le problème est de faire une prédiction sur l'apparence de la route à partir d'une image connaissant certains paramètres de la configuration robot-caméra, et d'étudier la variation de cette apparence en fonction de trois types de facteurs:

1) Variations dues à l'opérateur qui provoquent une modification de la distance focale  $f$ , ou de la hauteur  $h$  de la caméra.

2) Variations dues au programme qui agissent au niveau des variables  $\Theta$  et  $\Phi$  pour permettre une recherche d'un objet quelconque autour du robot.

3) Variations naturelles dues, à l'état de la route qui engendre des angles d'inclinaison du robot dans le cas d'une montée, d'une descente, ou d'un dévers .

#### IV.3.1 - Représentation de la carte

L'objectif global est le développement d'un système d'interprétation de l'image qui puisse coordonner et commander l'analyse et la construction du modèle de la scène au niveau de la base de connaissance. Il doit être capable de gérer les informations fournies par la carte et la caméra pour assurer les phases d'interprétation de scènes. Ainsi, la gestion des informations fournies par la carte doit permettre :

\* l'utilisation des contraintes spatiales pour contribuer à la formation des hypothèses durant l'analyse.

\* l'utilisation du modèle de la caméra pour permettre la projection des informations cartographiées sur le plan de l'image.

\* l'utilisation des modèles métriques pour déterminer la forme, la taille, la distance, et la position relative et absolue des objets.

Un milieu tel que le parc urbain est une organisation très complexe de structures étalées sur une surface assez importante. L'arrangement spatial des structures typiques telles que les routes, les bancs, les arbres, les constructions, etc., varie considérablement d'un parc à un autre. Néanmoins, les types de structures rencontrées sont assez bien connues.

Les techniques de traitement d'images sont traditionnellement entachées d'erreurs. Spécialement pour les scènes naturelles, les résultats de la segmentation ou de la détection des éléments linéaires sont parfois assez ambiguës puisque la correspondance entre les éléments détectés et les objets physiques est surjective. Les frontières entre les différents objets peuvent ne pas être distinguées à cause de l'occlusion, la similarité des propriétés spectrales, et la résolution intrinsèque de l'image. En conséquence, l'hypothèse qu'une primitive détectée, correspond directement à un objet physique est peu fiable; sauf si on procède à la reconstruction de l'objet original. En plus, l'analyse spatiale basée sur les sémantiques : au-dessus, au-dessous, à-droite, à-gauche, supérieur-à, etc., n'est pas assez rigoureuse pour l'interprétation générale d'une scène. Ces aspects nous ont encouragé à utiliser le principe de l'interprétation basé sur l'imagerie aérienne. Dans ce cas l'analyse spatiale est faite dans le repère de la carte au lieu de celui de l'image.

Pour mettre en évidence l'utilité métrique, le modèle du monde doit impérativement être lié à l'image analysée. Ceci est fait à l'aide de la correspondance image-carte en utilisant le modèle de la caméra. Plusieurs travaux ont été faits dans ce domaine, notamment celui de Brook basé sur le système ACRONYM [BRO-81] et celui de Hirose sur TITAN III [HIR-90]. Nagao a présenté un système de segmentation et d'interprétation des images aériennes infra-rouges contenant des routes, rivières et constructions [NAG-79]. Sa méthode se base sur l'adjacence des régions et les propriétés multi-spectrales. Un autre système appelé SPAM a été décrit par McKeown pour l'interprétation des images aériennes des scènes d'un aéroport [HSI-92] et [MCK-85].

Ainsi notre carte ressemble à une image aérienne de l'environnement prise à partir d'une certaine altitude. En fait, elle est construite en se basant sur un plan de l'environnement. L'objectif est d'obtenir une image sur laquelle sont dessinés tous les chemins du milieu ambiant, comportant certains objets utiles à la navigation. Ainsi des objets comme les arbres, murets, bancs...etc seront modélisés par des segments, lignes, points, cercles et ellipses. L'intérêt de fabriquer une image à partir d'une carte est de pouvoir attribuer une troisième dimension aux objets (la hauteur).

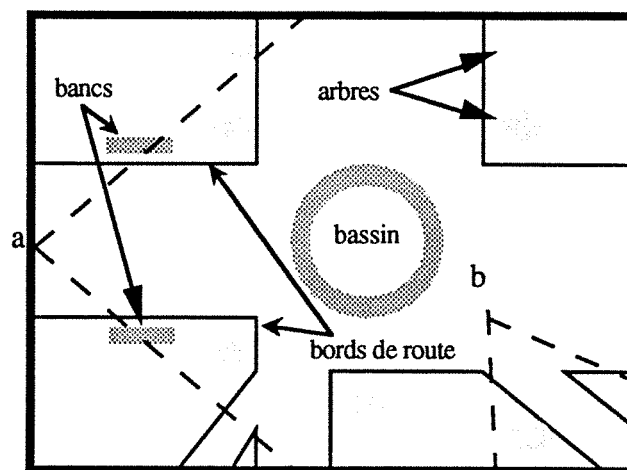


Figure IV.3.1.1 : exemple de carte.

La technique utilisée pour représenter la hauteur des objets est de forcer les niveaux des pixels appartenant aux primitives de l'objet à une valeur proportionnelle à leur hauteur géométrique. Par exemple si le pixel est codé sur 256 niveaux et si un mètre est représenté par 20 niveaux, les pixels représentant un arbre de 5 mètres auront un niveau de 100. Les segments formant les bords de route auront le plus bas niveau de pixel. Comme exemple la figure IV.3.1.1 représente un carrefour où les segments représentent les bords de route, les petits cercles représentent les arbres, les deux rectangles représentent les bancs et enfin, les deux cercles concentriques représentent l'entourage d'un bassin.

On remarque que le degré de précision de la représentation plane d'un objet

statique de l'environnement est proportionnel à la résolution utilisée. Evidemment, on représente avec plus de précision un objet, appartenant à une région de surface  $S$ , cartographiée sur une image de taille  $d \times d$  pixels qu'une image de  $d' \times d'$  pixels où  $d'$  est nettement inférieur à  $d$ . De la même manière, sur une image de  $d \times d$  pixels, un objet appartenant à une région  $S1$  et  $S2$  ( $S1$  incluse dans  $S2$ ) sera mieux représenté si  $S1$  est cartographiée au lieu de  $S2$ .

La représentation d'une zone de l'environnement de surface  $S = L \times L \text{ m}^2$  sur une image de  $d \times d$  pixels donne une résolution  $r$  exprimée par le rapport suivant :

$$r = \frac{d}{L} \quad (\text{pixels/mètre}).$$

En utilisant cette méthode de représentation de la carte, un grand nombre d'objets peuvent être inclus dans l'image pour décrire l'environnement. A cause des dimensions limitées de l'image à  $512 \times 512$ , une bonne résolution ne peut être obtenue que si les dimensions de la région à cartographier diminuent. Ceci limite considérablement la surface de la zone à cartographier. Cet handicap est surmonté en subdivisant l'environnement en régions successibles d'être cartographier sur une image unique avec une résolution acceptable. Ceci augmente le nombre d'images à utiliser et malheureusement la taille de l'espace mémoire requis.

Connaissant la surface de l'environnement à cartographier et la résolution désirée, le nombre des images peut être déterminé. Par conséquent, on aboutit à un ensemble d'images qui décrit d'une façon relativement précise l'environnement, à condition que celles-ci soient rangées et adressées dans un ordre correct. Pour cela on repère chaque image de la carte réelle par un nombre  $n_{ij}$  comme indiqué par la figure IV.3.1.2 .



		i			
		0	1	2	3
j	0	$n_{00}$	$n_{10}$	-	-
	1	$n_{01}$	$n_{11}$	-	-
	2	-	-	-	-
	3	-	-	-	$n_{33}$

Figure IV.3.1.2 : séquence des images coorespondant à la carte.

Connaissant la résolution  $r$  (pixels/mètre), les dimensions de l'image  $d \times d$  en pixels, et les coordonnées du robot  $(x_r, y_r)$  dans le repère du monde, on peut déterminer les indices  $i$  et  $j$  qui spécifient l'image contenant la portion de la carte sur laquelle le robot se trouve. Donc  $i$  et  $j$  sont données par :

$$i = E\left(\frac{x_c \cdot r}{d}\right)$$

$$j = E\left(\frac{y_c \cdot r}{d}\right)$$

où  $E()$  indique la partie entière.

Les coordonnées du robot  $(x_r, y_r)$  dans le repère du monde sont fournies par le système de localisation du robot, et c'est à partir d'elles que la paire  $(i, j)$  est calculée pour se repérer sur la carte réelle. La situation sur cette dernière n'est pas suffisante pour refléter l'apparence de l'environnement en ce point, car il faut avoir la direction du champ de vision  $\gamma$ , déterminée à partir de l'orientation du robot et de la caméra dans le repère du monde. Le champ visuel est limité par les paramètres de la caméra et son altitude  $h$  sur le

robot. C'est pour cette raison qu'il nous est apparu nécessaire d'utiliser une carte virtuelle qui est définie dans le paragraphe suivant.

### IV.3.2 - Gestion de la carte virtuelle

L'objectif principal est de construire une scène, à partir des données de la carte, qui reflète l'apparence de l'environnement du robot, en une position donnée. Par conséquent, les primitives qui apparaissent sur cette scène doivent nécessairement appartenir aux objets placés dans la région d'intersection du champ visuel et du plan de la carte. Ainsi, la carte virtuelle n'est autre que cette région d'intersection. Elle se déplace sur le plan de la carte suivant le mouvement du robot et modifie sa taille suivant les paramètres intrinsèques de la caméra (voir figure IV.3.2.1 ).

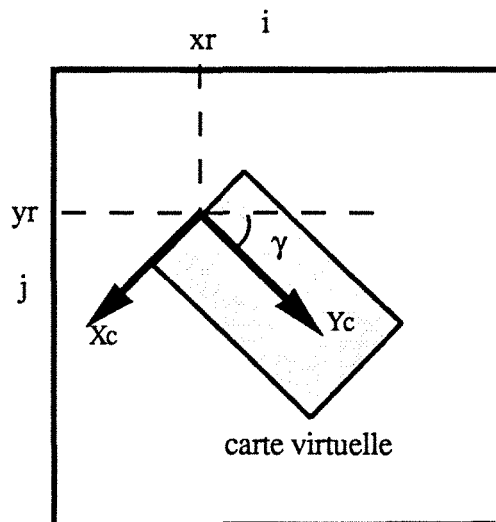


Figure IV.3.2.1 : carte virtuelle.

Enfin la définition de la carte virtuelle est basée sur les hypothèses suivantes :

- \* Son repère coïncide avec celui de la caméra.
- \* Ses dimensions sont variables en fonction des paramètres de la caméra.

\* Sa vitesse de déplacement sur la carte réelle est fonction de la vitesse du robot dans son environnement.

La construction de cette carte est assurée par logiciel, qui permet de la déplacer dans l'espace mémoire correspondant au terrain de navigation. Il faut noter que la carte réelle réside en mémoire sur disque. Donc pour diminuer le temps d'accès, on charge uniquement la carte virtuelle dans la mémoire dynamique pour interprétation, et pour l'extraire, il faut spécifier ses dimensions. On est donc amené à limiter d'abord le champ de vision à une distance  $k$  (en m) devant la caméra, et par conséquent la surface de celui-ci se trouve dépendante de la distance focale (voir figure IV.3.2.2).

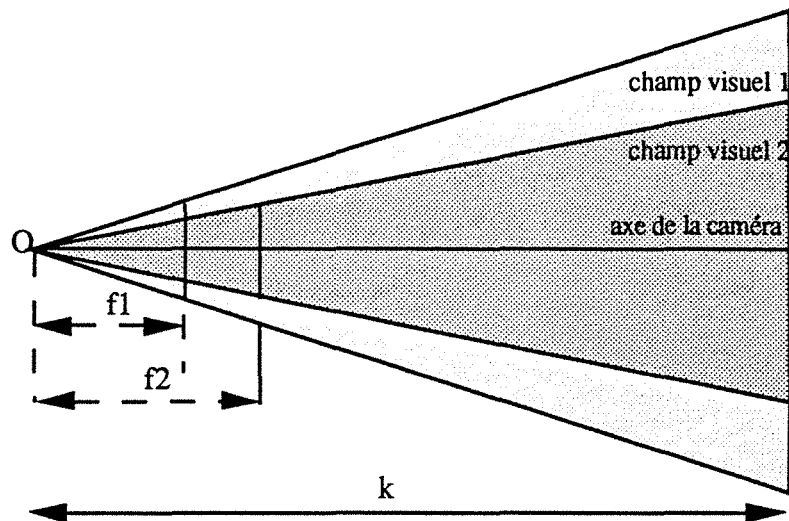


Figure IV.3.2.2 : variation du champ de vision en fonction de  $f$ .

Pour déterminer la surface de champ visuel on doit connaître l'espacement entre les pixels  $s$  ( $\mu\text{m}/\text{pixel}$ ), les dimensions de l'image  $d \times d$  pixels. Ainsi, on remarque sur la figure IV.3.2.3 que :

$$\frac{d_f}{k} = \frac{d \cdot s}{2f}$$

et

$$2d_f = \frac{k \cdot d \cdot s}{f}$$

Notons que  $k$ ,  $d$ , et  $s$  sont des constantes pour une configuration donnée, et  $f$  étant le seul paramètre variable qui peut modifier les dimensions de la carte virtuelle.

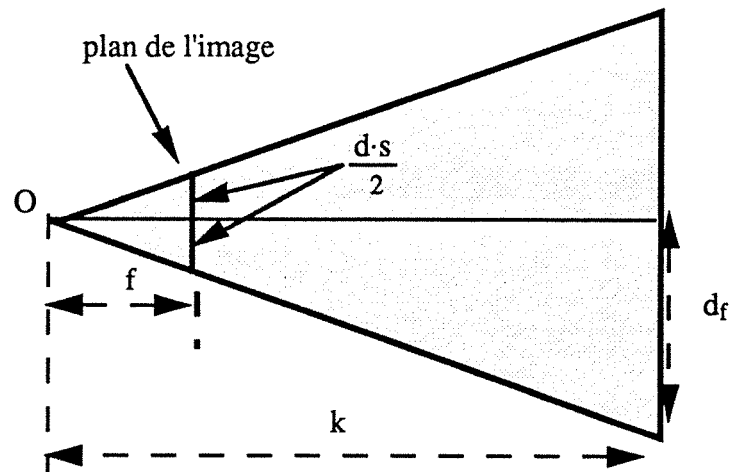


Figure IV.3.2.3 : dimensions du champ visuel.

Donc le champ de vision aura comme surface :

$$S = d_f \cdot k \text{ ( en m}^2\text{)}$$

Et on charge à partir de la carte réelle la carte virtuelle de dimensions:

$$2d_f \cdot r \text{ par } k \cdot r \text{ pixels.}$$

où  $r$  représente la résolution ( pixels/mètre).

### IV.3.3 - Prédiction de la scène à partir de la carte

Une fois la carte virtuelle déterminée et extraite de la carte réelle, on procède à la construction de la scène. Pour cela on utilise le repère de la caméra défini précédemment. Mais en pratique la transformation  $T$  des points de l'objet en point de l'image est très difficile à exploiter en raison du fait que le système de référence utilisé est centré sur la caméra. On utilise donc un système de référence globale correspondant à celui de l'environnement et un autre correspondant à celui de la plate-forme autour de laquelle les rotations  $\Theta$  et  $\Phi$  sont faites.

Prenons comme référentiel, le repère  $Re(Xe, Ye, Ze)$ , et comme repère de la plate-forme  $Rp(Xp, Yp, Zp)$  dont l'origine a pour coordonnées  $(x_r, y_r, h)$  dans  $Re$ . Les quantités  $x_r$  et  $y_r$  sont données par le système de localisation du robot car elles expriment la position du robot dans  $Re$  (la position du repère du robot dans celui de la plate-forme est supposée connue). Tandis que la quantité  $h$  est imposée par le montage de la caméra sur le robot. Puisqu'en réalité les rotations de la caméra se font autour des axes de la plate-forme, le repère de la caméra  $Rc(Xc, Yc, Zc)$  se trouve à une position  $(e_x, e_y, e_z)$  dans  $Rp$  (voir figure IV.3.3.1). Pour des raisons de simplification on suppose que le plan de l'image  $(Xc, Zc)$  est parfaitement parallèle au plan  $(Xp, Zp)$  de la plate-forme. Le centre de projection a pour coordonnées  $(0, -f, 0)$  dans le repère de la caméra  $Rc$ , où  $f$  exprime la distance focale de l'optique utilisée.

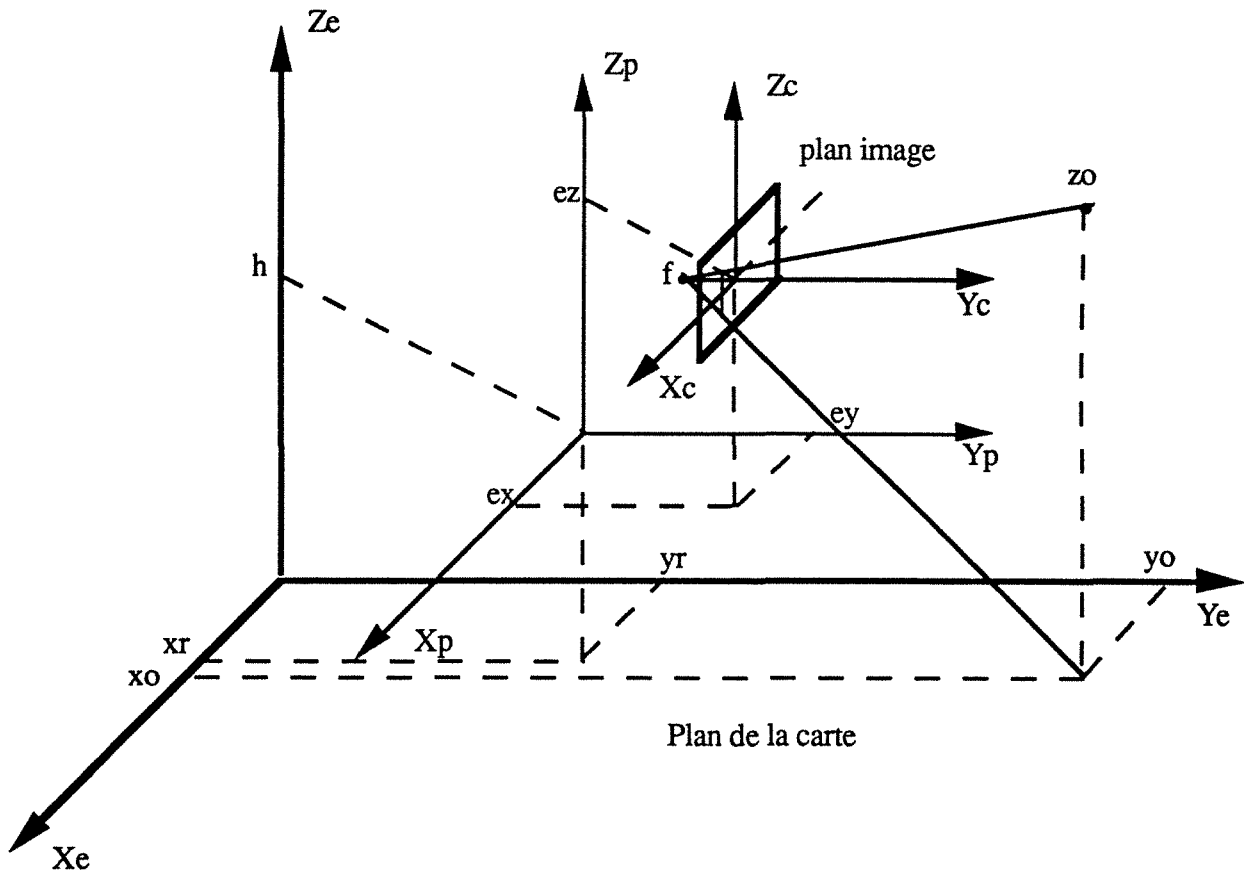


Figure IV.3.3.1 : Position de Rc et Rp par rapport à Re.

Connaissant un point M d'un objet ayant les coordonnées  $(x_o, y_o, z_o)$  dans  $R_e$ , quelles sont ses coordonnées sur le plan de l'image  $(X_c, Z_c)$  ? Le passage de  $R_e$  à  $R_p$  se fait par la transformation  $T_{e,p}$  composée de deux rotations et d'une translation ( on néglige donc le "dévers" qui se traduirait par une rotation autour de  $Y_p$  puisqu'on travaille dans l'hypothèse d'une route plate) :

$$T_{e,p} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Phi) & -\sin(\Phi) & 0 \\ 0 & \sin(\Phi) & \cos(\Phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\Theta) & -\sin(\Theta) & 0 & 0 \\ \sin(\Theta) & \cos(\Theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -x_r \\ 0 & 1 & 0 & -y_r \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Avec  $\Phi$  et  $\Theta$  des rotations autour des axes  $X_p$  et  $Z_p$  respectivement.

Le passage de  $R_p$  à  $R_c$  se fait par la transformation  $T_{p,c}$  composée d'une simple translation :

$$T_{p,c} = \begin{pmatrix} 1 & 0 & 0 & -ex \\ 0 & 1 & 0 & -ey \\ 0 & 0 & 1 & -ez \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ainsi la transformation du repère de l'environnement au repère de la caméra est donné par:

$$T_{e,c} = T_{e,p} \cdot T_{p,c}$$

donc:

$$T_{e,c} = \begin{pmatrix} \cos(\Theta) & -\sin(\Theta) & 0 & A \\ \sin(\Theta)\cos(\Phi) & \cos(\Theta)\cos(\Phi) & -\sin(\Phi) & B \\ \sin(\Theta)\sin(\Phi) & \cos(\Theta)\sin(\Phi) & \cos(\Phi) & C \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Avec les valeurs :

$$A = -xr \cos(\Theta) + yr \sin(\Theta) - ex$$

$$B = -xr \sin(\Theta)\cos(\Phi) - yr \cos(\Theta)\cos(\Phi) + h \sin(\Phi) - ey$$

$$C = -xr \sin(\Theta)\sin(\Phi) - yr \cos(\Theta)\sin(\Phi) - h \cos(\Phi) - ez$$

La projection perspective des points du repère  $R_c$  sur le plan de l'image est donnée par la

transformation  $T_{c,i}$  ( voir section IV.2.1 ) :

$$T_{c,i} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 1 & 0 & f \end{pmatrix}$$

Ainsi les coordonnées dans le plan de l'image de la projection perspective d'un point défini dans  $Re$ , sont obtenues par la transformation :

$$T_{e,i} = T_{e,c} \cdot T_{c,i} = \begin{pmatrix} f \cdot \cos(\Theta) & -f \cdot \sin(\Theta) & 0 & f \cdot A \\ f \cdot \sin(\Theta) \cos(\Phi) & f \cdot \cos(\Theta) \cos(\Phi) & -f \cdot \sin(\Phi) & f \cdot B \\ f \cdot \sin(\Theta) \sin(\Phi) & f \cdot \cos(\Theta) \sin(\Phi) & f \cdot \cos(\Phi) & f \cdot C \\ \sin(\Theta) \cos(\Phi) & \cos(\Theta) \cos(\Phi) & -\sin(\Phi) & B+f \end{pmatrix}$$

Ainsi, les coordonnées du point M sur le plan de l'image sont :

$$x_i = f \frac{x_o \cos(\Theta) - y_o \sin(\Theta) + A}{x_o \sin(\Theta) \cos(\Phi) + y_o \cos(\Theta) \cos(\Phi) - z_o \sin(\Phi) + B + f}$$

$$z_i = f \frac{x_o \sin(\Theta) \sin(\Phi) + y_o \cos(\Theta) \sin(\Phi) + z_o \cos(\Phi) + C}{x_o \sin(\Theta) \cos(\Phi) + y_o \cos(\Theta) \cos(\Phi) - z_o \sin(\Phi) + B + f}$$

En utilisant le principe de construction de carte décrit précédemment , l'objectif est



d'aboutir à une image qui reflète la scène perçue par la caméra à partir d'un point quelconque de la carte dans une direction donnée. Les objets qui apparaissent sur l'image seront ceux qui se trouvent sur l'intersection du champ visuel de la caméra avec le plan de la carte. En réalité, l'apparition de ces objets dépend aussi des paramètres de la configuration robot-caméra qui sont la hauteur de la caméra  $h$ , les déviations horizontale et verticale de la caméra  $\Theta$  et  $\Phi$  respectivement (voir figure IV.3.3.2).

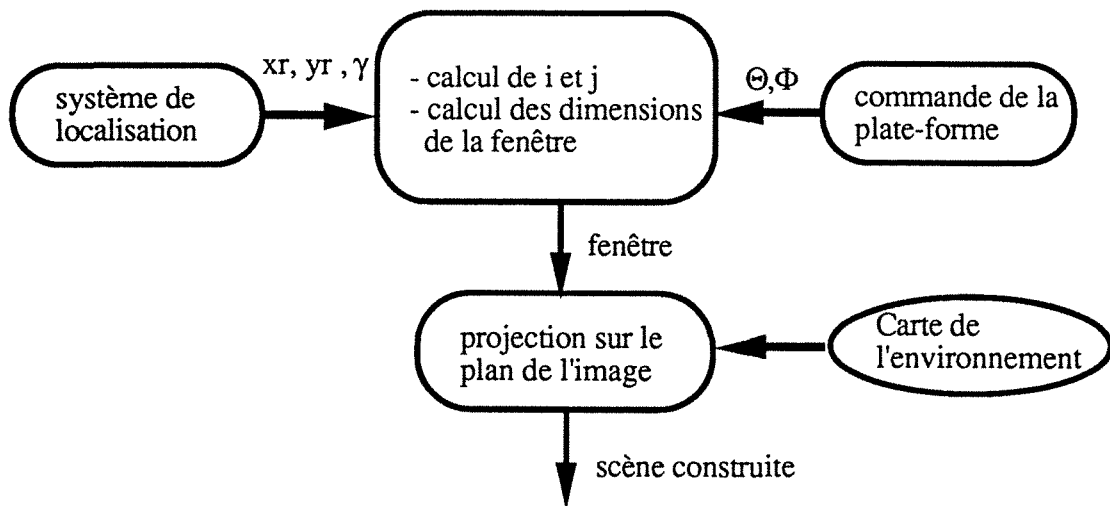


Figure IV.3.3.2 .

La figure IV.3.3.3 présente la prédiction d'une scène à partir des positions a et b représentées sur la figure IV.3.1.1 .

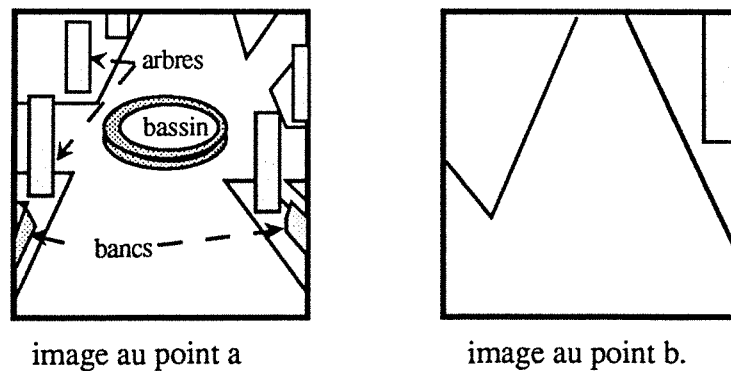


Figure IV.3.3.3 : images construites au points a et b

L'utilisation de la projection point par point de la scène sur le plan de l'image, permet de faire une prédiction sur des scènes très compliquées où figurent des intersections, virages, montées, descentes ou dévers. Le type d'images représentées sur la figure IV.3.3.4 peut être obtenu à partir d'un traitement du type précédent.

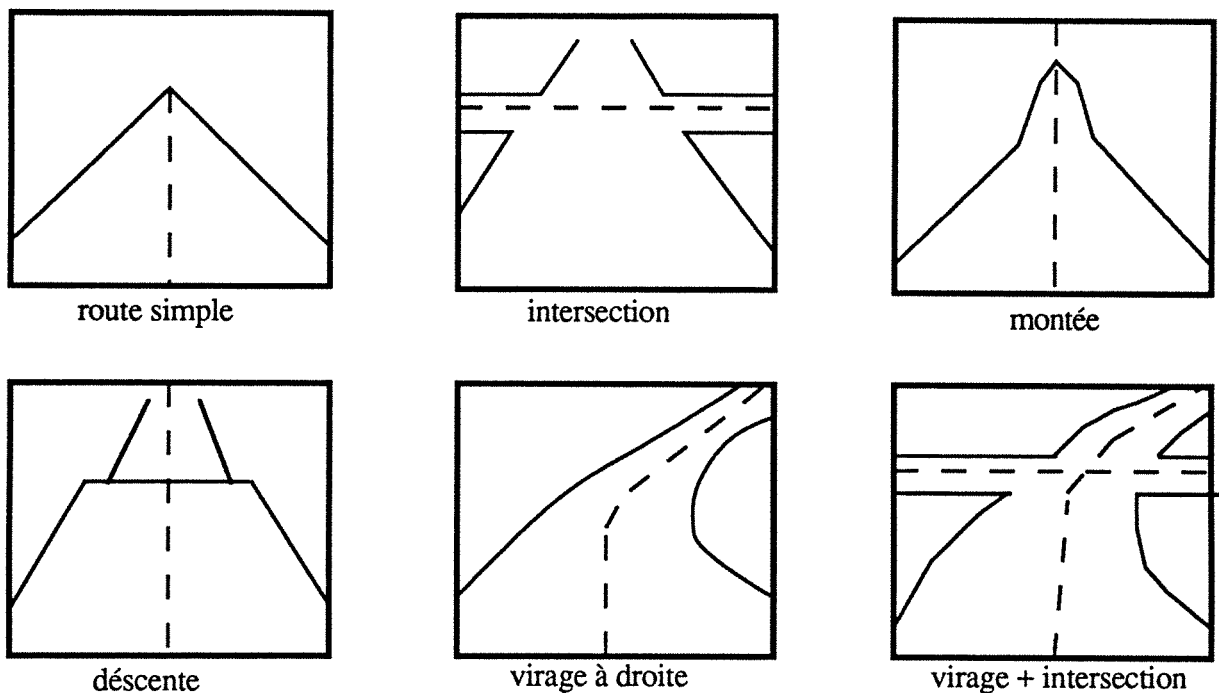


Figure IV.3.3.4 : exemple de scène traitée.

La limitation de cette méthode de représentation de carte est que les détails le long de la hauteur d'un objet ne peuvent pas être représentés. Par exemple, un panneau d'indication sera représenté par sa surface vue d'en haut et sa hauteur correspondante, alors que sur la scène construite le panneau est visualisé sans le message qu'il porte. A vrai dire cet algorithme est conçu pour aider le robot à naviguer en lui fournissant des informations utiles comme la position des objets statiques et des bords de route sans pour autant se préoccuper des détails figurant sur l'objet lui-même.

#### **IV.3.4 - Placement des fenêtres initiales en mode de suivi continu**

Le mode de suivi continu est constamment opérationnel après que la phase initiale est achevée. Après cette étape et à l'aide des informations fournies par les modules prédiction de scène, exploitation de la scène, et base de connaissance visuelle; le superviseur est normalement en possession du modèle de la scène initiale. Ceci va permettre au robot d'entamer la navigation autonome, et de se déplacer d'une distance  $d$  pré-déterminée par le générateur de trajectoire. Cette distance est parcourue en aveugle, et pendant ce temps le superviseur cherche continuellement une estimation des coordonnées et de l'orientation actuelles du robot. Une fois que la distance  $d$  est parcourue, le superviseur transmet cette information au module de prédiction de scène pour avoir, à partir de la carte, une scène prédite et déclenche automatiquement une nouvelle acquisition d'image. Pour permettre au module d'exploitation de la scène de fonctionner convenablement en mode de suivi continu, il faut que la scène prédite soit accompagnée de deux fenêtres qui indiquent les régions par lesquelles les bords droit et gauche de la route pénètrent le plan de l'image.

On a vu en section IV.3.1 que la carte représente la hauteur des objets sur l'image en leur attribuant un niveau de gris. Les bords de route ont donc un niveau fixe qu'on choisi lors de la construction de la carte, sur notre carte on leur a attribué un niveau très bas 1. Généralement, si les bords de route ont un niveau de gris  $g$ , leur détection est facilitée par une simple segmentation de l'image prédite autour de  $g$ . L'image binaire résultante décrit uniquement les bords de route et, par conséquent, l'application de la transformée de Hough au bas de l'image permet de déterminer en quelles positions les bords de route pénètrent la scène.

#### **IV.4 - GENERATION DE TRAJECTOIRES A PARTIR DE LA CARTE**

Les robots mobiles doivent être capables de s'adapter à l'environnement et avoir la

capacité de planifier et d'exécuter des tâches. Il est important que les robots mobiles aient la possibilité de comprendre le milieu dans lequel ils doivent évoluer librement.

Une des fonctions importantes est la génération d'une trajectoire à partir du point de départ jusqu'au point de destination sans heurter un obstacle et sans intervention de l'opérateur.

Quand cette idée est apparue dans le milieu des chercheurs en vision, les objectifs de recherches ont abandonné les solutions aux problèmes de vision de bas niveau et se sont orientés vers l'interprétation des données, la navigation et l'organisation du système de vision [BRO-85] et [KOC-85]. Ainsi, la planification de la trajectoire est devenue un processus lié à la détection et la cartographie. La stratégie de planification du chemin dépend du degré de connaissance du robot sur son environnement. Elle est basée, dans le cas d'une connaissance parfaite d'un milieu restreint, sur les méthodes graphiques. Dans le cas contraire, elle se base sur des heuristiques utilisant la télémétrie ou la vision pour localiser le robot puis identifier ensuite son entourage. Cette tâche, nécessitant l'information sur la profondeur du champ visuel, a rendu l'utilisation de la stéréo-vision obligatoire dans les applications de la vision en robotique mobile.

En fait, la propriété essentielle de la vision est l'information globale qu'elle peut fournir. Dans notre système l'information visuelle est utilisée principalement comme une information globale pour obtenir une estimation de la position et de l'orientation du robot par rapport à l'axe de la route. La méthode de construction de la carte utilisée permet de spécifier la position des obstacles statiques ainsi que leur hauteur. En examinant l'architecture du système de vision ( figure II.3.1.1 ) on remarque que le superviseur ne peut démarrer aucune tâche sans la connaissance préalable d'un chemin planifié par l'opérateur. Lorsque ce chemin est défini, il est communiqué au superviseur sous forme de fichier contenant les coordonnées des points qui caractérisent la forme du chemin sur le plan de la carte. Le nombre de ces points est fonction du nombre de changements de direction de l'axe de la route à partir du point de départ jusqu'au point d'arrivée. Figure IV.4.1.1.a montre un exemple d'un chemin dont le point de départ est A et la destination B.

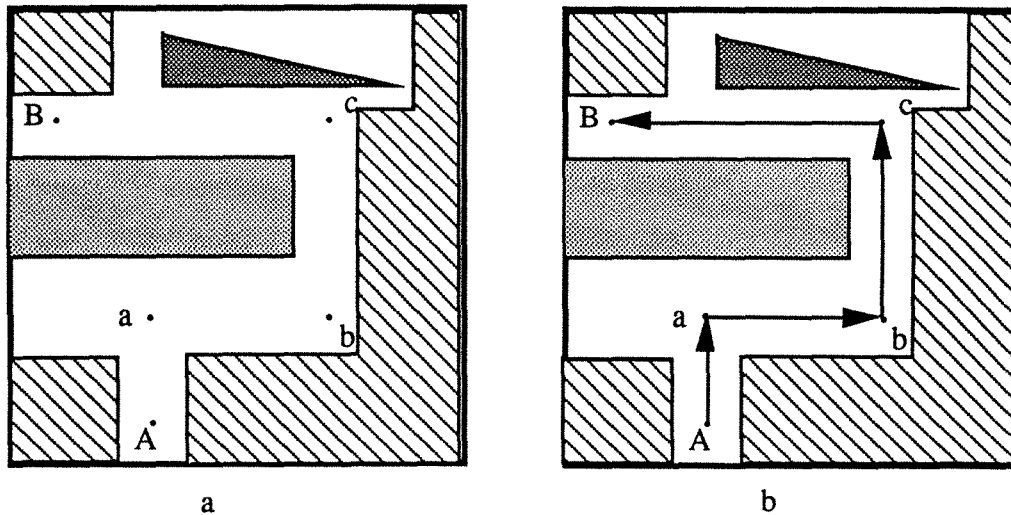


Figure IV.4.1.1: exemple d'un chemin.

Les points intermédiaires qui doivent être spécifiés, en plus de A et B, par l'opérateur sont a, b, et c. Ces points doivent être placés de telle façon qu'ils peuvent être reliés par des segments de droites complètement inclus dans le plan de la route ( sans traverser les bords de route ). Voir figure IV.4.1.1.b). Par contre, les segments de droites peuvent traverser le plan des obstacles statiques cartographiés qui sont situés sur le parcours. Figure IV.4.1.2 montre un exemple d'un obstacle situé sur le segment ab.

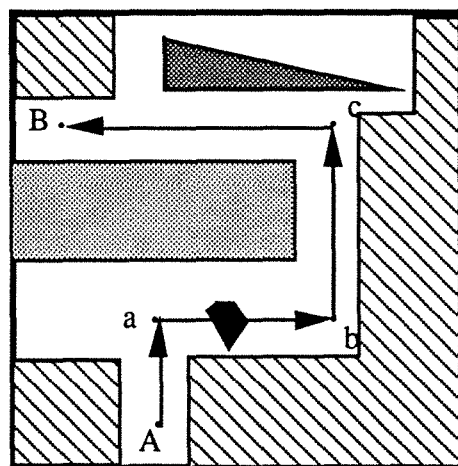


Figure IV.4.1.2: cas d'un obstacle entre a et b.

Lors de l'initialisation, le superviseur identifie les coordonnées du point A comme étant les coordonnées de la position actuelle du robot et transmet d'une manière

séquentielle les différents segments constituant le chemin ( figure IV.4.1.3). L'opérateur doit spécifier, en plus, pour chaque segment la distance  $D_{av}$  au bout de laquelle le robot doit faire une acquisition d'image après l'avoir parcouru en aveugle. Cette distance peut varier d'un segment à un autre en fonction de la structure et la complexité de la forme de la route. Ainsi, il est préférable de diminuer cette distance dans un virage par exemple afin d'assurer un suivi de trajectoire avec un minimum d'erreurs. D'autre part, le superviseur informe continuellement l'opérateur via l'écran sur l'état du robot. On distingue donc trois états différents:

- MARCHE : Cet état indique que le robot est entrain d'exécuter une tâche de navigation d'une manière autonome.
- ARRET : Cet état indique l'arrêt complet du robot et du processeur.
- BLOQUE: Dans cet état le robot est en arrêt, mais le processeur est en état d'attente d'une intervention de l'opérateur. Ceci se produit quand le système, après plusieurs tentatives, échoue dans la localisation des bords de route. Il attend donc la relance de l'extérieur. Cet état s'accompagne , en général , d'une alarme sonore.

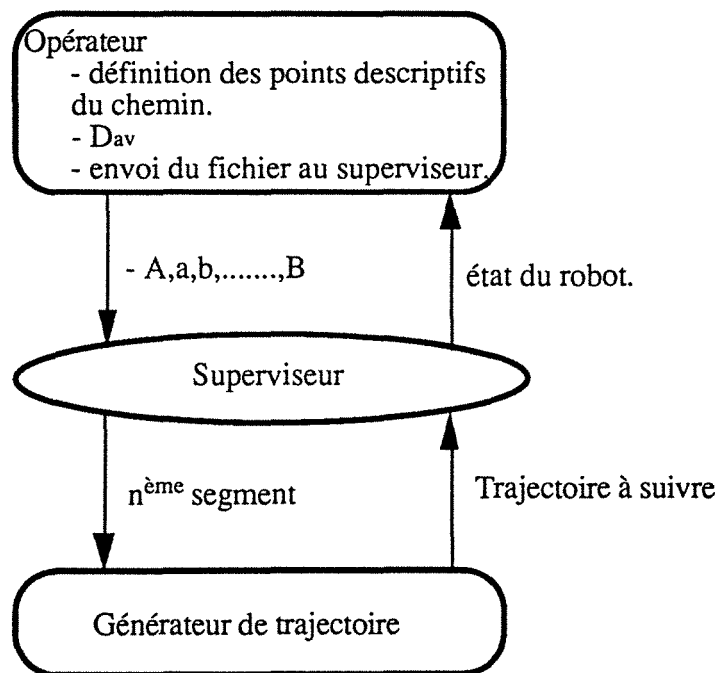


Figure IV.4.1.3: transfert séquentiel des segments de la trajectoire à suivre.

Le superviseur transmet un segment à la fois au générateur de trajectoire. Ce dernier accède à la carte et balaye la surface de la route à partir du point de départ jusqu'à la destination pour rechercher les obstacles statiques. Si aucun obstacle n'est trouvé sur le chemin, la longueur du segment est calculée en utilisant la résolution  $r$  ( pixels/mètre ) et les coordonnées des points que le robot doit atteindre et où il doit faire une acquisition d'image sont déterminés. Ces points sont obtenus en connaissant la distance de déplacement en aveugle  $D_{av}$  spécifiée par l'opérateur. Figure IV.4.1.4 montre la décomposition du segment Aa en intervalles de distance  $D_{av}$ .

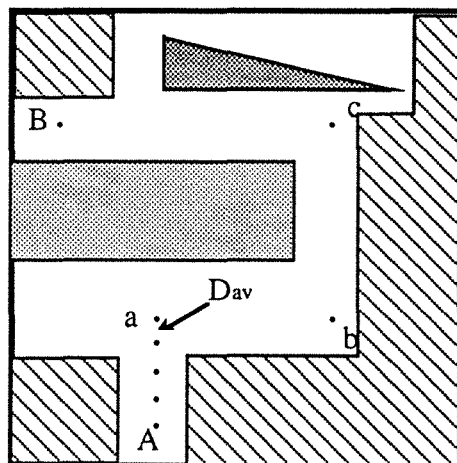


Figure IV.4.1.4: trajectoire sans obstacle.

Dans le cas d'existence d'un obstacle entre le point initial et le point final du segment, le générateur de trajectoire détermine l'ensemble des points de passage pour éviter l'obstacle avec un risque minimal. Ce risque est fonction de la distance de sécurité  $D_s$  qui est prédéterminée par l'opérateur. Ainsi la trajectoire est modifiée pour éviter l'obstacle dans deux cas ( figure IV.4.1.5 ) :

- 1- le point final du segment n'est pas visible à partir du point initial.
- 2- le point final est visible à partir du point initial mais la distance  $x_r$  de l'obstacle au segment est inférieure à la somme de la distance de sécurité  $D_s$  et la

moitié de la largeur du robot  $L_m$  ( voir figure IV.4.1.6 )

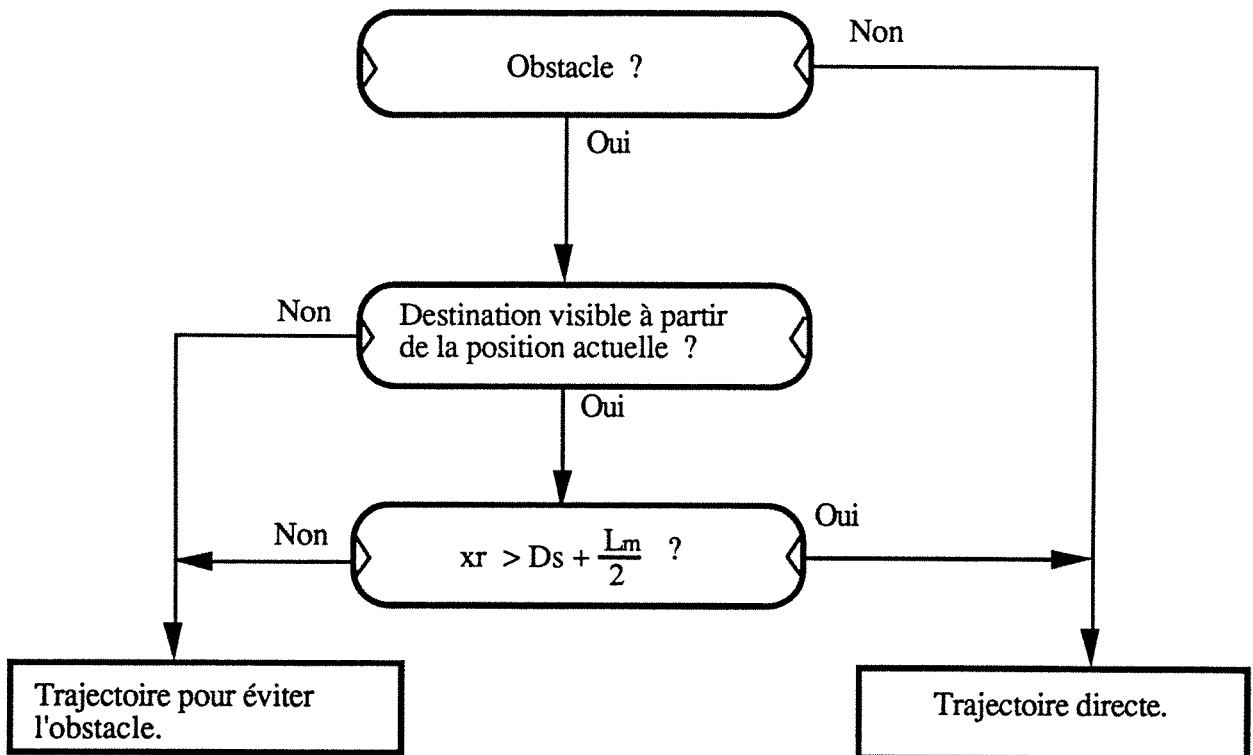


Figure IV.4.1.5

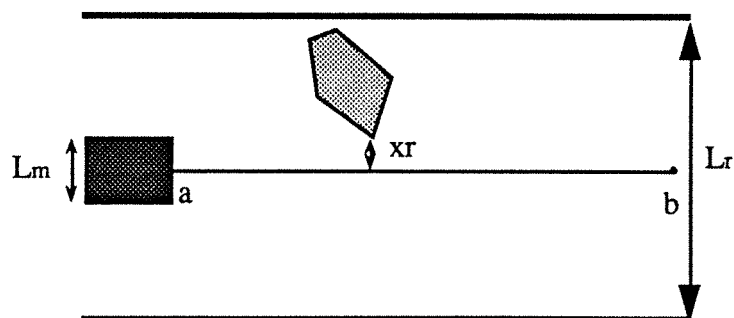


Figure IV.4.1.6

avec  $L_m$  la largeur du mobile et  $L_r$  la largeur de la route .

Dans le cas où un évitement est nécessaire, les points de passage formant la trajectoire seront modifiés et placés de telle manière que le robot contourne l'obstacle à une distance supérieure ou égale à  $D_s$  et une distance de déplacement en aveugle  $D_{av}$  égale



au dixième de celle prévue dans un cas d'absence d'obstacle. Cette nouvelle distance de déplacement en aveugle sera applicable à une distance d'environ 5m avant d'atteindre l'obstacle ( voir figure IV.4.1.7).

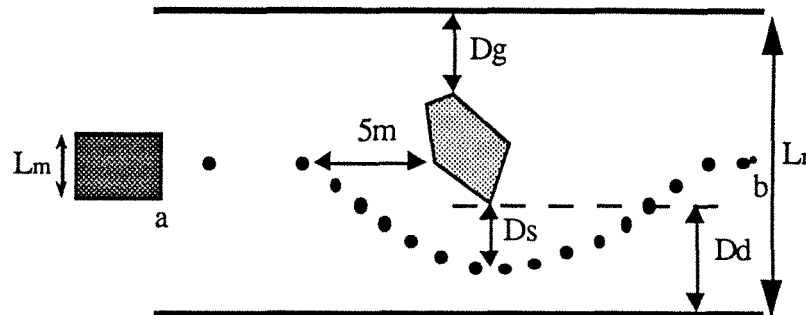


Figure IV.4.1.7 : évitement d'un obstacle statique.

Le choix de la direction à prendre pour éviter l'obstacle est fonction des distances  $D_d$  et  $D_g$  représentant respectivement les distances de l'obstacle aux bords droit et gauche de la route. Le passage se fait en respectant la distance de sécurité  $D_s$ .

Cette méthode est appliquée uniquement pour générer les points de passage afin d'éviter les obstacles statiques figurant sur la carte. Pour éviter autres types d'obstacles d'autres méthodes efficaces de contournement d'obstacles existent et qui ne font pas l'objet de cette thèse.

## IV.5 - CONCLUSION

Sur le plan de la cartographie on s'est basé sur le principe de l'imagerie aérienne pour numériser la carte de l'environnement dans plusieurs images facilement accessibles. Pour faciliter l'interprétation, on a étiqueté ces images en fonction de la nature et la hauteur des objets cartographiés. L'utilisation de la carte virtuelle permet une reconstruction de la scène pour une analyse ultérieure. On note que le traitement à ce niveau s'est concentré surtout sur la position des bords de route et le placement des fenêtres initiales pour le déclenchement du mode de suivi continu. La scène construite à partir des données de la carte sera utilisée ultérieurement pour fournir un modèle prédit qui permet de vérifier les données acquises. La vérification est correcte seulement si la position du robot est assez précise, ceci constitue le problème majeure de cette méthode.

Néanmoins des techniques peuvent être envisagées pour réduire cette inconvénient, notamment par l'utilisation des erreurs du système de localisation pour évaluer et minimiser les erreurs associés au modèle prédit. La navigation dans un milieu extérieur tel que le parc urbain nécessite parfois la connaissance d'autres types d'informations telle que la structure de certains objets de l'environnement. Par conséquent, une extension à notre carte serait d'associer, par une représentation symbolique, à chaque primitive quelques éléments d'information sur ses caractéristiques.



**CHAPITRE V**

*ROLE DE LA BASE DE  
CONNAISSANCE VISUELLE DANS LE  
SYSTEME DE VISION.*

## **V.1 - INTRODUCTION**

Le modèle de la scène est la composante centrale du système de vision, il représente l'interprétation de l'environnement par le robot. Un modèle de scène incorrect conduit à un échec total de la navigation autonome. Ainsi, la production du modèle de la scène doit être très fiable. L'objectif de la base de connaissance visuelle est de créer un modèle de la scène en utilisant les caractéristiques géométriques déduites à partir des segments de droite fournis par le module d'exploitation de la scène. En conséquence, les résultats du module d'exploitation de la scène représentent ce que le système de vision "voit". Alors que les résultats du traitement des données de la base de connaissance sont ce que le système "perçoit". C'est à ce niveau que les défaillances de l'exploitation de la scène liées à la variabilité de celle-ci et à d'autres sources d'erreurs sont détectées et éventuellement corrigées par des techniques géométriques s'appuyant sur les connaissances enregistrées à priori.

## **V.2 - CONTRAINTES GEOMETRIQUES SUR LA STRUCTURE DE LA ROUTE**

Pour pouvoir interpréter le sens des droites détectées par le module d'exploitation de l'image, on doit étudier leurs propriétés ainsi que leurs relations avec l'environnement. Ceci oblige le passage du plan 2-D de l'image au repère tri-dimensionnel de l'environnement. Ce passage constitue un problème mal posé en vision artificielle car il nécessite l'introduction de contraintes géométriques pour aboutir à une solution unique. Dans notre cas ces contraintes sont les hypothèses émises sur la structure de la route .

Contrairement à la géométrie directe, la géométrie inverse donne une solution unique au problème; aucune contrainte n'est nécessaire pour le rendre résoluble. La combinaison des méthodes de la géométrie directe et de la géométrie inverse permet l'enregistrement des caractéristiques d'une image à l'autre aussi bien que de faire des prédictions sur la continuité de la route.

Des travaux effectués sur le système de vision VITS [TUR-88] ont montré qu'un modèle de route sur terrain plat ("flat-earth road model") permet une résolution rapide du problème par la géométrie directe. Ce système utilise la géométrie inverse pour résoudre le problème, en déterminant la position sur le plan de l'image d'un point dont la position tri-dimensionnelle est connue. Par contre, cette hypothèse très utile dans certaines circonstances, n'est pas assez précise dans la totalité des applications de suivi de route. Une autre approche utilise un modèle fondé sur le concept géométrique "colline-et-vallée", et utilise une méthode rapide de "shape-from-contour" pour résoudre le problème de la géométrie directe; c'est une légère modification du modèle de route en terrain plat.

Dans notre cas, compte tenu de la topographie du parc urbain où évoluera le robot, l'établissement du modèle de la scène se base sur l'hypothèse fondamentale que le terrain sur lequel le robot est plat. Par conséquent, la structure de la route est soumise aux hypothèses suivantes:

**\* *Contrainte de consistance :***

Chaque route est formée de deux bords qui sont localement parallèles et séparés d'une distance constante correspondant approximativement à la largeur de celle-ci.

**\* *Contrainte de continuité :***

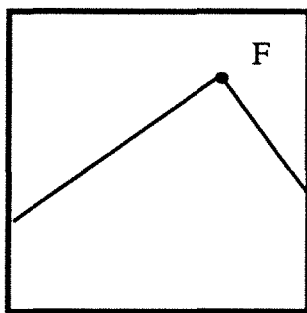
La continuité des bords de route dans l'image implique leur continuité dans l'environnement. Cette contrainte est aussi appliquée entre images successives pour renforcer la consistance spatio-temporelle. Cette contrainte permet de prédire la position d'un bord de route manquant sur la scène .

### V.3 - SELECTION DES BORDS DE ROUTE

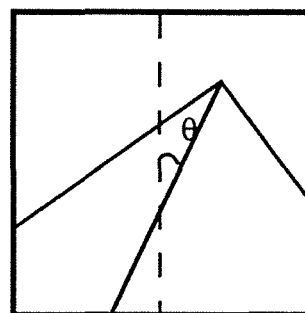
Un couple de segments de droites représentant les bords droit et gauche de la route, peut être décrit par quatre paramètres descriptifs :

*a) Le point de fuite :*

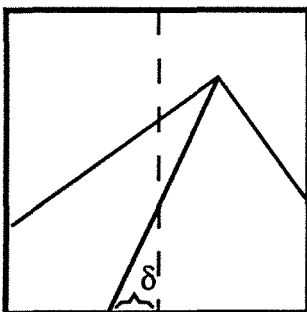
Ce point est déterminé par l'intersection des deux segments de droites du couple représentant les bords de route. Ce point peut être à l'intérieur ou à l'extérieur du plan de l'image en fonction de la configuration robot-caméra (voir figure V.3.1.a).



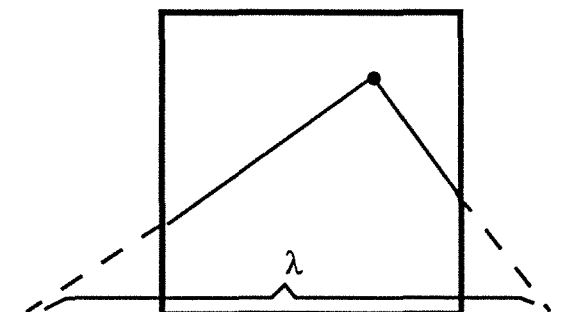
a: point de fuite F



b: erreur d'orientation  $\theta$



c: erreur de décalage  $\delta$



d: largeur de la route  $\lambda$

Figure V.3.1: paramètres descriptifs d'un couple de droites représentant des bords de route.

***b) L'erreur d'orientation  $\theta$  :***

En sélectionnant un point du bord de route droit et de son homologue du bord gauche ( sur la même ligne horizontale de l'image ), on peut déterminer leur point milieu. Le segment de droite passant par ce point et le point de fuite F constitue l'axe de la route. L'erreur d'orientation  $\theta$  est l'angle entre l'axe de la route et l'axe vertical du plan de l'image (voir figure V.3.1.b). Il en résulte que, connaissant  $\theta$  et l'orientation du repère de la plate-forme par rapport au repère du robot, on peut déterminer l'erreur d'orientation du robot par rapport à l'axe de la route.

***c) L'erreur de décalage  $\delta$  :***

Cet écart en pixels est mesuré au bas de l'image entre l'axe vertical du plan de la caméra et l'axe de la route, comme illustré par la figure V.3.1.c. La caméra étant calibrée, on peut donc déterminer à partir de  $\delta$  l'erreur de position du robot par rapport à l'axe de la route.

***d) La largeur de la route  $\lambda$  :***

Cette quantité, en pixels, mesurée sur le plan de l'image permet d'évaluer la largeur de la route ( voir figure V.3.1.d ). Après calibrage de la caméra, le passage du repère de celle-ci à celui de l'environnement, permet d'aboutir à une mesure approximative de la largeur réelle de la route.

## **V.4 - MODELE DE LA SCENE**



Dans ce mode, le module "exploitation de la scène" fournit au plus 36 couples de bords de route possibles. Chaque bord est décrit par l'équation:  $z = a_i x + b_i$ . Puisqu'on suppose que le robot est à peu près sur l'axe de la route au moment de l'initialisation, chaque paire contient deux segments de droites ayant des pentes de signes opposés.

La transformée de Hough attribue un poids  $\omega$  à chaque segment de droite ( voir chapitre III ). En conséquence, à chaque paire constituée de deux segments  $i$  et  $j$  correspondant respectivement à la combinaison d'un bord droit et d'un bord gauche, on associe un poids  $\omega_p$  donné par la moyenne des poids individuels:

$$\omega_p = \frac{\omega_i + \omega_j}{2}$$

Il existe donc au maximum 36 paires car :  $1 \leq i \leq 6$  et  $1 \leq j \leq 6$ .

La description d'une paire de bords de route est faite par l'association des cinq paramètres décrit précédemment  $F, \theta, \delta, \lambda$  et  $\omega_p$ . On génère donc un espace à cinq dimensions  $\mathcal{R}^5$  sur lequel chaque élément est décrit par :

$$\mathcal{P}_i = ( F_i, \theta_i, \delta_i, \lambda_i, \omega_{p_i} ) \quad 1 \leq i \leq 36.$$

La prédiction de scène fournit un modèle décrivant la scène à partir de la carte; celui-ci est donné par :

$$\mathcal{M}_p = ( F_p, \theta_p, \delta_p, \lambda_p )$$

Dans ce modèle prédit, on trouve la position du point de fuite, l'erreur d'orientation, l'erreur de décalage, et enfin la largeur de la route.

L'étude de la correspondance entre l'image et le modèle est basée sur les différences entre les paramètres du modèle prédit  $\mathcal{M}_p = (F_p, \theta_p, \delta_p, \lambda_p)$  et les paires  $\mathcal{P}_i = (F_i, \theta_i, \delta_i, \lambda_i, \omega_{pi})$  avec  $1 \leq i \leq 36$ . On tolère que la position du point de fuite soit située sur une surface  $\mathcal{S}$  du plan de l'image, de dimension  $n \times n$  pixels, et associée à une erreur maximale  $\Delta\theta_{\max}$ ,  $\Delta\delta_{\max}$  et  $\Delta\lambda_{\max}$  pour les paramètres  $\theta_i$ ,  $\delta_i$  et  $\lambda_i$ .

Donc la condition de validation de la paire est alors :

$$\begin{array}{ll} \text{si} & F_i \in \mathcal{S} , \\ & |\theta_p - \theta_i| \leq \Delta\theta_{\max} , \\ & |\delta_p - \delta_i| \leq \Delta\delta_{\max} , \\ \text{et} & |\lambda_p - \lambda_i| \leq \Delta\lambda_{\max} . \end{array}$$

la paire  $\mathcal{P}_i$  est choisie.

Il y a échec de la procédure si aucune paire ne satisfait le critère. Les causes principales d'échec résultent éventuellement des conditions extérieures ou de l'orientation de la caméra, ou des deux. Par conséquent, afin de reprendre la navigation, la caméra doit être réorientée et le processus de détection de bords de route appliqué à nouveau .

En général, il peut y avoir plusieurs paires qui satisfont ce critère. Dans ce cas, ces paires seront remplacées par une paire équivalente dont les paramètres seront des moyennes pondérées . La pondération se fait par  $\omega_p$  car dans  $\mathcal{R}^5$ , parmi les cinq paramètres d'une paire  $\mathcal{P}_i = (F_i, \theta_i, \delta_i, \lambda_i, \omega_{pi})$ , seule la variable  $\omega_{pi}$  est liée à la qualité du résultat de l'exploitation de la scène . Donc , si  $n$  paires ont satisfait le critère, la paire équivalente  $\mathcal{P}_{eq}$  aura comme paramètres:

$$\begin{aligned} \theta_{eq} &= \frac{\sum \omega_{pj} \cdot \theta_j}{\sum \omega_{pj}} , \\ \delta_{eq} &= \frac{\sum \omega_{pj} \cdot \delta_j}{\sum \omega_{pj}} , \end{aligned}$$

$$\lambda_{eq} = \frac{\sum \omega_{pj} \cdot \lambda_j}{\sum \omega_{pj}}$$

et le point de fuite aura comme coordonnées :

$$F_{eq} = (x_{eq}; y_{eq}).$$

avec :

$$x_{eq} = \frac{\sum \omega_{pj} \cdot x_j}{\sum \omega_{pj}}$$

$$y_{eq} = \frac{\sum \omega_{pj} \cdot y_j}{\sum \omega_{pj}}$$

La variable j, prend les n valeurs de i correspondant aux paires qui ont satisfait le critère.

Enfin, la paire sélectionnée prendra en compte les paramètres du modèle prédit et ceux de la paire équivalente. Ses paramètres seront donc:

$$\theta_{moy} = \frac{\theta_{eq} + \theta_p}{2},$$

$$\delta_{moy} = \frac{\delta_{eq} + \delta_p}{2},$$

$$\lambda_{moy} = \frac{\lambda_{eq} + \lambda_p}{2}.$$

$$F_{moy} = \left( \frac{x_{eq} + x_p}{2}, \frac{y_{eq} + y_p}{2} \right).$$

La sélection sera alors confirmée au système de vision et le processus de détection de bords de route s'arrêtera.

Dans l'exemple traité au tableau III.3.3.1, les bords de route sélectionnés correspondent aux droites d'équations :

$$\text{bord droit : } y = 0,7 x - 55$$

$$\text{bord gauche : } y = - x + 280$$

l'erreur d'orientation entre l'axe du robot et l'axe de la route est :

$$\theta = 12,11^\circ.$$

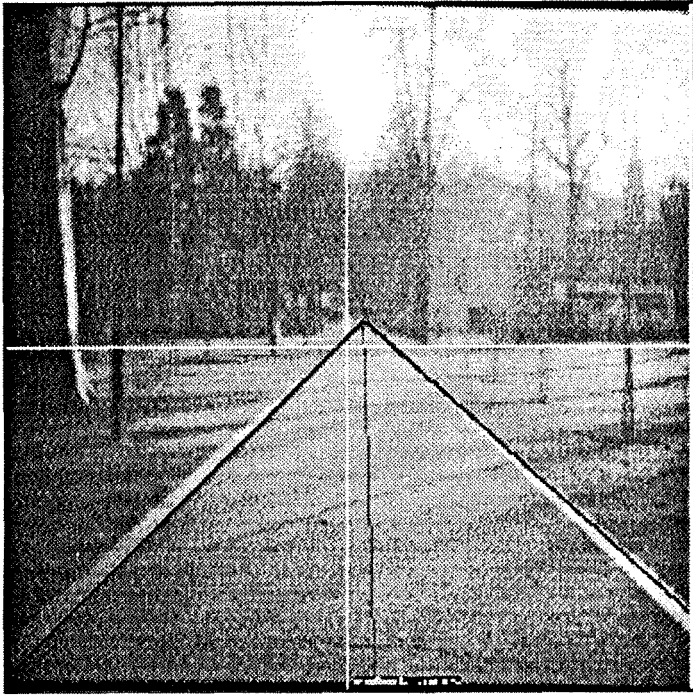
Les bords de route sélectionnés correspondent aux droites d'équation:

$$\text{bord droit : } y = 0,6 x - 20$$

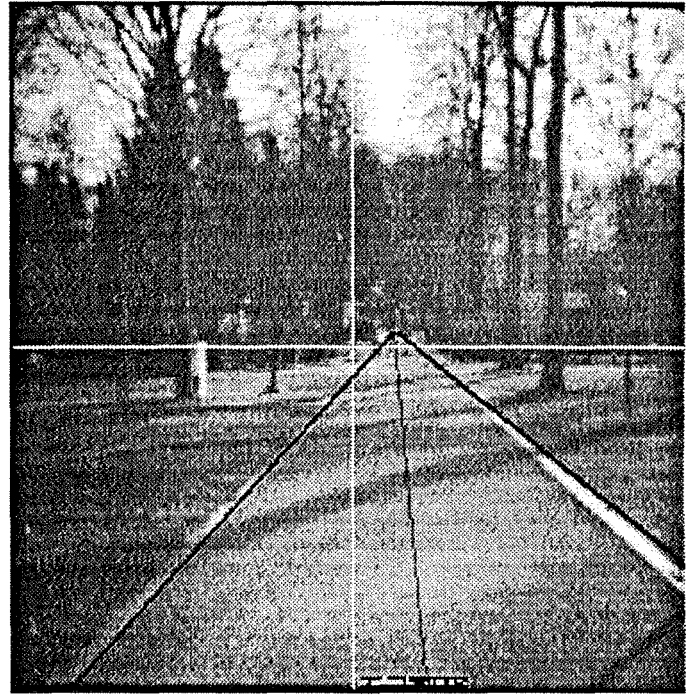
$$\text{bord gauche : } y = - x + 275$$

l'erreur d'orientation entre l'axe de la route et l'axe du robot est :

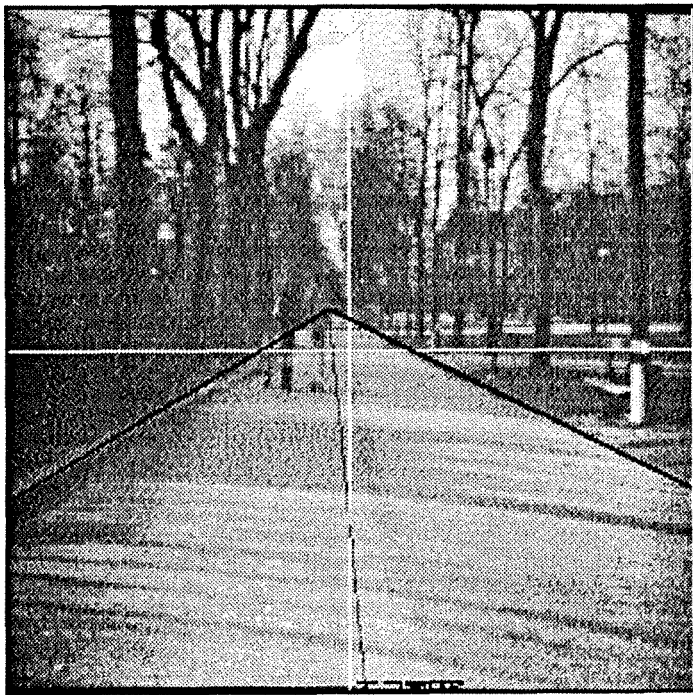
$$\theta = 18,40^\circ$$



a : Scene 1



b: Scene 2



c : Scène 3

## V.5 - DETERMINATION DES PARAMETRES PHYSIQUES

Le modèle de la scène, qui est une description du chemin observé est fourni par le système de vision après le traitement de chaque image acquise. Il donne ainsi une idée sur l'environnement du robot à l'instant de l'acquisition. En particulier, il contient des informations sur l'orientation du robot  $\theta$  par rapport à l'axe de la route, sur l'erreur de position ou décalage  $\delta$  par rapport à l'axe de la route, et enfin sur la largeur de la route  $\lambda$ .

Ce modèle s'écrit :

$$\mathcal{M}_s = \{ \theta, \delta, \lambda \} .$$

L'algorithme de sélection des bords de route nous fournit une paire valide  $\mathcal{P} = (F_{moy}, \theta_{moy}, \delta_{moy}, \lambda_{moy})$  qui décrit la route sur le plan de l'image. Mais  $F_{moy}$ ,  $\theta_{moy}$ ,  $\delta_{moy}$  et  $\lambda_{moy}$  sont des quantités exprimées dans le repère de l'image. Puisque le modèle de la scène doit être exploité par le module de supervision, il faut, à des fins de navigation, exprimer ces paramètres dans le repère de l'environnement. Par conséquent, à partir du triplet  $\{ \theta, \delta, \lambda \}$  donné par le modèle de la scène, on détermine un triplet  $\{ \Theta, \Delta, \Lambda \}$  exprimant les paramètres réels dans le repère de l'environnement. Pour exprimer ces paramètres il faut considérer la transformation qui permet le passage du repère de l'environnement à celui de l'image.

Avec la même configuration robot-caméra utilisée au chapitre IV, on considère que l'axe de la route est à une distance  $dx$  de la position du robot,  $x_r$  (voir figure V.5.1).

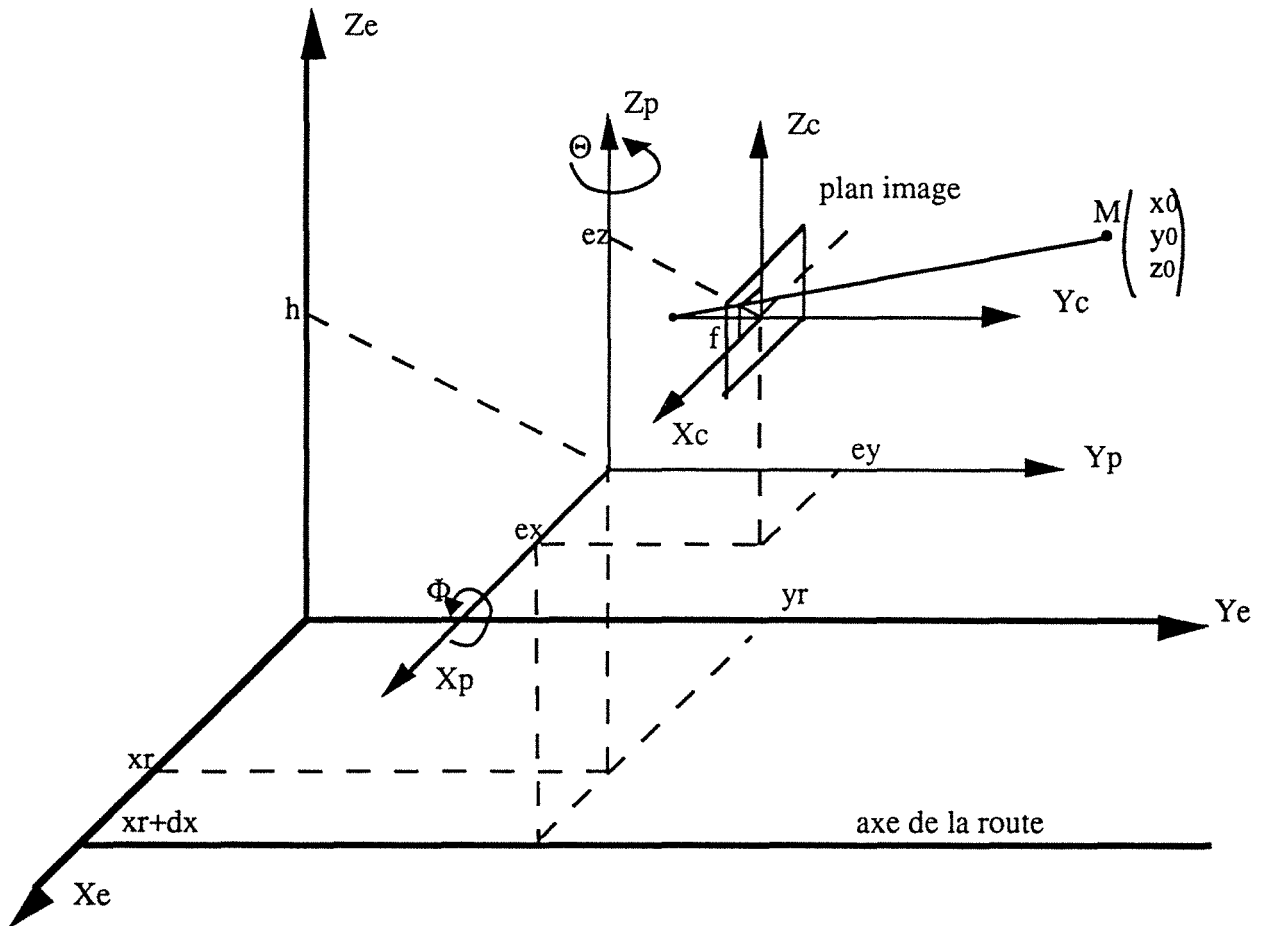


Figure V.5.1 : Position de Rc et Rp par rapport à Re.

On a trouvé que l'image d'un point M, dont les coordonnées sont  $(x_0, y_0, z_0)$  sur le plan de l'environnement, a pour coordonnées sur le plan de l'image ( section IV.3.3) :

$$x_i = f \frac{x_0 \cos(\Theta) - y_0 \sin(\Theta) + A}{x_0 \sin(\Theta) \cos(\Phi) + y_0 \cos(\Theta) \cos(\Phi) - z_0 \sin(\Phi) + B + f} \quad (V.51)$$

$$z_i = f \frac{x_0 \sin(\Theta) \sin(\Phi) + y_0 \cos(\Theta) \sin(\Phi) + z_0 \cos(\Phi) + C}{x_0 \sin(\Theta) \cos(\Phi) + y_0 \cos(\Theta) \cos(\Phi) - z_0 \sin(\Phi) + B + f} \quad (V.52)$$

Rappelons les expressions des paramètres A, B, et C :

$$A = -x_r \cos(\Theta) + y_r \sin(\Theta) - e_x$$

$$B = -x_r \sin(\Theta)\cos(\Phi) - y_r \cos(\Theta)\cos(\Phi) + h \sin(\Phi) - e_y$$

$$C = -x_r \sin(\Theta)\sin(\Phi) - y_r \cos(\Theta)\sin(\Phi) - h \cos(\Phi) - e_z$$

Pour étudier la position du robot par rapport à la route, on considère que l'hypothèse de planéité du terrain impose la condition  $z_o=0$ . Ainsi, les coordonnées dans le plan de l'image deviennent:

$$x_i = f \frac{x_o \cos(\Theta) - y_o \sin(\Theta) + A}{x_o \sin(\Theta)\cos(\Phi) + y_o \cos(\Theta) \cos(\Phi) + B + f}$$

(V.53)

$$z_i = f \frac{x_o \sin(\Theta) \sin(\Phi) + y_o \cos(\Theta)\sin(\Phi) + C}{x_o \sin(\Theta)\cos(\Phi) + y_o \cos(\Theta) \cos(\Phi) + B + f}$$

(V.54)

### V.5.1 - Détermination de $\Theta$ :

L'erreur d'orientation mesurée sur le plan de l'image est  $\theta$ , correspondant à l'angle entre l'axe de la route et l'axe vertical de la scène sur le plan de l'image. Pour trouver l'orientation réelle  $\Theta$  entre le repère de la plate-forme  $R_p$  et celui du monde  $R_e$ , on maintient fixe l'angle de rotation  $\Phi$  du repère  $R_p$  autour de l'axe  $X_p$ . On remarque, en particulier, que l'angle  $\Phi$  n'influe pas sur l'orientation  $\Theta$ .

Supposons que le robot soit centré sur l'axe de la route, ainsi l'axe de la route aura comme équation dans le repère  $R_e$  :

$$x_o = x_r + e_x.$$



La représentation de cet axe sur le plan de l'image s'obtient en faisant tendre  $y_0$  vers l'infini pour avoir le point de fuite. Son orientation par rapport à l'axe vertical du plan de l'image varie en fonction de l'orientation de la plate-forme  $\Theta$  et de la distance focale  $f$  ( voir figure V.5.1.1 ).

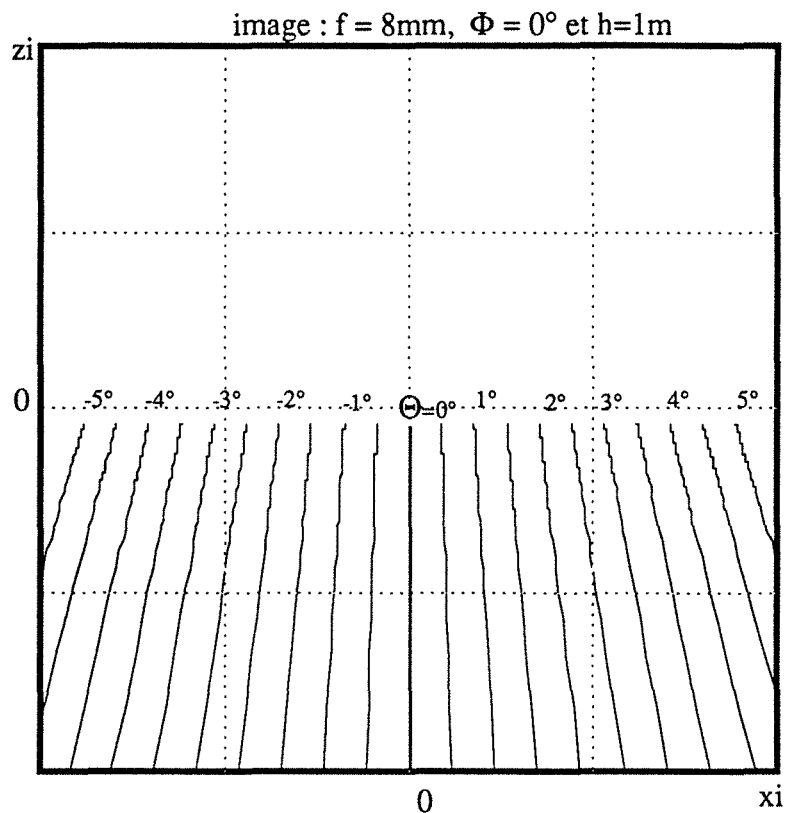


Figure V.5.1.1 : orientation de l'axe de la route en fonction de l'orientation  $\Theta$

En fait , sur le plan de l'image, l'axe de la route est décrit par une équation de la forme:  $z_i = \tan (\alpha) \cdot x_i + b$ , donc , on peut déterminer les paramètres  $\tan (\alpha)$  et  $b$ .

En plus , on remarque que sur la figure V.5.1.2 :

$$\alpha = \frac{\pi}{2} - \theta.$$

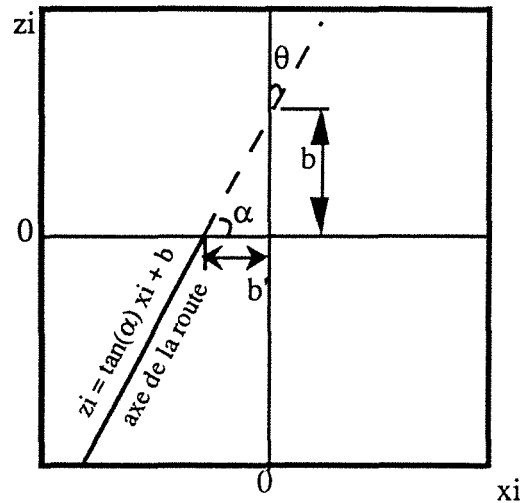


Figure V.5.1.2 : relation entre la pente  $\alpha$  et l'erreur d'orientation  $\theta$ .

On note aussi que:

$$\tan(\alpha) = \frac{b}{b'}$$

où  $b$  est la valeur de  $z_i$  à  $x_i = 0$ . L'abscisse  $x_i$  s'annule quand:

$$x_o \cos(\Theta) - y_o \sin(\Theta) + A = 0 \quad \Rightarrow \quad y_o = \frac{x_o \cos(\Theta) + A}{\sin(\Theta)}$$

en remplaçant  $y_o$  par sa valeur dans  $z_i$ , on obtient :

$$b = f \frac{x_o \sin(\Phi) + A \cos(\Theta) \sin(\Phi) + C \sin(\Theta)}{x_o \cos(\Phi) + A \cos(\Theta) \cos(\Phi) + (B+f) \sin(\Theta)} \quad (V.55)$$

En remplaçant  $A$ ,  $B$  et  $C$  par leurs expressions, on obtient :

$$b = f \frac{ex \sin(\Phi)[1-\cos(\Theta)] - h \cos(\Phi) \sin(\Theta) - ez \sin(\Theta)}{ex \cos(\Phi)[1-\cos(\Theta)] + h \sin(\Phi) \sin(\Theta) - (ey - f) \sin(\Theta)} \quad (V.55')$$

Et  $b'$  est la valeur de  $x_i$  quand  $z_i = 0$ . C'est à dire quand :

$$x_0 \sin(\Theta) \sin(\Phi) + y_0 \cos(\Theta) \sin(\Phi) + C = 0 \quad \Rightarrow \quad y_0 = \frac{-C - x_0 \sin(\Theta) \sin(\Phi)}{\cos(\Theta) \sin(\Phi)} \quad (\text{V.56})$$

En portant cette valeur dans l'expression de  $x_i$ , on obtient :

$$b' = f \frac{x_0 \sin(\Phi) + C \sin(\Theta) + A \cos(\Theta) \sin(\Phi)}{-C \cos(\Theta) \cos(\Phi) + (B+f) \sin(\Phi) \cos(\Theta)} \quad (\text{V.57})$$

En remplaçant A, B et C par leurs expressions, on obtient :

$$b' = f \frac{ex \sin(\Phi) [1 - \cos(\Theta)] - h \cos(\Phi) \sin(\Theta) - ez \sin(\Theta)}{h \cos(\Theta) - (ey - f) \sin(\Phi) \cos(\Theta) + ez \cos(\Theta) \cos(\Phi)} \quad (\text{V.57'})$$

Donc le rapport  $b/b'$  devient :

$$\frac{b}{b'} = \frac{\cos(\Theta) [h - (ey - f) \sin(\Phi) + ez \cos(\Phi)]}{ex \cos(\Phi) [1 - \cos(\Theta)] + \sin(\Theta) [h \sin(\Phi) - ey + f]} = \tan(\alpha)$$

Ainsi , on obtient une relation entre  $\Theta$ ,  $\Phi$  et  $\theta$  qui est la suivante :

$$\tan\left(\frac{\pi}{2} - \theta\right) = \frac{\cos(\Theta) [h - (ey - f) \sin(\Phi) + ez \cos(\Phi)]}{ex \cos(\Phi) [1 - \cos(\Theta)] + \sin(\Theta) [h \sin(\Phi) - ey + f]}$$

(V.57'')

Pour étudier les variations de  $\theta$  en fonction de  $\Theta$  et  $\Phi$ , il faut maintenir une des variables  $\Theta$  ou  $\Phi$  constante et faire varier l'autre.

Considérons un cas particulier où  $\Phi$  est maintenu constant afin d'étudier la relation entre  $\Theta$  et  $\theta$ . On peut choisir  $\Phi=0^\circ$  pour simplifier les expressions de  $b$  et  $b'$ .

Les paramètres  $A$ ,  $B$  et  $C$  deviennent alors si  $\Phi=0^\circ$  :

$$\begin{aligned} A &= -xr \cos(\Theta) + yr \sin(\Theta) - ex \\ B &= -xr \sin(\Theta) - yr \cos(\Theta) - ey \\ C &= -(h + ez) \end{aligned}$$

Et :

$$\begin{aligned} b &= f \frac{-(h+ez)\sin(\Theta)}{xo+(-xr \cos(\Theta) + yr \sin(\Theta) - ex) \cos(\Theta) + (-xr \sin(\Theta) - yr \cos(\Theta) - ey+f) \sin(\Theta)} \\ &= f \frac{-(h+ez)\sin(\Theta)}{xo - xr - ex \cos(\Theta) - (ey - f) \sin(\Theta)} \end{aligned} \quad (V.58)$$

et

$$\begin{aligned} b' &= f \frac{\sin(\Theta)}{-\cos(\Theta)} \\ &= -f \tan(\Theta) \end{aligned} \quad (V.59)$$

On remplace  $b'$  par sa valeur, on obtient :

$$\tan(\alpha) = \frac{b}{-f \tan(\Theta)} = \tan\left(\frac{\pi}{2} - \theta\right)$$

d'où :

$$\Theta = \arctan \left( -b / f \cdot \tan \left( \frac{\pi}{2} - \theta \right) \right)$$

(V.60)

Enfin, ayant obtenu la valeur de  $\theta$  pour le modèle de la scène on peut trouver l'erreur d'orientation de la plate-forme et donc du robot par rapport à l'axe de la route. La figure V. 5.1.3 nous donne une idée des variations de  $\theta$  quand l'orientation de la plate-forme  $\Theta$  passe de  $-30^\circ$  à  $30^\circ$ , avec la configuration suivante :

$e_x=0,01\text{m}$  ;  $e_y=0,01\text{m}$  ;  $e_z=0,01\text{m}$  ;  $f=8\text{mm}$ ; et  $h=1\text{m}$  .

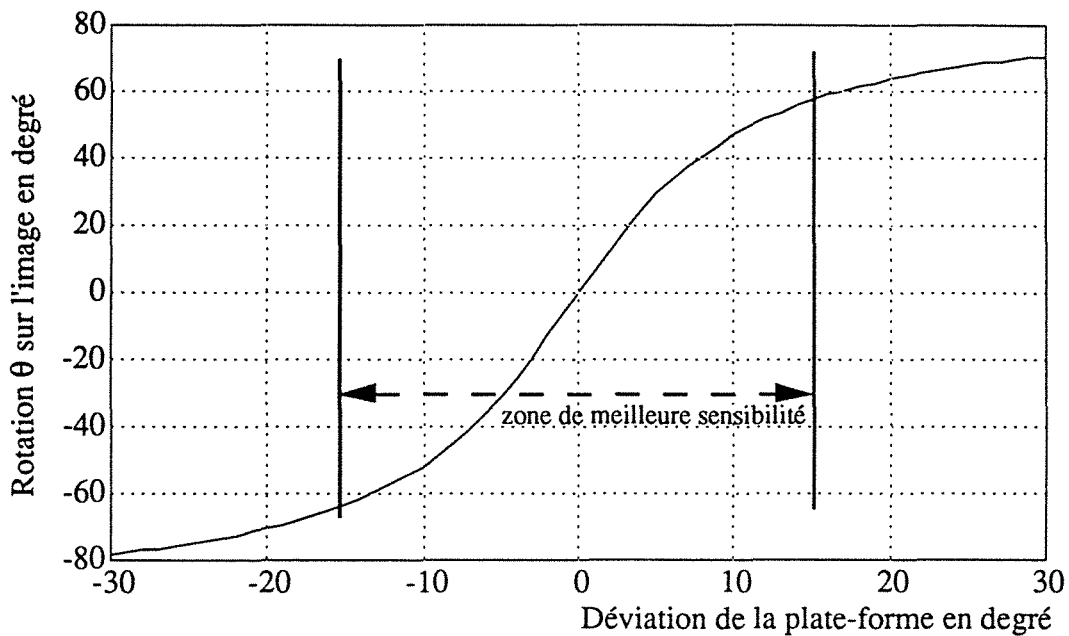


Figure V.5.1.3 : variation de  $\theta$  en fonction de  $\Theta$ .

On remarque que quand  $\Theta=0^\circ$  l'axe  $Y_c$  de la caméra est parallèle à l'axe de la route dont l'image coïncide avec l'axe vertical de la caméra donc  $\theta=0^\circ$ . La courbe n'est pas symétrique par rapport à l'origine car la rotation  $\Theta$  se fait autour de l'axe  $Z_p$  du repère de la plate-forme et non pas autour de  $Z_c$ .

V.5.2 - Détermination de  $\Delta$  :

Sur le plan de l'image  $z_i$  varie entre  $-z_{i\max}$  et  $z_{i\max}$ , l'erreur de décalage  $\delta$  correspond à la valeur de  $x_i$  quand  $z_i = -z_{i\max}$  ( voir figure V.5.2.1 ).

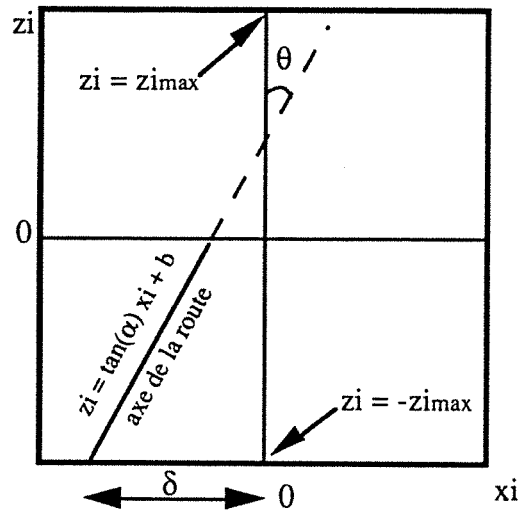


Figure V.5.2.1 : Décalage de l'axe de la route par rapport à celui de l'image.

Ainsi, quand  $z_i = -z_{i\max}$ , la valeur  $y_o$  est égale à:

$$-z_{i\max} = f \frac{x_o \sin(\Theta) \sin(\Phi) + y_o \cos(\Theta) \sin(\Phi) + C}{x_o \sin(\Theta) \cos(\Phi) + y_o \cos(\Theta) \cos(\Phi) + B + f} \quad (V.61)$$

soit

$$y_o = - \frac{x_o \sin(\Theta) [ f \sin(\Phi) + z_{i\max} \cos(\Phi) ] + z_{i\max}(B+f) + f C}{z_{i\max} \cos(\Theta) \cos(\Phi) + f \cos(\Theta) \sin(\Phi)}$$

(V.62)

En portant cette valeur dans l'expression de  $x_i$ , on obtient le décalage:

$$\delta = \frac{V1}{(B+f)\cos(\Theta)\sin(\Phi) - C \cos(\Theta)\cos(\Phi)} \quad (V.63)$$

Avec :

$$V1 = x_0 \left[ f \sin(\Phi) + z_{\max} \cos^2(\Theta) \cos(\Phi) \right] + z_{\max} \left[ \sin(\Theta) \cos(\Phi) + (B+f) \sin(\Theta) \right. \\ \left. + A \cos(\Theta) \cos(\Phi) \right] + f C \sin(\Theta) + A f \cos(\Theta) \sin(\Phi) \quad (V.64)$$

On peut aussi écrire  $\delta$  sous la forme suivante:

$$\delta = \frac{f \sin(\Phi) + z_{\max} \cos^2(\Theta) \cos(\Phi)}{(B+f)\cos(\Theta)\sin(\Phi) - C \cos(\Theta)\cos(\Phi)} x_0 + \\ \frac{z_{\max} \left[ \sin(\Theta) \cos(\Phi) + (B+f) \sin(\Theta) + A \cos(\Theta) \cos(\Phi) \right] + f C \sin(\Theta) + A f \cos(\Theta) \sin(\Phi)}{(B+f)\cos(\Theta)\sin(\Phi) - C \cos(\Theta)\cos(\Phi)} \quad (V.64')$$

Comme on le remarque sur cette expression (V.64'), l'erreur de décalage  $\delta$  est composée d'un terme dépendant du décalage généré par la position  $x_0$  de l'axe de la route, et un autre dû aux rotations  $\Phi$  et  $\Theta$ . La variation de  $x_0$  autour de  $x_r + e_x$  induit une rotation de l'axe de la route autour du point de fuite. La figure V.5.2.2 montre cette variation avec  $\Theta = \Phi = 0^\circ$ . En comparant cette figure avec la figure V.5.1.1, on remarque qu'une désorientation  $\Theta$  induit un décalage et modifie la position du point de fuite, tandis qu'un décalage  $\Delta$  induit un décalage  $\delta$  sur le plan de l'image et préserve la position du point de fuite.

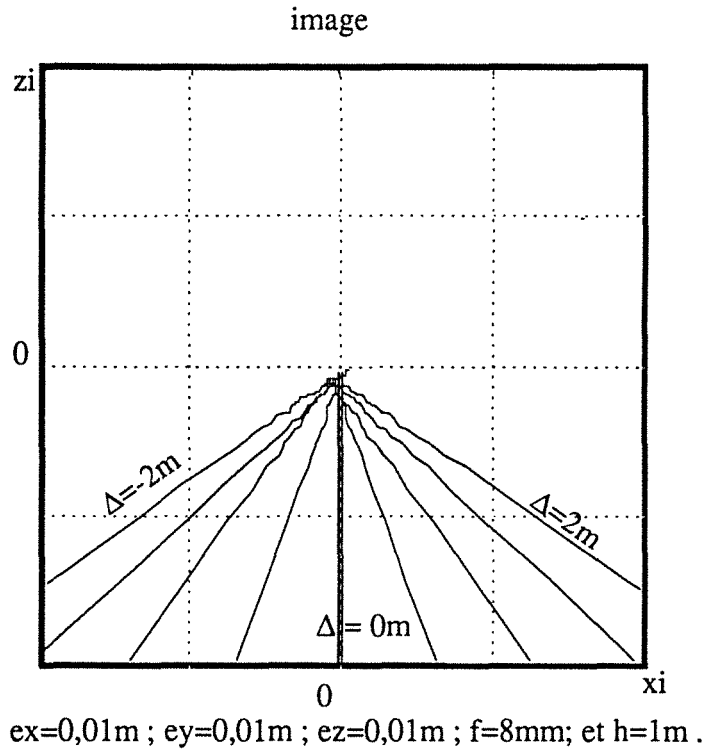


Figure V.5.2.2 : rotation de l'axe de la route en fonction du décalage  $\Delta$  avec  $\Phi=\Theta=0^\circ$ .

Un axe de route parallèle à celui de la caméra (alignement) est décrit par son équation dans le repère du monde :

$$x_0 = x_I + e_x. \quad (V.65)$$

A partir des expressions (V.63) et (V.64) on tire  $x_0$ , on aura donc :

$$x_0 = \frac{V_2}{f \sin(\Phi) + z_{\max} \cos^2(\Theta) \cos(\Phi)} \quad (V.66)$$

Avec :

$$V_2 = \delta \left[ (B+f) \cos(\Theta) \sin(\Phi) - C \cos(\Theta) \cos(\Phi) \right] - z_{\max} \left[ \sin(\Theta) \cos(\Phi) + (B+f) \sin(\Theta) + A \cos(\Theta) \cos(\Phi) \right] - f \cdot C \sin(\Theta) - A f \cos(\Theta) \sin(\Phi)$$



$$V2 = \delta \cos(\Theta) [ h - (ey - f) \sin(\Phi) + ez \cos(\Phi) ] - z_{\max} [ -xr \cos(\Phi) + h \sin(\Theta) \sin(\Phi) - (ey - f) \sin(\Theta) - ex \cos(\Theta) \cos(\Phi) + \sin(\Theta) \cos(\Phi) ] + f [ xr \sin(\Phi) + h \cos(\Phi) \sin(\Theta) + ex \cos(\Theta) \sin(\Phi) + ez \sin(\Phi) ] \quad (V.67)$$

S'il y a un décalage  $\Delta$  entre le robot et l'axe de la route celui-ci aura comme expression:

$$\Delta = x_o - x_r - ex$$

donc:  $x_o = \Delta + x_r + ex \quad (V.68)$

En remplaçant  $x_o$  par sa valeur , on obtient donc :

$$\Delta = \frac{V2}{f \sin(\Phi) + z_{\max} \cos^2(\Theta) \cos(\Phi)} - x_r - ex \quad (V.69)$$

où  $V2 = \delta \cos(\Theta) [ h - (ey - f) \sin(\Phi) + ez \cos(\Phi) ] - z_{\max} [ -xr \cos(\Phi) + h \sin(\Theta) \sin(\Phi) - (ey - f) \sin(\Theta) - ex \cos(\Theta) \cos(\Phi) + \sin(\Theta) \cos(\Phi) ] + f [ xr \sin(\Phi) + h \cos(\Phi) \sin(\Theta) + ex \cos(\Theta) \sin(\Phi) + ez \sin(\Theta) ]$

Comme précédemment, on considère le cas le plus simple où on pose  $\Phi=0^\circ$  pour simplifier l'expression de  $\Delta$  en fonction de  $\delta$ . On obtient :

$$\Delta = \frac{h+ez}{z_{\max} \cos(\Theta)} \delta + \frac{x_r + (ey - f - 1) \sin(\Theta) + ex \cos(\Theta)}{\cos^2(\Theta)} + \frac{(h+ez) f \sin(\Theta)}{z_{\max} \cos^2(\Theta)} - x_r - ex \quad (V.70)$$

Ainsi  $\Delta$  est une fonction linéaire de  $\delta$  dont la pente:  $\frac{(h+ez)}{z_{\max} \cos(\Theta)}$  varie uniquement en fonction de  $\Theta$  puisque  $h$ ,  $ez$ , et  $z_{\max}$  sont des constantes. En utilisant une caméra CCD dont les distances verticale et horizontale entre les pixels sont  $18 \mu\text{m}$  et  $30 \mu\text{m}$  respectivement, et en fixant  $f$  à  $8\text{mm}$ , sur une image de  $512 \times 512$  pixels, la variation de  $\Theta$  de  $-4^\circ$  à  $4^\circ$  donne des droites presque parallèles (voir figure V.5.2.3).

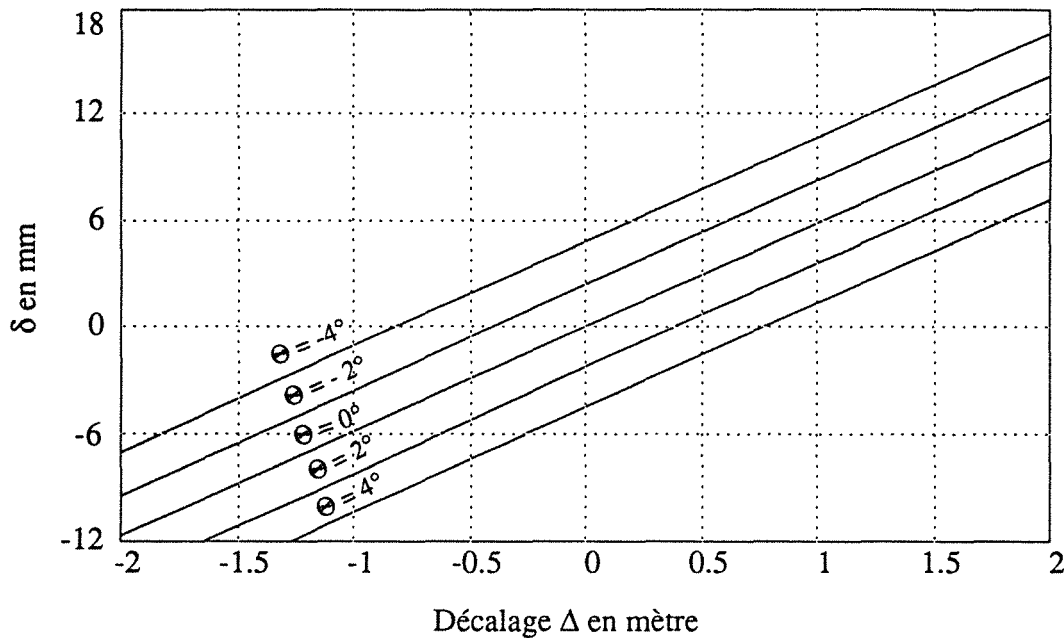


Figure V.5.2.3 : décalage  $\delta$  sur le plan de l'image en fonction du décalage  $\Delta$  avec  $\Phi=0^\circ$ .

Avec cette configuration, un décalage  $\Delta = 0 \text{ m}$  et une orientation de la plateforme  $\Theta$  de  $-2^\circ$ , donnent approximativement une erreur de décalage  $\delta$  de  $3 \text{ mm}$  équivalent à  $100$  pixels.

### V.5.3 - Détermination de $\Lambda$ :

En examinant la figure V.5.3.1, on remarque que pour trouver la largeur de la route il suffit de trouver le décalage engendré par chaque bord de route.

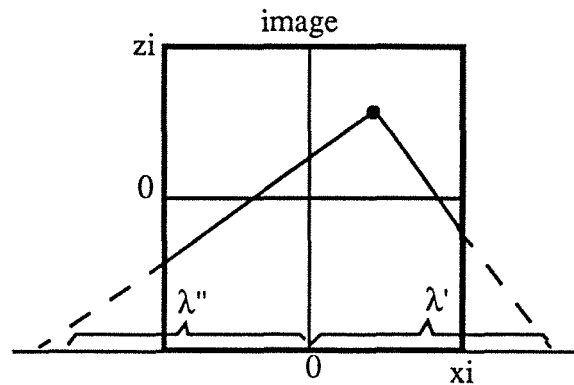


Figure V.5.3.1 : largeur de la route

On remarque que sur le plan de l'image:

$$\lambda = \lambda' - \lambda''.$$

Or, sur le repère du monde  $\lambda'$  et  $\lambda''$  reflètent des décalages physiques  $\Lambda'$  et  $\Lambda''$  respectivement qui sont donnés par  $\Delta$  et  $\Delta'$  obtenues à partir de la relation (V.69):

$$\Delta' = \frac{V'^2}{f \sin(\Phi) + z_{\max} \cos^2(\Theta) \cos(\Phi)} - x_r - e_x \quad (V.69)$$

où  $V'^2 = \delta' \cos(\Theta) [ h - (e_y - f) \sin(\Phi) + e_z \cos(\Phi) ] - z_{\max} [ -x_r \cos(\Phi) + h \sin(\Theta) \sin(\Phi) - (e_y - f) \sin(\Theta) - e_x \cos(\Theta) \cos(\Phi) + \sin(\Theta) \cos(\Phi) ] + f [ x_r \sin(\Phi) + h \cos(\Phi) \sin(\Theta) + e_x \cos(\Theta) \sin(\Phi) + e_z \sin(\Theta) ]$

Et :

$$\Delta'' = \frac{V''^2}{f \sin(\Phi) + z_{\max} \cos^2(\Theta) \cos(\Phi)} - x_r - e_x \quad (V.69)$$

où  $V''^2 = \delta'' \cos(\Theta) [ h - (ey - f) \sin(\Phi) + ez \cos(\Phi) ] - z_{\max} [ -xr \cos(\Phi) + h \sin(\Theta) \sin(\Phi) - (ey - f) \sin(\Theta) - ex \cos(\Theta) \cos(\Phi) + \sin(\Theta) \cos(\Phi) ] + f [ xr \sin(\Phi) + h \cos(\Phi) \sin(\Theta) + ex \cos(\Theta) \sin(\Phi) + ez \sin(\Theta) ]$

Mais, physiquement la largeur de la route est donnée , par :

$$\Lambda = \Lambda' - \Lambda'' = \Delta' - \Delta''$$

donc :

$$\Lambda = \Delta' - \Delta'' = \frac{[h - (ey - f) \sin(\Phi) + ez \cos(\Phi)] \cos(\Theta)}{f \sin(\Phi) + z_{\max} \cos^2(\Theta) \cos(\Phi)} (\delta' - \delta'') \quad (V.71)$$

et puisque :

$$\delta' = \lambda', \delta'' = \lambda'' \text{ donc } \delta' - \delta'' = \lambda' - \lambda'' = \lambda$$

d'où :

$$\Lambda = \frac{[h - (ey - f) \sin(\Phi) + ez \cos(\Phi)] \cos(\Theta)}{f \sin(\Phi) + z_{\max} \cos^2(\Theta) \cos(\Phi)} \lambda \quad (V.72)$$

Si l'on reste dans la configuration précédente avec  $\Phi=0^\circ$ , on obtient:

$$\Lambda = \frac{h + ez}{z_{\max} \cos(\Theta)} \lambda$$

et la seule variable de cette expression est  $\Theta$ . Puisque la distance verticale entre les pixels est  $18 \mu\text{m}$ , une image de  $512 \times 512$  pixels pour  $z_{\text{max}} : 4,608 \text{ mm}$ .

La figure V.5.3.2 décrit la variation  $\lambda$  sur le plan de l'image en fonction de la variation de la largeur de la route avec différentes valeurs de l'orientation de la plateforme  $\Theta$ .

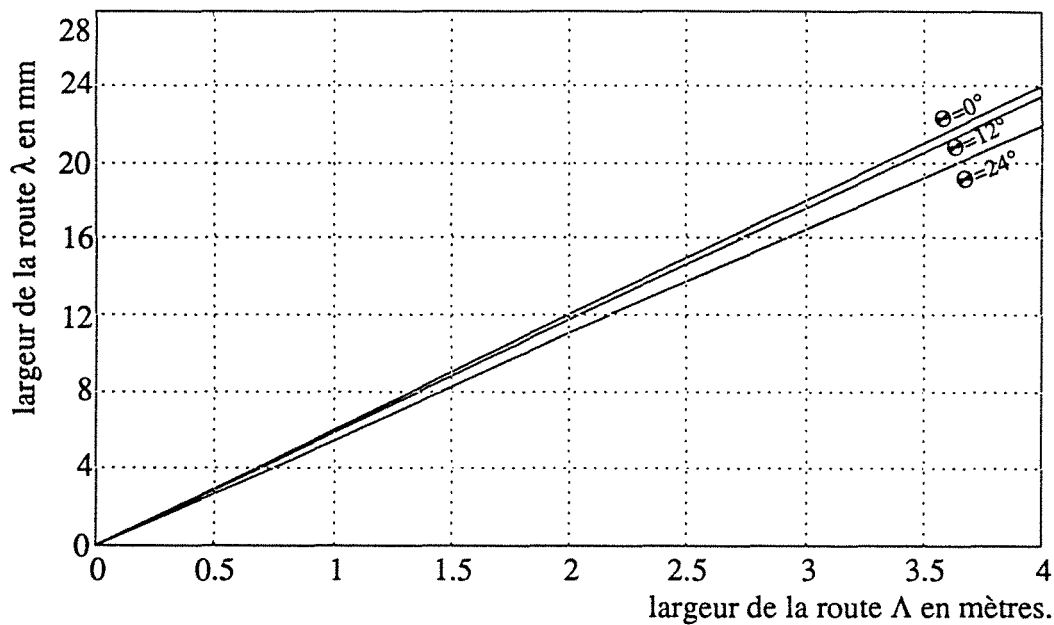


Figure V.5.3.2 : variation de  $\lambda$  en fonction de  $\Lambda$ .

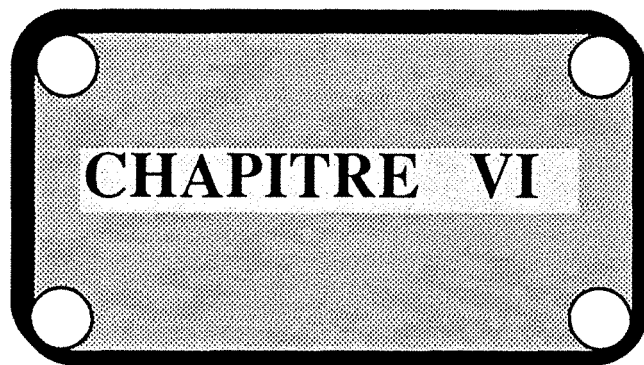
En remarque donc que seul  $\theta$  ne varie pas linéairement avec  $\Theta$  alors que  $\delta$  et  $\lambda$  sont des fonctions linéaires de  $\Delta$  et  $\Lambda$  pour l'axe de la caméra maintenu horizontalement.

## V.6 - CONCLUSION

La base de connaissance décrite dans ce chapitre permet d'établir une sélection rigoureuse des bords de route parmi les différents éléments dominants détectés au

niveau de l'exploitation de la scène. Elle détermine donc le modèle de la scène qui comprend la position et la direction du robot, et la largeur de la route. Ce modèle sera exploité par les modules assurant la pilotage du robot.

En utilisant le modèle de la caméra, on a étudié le passage des paramètres du modèle de la scène aux paramètres physiques, et la variation des uns par rapport aux autres. Ceci a permis de calibrer la caméra et de vérifier la robustesse du système de vision. Plusieurs extensions sont envisagées sur ce module notamment l'introduction et la combinaison des résultats de la segmentation avec ceux qu'on a actuellement, et l'élaboration d'un processus de recalage du robot par vision.



APPLICATION : TRAITEMENT  
D'UNE SEQUENCE D'IMAGES  
DECRIVANT UN PARCOURS DU  
ROBOT.



## VI.1 - CALIBRATION DE LA CAMERA

La calibration de la distance focale de la caméra est établie sur le principe des triangles semblables appliqué à une image grille. On suppose que la longueur réelle d'un côté d'un carré de la grille est  $l$ , et sa longueur observée sur le plan de l'image  $k$ . On considère aussi la distance  $d$  qui sépare l'objectif de la caméra à la grille [TSA-86] (comme indiqué par la figure VI.1.1). Par conséquent, la distance focale  $f$ , définie par la distance entre le centre de la lentille et le plan de l'image, est exprimée par :  $f = k \frac{d}{l}$ .

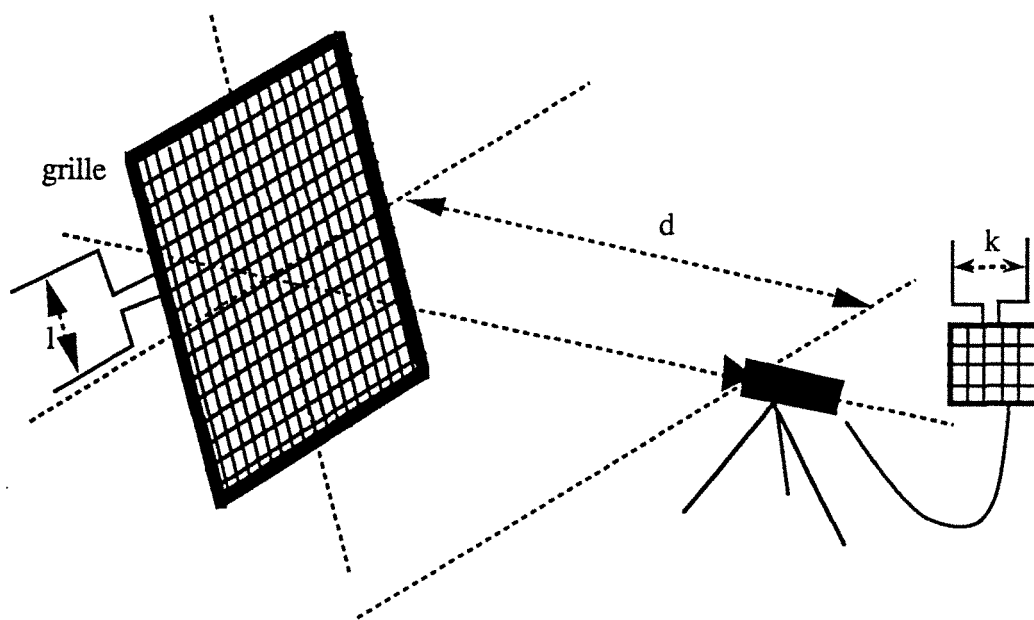


Figure VI.1.1: Calibration de la distance focale.

Le centre de l'image, défini par l'intersection du plan de l'image avec l'axe optique de la caméra, est utilisé comme origine du repère de l'image. La méthode utilisée pour calibrer sa position est celle du point de fuite. Comme indiqué par la figure VI.1.2, une forme rectangulaire noire est disposée de telle façon qu'elle soit parallèle à l'axe optique de la caméra, et que son axe horizontal soit à la même hauteur  $h$  que celui de la caméra.

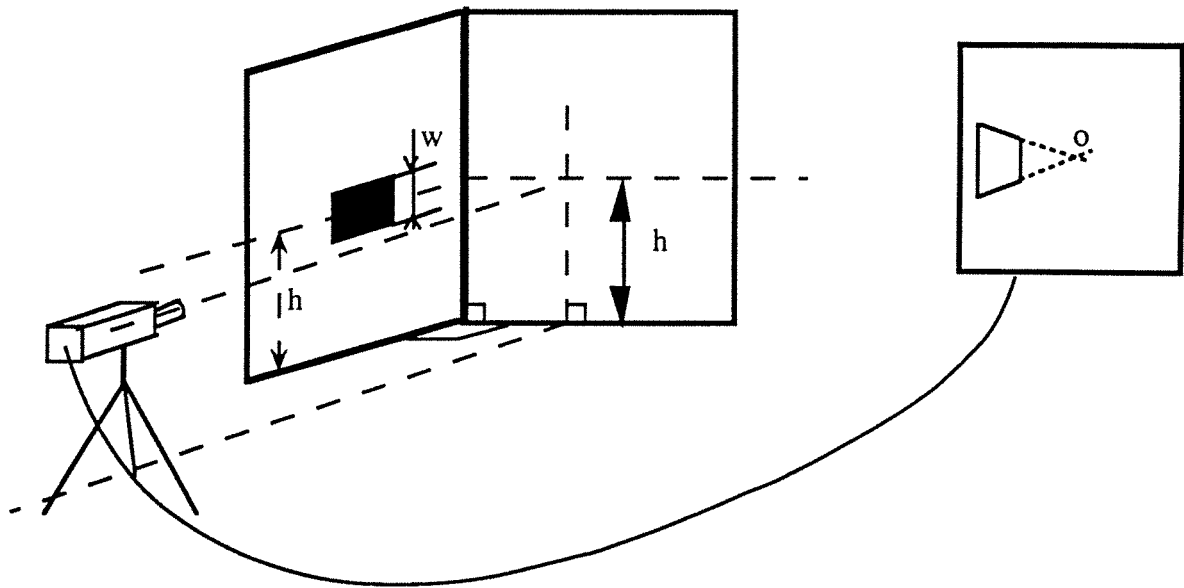


Figure VI.1.2: Calibration du centre de l'image.

Puisque les côtés haut et bas du rectangle sont parallèles à l'axe optique, leur intersection sur le plan de l'image doit, selon le principe du point de fuite, coïncider avec le centre du plan de l'image.

En négligeant la distortion géométrique ( $K_d \approx 0$ ), le centre de l'image  $O$ , a pour coordonnées  $(252, 257)$  ce qui est légèrement déplacé par rapport au centre spécifié par le constructeur. Les dimensions d'un pixel données par le constructeur sont  $l_x = 30 \mu\text{m}$  et  $l_y = 18 \mu\text{m}$ , et la distance focale  $f$  est en moyenne  $8,51 \text{mm}$ .

## VI.2 - DETERMINATION DES PARAMETRES PHYSIQUES

Pour tester les performances de notre système dans la détermination des paramètres physique  $\Theta$ ,  $\Delta$  et  $\Lambda$  à partir des quantités  $\theta$ ,  $\delta$  et  $\lambda$  mesurées sur le plan de l'image, on a utilisé la configuration indiquée par la figure VI.2.1. La plateforme ne peut exécuter qu'une rotation autour de l'axe  $Z_p$ , ce qui correspond à la déviation horizontale  $\Theta$  de la caméra. Ainsi, la déviation verticale  $\Phi$  se trouve approximativement nulle ( $\Phi=0^\circ$ ) sur un terrain plat. La position du repère de la caméra par rapport à celui de la plateforme est donnée par les coordonnées  $e_x$ ,  $e_y$  et  $e_z$ . Pour simplifier, l'axe  $Z_e$  du repère de l'environnement coïncide avec l'axe  $Z_p$  et donc les coordonnées du repère  $R_p$  de la plateforme par rapport à celui de l'environnement  $R_e$  sont  $x_r=y_r=0$  et  $h=104$  cm. La distance focale est de 8,5 mm.

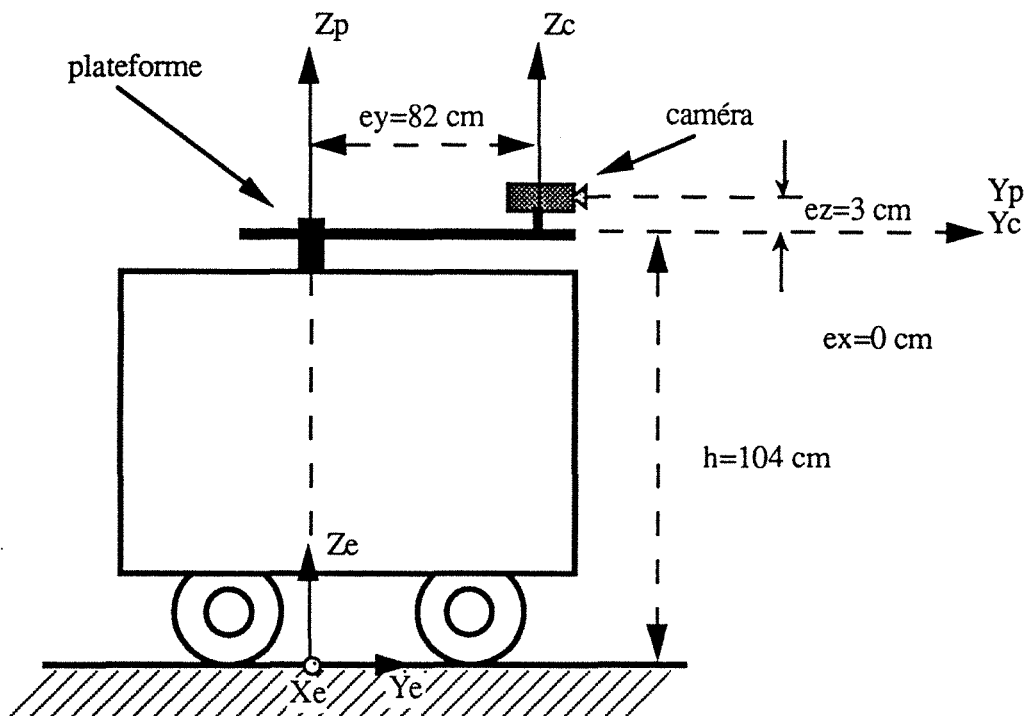


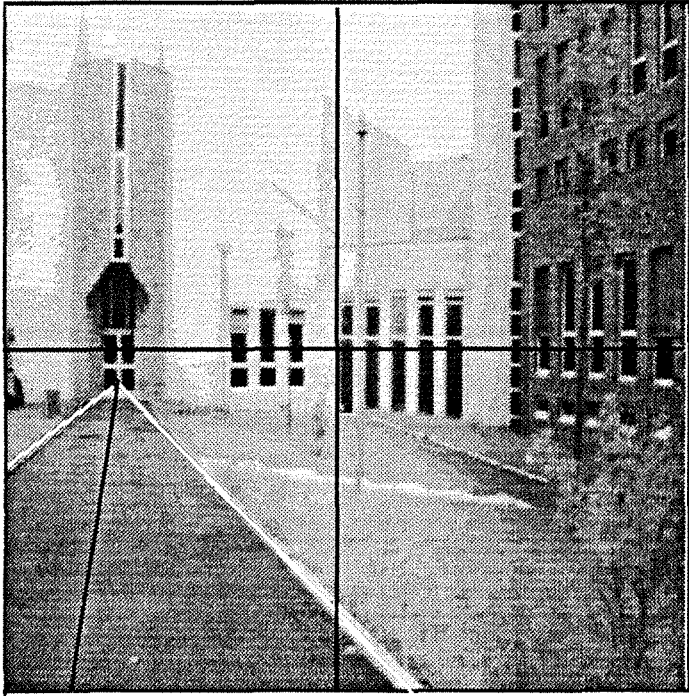
Figure VI.2.1: Configuration pour l'évaluation des paramètres physiques.

On a choisi une scène typique comprenant une route, et on a établi 8 acquisitions d'image correspondant à huit orientations différentes de la caméra ( $-20^\circ$ ,  $-15^\circ$ ,  $-10^\circ$ ,  $-5^\circ$ ,  $0^\circ$ ,  $5^\circ$ ,  $10^\circ$ ,  $15^\circ$ ). À partir de ces images (voir figure VI.2.2) l'algorithme détermine les bords et l'axe de la route, mesure les quantités  $\theta$ ,  $\delta$  et  $\lambda$ , et

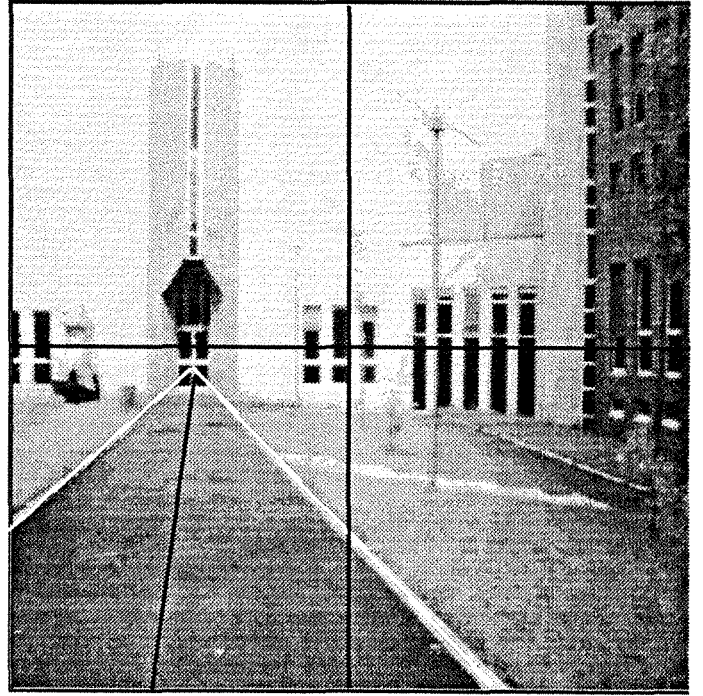
évalue les paramètres physiques de la configuration. Le tableau suivant montre les valeurs obtenues:

	$\theta$ en ( $^{\circ}$ )	$\Theta$ ( $^{\circ}$ ) calculée	$\Theta$ ( $^{\circ}$ ) théorique	$\delta$ en px	$\Delta$ (m) calculée	$\Delta$ (m) théorique	$\lambda$ en px	$\Lambda$ (m) calculée	$\Lambda$ (m) théorique
scène 1	-9,6887	-18,541	-20	(-) 213	-0,2112	0	498	3,5061	3,30
scène 2	-7,6433	-13,550	-15	(-) 150	-0,2802	0	498	3,5067	3,30
scène 3	-5,5722	-8,221	-10	(-) 93	-0,2547	0	499	3,4121	3,30
scène 4	-4,4462	-4,182	-5	(-) 39	-0,2776	0	498	3,4409	3,30
scène 5	0	0	0	20	0,1389	0	506	3,3333	3,30
scène 6	1,3980	6,7038	5	76	0,2415	0	500	3,4300	3,30
scène 7	2,7617	12,009	10	130	0,1276	0	512	3,4066	3,30
scène 8	4,0394	16,075	15	187	0,1905	0	503	3,4293	3,30

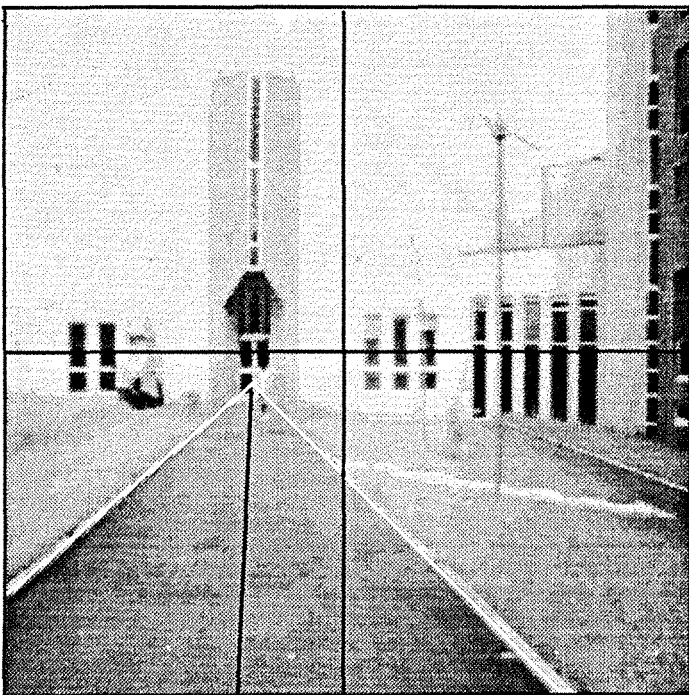
On constate que les erreurs relatives moyennes sur les paramètres  $\Theta$ ,  $\Delta$  et  $\Lambda$  sont respectivement 14,5056%, 21,05% et 4,06%. Ces erreurs sont dues en grande partie à l'hypothèse  $\Phi=0^{\circ}$ , car on peut remarquer sur toutes les scènes que le point de fuite ne se trouve pas sur l'axe des abscisses  $x_i$  mais en dessous. Ceci implique l'existence d'une déviation verticale vers le bas ( $\Phi < 0$ ), ou d'un manque de planéité du sol. En outre, La plateforme que nous avons utilisée ( voir figure VI.2.1) ne nous a pas permis de commander l'angle  $\Phi$



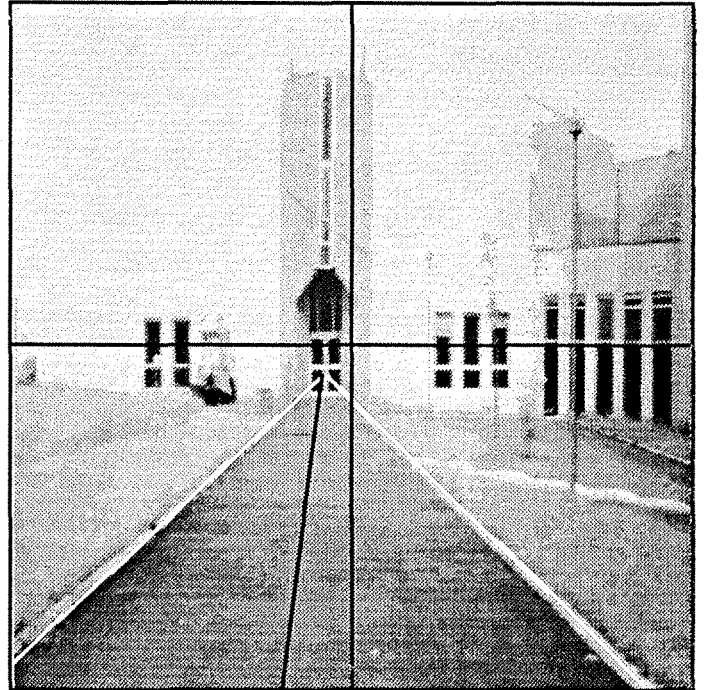
scène 1



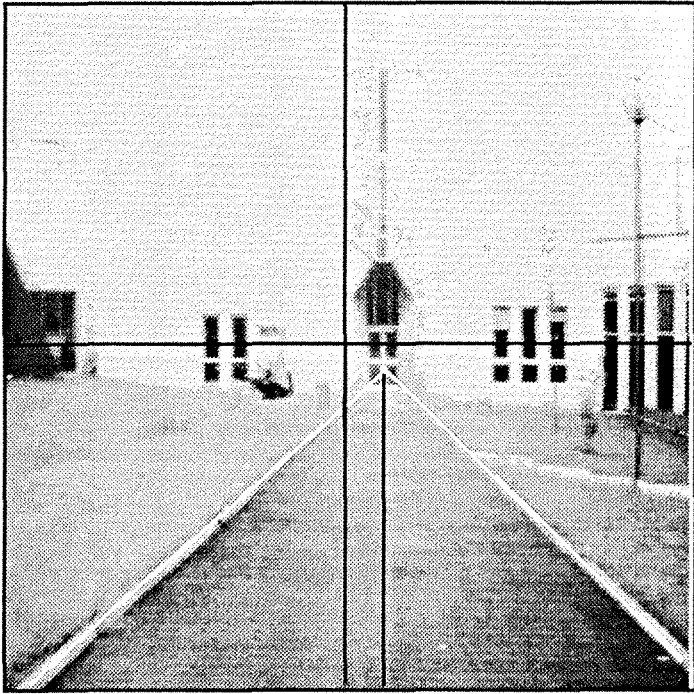
scène 2



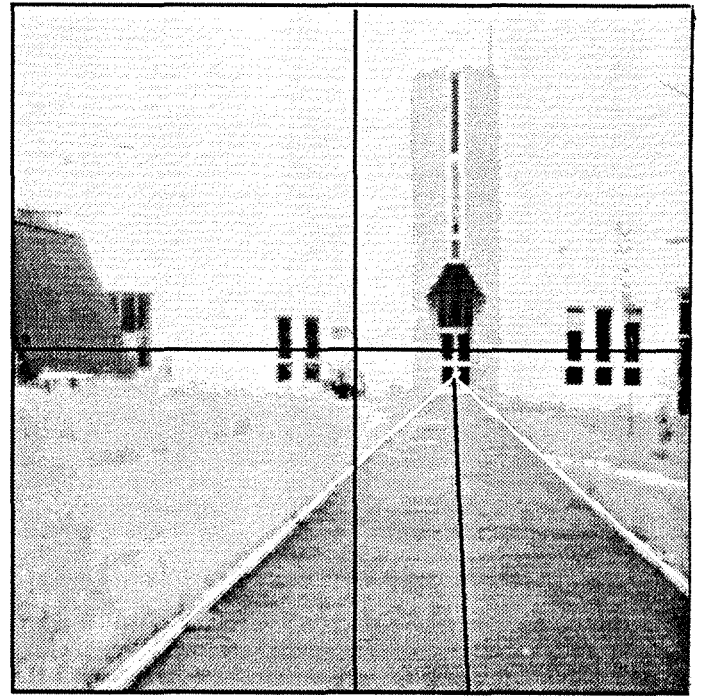
scène 3



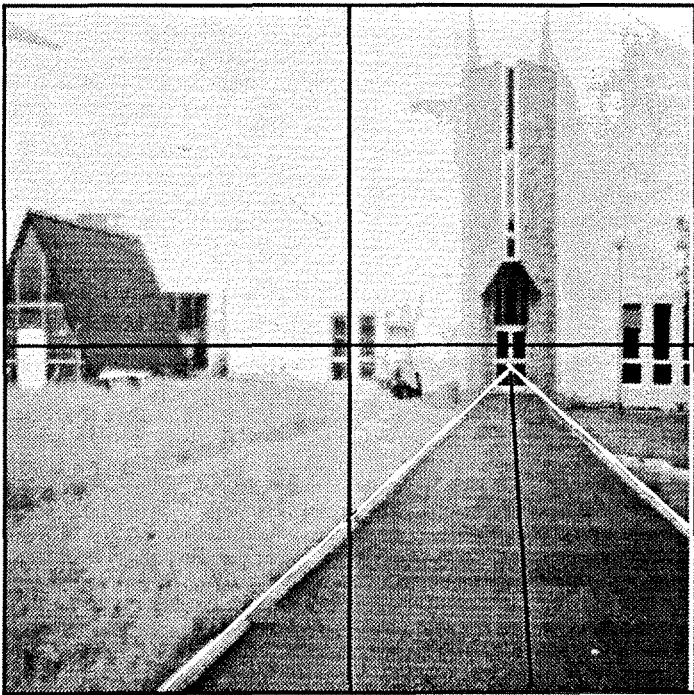
scène 4



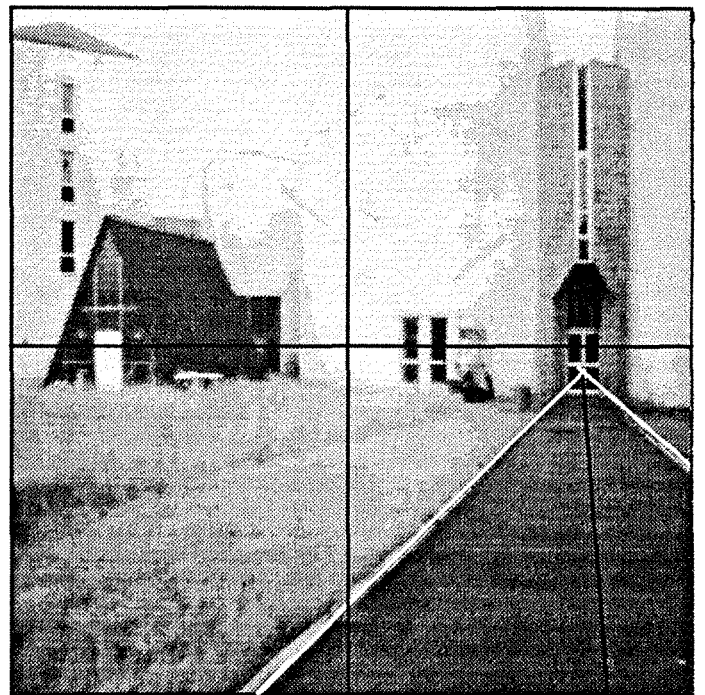
scène 5



scène 6



scène 7



scène 8

### VI.3-TRAITEMENT D'UNE SEQUENCE D'IMAGES ( SUIVI DE ROUTE)

Pour tester les performances de notre système dans la tâche de suivi de route, on a utilisé, pour établir l'acquisition d'images dans le parc urbain, le robot mobile dont la configuration est illustrée par la figure VI.3.1 . Celui-ci a été guidé manuellement pendant l'expérience. L'acquisition des images est assurée par une carte CYCLOPE montée sur un PC ( VICTOR-286 ). La séquence d'images est sauvegardée sur disque dur pour être traitée ensuite sur une station SUN 3/60.

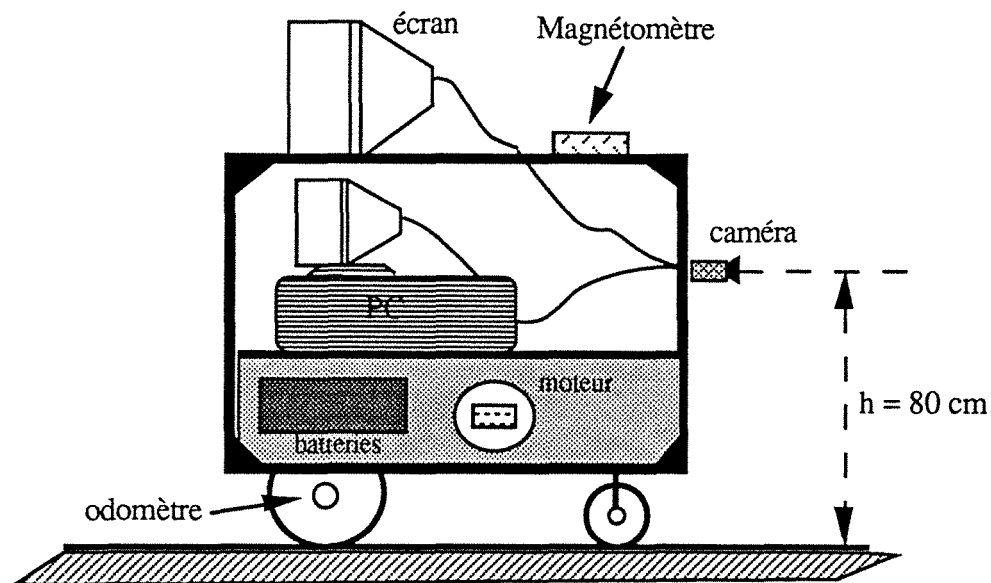
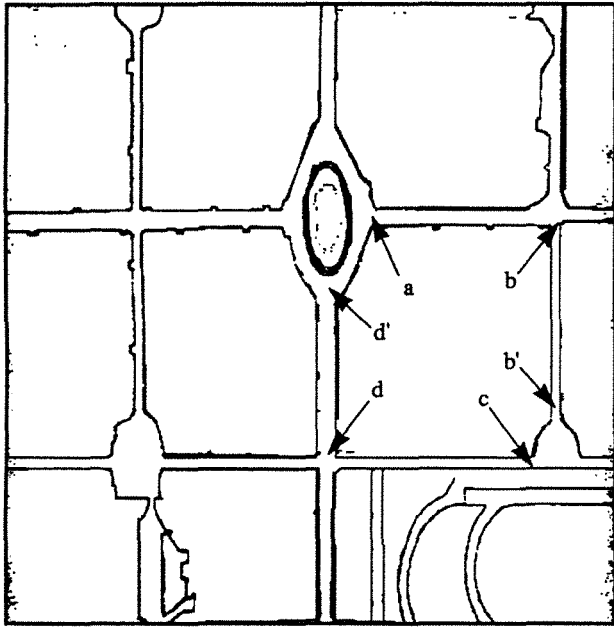
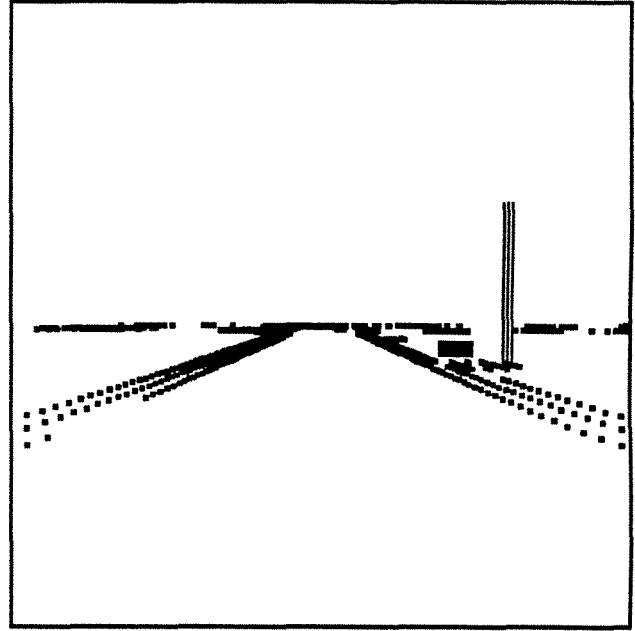


Figure V.3.1 : Robot mobile utilisé.

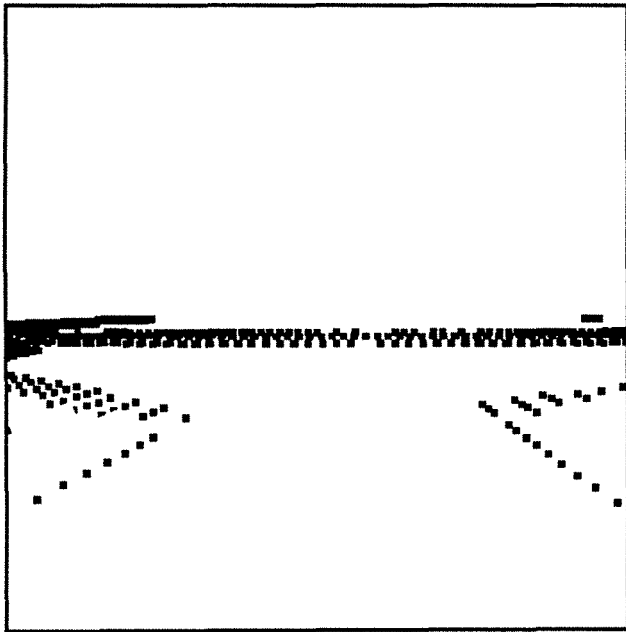
On a choisi d'effectuer les prises d'images le long d'un parcours type au sein du parc urbain . Ainsi on a pris 33 images à un intervalle fixe de 10m. Le parcours fermé est décrit par les points de passage abb'cdd'a comme indiqué sur la carte ( voir figure VI.3.2.a ) . Cette carte représente la région sud du parc avec une résolution de 2,048 px/m. En limitant le champ de vision à  $k=100m$ , la carte virtuelle aura comme dimensions 370 x 205 pixels puisque  $f=8,5mm$  et  $d_f=90,3529m$  ( voir section IV.3.2). Sur cette carte, on repère avec le maximum de précision possible la position des bords de route, des lampadaires et des bancs.



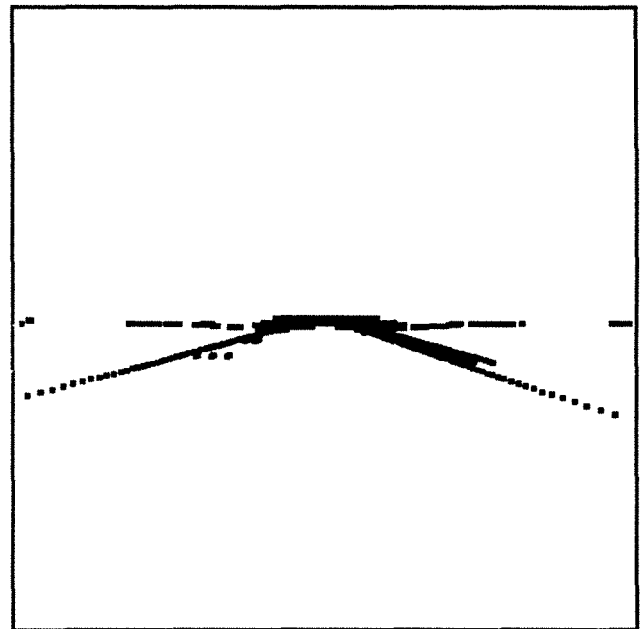
a: carte d'une région du parc urbain



b: scène correspondant à l'image initiale (en a).



c: scène correspondant à la 2ème intersection (en b')



d: scène correspondant à la 3ème intersection (en c)

Figure VI.3.2: carte et scènes prédites



Le robot entame son parcours au point (a) par l'acquisition de la première image pour pouvoir se situer par rapport aux bords de route. Cette image correspond à  $i_1$  sur la figure VI.3.3. Le traitement de la phase initiale est choisi pour améliorer cette image et extraire l'image orientation du gradient Sobel (voir  $i_2$ ). Après analyse de l'histogramme,  $i_2$  est segmentée et filtrée pour reconstruire une nouvelle image d'orientation du gradient  $i_3$  sur laquelle figurent seulement les éléments linéaires désirés. Sur  $i_1$  on constate que le bord gauche est de catégorie (a) alors que celui de droite est de catégorie (b) (voir chapitre III). Ceci peut aussi être constaté sur l'image reconstruite  $i_3$  par la présence de deux segments de droite représentant le bord gauche alors qu'il n'y en a qu'un seul pour le bord droit. La segmentation de  $i_3$  permet de représenter chaque direction dans une image comme le montrent les images binaires  $i_4$ ,  $i_5$ ,  $i_6$ , et  $i_7$ . L'application de la transformée de Hough à chaque image binaire permet d'associer à chaque élément linéaire une équation qui le représente sur le plan de l'image et un poids qui exprime son degré de linéarité. Ayant les coordonnées du point a sur la carte, le module de prédiction de scène permet d'avoir une image qui reflète la position des bords de route sur  $i_1$  ( voir figure VI.3.2.b). Cette image nous montre la présence d'une lampadaire et d'un banc à droite du bord droit de la route. L'application de la transformée de Hough à cette image permet d'établir un modèle prédit. En se basant sur ce dernier, le module de base de connaissance fait une comparaison avec ce qui a été détecté pour sélectionner enfin les bords de route (voir  $i_8$  ). Le modèle de la route est donc validé pour évaluer les paramètres physiques  $\Theta$ ,  $\Delta$ , et  $\Lambda$ .

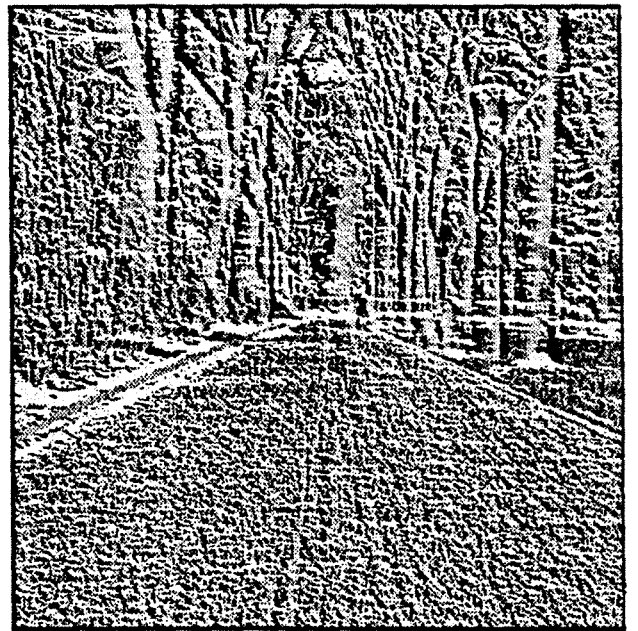
Quand le robot est en possession des paramètres physiques, il peut entamer la navigation et se déplacer d'une certaine distance  $D_{av}$ , en aveugle qui a été fixée dans notre cas à 10m. Durant son déplacement le robot n'utilise pas l'information visuelle mais l'information issue du magnétomètre et de l'odomètre. Au bout de la distance  $D_{av}$  le système entre en mode de suivi continu de chemin. Le module de prédiction de scène fournit une image semblable à celle de la figure V.3.2.b pour déterminer la position des fenêtres initiales à travers lesquelles les bords droit et gauche de la route pénètrent dans le plan de l'image. Les fenêtres sont ensuite placées sur la deuxième image acquise et le traitement en mode de suivi continu débute ( voir  $i_9$ ). Le placement des fenêtres se fait d'une manière séquentielle en autorisant une désorientation maximale de  $5^\circ$  du bord de route d'une fenêtre à l'autre jusqu'à ce que cette dernière atteigne 60% de l'image. Quand une fenêtre ne présente aucun segment, une autre fenêtre décalée de 10 pixels est placée et le traitement recommence sans contrainte de désorientation comme indiqué par les bords

gauche des images i9 et i11. Ce décalage permet de corriger une fausse détection notamment après la fenêtre initiale comme le cas du bord gauche de l'image i15 ou d'une divergence des fenêtres comme le cas du bord gauche de l'image i32. Pour une bonne visibilité, certaines fenêtres ont été retracées en noir.

Le traitement de l'image i17, qui correspond au point b sur la carte, est mis en l'échec à cause de l'absence du bord gauche de la route. Le bord droit a été correctement détecté par le placement séquentiel des trois fenêtres, mais le manque du bord gauche empêche d'établir le modèle de la scène et le traitement est ainsi abandonné. La deuxième intersection indiquée par b' sur la carte est rencontrée sur les images i25 et i26. On constate que la détection est correcte à cause des parties visibles des bords de route, et on peut voir sur la scène prédite (figure VI.3.2.c) l'élargissement de la route. D'autre part, rien n'a été détecté sur i26 qui est identique à i17 à cause de l'intersection. Ceci est dû à l'invisibilité du bord gauche et la fausse orientation du bord droit. L'image i36 a été prise en d après la troisième intersection, et on constate une détection incorrecte des bords de route à cause de la présence d'une perturbation (pietons) dans la fenêtre initiale gauche et la présence d'un objet inconnu dans la fenêtre initiale droite masquant complètement le bord droit. Dans la dernière scène i40, le système confond le bord du bassin avec le bord gauche de la route, et ceci induit un axe de route fictif qui a permis dans ce cas de guider correctement le robot vers la position initiale en a.



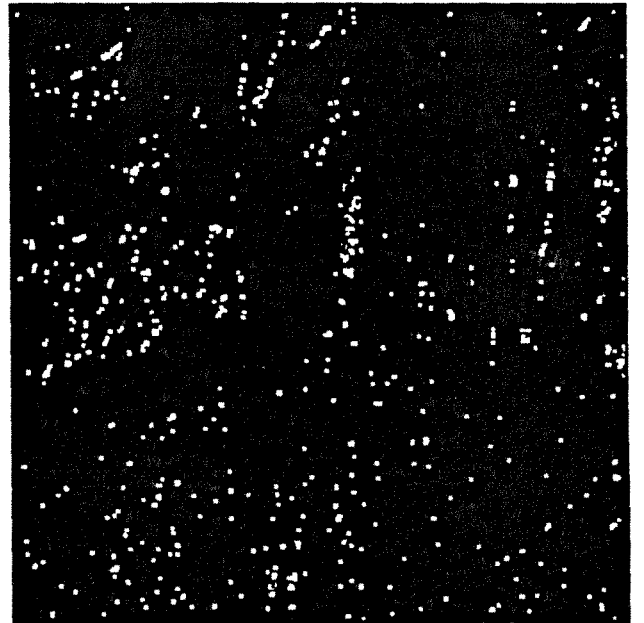
i1



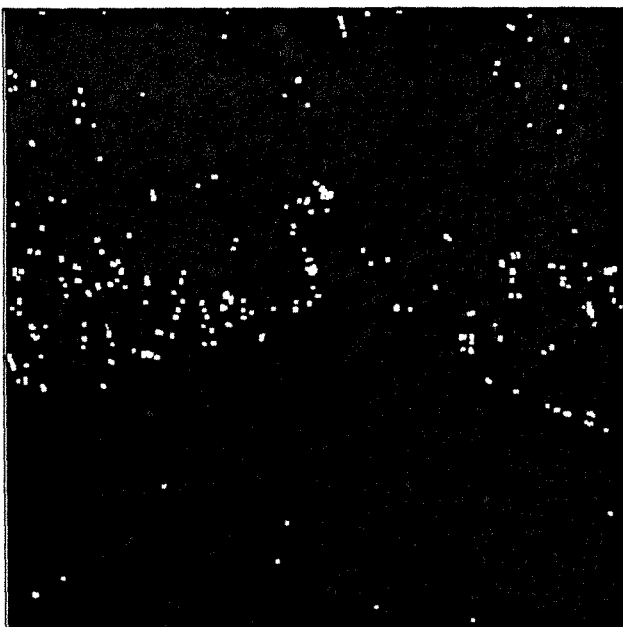
i2



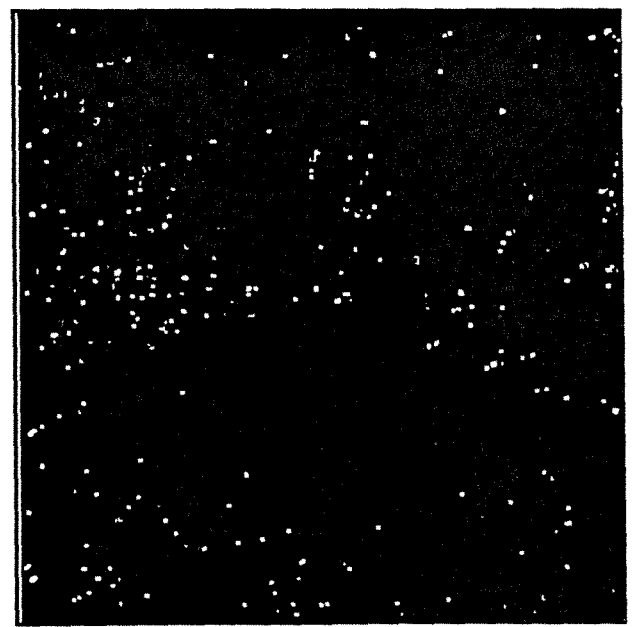
i3



i4



i5



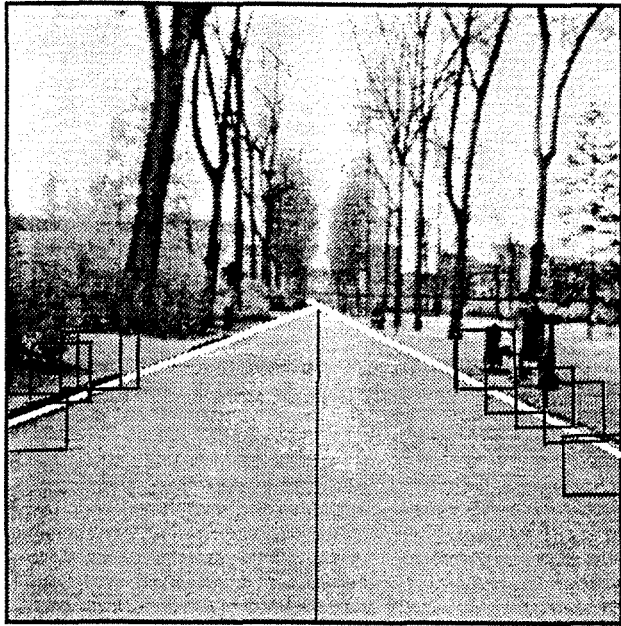
i6



i7



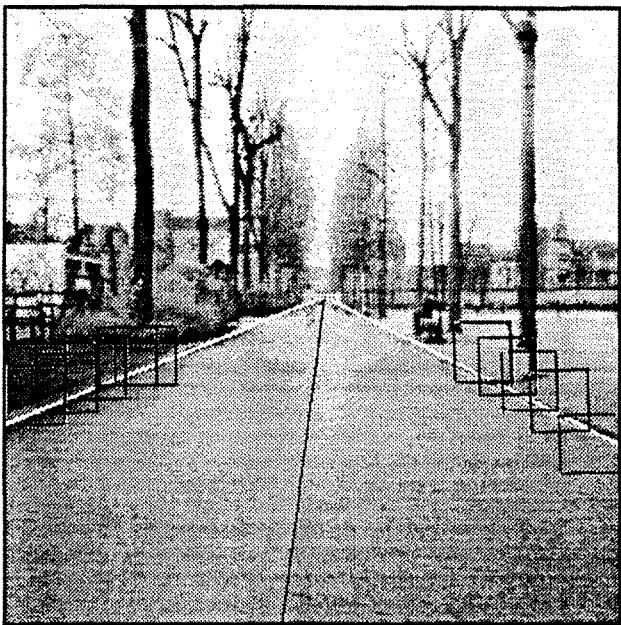
i8



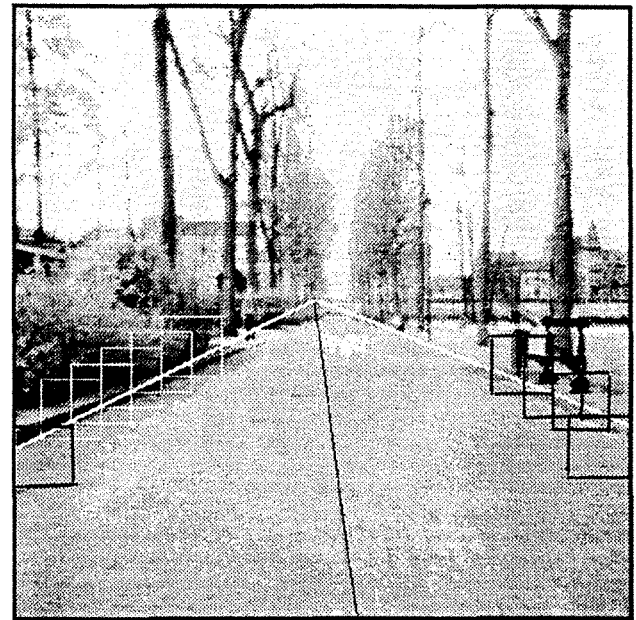
i9



i10

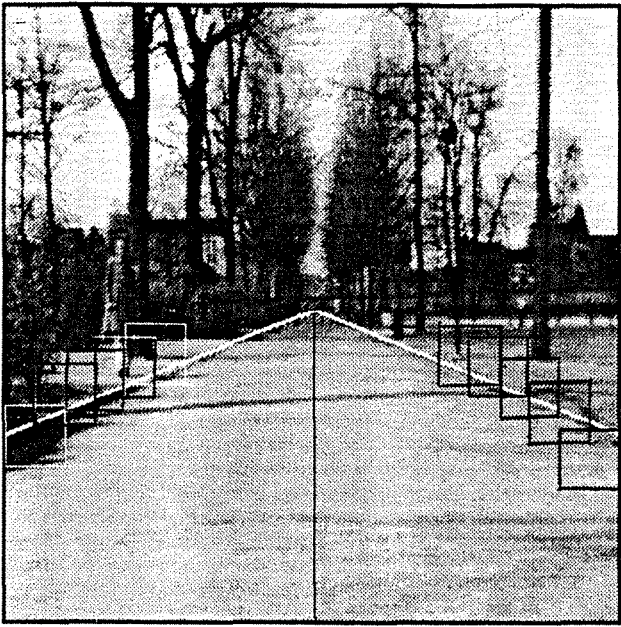


i11

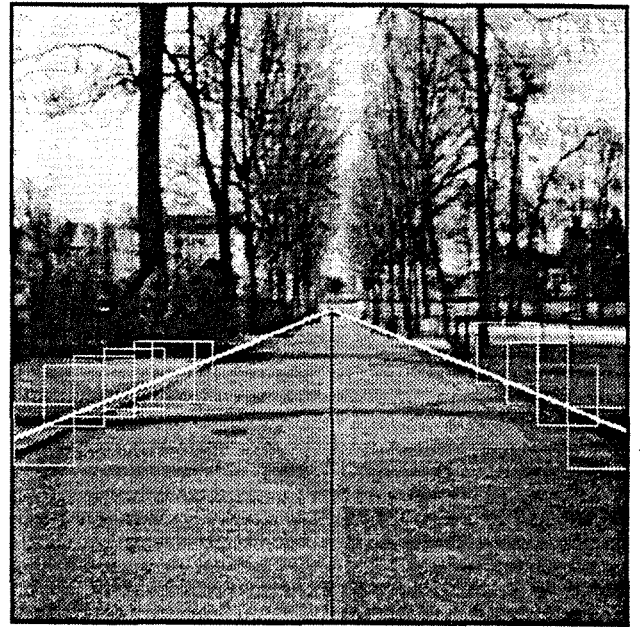


i12

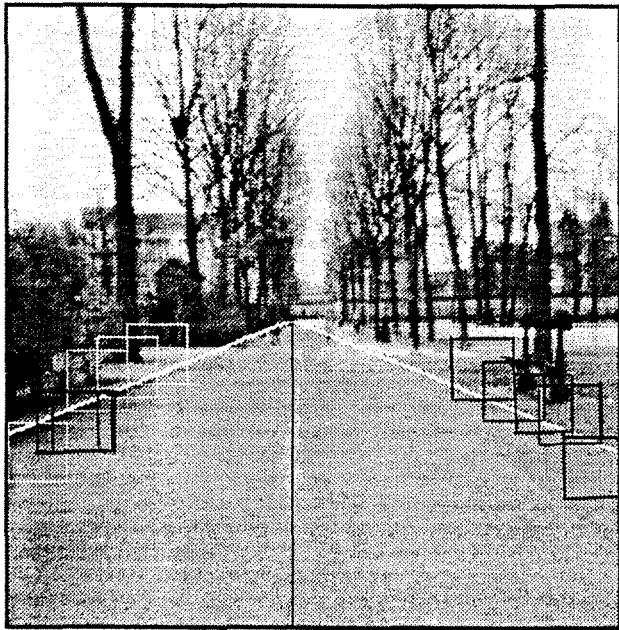




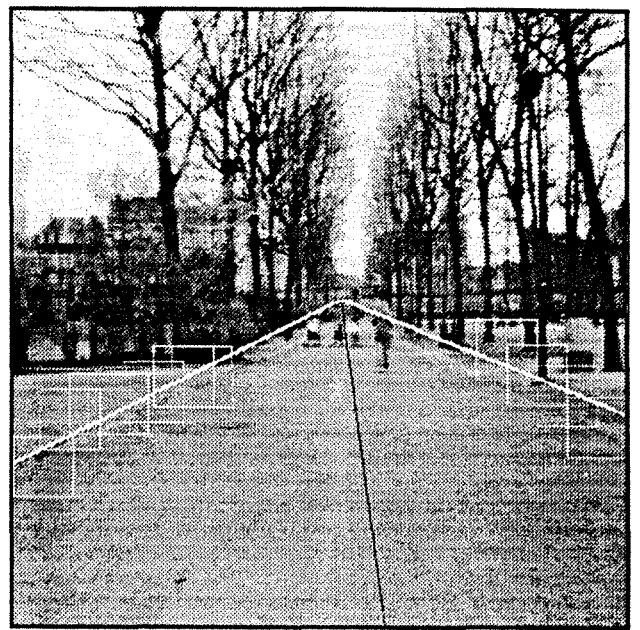
i13



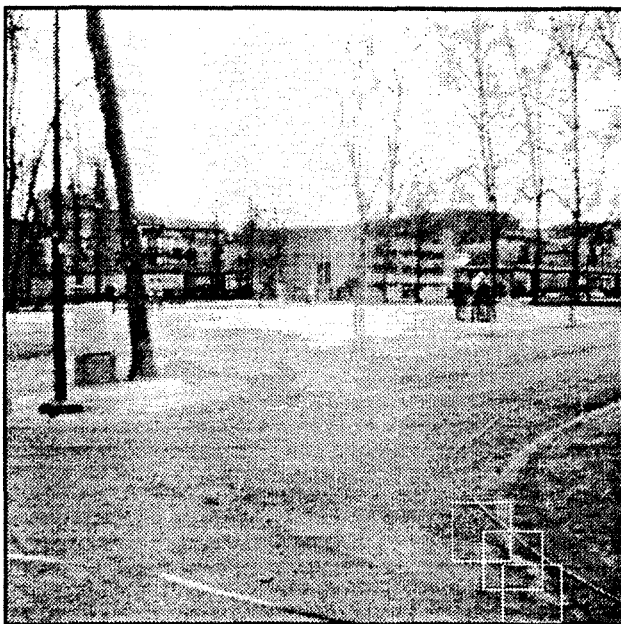
i14



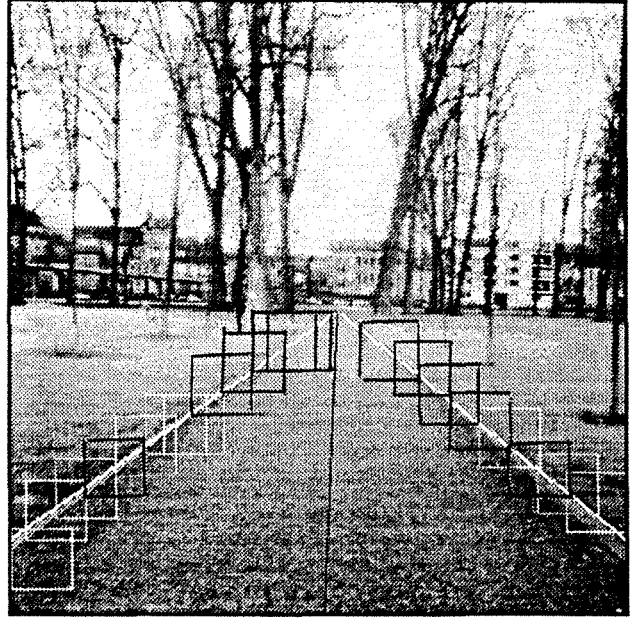
i15



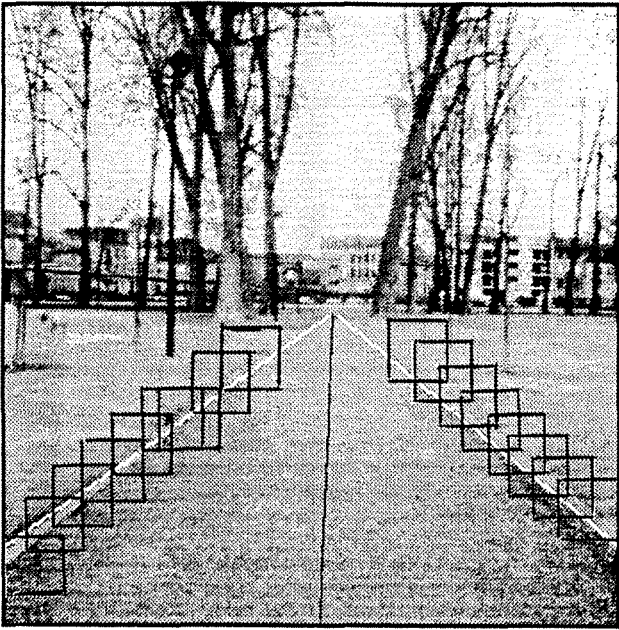
i16



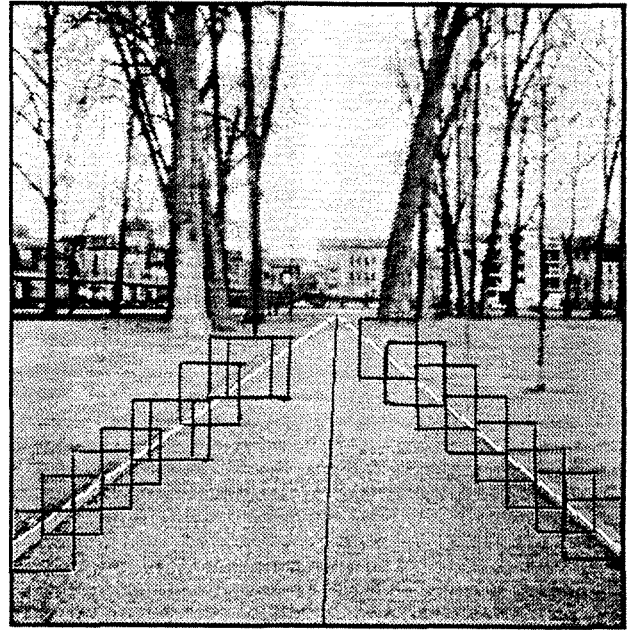
i17



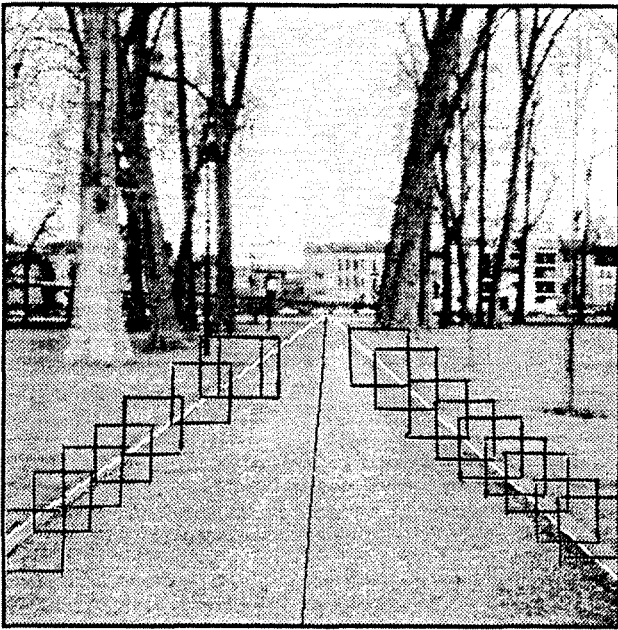
i18



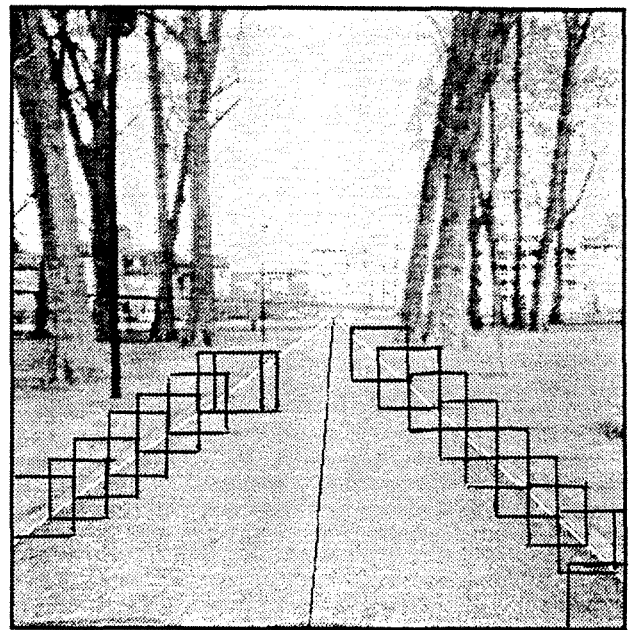
i19



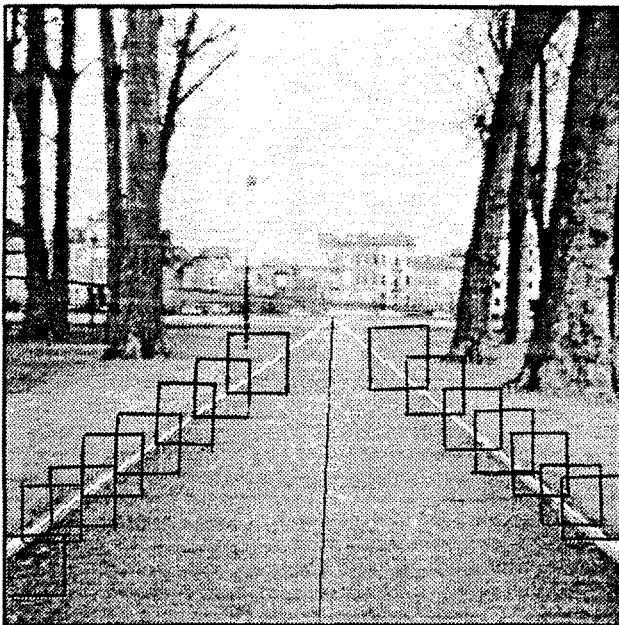
i20



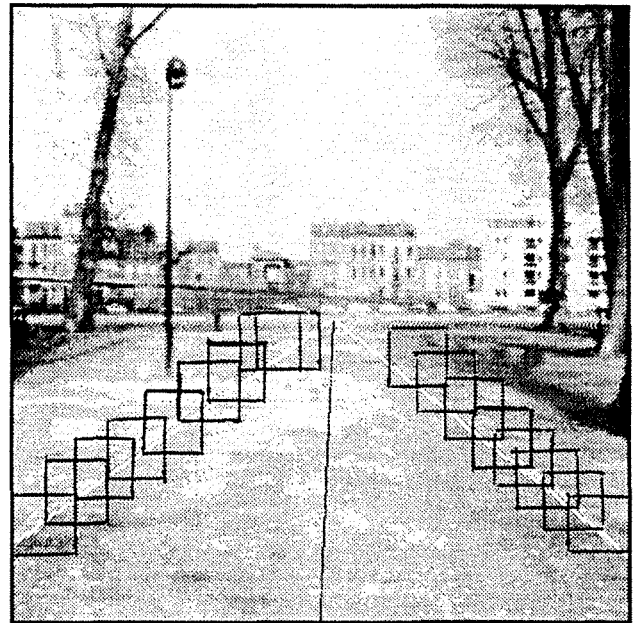
i21



i22

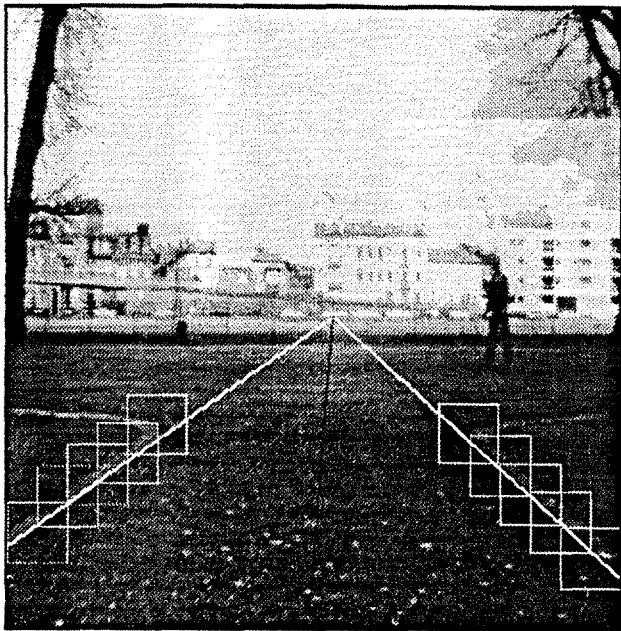


i23



i24

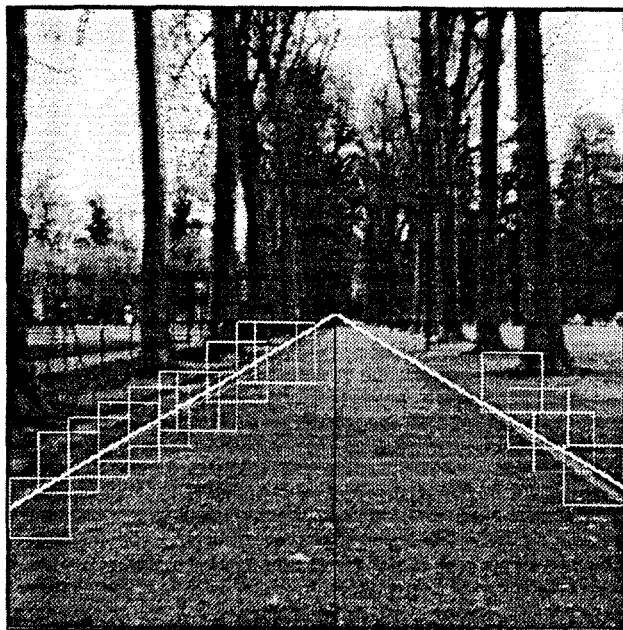




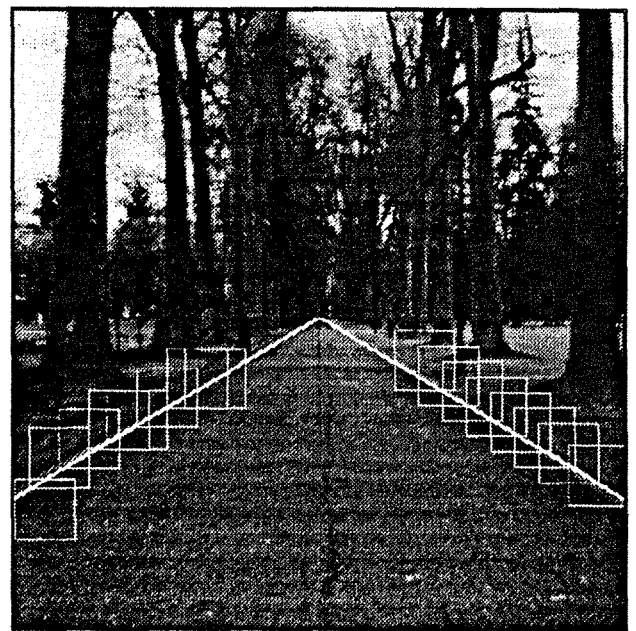
i25



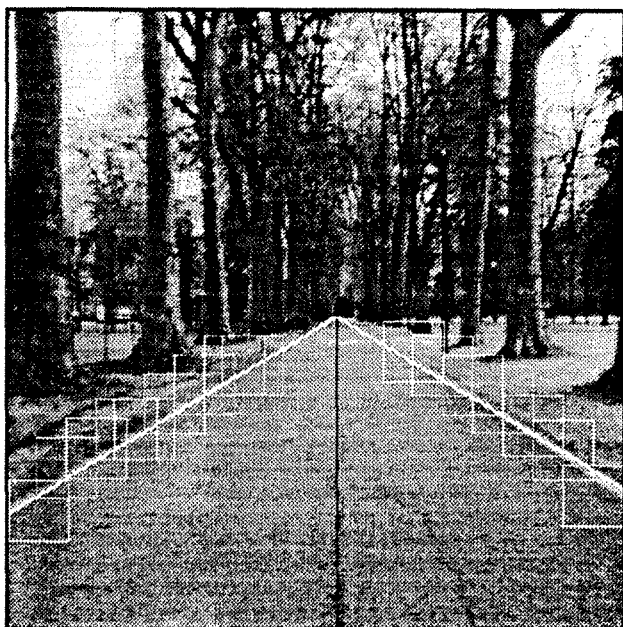
i26



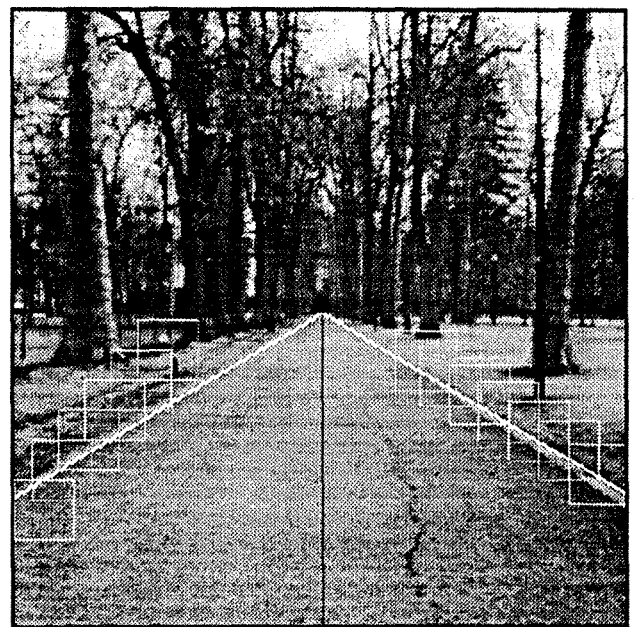
i27



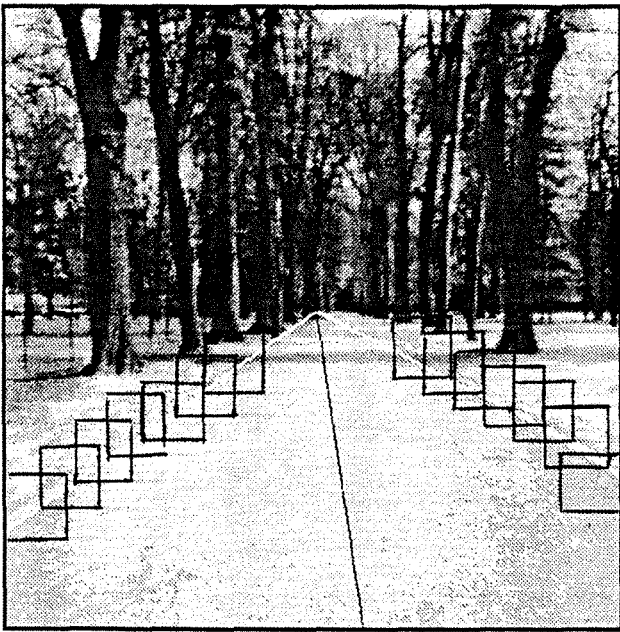
i28



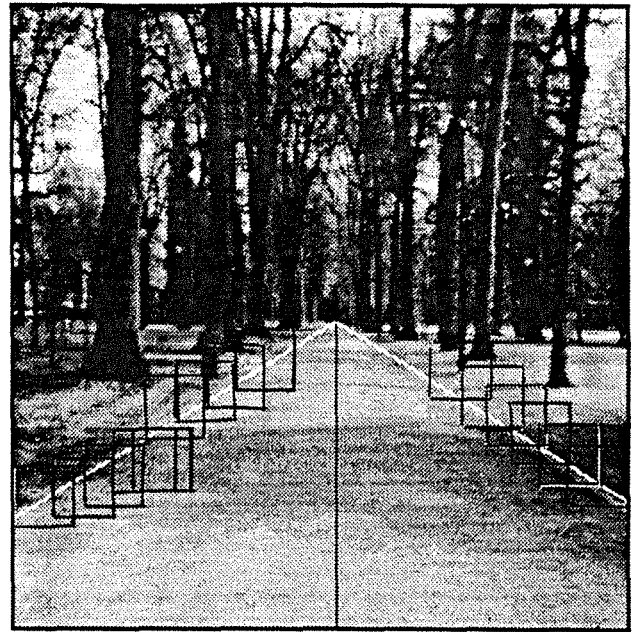
i29



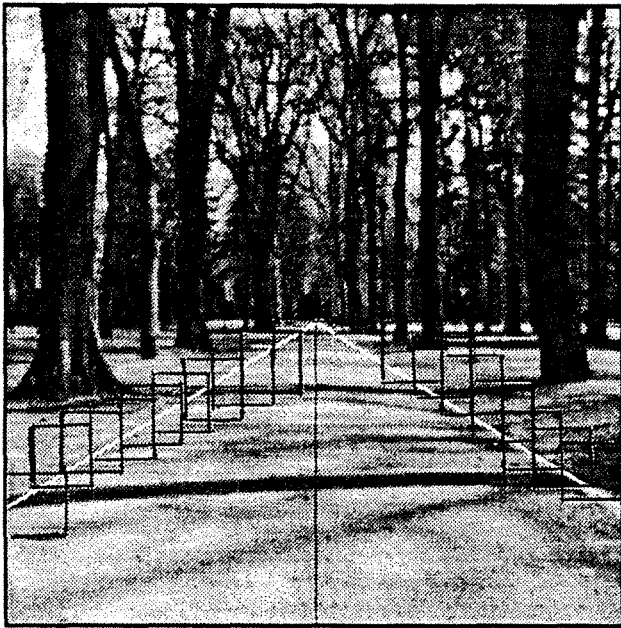
i30



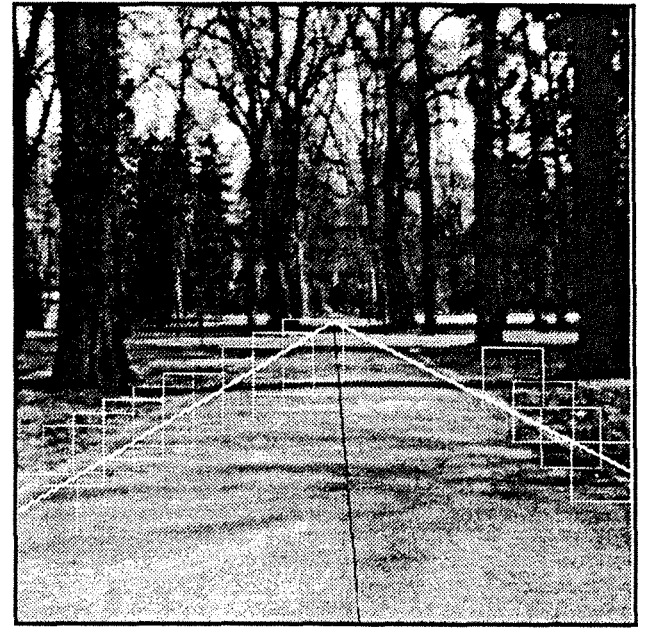
i31



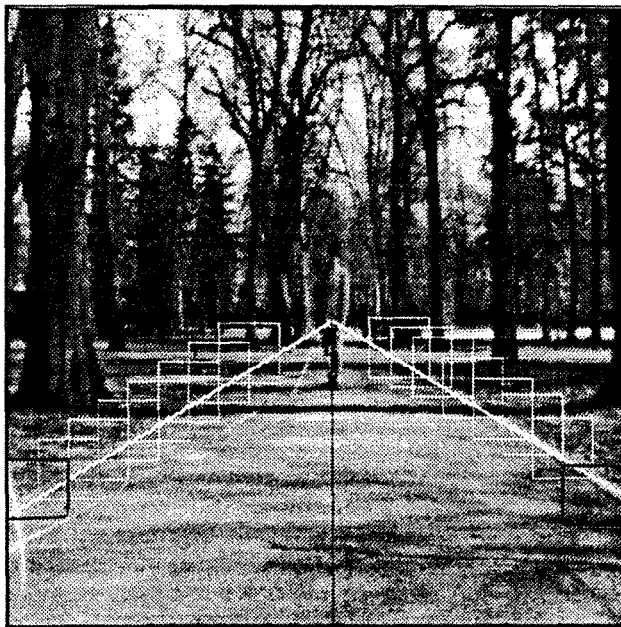
i32



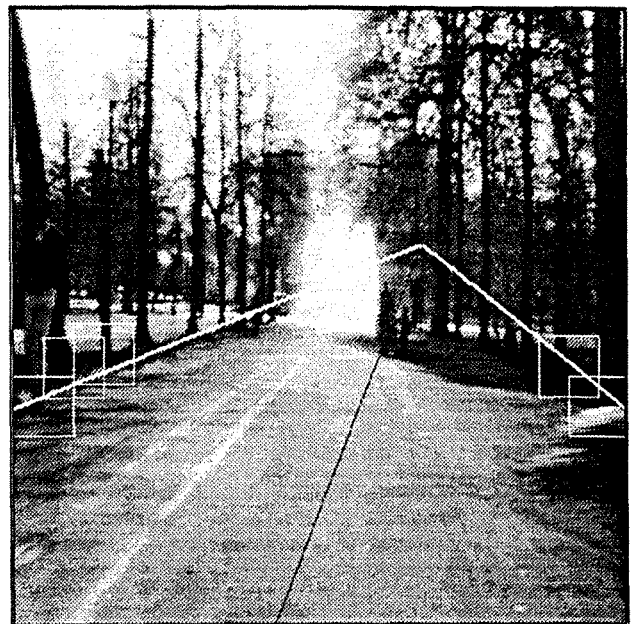
i33



i34

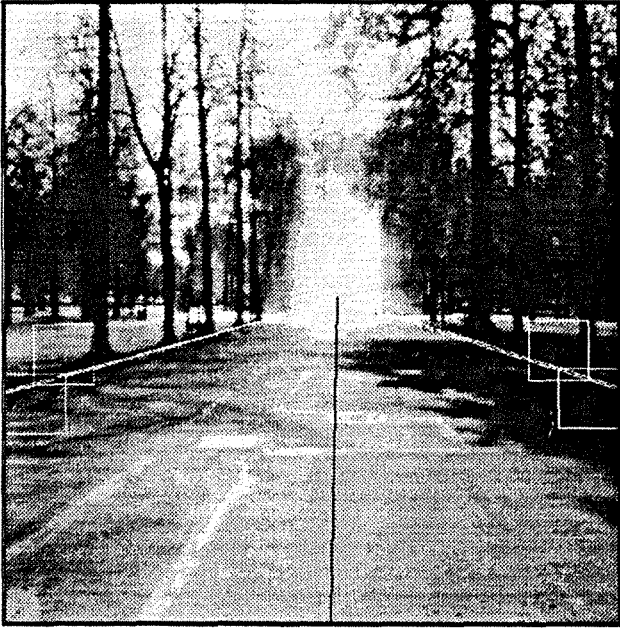


i35



i36





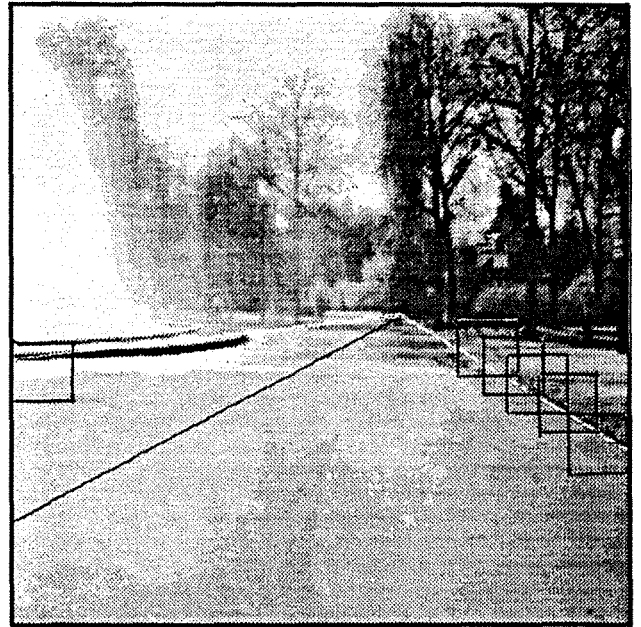
i37



i38



i39



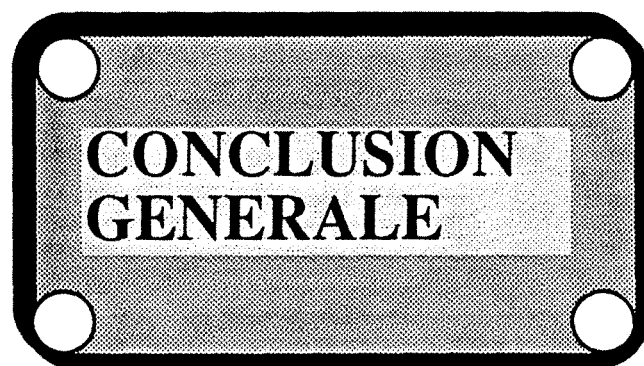
i40

En ce qui concerne la précision de détection des bords de route , le traitement en phase initiale est meilleur que celui du suivi continu. Ceci est essentiellement dû au fait qu'en mode de suivi continu les bords de route sont détectés à partir de plusieurs parties dont l'orientation varie d'une fenêtre à l'autre. On remarque ce genre d'erreur sur les images i12 (bord gauche), i16 (bord droit et gauche), i27 (bord droit) et i34 (bord droit).

Les algorithmes ont été exécutés sur une station SUN 3/60 avec l'aide du logiciel VISILOG. Le temps de traitement pendant la phase initiale varie entre 12 et 32 minutes en fonction du nombre des images binaires extraites après segmentation de l'image orientation. Plus de 60% de ce temps est consacré à la transformée de Hough.

En mode de suivi continu, le temps de calcul dépend totalement du nombre de fenêtres. Ainsi, le temps moyen de traitement d'une fenêtre est de 10 s. D'autre part, le nombre de fenêtres dépend de la visibilité des bords de route ainsi que leur orientations sur le plan de l'image. Ainsi, en limitant le placement des fenêtres à 60% de l'image, le nombre de fenêtres est d'autant plus élevé que la largeur de la route est faible. On remarque ceci en comparant les images i18 - i24 avec les images i37 et i38. La moyenne dans notre séquence d'images est de 14 fenêtres, et par conséquent le temps moyen de traitement d'une image en ce mode est de 5,33 minutes car il faut considérer le temps requis par la transformée de Hough qui est en moyenne 3 minutes par image ( donc  $10s \times 14 + 180s = 320 s$  ). On remarque donc l'utilité de ce mode en ce qui concerne le temps de calcul puisqu'il permet de réduire de 83,33% le temps d'exécution requis par la phase initiale.

On note que les contraintes de temps réel n'ont pas été prise en compte dans cette étude et que les temps indiqués n'ont qu'une valeur indicative ne concernant que l'aspect simulation. Il faut noter qu'il existe sur le marché du matériel de vision qui permet d'exécuter les opérations de traitement d'images et même la transformée de Hough en temps réel. Ainsi des architectures spécialisées permettent d'atteindre des performances acceptables tels les systèmes de vision à très haute vitesse d'acquisition associés à des transputers réalisant la parallélisation des tâches.



Les travaux que nous avons présenté dans ce mémoire traitent de l'application de la vision au guidage d'un robot mobile dans des milieux extérieurs naturels. Les résultats obtenus mettent en évidence la faisabilité du système de vision à structure modulaire dans le guidage du robot mobile. Pour parvenir à ces résultats, nous avons mis l'accent sur les points suivants :

- \* établissement d'une structure logicielle du système de vision.
- \*\* exploitation de la scène afin de déterminer les bords de route.
- \*\*\* construction et utilisation de la carte.
- \*\*\*\* développement d'une base de connaissance.

Nous avons , sans être exhaustif, apporté des contributions dans ces domaines dans les chapitres I à V.

\* Nous avons développé notre architecture logicielle d'un système de vision pour la navigation autonome d'un robot mobile dans un milieu naturel. La modularité de ce système permet de distinguer les rôles des différents modules et de contrôler le flot des commandes entre eux. Ce système nous a permis de détecter les bords de route dans la majorité des scènes naturelles sous différentes conditions, et de positionner le robot d'une façon assez précise par rapport au chemin. Ce positionnement se fait par la détermination de l'orientation du robot et le décalage de sa position par rapport à l'axe de la route.

Notons que le module d'interprétation géométrique n'est pas encore suffisamment exploité par le système de vision. Son développement contribuera à la sélection des bords de route au niveau de la base de connaissance et surtout au processus de recalage du

robot. Ce processus est en cours d'étude et se base sur la position dans la scène des objets naturels ou artificiels qui peuvent en collaboration avec les données de la carte, établir une correction des coordonnées du robot sur le repère de l'environnement.

\*\* L'algorithme de détection des éléments dominants de la scène a donné des résultats très satisfaisants pour différentes conditions extérieures comme la présence des ombres , des humains, le changement climatique et la présence des objets de formes quelconques. L'extension à apporter à ce module pour renforcer sa robustesse est l'utilisation de la segmentation en niveau de gris pour séparer la route du reste de la scène . L'étude de la segmentation doit se concentrer sur l'établissement d'un seuil adaptatif aux variations des caractéristiques de la scène et surtout dans la mesure du possible adapter le traitement qui suit la segmentation à ce seuil.

\*\*\* La cartographie nous a été un outil précieux pour surmonter les problèmes éventuels liés à la qualité des scènes . La prédiction de scène à partir de la carte a presque éliminé toute erreur dans la détection de bord de route. Cependant , le point faible de cette méthode réside dans le fait que sa précision est fortement liée à la précision du système de localisation du robot. D'autre part, nous pensons que la méthode de représentation de la carte proposée et de reconstruction de scène liée à un système de localisation assez précis, peut être généralisée et utilisée d'une manière efficace.

Si le recalage doit se faire, en utilisant des objets naturels ou artificiels qui doivent être cartographiés , nous pensons qu'il est nécessaire d'associer aux informations fournies par cette carte la modélisation de ces objets. Cette modélisation doit inclure en principe la forme et éventuellement le message que les objets portent s'ils sont artificiels. En outre, la génération de trajectoire du point de départ à la destination était malheureusement limitée à l'évitement des obstacles statiques cartographiés. Ainsi, le

résultat est aussi lié à la précision du système de localisation du robot. Une amélioration à ce niveau, concerne en grande partie l'évitement des obstacles dynamiques détectés.

\*\*\*\* La base de connaissance représente le centre d'intelligence du système de vision puisqu'elle reflète ce que le système est capable de percevoir. A ce niveau les modèles mathématiques des objets sont établis. Jusqu'ici le rôle de ce module est d'établir le modèle de la scène qui inclut la position des bords de route et l'orientation de son axe. L'extension majeure à apporter ce modèle est la capacité de combiner des descriptions symboliques indépendantes qui existent au niveau exploitation de scène. Par exemple utiliser les bords de route détectés par la dominance de l'orientation du gradient, pour déterminer le seuil de segmentation, afin d'étiqueter la scène pour certaines tâches comme la planification du chemin.

# ANNEXE 1

## Evaluations des performances des opérateurs de détection de contours.

Il existe dans la littérature plusieurs méthodes d'évaluation de performance d'un opérateur de détection de contours. Ceci rend l'évaluation très complexe à cause des difficultés de déterminer les meilleurs paramètres associés à chaque méthode et la manque d'un critère de performance global et définitif.

Pratt a développé un critère de performance en se basant sur un facteur de confiance attribué au degré de compatibilité entre le modèle du bord idéal et les informations fournies par l'image. Généralement, il y a trois types d'erreurs associés à la localisation d'un bord :

- 1) manque des points appartenant au bord.
- 2) échec de localisation des points du bord.
- 3) classification des points générés par le bruit comme points du bord.

La précision de la localisation d'un bord est donc évaluée par le facteur de mérite qui est défini par :

$$R = \frac{1}{I_n} \sum_{i=1}^{I_a} \frac{1}{1 + \alpha d^2}$$

Avec  $I_n = \max(I_i, I_a)$  où  $I_i$  et  $I_a$  représentent respectivement le nombre des points appartenant au bord idéal et réel,  $\alpha$  est un facteur d'échelle, et  $d$  la distance entre un point du bord détecté et la ligne passant par le bord idéal.

Le facteur de mérite est normalisé de telle façon que  $R=1$  pour un bord parfaitement détecté. Le facteur  $\alpha$  peut être ajusté pour pénaliser les bords localisés avec un décalage par rapport à la position désirée. La normalisation par le maximum des points du bord idéal et réel assure une pénalisation pour les bords flous ou fragmentés.

Cette méthodologie a été appliquée à quelques opérateurs de détection de bord en utilisant une image test de dimension  $64 \times 64$  pixels qui sont codés sur 256 niveaux et comportant au centre un bord de contraste et de pente variables. Un bruit gaussien

indépendant d'écart type  $\sigma_n$  a été ajouté, et l'image résultante normalisée sur l'intervalle 0-255. Le rapport signal-bruit est donné par :

$$RSB = \frac{h^2}{\sigma_n^2}$$

$h$  est la hauteur du bord comme indiqué par la figure A1.

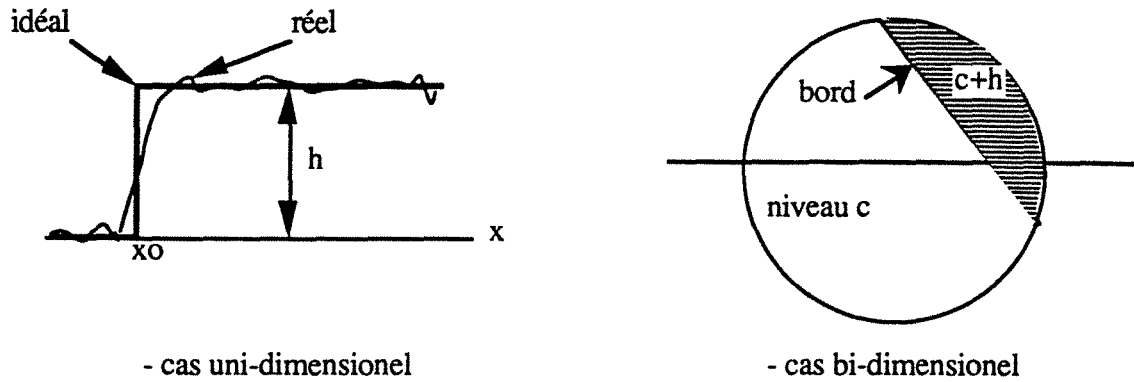


Figure A1 : hauteur d'un bord.

Figure A2 montre le facteur de mérite en fonction du rapport signal-bruit pour plusieurs détecteurs de bord avec  $\alpha = \frac{1}{9}$ . Figure A3 montre aussi le facteur de mérite en fonction de la largeur du bord.

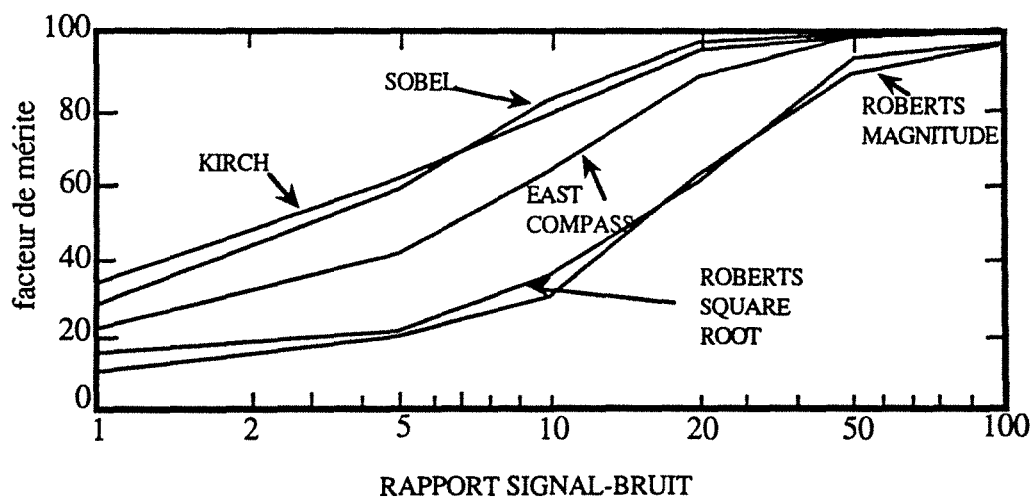


Figure A2 .



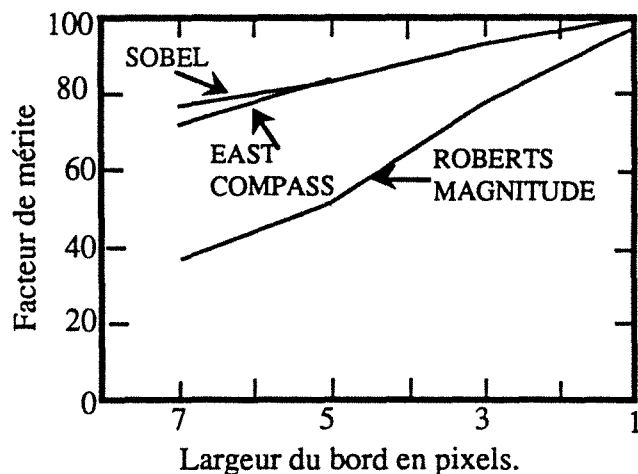


Figure A3.

Ces courbes montrent que parmi les différents opérateurs, ceux de Sobel et Prewitt donnent un facteur de mérite plus élevé que celui de Roberts. En ce qui concerne l'orientation, l'opérateur de Prewitt montre un facteur de mérite plus important pour un bord vertical, par contre il est approximativement identique à celui de Sobel pour un bord diagonal.

Ayant un modèle du bord, des calculs simples peuvent déterminer la réponse de l'orientation du bord détecté en fonction de l'orientation réelle de celui-ci comme indiqué par figure A4.

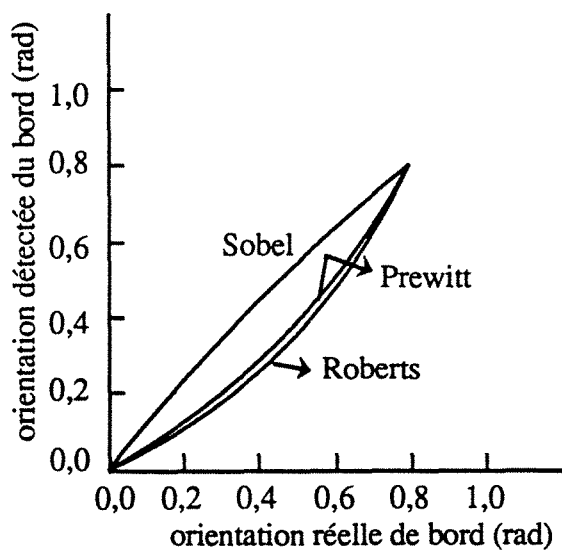
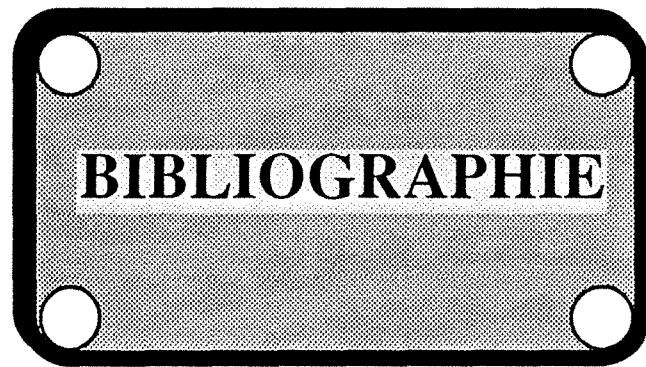


Figure A3

La figure A3 montre que l'opérateur de Sobel fournit la réponse la plus linéaire entre l'orientation réelle et l'orientation détectée du bord.

Les méthodes d'évaluation développées par Pratt et Abdou permet d'avoir les deux conclusions importantes qui sont :

- 1) Les opérateurs de taille 3x3 sont plus performant que ceux de taille 2x2.
- 2) Les opérateurs de Sobel et Prewitt de taille 3x3 sont les meilleures des opérateurs de tailles semblables.



## BIBLIOGRAPHIE

### OUVRAGES ET THESES:

**[ALAH] - MOUHAMED ALAHMED.**

Developpements de méthodes de vision par ordinateur: extraction de primitives géométriques.

THESE DOCTORAT D'ETAT. STRASBOURG. 1986

**[DHOM] - MICHEL DHOME.**

Contours et reconnaissance des formes pour l'analyse des scènes en robotiques

THESE DE DOCTEUR INGENIEUR. CLERMONT FERRAND. 1984.

**[DUDA] - RICHARD DUDA, PETER E. PRATT.**

Pattern Classification and Scene Analysis.

JOHN WILEY & SONS. 1973.

**[GARC] - PASCAL GARCIA.**

Contribution à la reconnaissance et à la localisation d'objets 2-D dans une image.

THESE DOCTORAT 3E CYCLE. NANTES. 1986.

**[HALL] - ERNEST L.HALL.**

Computer Image Processing and Recognition.

ACADEMIC PRESS, INC. 1979.

**[LAHA] - JEAN-CLAUDE LAHAYE.**

Etude et réalisation d'un système de vision temps réel par reconnaissance  
d'éléments rectilignes.

THESE DOCTORAT INP GRENOBLE. 1986.

**[MARR] - D. MARR.**

Vision.

San Fransico , CA; Freeman, 1982.

**[PRAT] - WILLIAM K. PRATT.**

Digital Image Processing.

JOHN WILEY & SONS. 1978.

**[ROSE] - AZRIEL ROSENFELD, AVINASH C. KAK.**

Digital Image Processing.

ACADEMIC PRESS, INC. 1982.

**[ZIOU] - DJEMEL ZIOU.**

La Détection de Contours dans des Images à Niveaux de Gris: Mise en Oeuvre  
et Sélection de Détecteurs.

THESE DOCTORAT INPL . NANCY. OCT 1991.

**PUBLICATIONS:**

**[BRO-81] - R. A. BROOKS.**

Symbolic Reasoning among 3-D Models and 2-D Images.

ARTIFICIAL INTELLIGENCE. Vol 17. 1981. pp285-348.

**[BRO-85] - R. A. BROOKS.**

Visual Map Making for a Mobile Robot.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND  
AUTOMATION. St Louis, MO. Mars 1985. pp 824 -829

**[BRZ-89] - D. BRZAKOVIC, L. HONG.**

Road Edge Detection for Mobile Robot Navigation.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND  
AUTOMATION. Scottsdale 15-19 May 1989. pp 1143-1147.

**[BUR-86] - J.BRIAN BURNS, ALLEN R. HANSON.**

Extracting Straight Lines.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. PAMI-8, N° 4, July 1986. pp 425-455.

**[CAN-83] - J.F CANNY.**

Finding Edges And Lines in Images.

MIT Technical Report N° 720, 1983.

**[CAS-89] - S. CASTAN, J. ZHAO, J. SHEN.**

Une Famille de Détecteurs de Contours Basée sur le Filtre Exponentiel Optimal.

7<sup>ème</sup> CONGRES DE RECONNAISSANCE DES FORMES ET

INTELLIGENCE ARTIFICIELLE. 29 Nov- 1<sup>er</sup> Decembre. pp 23-36.

**[CHE-89] - J.S CHEN, G. MEDIONI.**

Detection, Localisation, and Estimation of Edges

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 11, N° 2, February 1989.pp 191-198.

**[CHE-89] - FANG-HSUAN CHEN, WEN-HSING HSU, MEI-YING  
CHEN.**

Recognition of Handwritten Chinese Characters by Modified Hough  
Transform Techniques.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 11, N° 4, April 1989.pp 429-438.

**[CHE-91] - SHU-YUAN CHEN, WEN-HSIANG TSAI.**

Determination of robot locations by Common Object Shapes.

IEEE JOURNAL OF ROBOTICS AND AUTOMATION. Vol 7, N° 1,  
February 1991. pp 149-157.

**[CRI-91] - JILL. CRISMAN, JON. WEBB.**

The Warp Machine on Navlab.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE. Vol 13, N° 5, May 1991.pp 451-465.

**[CRO-85] - JAMES L. CROWLEY**

Navigation for an Intelligent Mobile Robot.

IEEE JOURNAL OF ROBOTICS AND AUTOMATION, vol. RA-1, N° 1,  
March 1985.pp 31-41.

**[CRO-89] - JAMES L. CROWLEY**

World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Scottsdale 15-19 May 1989. pp 674-688.

**[DAV-86] - LARRY S. DAVIS, TODD KUSHNER, JACQUELINE LEMOIGNE, ALLEN WAXMAN.**

Road Boundary Detection for Autonomous Vehicle Navigation.

OPTICAL ENGINEERING. Vol. 25 N°3 .March 1986. pp 409-414.

**[DEM-90] - DANIEL DEMENTHON, LARRY S. DAVIS.**

Reconstruction of a Road by Local Image Matches and Global 3-D Optimisation.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION . May 13-18, 1990 Cincinnati.

**[DER-87] - RACHID DERICHE.**

Optimal Edge Detection using Recursive Filtering.

Proc. First International Conference on Computer Vision, London, June 8-12 1987.

**[DER-90] - RACHID DERICHE.**

Fast Algorithms for Low-Level Vision.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 12, N° 1, January 1990. pp 78-87.



**[DHO-89] - MICHEL DHOME, MARC RICHTIN, JEAN-THIERRY  
LAPRESTE, GERARD RIVES.**

Determination of the Attitude of 3-D Objects from a Single Perspective View.  
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 11, N° 12, December 1989. pp 1265-1279.

**[DIC-87] - S. DICKINSON, L S. DAVIS.**

An Expert Vision System for Autonomous Land Vehicle Road Following.  
TECHNICAL REPPORT. Center for Automation Research. University of  
Maryland College Park, MD 20742.

**[DIC-85] - E.D DICKMANS, A. ZAPP.**

Guiding Land vehicules along Roadways by Computer Vision.  
CONGRES AUTOMATIQUE AFCET. 23-25 octobre 1985. pp 233-245.

**[ELA-90] - M. BOUDHIR ELARBI, MICHEL DUFAUT, RENE HUSSON.**

Guidage de Robot Mobile par Détection de Bord de Route. Etude de la Phase  
Initiale de Navigation.  
REVUE D'AUTOMATIQUE ET DE PRODUCTIQUE APPLIQUEES. Vol 3.  
N°2/ 1990. pp 67-81.

**[ELA-91] - M. BOUDHIR ELARBI, MICHEL DUFAUT, RENE HUSSON.**

Map Database Construction and Scene Prediction for Visual Navigation.  
THE EUROPEAN ROBOTICS AND INTELLIGENT SYSTEMS  
CONFERENCE. June 23-38, 1991. pp 357-365.

**[FAN-88] - ZHIGANG FAN, FERNAND S. COHEN.**

Textured Image Segmentation as a Multiple Hypothesis Test.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 35, N° 6, June 1988.pp 691-701.

**[FED-91] - JOHN. T FEDDEMA, GEORGE LEE, OWEN ROBERT MITCHELL**

Weighted Selection of Image Features for Resolved Rate Visual Feedback  
Control.

IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, vol. 7, N° 1,  
February 1991.pp 31-47.

**[HAR-87] - SCOTT Y. HARMON.**

The Ground Surveillance Robot (GSR): An Autonomous Vehicule Designed to  
Transit Unknown Terrain.

IEEE JOURNAL OF ROBOTICS AND AUTOMATION. Vol RA-3, N° 3, June  
1987. pp 266-279.

**[HEB-89] - MARTIAL HEBERT.**

Building and Navigating Maps of Road Scenes using Active Sensor.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND  
AUTOMATION. Scottdale 15-19 May 1989. pp 1136-1142.

**[HIR-90] - S. HIROSE, K. YOSHIDA, Y. TORATANI.**

The Study of a Map Realization System : Consideration of Real-Time Map  
Generation.

ADVANCED ROBOTICS. Vol 4, N° 3. pp 223-242.

**[HSI-92] - Y.C. HSIEH, D.M. McKEOWN, F.P PERLANT.**

Performance Evaluation of Scene Registration and Stereo Matching for  
Cartographic Feature Extraction.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 14, N° 2, February 1992.pp 214-238.

**[INI-84] - R.M INIGO, E. S. McVEY, B.J. BERGER, M.J. MITZ**

Machine Vision applied to vehicle guidance.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. PAMI-6, N° 6, November 1984.pp 820-826.

**[ILL-87] - J. ILLINGWORTH, J. KITTLER.**

The adaptative Hough Transform.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. PAMI-9, N° 5, September 1987.pp 690-698.

**[KAH-90] - P. KAHN, L. KITCHEN, E.M RISEMAN.**

A Fast Line Finder for Vision-Guided Robot Navigation.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 12, N° 11, November 1990.pp 1098-1102.

**[KAN-90] - KENICHI KANATANI, KAZUNARI WATANABE.**

Reconstruction of 3-D Road Geometry from Images for Autonomous Land  
Vehicles.

IEEE TRANSACTION ON ROBOTICS AND AUTOMATION. Vol 6, N° 1,  
February 1990. pp 127-133.

**[KEO-85] - DAVID M. McKEOWN, JR., WILSON A. HARVEY, JR., JOHN McDERMOTT.**

Rule-Based Interpretation of Aerial Imagery.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. PAMI-7, N° 5, September 1985. pp 570-585.

**[KIR-91] - N. KIRYATI, A.M BRUCKSTEIN.**

On Navigating Between Friends and Foes.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 13, N° 6, June 1991. pp 603-605.

**[KIT-89] - L.J. KITCHEN, J. MALIN**

The Effect of Spatial Discretization on the Magnitude and Direction Response of Simple Differential Edge Operators on a Step Edge.

COMPUTER VISION , GRAPHICS AND IMAGE PROCESSING, vol 47, 1989. pp 243-258.

**[KLU-89] - K. KLUGE, C. THORPE.**

Explicit Models for Robot Road Following.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION. Scottdale 15-19 May 1989. pp 1148-1159.

**[KOC-85] - E. KOCH, G. HILLEL, A. MEYSTEEL, C. ISIK.**

Simulation of Path Planning for a System with Vision and Map Updating.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION. St Louis, MO. Mars 1985. pp 146-160.

**[KRO-89] - ERIC KROTKOV.**

Mobile Robot Localization Using a Single Image.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND  
AUTOMATION. Scottsdale 15-19 May 1989. pp 978-983.

**[KUA-88] - DARWIN KUAN, GARY PHIPPS, A. CHUAN HSUEH.**

Autonomous Robotic Vehicule Road following.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 10, N° 5, September 1988. pp 648-658.

**[LAH-86] - J.C LAHAYE, A. CHEHIKIAN.**

Localisation d'objets en temp réel par transformée de Hough.

DEUXIEME COLLOQUE IMAGE. NICE, Avril 1986. pp 166-172.

**[MAI-86] - HENRI MAITRE.**

Contribution to the Prediction of performances of the Hough Transform.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. PAMI-8, N° 5, September 1986. pp 669-674.

**[MAT-90] - TSUTOMU MATSUMOTO, SHIN'ICHI YUTA.**

Autonomous Navigation System for Mobile Robots using a Route Map.

ADVANCED ROBOTICS Vol. 4, N° 3. 1990. pp 243-261.

**[MCK-85] - D. McKEOWN, W. HARVEY, J. McDERMOTT.**

Rule-Based Interpretation of Aerial Imagery.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. PAMI-7, N° 5, September 1985. pp 570-585.

**[MOH-88] - ROGER MOHR.**

Sur l'Appariement Modèle-Perception.

2E ATELIER SCIENTIFIQUE: Traitement d'Images. Aussois, Savoie 19-22  
Avril 1988. pp 11-17.

**[MOI-88] - JACKELINE LE MOIGNE.**

Domain-Dependent Reasoning for Visual Navigation of Roadways.

IEEE JOURNAL OF ROBOTICS AND AUTOMATION. Vol 4, N° 4, August  
1988. pp 419-427.

**[MOR-83] - H.P. MORAVEC.**

The Stanford Cart and the CMU Rover.

IEEE PROCEEDING. Vol 71, N° 7. July 1983. pp 872-884.

**[NAG-79] - M.NAGAO, T. MATSUYAMA.**

Structural Analysis of Complex Aerial Photographs.

6th IJCAI PROCEEDING. TOKYO, JAPAN. AUG 1979. pp 610-616.

**[NOR-89] - F.R. NOREILS, R.G. CHATILA.**

Control of Mobile Robot Actions.

IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND  
AUTOMATION, Scottsdale 15-19 May 1989. pp 701-707.

**[SHA-88] - UMA KANT SHARMA, LARRY S. DAVIS.**

Road Boundary Detection in Range Imagery for an Autonomous Robot.

IEEE JOURNAL OF ROBOTICS AND AUTOMATION. Vol 4, N° 5, October  
1988. pp 515-523.

**[TAN-92] - H. LEONG TAN, SAUL. B, E. DELP.**

Acost Minimisation Approach to Edge Detection using Simulated Annealing.  
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 14, N° 1, January 1992. pp 3-19.

**[THO-77] - A.M. THOMPSON.**

The navigation system of the JPL robot.  
Fifth Proceeding IJCAI. 1977. pp 749-757.

**[THO-88] - CHARLES THORPE, MARTIAL H. HEBERT, TAKEO KANADE,  
STEVEN A. SHAFER.**

Vision and Navigation for the Carnegie-Mellon Navlab.  
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE  
INTELLIGENCE, vol. 10, N° 3, May 1988. pp 362-372.

**[TSA-86] - WEN-HSIANG TSAI, HUEI-LIN CHOU.**

A new approach to robot location by house corners.  
Pattern Recognition, vol 19, N°6, 1986. pp 439-451.

**[TSU-79] - S. TSUGAWA, T. YATABE, T. HIROSE, S. MATSUMUTO.**

An automobile with artificial intelligence.  
SIXTH PROCEEDING IJCAI. 1979. pp 893-895.

**[TUR-87] - MATTHEW A. TURK, DAVID G. MORGENTHALER, KEITH D. GREMBAN, MARTIN MARRA.**

Video Road Following for the Autonomous Land Vehicle.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 10, N° 3, May 1987. pp 273-280.

**[TUR-88] - MATTHEW A. TURK, DAVID G. MORGENTHALER, KEITH D. GREMBAN, MARTIN MARRA.**

VITS- A Vision System for Autonomous Land Vehicle Navigation.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 10, N° 3, May 1988. pp 342-359.

**[VEY-86] - E.S. MC VEY, K. C. DRAKE, R. M. INIGO.**

Range Measurements by Mobile Robot Using a Navigation Line.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. PAMI-8, N° 1, January 1986. pp 105-109.

**[VIN-86] - A.R DE SAINT VINCENT.**

A 3-D perception system for the mobile robot Hilare.

IEEE PROCEEDING. INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION. San Francisco, CA. April 1986. pp 1105-1111.

**[WAN-91] - LING. L. WANG, W. HSIANG TSAI.**

Camera Calibration by Vanishing Lines for 3-D Computer Vision.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 13, N° 4, April 1991. pp 371-376.



**[WAX-87] - ALLEN M. WAXMAN, JACQUELINE J. LEMOIGNE, LARRY S.  
DAVIS, BABU SRINIVASAN, TODD R. KUSHNER, ELI  
LIANG, THARAKESH SIDDALINGAIAH.**

A Visual Navigation System for Autonomous Land Vehicules.

IEEE JOURNAL OF ROBOTICS AND AUTOMATION. Vol RA-3, N° 2,

April 1987. pp 124-141.

**AUTORISATION DE SOUTENANCE DE THESE  
DU DOCTORAT DE L'INSTITUT NATIONAL POLYTECHNIQUE DE  
LORRAINE**

o0o

VU LES RAPPORTS ETABLIS PAR :

**Monsieur WENDEL, Professeur, E.N.S.P. Strasbourg.**

**Monsieur POSTAIRE, Professeur, Centre d'Automatique de Villeneuve d'Ascq.**

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

**Monsieur ELARBI-BOUDIHIR Mohamed**

à soutenir devant l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, une thèse intitulée :

**"Contribution au guidage des robots mobiles par vision"**

en vue de l'obtention du titre de :

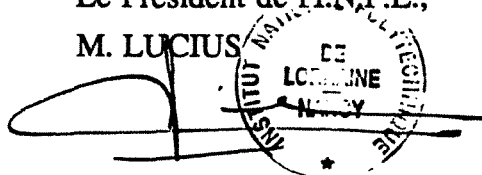
**DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE**

Spécialité : **"GENIE ELECTRIQUE"**

Fait à Vandoeuvre le, 1<sup>o</sup> Octobre 1992

Le Président de l'I.N.P.L.,

M. LUCIUS



INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

## RESUME

Cette thèse concerne l'étude et le développement d'une architecture logicielle d'un système de vision pour le guidage d'un robot mobile dans un milieu extérieur. Ce système est composé de différents modules assurant chacun une tâche spécifique, et l'ensemble des tâches est coordonné par un module central jouant le rôle de superviseur. Dans cette étude, l'accent est principalement mis sur le module d'analyse de scène qui est basé sur un algorithme de détection des éléments linéaires dans l'image numérique. Ce module est fonctionnel sous deux modes différents: la phase initiale qui consiste en un traitement de l'image entière pour initialiser la navigation, et la phase de suivi continu qui assure le traitement sur des fenêtres extraites des images prises périodiquement. Pour diminuer le risque d'erreur lié à ce type de technique, une carte détaillée de l'environnement a été établie dans le but de réaliser une prédiction de scène à partir de données visuelles. La scène prédite est utilisée pour valider les éléments détectés au niveau du module de la base de connaissance. Ce dernier établit un raisonnement à partir des informations acquises et prédites pour construire un modèle de la scène. Le module de pilotage génère à partir de ce modèle une trajectoire constituée d'une suite de segments de droite qui sont ensuite interprétés séquentiellement par le dispositif de navigation.

**Mot Clés:** Vision ordinateur, robot mobile, détection de bords, navigation visuelle, transformée de Hough, détection des éléments linéaires.

## ABSTRACT

A new vision system software architecture has been developed to support the visual navigation of an autonomous mobile robot in an outdoor environment. The system consists of different modules ensuring each a specific task. The different tasks are coordinated by the central module called the supervisor. Most of the processing time is devoted to the image processing module which is based on an algorithm to extract the dominant linear features. This module operates into two modes: the initial phase which forms the type of processing applied to the first acquired image in order to initiate the navigation, and the continuous following mode which ensures the processing of the subsequent images taken periodically. In order to rely less on the visual data, a detailed map of the environment has been established and an algorithm is used to make a scene prediction. The predicted scene is used to validate the elements detected at the level of the knowledge base. The latter reasons about the acquired and predicted data to construct a scene model. This model is consequently used by the geometrical reasoning module to end up with a 3-D interpretation useful to the pilot module. Moreover, a simple trajectory generation algorithm is used to decompose the path into segments which are transmitted to the pilot module one after the other.

**Keywords:** computer vision, mobile robot, edge detection, visual navigation, Hough transform, linear feature detection.