

UNIVERSITE DE SAAD DAHLED DE BLIDA

Faculté des Sciences de l'Ingénieur

Département d'électronique

MEMOIRE DE MAGISTER

Spécialité : Image et Parole

**Construction incrémentale de l'environnement du robot mobile
ATRV2**

Par

BOUTICHE Yamina

Devant le jury composé de :

P. GUESSOUM	Professeur, U. de Blida	Président
M. BELHOCINE	Maître de recherche, CDTA	Examineur
N. Benblidia	Chargé de cours, U. de Blida	Examineur
K. Achour	Directeur de recherche, CDTA	Rapporteur
A. Djekoune	Chargé de recherche, CDTA	Co-rapporteur

Blida, Octobre 2007

RESUME

La principale ligne actuelle de recherche pour la robotique mobile en environnement naturel est de réussir à faire déplacer un robot mobile de manière autonome dans son environnement, pour lui faire accomplir un certain nombre de tâches. Ces tâches sont par exemple le déplacement du robot vers une cible fixe ou mobile, l'évitement d'obstacles, le suivi d'une route, l'exploration et l'intervention dans un milieu hostile. Pour cela, les problèmes à résoudre sont parfois assez complexes. L'un d'eux la détermination de la carte locale de l'environnement du robot au moyen de la perception.

C'est dans ce contexte que se situe ce projet. Son objectif est de construire de manière autonome la carte tridimensionnelle de l'environnement d'un robot mobile équipé d'un banc stéréoscopique. Le robot mobile se déplace dans son environnement en évitant les obstacles, sachant qu'il ne dispose d'aucune information a priori du lieu où il se trouve. Il doit donc construire une carte du lieu à partir des seules images fournies par le banc stéréoscopique, dans ce cas on parle de la stéréovision binoculaire.

Le principe de base de la stéréovision est d'apparier deux images de la même scène prises sous des angles de vues différentes. Afin d'aboutir à la reconstruction d'une carte de l'environnement plusieurs étapes sont nécessaires. Il s'agit tout d'abord, d'extraire les informations caractérisant l'environnement du robot mobile appelé primitives. Ces dernières sont extraites par segmentation. Elles seront ensuite appariées, puis reconstruites en 3D.

Afin que le robot mobile puisse exploiter les résultats de la reconstruction, ces dernières seront projetés verticalement sur une carte à deux dimensions (le sol), où l'on voit apparaître des aires élémentaires vides, occupées et inexplorées.

Mots clefs: robotique mobile, vision par ordinateur, segment de droite, mise en correspondance, construction 3D, segmentation d'images.

ملخص

أهم مجالات البحث حاليا في ميدان الآليات المتحركة في محيط طبيعي هو الوصول إلي إعطاء الآلي المتحرك الاستقلالية التامة ليتحرك في محيط ثلاثي الأبعاد وذلك ليقوم بالمهام الموكلة له. هذه المهام قد تكون على سبيل المثال تحرك الآلي نحو هدف معين ثابت أو متحرك؛ تجنب الاصطدام بالعوائق؛ إتباع طريق معين؛ و استغلال الآلي لتدخل في المناطق المحظورة على الإنسان (الفضاء؛ المناطق النووية). للوصول إلى هذا الهدف هناك مشاكل معقدة مطروحة للحل. من أهمها إنشاء بطاقة لما يحيط بالآلي المتحرك باستعمال الرؤية الاصطناعية . في هذا المجال يندرج مشروعنا هذا؛ هدفه إنشاء بطريقة مستقلة بطاقة ثلاثية الأبعاد لما يحيط بالآلي المتحرك؛ المزود بجهاز البصر المجسادي (الستريوسكوبي). المحيط غير معروف مسبقا و غير منظم يتحرك الآلي في هذا الأخير مع تفادي الاصطدام لذلك يجب عليه إنشاء بطاقة للمكان الموجود في مجال رؤيته انطلاقا فقط من المعلومات المعطاة من جهاز البصر المجسادي (الستريوسكوبي). في هذه الحالة نتكلم عن البصر المجسادي الثنائي.

المبدأ الأساسي في البصر المجسادي هو إزواج صورتين لنفس المشهد؛ مؤخودتين من زوايا نظر مختلفتين. للحصول علي هذه البطاقة ثلاثية الأبعاد؛ عدة مراحل مهمة يجب إنجازها؛ تتمثل أولا في استخلاص المعلومات التي تمثل الشكل العام للمحيط؛ تسمى "أوليات" تستخلص هذه الأخيرة عن طريق عملية تجزئة الصورة. ثم نقوم بإزواج أوليات المتجانسة في الصورتين و أخيرا نقوم بإنشاء الصورة ثلاثية الأبعاد للمشهد الذي يراه الآلي.

لكي يستطيع الآلي استعمال الصورة ثلاثية الأبعاد المحيط المتحصل عليها نقوم بإسقاطها عموديا علي بطاقة ثنائية الأبعاد (الأرض) أين تظهر المناطق الفارغة و المستعملة.

كلمات المفتاح

الآليات المتحركة؛ الرؤية الاصطناعية؛ إزواج؛ قطعة مستقيمة؛ إنشاء ثلاثي الأبعاد؛ تجزئة

الصورة

ABSTRACT

The main present line of research for the mobile robotics in natural environment is to succeed in making displace a mobile robot in an autonomous manner in its environment, to make accomplish it a certain number of tasks. These tasks are for example the displacement of the robot toward a stationary or mobile target, the obstacles avoidance, the follow-up of a road, the exploration and intervention in a hostile environment. For it, the problems to solve are sometimes complex enough. One of them the determination of the environment local card of the robot by means of the perception.

It is in this context that is located this project. Its objective is to construct autonomous manner the three-dimensional card of the environment of a mobile robot equipped with a stereoscopic bench. The mobile robot moves in its environment while avoiding the obstacles, knowing that it having no information a priori of the place where it is. It must construct a card of the place therefore from the only image provided by the stereoscopic bench, indeed, we speak in this case of the binocular stereovision.

The principle of basis of the stereovision is to match two images of the same scene taken under different angles views. To end to succeed to the reconstruction of a card of the environment several stages are necessary. It is first of all about, of excerpts the information characterizing the mobile robot environment named primitive. These last are extracted by segmentation. They will be matched, then rebuilt in 3D.

So that the mobile robot can exploit the results of the reconstruction, this last will be projected vertically on a card to two dimensions (ground), where one sees to appear of the empty, occupied and uncharted elementary areas.

Words keys: mobile robotics, vision by computer, segment of right, Matching, construction 3D, images segmentation.

Remerciements

Je tiens à remercier tout d'abord très vivement les membres de mon jury pour l'honneur qu'ils me font, *M^r. Guessoum*, qui le préside. *M^{me} Benblidia* et *M^r. Belhocine*, qui ont bien voulu être examinateurs.

Je remercie très chaleureusement *Dr Achour Karim* d'avoir accepté d'encadrer ce travail, et de m'avoir accueilli dans leur laboratoire.

M. Djekoune A. Oualid a joué un rôle tout particulier dans le développement de ce travail. Il a collaboré depuis son début, et a su l'encourager en permanence. Son travail théorique, son enthousiasme, et sa confiance ont été déterminants dans la conduite de cette mémoire. Je tiens à le remercier très chaleureusement, et à lui exprimer toute ma gratitude.

Je remercie profondément, tous les membres de l'équipe de recherche « Equipe Vision Artificielle et ANalyse d'IMages » du centre de recherche CDTA pour leur bonne humeur, leur aide, et disponibilité.

Je remercie très chaleureusement, tous mes enseignants de l'institut d'électronique, université de Blida.

Je remercie également, mes ami(e)s pour leur encouragement, et soutien moral.

Enfin, j'adresse un immense merci à mes Parents et à mes frères et sœurs qui ont sans aucun doute contribué d'une façon ou d'une autre à ma passion durant toutes mes d'études.

Merci

TABLE DES MATIERES

RESUME

REMERCIEMENTS

TABLE DES MATIERES

LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

INTRODUCTION GENERALE	14
1. PERCEPTION ET MODELISATION DE L'ENVIRONNEMENT	
1.1 Introduction	17
1.2 Acquisition des informations tridimensionnelles	18
1.2.1 Les capteurs actifs	18
1.2.2 Les capteurs passifs	19
1.3 Perception de l'environnement par la vision	20
1.3.1 La vision chez les êtres vivants	20
1.3.2 La vision artificielle	20
1.3.3 La stéréovision	21
1.3.4 Quelques applications de la vision artificielle	22
1.4 Modélisation géométrique d'une caméra	23
1.4.1 Modéliser une caméra	23
1.4.2 Paramètres extrinsèques	25
1.4.3 Paramètres intrinsèques	29
1.4.3.1 La projection perspective	29
1.4.3.2 Transformation caméra/ image	30
1.4.4 Le modèle complet sans distorsion	33
1.4.5 Relation géométrique entre images	36
1.4.6 Rectification d'images	37
1.5 Modélisation de l'environnement	38
1.5.1 Carte métrique	39
1.5.2 Carte topologique	40
1.6 Conclusion	41

2. SEGMENTATION D'IMAGE ET CHOIX DE PRIMITIVES

2.1 Choix de primitives	42
2.1.1 Introduction	42
2.1.2 Points d'intérêts	43
2.1.3 Les régions	44
2.1.4 Points contours	45
2.1.5 Les segments de droite	45
2.1.6 Discussion	47
2.2 Segmentation d'images	48
2.2.1 Introduction	48
2.2.2 Extraction de contours	48
2.2.2.1 Définition	48
2.2.2.2 Les approches classiques	49
2.2.2.3 Les approches analytiques	51
2.2.3 Le chaînage	52
2.2.4 Segmentation de contours	53
2.2.4.1 La transforme de Hough	53
2.2.4.2 L'approximation polygonale	55
a. Pourquoi ?	55
b. Approximation polygonale par découpage récursif	55
c. Approximation polygonale par fusion	57
2.2.4.3 Discussion	60
2.3 Conclusion	61
3. MISE EN CORRESPONDANCE STÉRÉOSCOPIQUE	
3.1 Introduction	62
3.2 Définition	62
3.3 Problèmes liés à la mise en correspondance	63
3.3.1 Le bruit de l'image	63
3.3.2 Variation photométrique	63
3.3.3 Distorsion perspective	64
3.3.4 Répétition ou absence de textures	64
3.3.5 Les occultations	64
3.4 Les contraintes de la mise en correspondance	65

3.4.1	Contrainte Lambertienne	65
3.4.2	Contrainte épipolaire	65
3.4.3	Contrainte d'orientation	66
3.4.4	Contrainte d'ordre	66
3.4.5	Contrainte d'unicité	67
3.4.6	Contrainte de continuité	68
3.4.7	Contrainte de voisinage	68
3.5	Principales approches d'appariement	69
3.5.1	Méthodes surfaciques	69
3.5.2	Méthodes à base de primitives	71
3.5.2.1	Mise en correspondance hiérarchique	72
3.5.2.2	Mise en correspondance par programmation dynamique	73
3.5.2.3	Mise en correspondance par relaxation	73
3.5.2.4	Mise en correspondance par isomorphisme de graphe	74
3.5.2.5	Mise en correspondance par invariants	74
3.6	L'appariement des segments de droite (approche adoptée)	75
3.6.1	Pourquoi apparier des segments	75
3.6.2	Principe de l'algorithme	75
3.6.3	La fonction d'appariement	76
3.6.4	Difficultés	76
3.6.5	Calcul de voisinage	77
3.6.6	Algorithme d'appariement des segments de droite	78
3.7	Conclusion	82
4. LA RECONSTRUCTION TRIDIMENSIONNELLE		
4.1	Introduction	83
4.2	Approches de reconstruction	84
4.2.1	Approches mono-image	84
4.2.2	Approches multi-images	85
4.2.3	Discussion	86
4.3	Méthodes de reconstruction 3D stéréoscopique	86
4.3.1	Reconstruction par la distance minimale entre deux rayons optiques	87
4.3.2	Reconstruction par triangulation	88
4.3.3	Reconstruction en déterminant la projection des points 3D	

sur les deux plans images	89
4.4 Discussion	92
5. RESULTATS EXPERIMENTAUX	
5.1 Introduction	93
5.2 L'interface graphique du logiciel développé	94
5.3 Choix et extraction de la primitive	95
5.3.1 Extraction de contours	96
5.3.2 Chaînage des contours	98
5.3.3 Approximation polygonale	102
5.4 Mise en correspondance	108
5.5 Reconstruction 3D de la scène	113
5.6 Conclusion	119
CONCLUSION GENERALE	120
APPENDICE A : Le robot ATRV2	
A.1 Présentation robot mobile <i>ATRV2</i>	122
A.2 Architecture Matérielle du robot Mobile <i>ATRV2</i>	124
A.3 Architecture Logicielle du robot Mobile <i>ATRV2</i>	125
A.4 Géométrie du robot mobile <i>ATRV2</i>	126
A.5 Capacité de mouvement du robot mobile <i>ATRV2</i>	127
APPENDICE B : Calibrage	
B.1 Principe de calibrage	128
B.2 Méthodes d'étalonnage	129
B.2.1 Etalonnage selon Tsai	129
B.2.2 Etalonnage selon Faugeras et Toscani	129
B.2.3 Etalonnage selon Zhang	130
B.3 Les mires de calibration	131
B.3.1 Les mires à motifs rectangulaires	131
B.3.2 Les mires à motifs circulaires	132
B.3.3 Les autres types de mires	133
REFERENCES	134

LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

Figure 1.1: Tâches effectuées par le robot dans un cycle.....	18
Figure 1.2 : Capteurs actifs.....	19
Figure 1.3 :Capteurs passifs	19
Figure 1.4 : Acquisition d'une scène par caméra. Le centre optique est situé entre la scène et le plan image	24
Figure 1.5 : Acquisition d'une scène par caméra. Le centre optique est représenté en arrière du plan image	24
Figure1.6 : Représentation géométrique des repères utilisés pour la détermination du modèle de la caméra	26
Figure 1.7 : Les trois rotations simples	27
Figure1.8 : Projection perspective cas du plan vertical	30
Figure 1.9 : Principe de calibration d'une caméra.	34
Figure 1.10 : Relation géométrique entre une paire d'images : géométrie epipolaire	36
Figure 1.11: Rectification d'images.	38
Figure 1.12: Illustration de la distinction classique entre les cartes métriques et les cartes topologiques	39
Figure 1.13: Carte topologique	40
Figure 2.1: Représentation d'une droite dans l'espace de Hough	53
Figure 2.2 : Détection de droites avec la transformée de Hough	54
Figure 2.3 : Principe de l'approximation polygonale par découpage récursif.....	56
Figure 2.4 : Calcul de la distance e_m	56
Figure 2.5 : Principe de l'approximation polygonale par fusion	58
Figure 2.6 : Les chaînes initiales d'une scène de bureau	59
Figure 2.7: Approximation avec la méthode de découpage récursive l'écart $e_m = 3$	59
Figure 2.8 : Approximation avec la méthode de fusion selon l'écart $e_m = 3$	60
Figure 3.1 : Deux pixels P_1 et P_2 correspondants	63

Figure 3.2 : Exemple du problème d'occultation dans un paire d'images	
Stéréoscopiques	64
Figure 3.3: Géométrie de la stéréovision binoculaire	65
Figure 3.4: Contrainte d'ordre	66
Figure 3.5 : Cas où la contrainte d'ordre n'est pas respectée.	67
Figure 3.6 : Cas où la contrainte d'unicité n'est pas respectée	67
Figure 3.7 : Contrainte de voisinage	69
Figure 3.8 : Recherche d'un correspondance par corrélation	70
Figure 3.9 : Algorithme de mise en correspondance hiérarchique	72
Figure 3.10: Différence de segmentation entre image gauche et droite	77
Figure 3.11: Calcul des voisinages	78
Figure 3.12: Construction de la fenêtre de recherche	79
Figure 3.13: Contrainte de compatibilité entre les appariements	
gauche-droite et droite-gauche	81
Figure 3.14: Cas de recouvrement entre deux segments de droites	82
Figure 4.1: Principe de la reconstruction par la distance minimale entre	
deux rayons optique	88
Figure 4.2: Reconstruction en déterminant la projection des points 3D	
sur les deux plans images	89
Figure 5.1: Fenêtre principale du logiciel	94
Figure 5.2: Boite de dialogue pour l'ouverture d'une image	95
Figure 5.3: Boite de dialogue pour l'ouverture d'une paire stéréo d'images ..	95
Figure 5.4: Image originale « 2des » en niveau de gris	96
Figure 5.5: Extraction de contours pour différente valeurs de α .	
Cas de l'image «2des»	97
Figure 5.6: Image originale « ball » en niveau de gris	97
Figure 5.7: Extraction de contours pour différentes valeurs de α .	
Cas de l'image « ball »	98
Figure 5.8: Interface de l'algorithme de chaînage. Résultat pour $Long = 5$	99
Figure 5.9: Résultat de l'algorithme de chaînage pour différentes valeurs	
de « Long » appliqué sur l'image contours « 2des »	100
Figure 5.10: Résultat de l'algorithme de chaînage pour différentes valeurs	
de « Long » appliqué sur l'image contours «ball»	100

Figure 5.11: Boite de dialogue pour saisie des paramètres de l'approximation polygonale	102
Figure 5.12 : L'interface de l'algorithme de l'approximation polygonale.	103
Figure 5.13 : Résultats de traitements de l'image « disque »	104
Figure 5.14 : Résultats de l'approximation polygonale pour différentes valeurs de e_m	104
Figure 5.15 : Résultats de traitement de l'image « maison »	105
Figure 5.16 : Résultats de l'approximation polygonale pour différentes valeurs de e_m	105
Figure 5.17: résultats de calcul de voisinage pour différentes valeurs de résolution de la grille	107
Figure 5.18 : Boîte de dialogue de mise en correspondance	108
Figure 5.19: l'interface de l'algorithme de mise en correspondance	109
Figure 5.20 : Résultat de l'appariement de la paire stéréo « 2dés » pour $Tfen = 5$	110
Figure 5.21 : Résultat de l'appariement de la paire stéréo « maison » pour $Tfen = 5$	110
Figure 5.22 : Résultat de l'appariement de la paire stéréo « 2dés » pour $Tfen = 20$	111
Figure 5.23 : Résultat de l'appariement de la paire stéréo « maison » pour $Tfen = 20$	111
Figure 5.24 : Résultat de l'appariement de la paire stéréo « 2dés » pour $Tfen = 40$	112
Figure 5.25 : Résultat de l'appariement de la paire stéréo « maison » pour $Tfen = 40$	112
Figure 5.26 : Boite de dialogue pour la saisie des paramètres du banc stéréoscopique	114
Figure 5.27 : L'interface de la phase de reconstruction 3D.	115
Figure 5.28 : La paire stéréoscopique «2dés» originale en niveau de gris....	115
Figure 5.29: Résultat de reconstruction de la paire stéréo « 2dés », le premier cas	116
Figure 5.30: Résultat de reconstruction de la paire stéréo « 2dés », deuxième cas	116

Figure 5.31 : La paire stéréoscopique «maison» originale en niveau de gris.....	117
Figure 5.32: Résultat de reconstruction de la paire stéréo «maison», le premier cas.....	117
Figure 5.33: Résultat de reconstruction de la paire stéréo « maison», deuxième cas	118
Figure A.1 : Le robot mobile <i>ATRV2</i>	123
Figure A.2 : vue de haut du robot mobile <i>ATRV2</i>	123
Figure A.3 : Vue de l'intérieur du robot mobile <i>ATRV2</i>	124
Figure A.4 : vue arrière du robot mobile <i>ATRV2</i>	125
Figure A.5: Le banc stéréoscopique.	125
Figure A.6 : La géométrie du robot mobile <i>ATRV2</i>	126
Figure A.7 : Quatre différents types de mouvement réalisable par le robot mobile <i>ATRV2</i>	127
Figure B.1 : La mire utilisée par R. Y. Tsai et le schéma de l'acquisition	131
Figure B.2 : La mire utilisée à l'INRIA	132
Figure B.3 : La mire à motifs circulaires	132
Figure B.4 : La mire bi-plan à motifs circulaires	133
Figure B.5 : Quelques autres formes des mires	133

Tableaux :

Tableau 5.A : Résumer des résultats obtenu de l'algorithme de l'approximation polygonale pour différentes valeurs de e_m	106
Tableau 5.A : Les résultats expérimentaux obtenu on appliquons l'algorithme d'appariement sur deux paires d'images stéréoscopique	113

INTRODUCTION GENERALE

Depuis sa naissance dans les années 60, à partir des travaux de David MARR [1], la vision artificielle a pour vocation de permettre à des systèmes automatiques ou robotiques d'acquérir et de traiter des informations visuelles en reproduisant le fonctionnement complexe de la vision humaine.

Pourquoi la vision artificielle ?

Comme nous savons, la vision est un de nos sens les plus puissants. Elle nous fournit une quantité importante d'informations qui nous permettent d'interagir intelligemment avec notre environnement. Grâce à elle, nous sommes capables en un instant d'identifier la plupart des objets situés dans notre champ visuel, de repérer leur position et enfin d'ébaucher une réflexion sur les tâches que nous devons effectuer, mais c'est aussi un de nos sens les plus compliqués. Son mécanisme reste mal connu malgré les progrès réalisés par la psychologie cognitive.

D'une manière similaire, la vision artificielle, dite aussi vision par ordinateur, et en particulier la vision en robotique, vise à donner aux robots mobiles la capacité de percevoir et comprendre l'espace qui les entoure. Cependant, malgré les efforts des chercheurs, à nos jours aucune machine ne parvient à rivaliser - et de loin - avec la vision humaine. Elle vise plutôt à arriver par des moyens informatiques à des résultats similaires.

Les progrès connus ces dernières années, dans la robotique mobile sont les résultats des développements technologiques des capteurs et des ordinateurs; ces derniers ont permis leur implantation directement dans l'architecture embarquée des robots mobiles ou manipulateurs. Cependant les portes sont ouvertes devant les chercheurs pour l'exploitation de ces technologies, et atteindre l'objectif de remplacer les hommes par des robots dans des sites hostiles à la présence humaine, comme l'espace et les sites nucléaires, etc.

L'utilisation de la vision en robotique va non seulement augmenter l'autonomie et l'efficacité des robots, mais également de les doter des capacités de reconnaissance des objets et de raisonnement, ce qui les rend capables de prendre des décisions c'est-à-dire les rendre intelligents. Certes, ces objectifs ont posé de

nouveaux problèmes en robotique tel que le traitement et l'analyse d'images, la reconnaissance d'objets, etc.

Sur un pixel de l'image peuvent se projeter une infinité de points de la scène [2]. Une seule image ne permet pas de remonter à l'information tridimensionnelle de l'environnement, par contre l'utilisation d'une deuxième image nous permet de trouver le point de la scène correspondant à ce pixel. La reconstruction de l'environnement est alors possible; on parle dans ce cas de la *stéréovision*

La progression de l'autonomie d'un robot mobile à base de la stéréovision consiste à répéter les étapes suivantes dans un cycle [3] :

- Acquisition d'images stéréoscopiques;
- L'extraction des primitives (ou indice visuel) à partir des images acquises;
- Mise en correspondance de ces primitives;
- Reconstruction de l'environnement; afin d'estimer le terrain visible dans le champ de vision.
- Planification du chemin permettant au robot mobile d'atteindre son but, tout en évitant les obstacles.
- Exécution des étapes précédentes le long du chemin planifié jusqu'à l'arrivée au but.

C'est dans ce contexte que se situe ce projet, il s'intègre dans les travaux de l'équipe « Equipe Vision Artificielle et ANalyse d'IMages » du centre de recherche CDTA « Centre de Développement des Technologies Avancées ». L'objectif principal de notre travail est la mise en œuvre d'un algorithme permettant la reconstruction de l'environnement (détermination du relief de la scène observée) à partir de deux images acquises par un banc stéréoscopique calibré. A cet effet, plusieurs étapes sont nécessaires qui seront détaillées le long de cette mémoire.

Ce travail est structuré de la manière suivante:

Le premier chapitre est consacré aux différents modes de perception de l'environnement, à savoir la perception: par capteurs passifs, et par capteurs actifs. Par la suite le chapitre cible la perception de l'environnement par la vision.

L'utilisation de la caméra nécessite leur modélisation mathématique, cependant une section de ce chapitre est consacrée à la modélisation géométrique de cette

dernière. Un bref survol sur les approches de modélisation de l'environnement est en suite apporté à la fin du chapitre.

Le deuxième chapitre comporte deux parties : la première partie présente une étape cruciale dans la boucle de la reconstruction, il s'agit du choix de primitives ou indices visuels, nous signalons l'intérêt d'un bon choix et sa relation avec la tâche à réaliser. La deuxième partie est consacrée à une étude générale de la segmentation d'image. Les algorithmes qui permettent de passer d'une image brute à une image bien structurée sous forme de liste de segments de droite sont aussi détaillés dans ce chapitre.

Dans le troisième chapitre, nous nous intéresserons à l'étape la plus délicate dans un système de vision stéréoscopique, en premier lieu nous citerons les problèmes qui rendent l'appariement un exercice difficile à résoudre, puis, les contraintes souvent utilisées pour surmonter ces problèmes. Nous donnerons aussi un survol sur les techniques de mise en correspondance rencontrées dans la littérature, nous exposons les principales approches d'appariement : celles basées sur l'appariement des indices visuels extraits de chaque image, et celles dites d'appariement surfacique. A la fin, une description approfondie de l'algorithme choisi pour l'appariement des segments de droite, primitives que nous avons utilisées.

Le propos du chapitre quatre, est de présenter la méthode utilisée pour la reconstruction des informations tridimensionnelles d'une scène à partir des couples de segments de droite homologues déterminés, ainsi que les paramètres intrinsèques et extrinsèques des caméras.

Au cours du chapitre cinq, nous présentons et analysons les résultats expérimentaux de chaque algorithme développé. Nous discutons les l'impacte des paramètres utilisés sur la qualité des résultats.

Enfin, la conclusion générale présente un résumé du travail accompli, ainsi que les perspectives liées à ce travail.

CHAPITRE 1

PERCEPTION ET MODÉLISATION DE L'ENVIRONNEMENT

Dans ce chapitre nous allons décrire brièvement le problème de la perception et de la compréhension automatique de l'environnement. Nous allons montrer les différents capteurs généralement utilisés dans la perception de l'environnement d'un robot mobile; et essentiellement ceux utilisés dans la vision artificielle.

La modélisation d'une caméra avec son modèle le plus fréquemment utilisé et la modélisation de l'environnement sont aussi décrites dans ce chapitre.

1.1 Introduction :

La réalisation d'une tâche en robotique nécessite le savoir d'acquérir et de manipuler les modèles de l'environnement à partir des informations fournies par les capteurs embarqués des robots mobiles. Parfois, la finalité même de cette tâche est la reconstruction de ces modèles. D'autres fois, ces modèles serviront d'entrée à un module de planification de trajectoire. Dans tous les cas, la fiabilité des représentations, et donc des fonctions de perception est une condition essentielle à la bonne réalisation d'une tâche en robotique.

La génération des déplacements d'un robot mobile dans un environnement inconnu ou imparfaitement connu nécessite généralement, un passage, dans un cycle, à travers plusieurs couches fonctionnelles [4]: perception, modélisation, planification et action (figure 1.1).

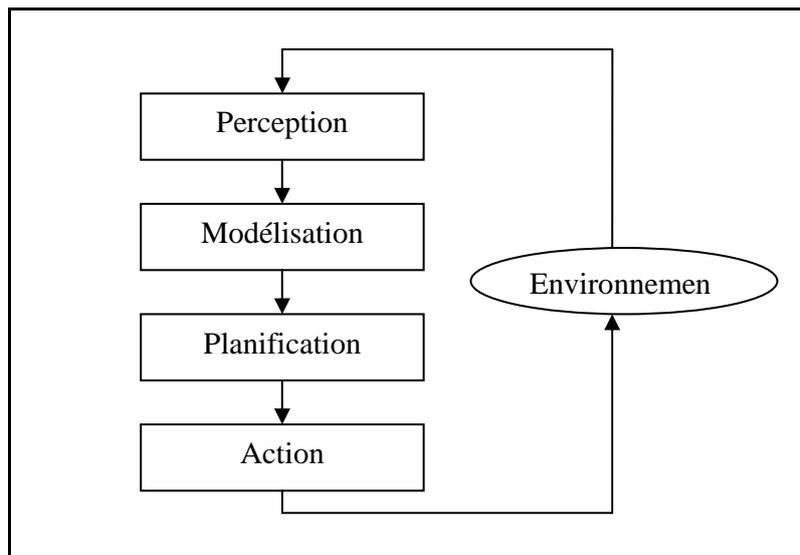


Figure 1.1: Tâches effectuées par le robot dans un cycle.

1.2 Acquisition des données de l'environnement :

En pratique, deux types de capteurs sont utilisés pour acquérir les informations d'une scène dans un environnement quelconque [5]:

- Les capteurs actifs;
- Les capteurs passifs.

Pour une application donnée, le choix du capteur adéquat dépend du type même de la tâche à réaliser (navigation, inspection, etc.), de la précision requise, et bien évidemment du type de l'environnement à explorer (scènes d'intérieur, environnement hostile, environnement naturel, etc.).

1.2.1 Les capteurs actifs :

Le principe de ce type de capteur est le calcul du temps de propagation des ondes qu'il émet. Ce temps mis par une onde pour parcourir la distance aller-retour entre la source du capteur et un point de l'objet de la scène permet de calculer la position de ce point dans un plan. Ils sont appelés capteurs actifs car ils fournissent de l'énergie pour interagir avec l'environnement. Les exemples les plus répandus de tels systèmes sont le télémètre laser et la sonde à ultrasons, qui sont utilisés souvent pour la navigation et la localisation des robots mobiles.



(a): Ultra sons



(b): LMS 291-S05



(c) : SICK LMS_200

(a): capteur ultra sons

(b), (c) : capteurs télémètres lasers

Figure 1.2 : Capteurs actifs

1.2.2 Les capteurs passifs:

A la différence des capteurs précédents, ces capteurs n'ont aucune influence sur l'environnement étudié (ne transmettent aucune énergie). Leur principe est simple. Ils reçoivent les signaux émis ou réfléchis par l'environnement.

Nous distinguons principalement *les caméras* ; qui sont très utilisées dans de nombreuses applications, et permettent de remonter à une représentation 3D de la scène perçue en utilisant le plus souvent des algorithmes de stéréovision [5]. Deux images au moins sont en principe nécessaires pour contribuer à la compréhension du monde tridimensionnel.

L'utilisation de telles méthodes de perception rentre dans le domaine de la vision artificielle, domaine qui a posé et continue de poser de nombreux problèmes et défis.



(a)



(b)



(c)

(a),(b): Perception par deux caméras (stéréovision)

(c) : Perception par une caméra (vision monoculaire)

Figure 1.3 : Capteurs passifs

1.3 Perception de l'environnement par la vision :

1.3.1 La vision chez les êtres vivants :

La vision est un processus de traitement de l'information, en l'occurrence des images. Ces images peuvent être de types variées et le résultat de la vision est une interprétation de l'environnement, également variable (la mouche, la poule et homme « voient » et pourtant ils n'ont pas les mêmes rétines, donc les mêmes images ; et ils n'ont certainement pas les mêmes interprétations des objets vus).

Concernant la vision humaine, l'information qui est réellement disponible sur la rétine des yeux n'est ni plus ni moins qu'une collection de points (environ un million d'après les neurologistes). En chaque point, il a tout simplement une information qui donne une indication quant à la quantité de lumière et de couleur qui proviennent de l'espace environnant et qui ont été projetés en cet endroit de la rétine [6]. Pourtant l'homme voit une table, une voiture, un bureau etc., en plus il en perçoit les distances. Pour l'homme, la vision est une expérience permanente et d'une extrême facilité. Cette facilité n'est cependant qu'une apparence, la vision humaine faisant partie d'un système très perfectionné mais aussi très complexe. On sait maintenant qu'une douzaine de centres cérébraux, soit environ 60% du cortex, participent à l'activité de la vision. La vue est donc d'une importance capitale pour l'intelligence humaine et du fait de cette liaison intime entre l'œil et le cerveau, il est difficile, sinon impossible, de séparer la vision des autres activités intellectuelles.

La vision a suscité l'intérêt de nombreux scientifiques et philosophes depuis fort longtemps. Parmi ceux-ci, les neurobiologistes mènent des recherches théoriques et expérimentales afin d'essayer de comprendre l'anatomie et le fonctionnement du cerveau, qui est loin cependant de révéler tous ses secrets.

1.3.2 La vision artificielle :

La vision artificielle, s'intègre dans le domaine de *l'intelligence artificielle*. C'est la dernière née des disciplines relevant des théories de la perception visuelle. Elle est la science qui développe les bases algorithmiques et théoriques grâce auxquelles l'information utile relative à l'environnement peut être automatiquement extraite et analysée à partir d'une ou de plusieurs images. Les premières années de cette

nouvelle discipline ont été marquées par un essor important grâce aux débouchés applicatifs multiples et sources de grands challenges scientifiques.

La vision artificielle constitue aujourd'hui une discipline que nous pouvons qualifier de délicate du fait de la difficulté de s'inspirer d'un acte instinctif et de le simuler, dans la mesure où nous n'avons que peu de renseignements sur cet acte [7]. Les actes raisonnés sont plus faciles à modéliser en intelligence artificielle que les actes instinctifs comme la vision, à laquelle nous nous intéressons.

Le but final des chercheurs dans ce domaine est d'imiter le système visuel humain, un but qui est loin d'être réalisé. En attendant, ils essaient de créer un modèle qui, vu de l'extérieur, possède des propriétés semblables au fonctionnement de la vision biologique.

En robotique mobile, Il est important de noter que les innovations technologiques concernant la fabrication des caméras et l'évolution des ordinateurs ont permis d'intégrer des systèmes complexes de vision dans les systèmes embarqués. Cependant, les portes sont ouvertes aux chercheurs pour décrire des méthodes et des algorithmes qui exploitent les informations fournies par les systèmes de vision pour donner plus d'autonomie aux robots.

1.3.3 La stéréovision :

Les caméras sont des sources d'information très riches, mais le champ de vue partiel fourni par les objectifs classiques est une limite à leur utilisation. Des applications telle que la surveillance, la réalité virtuelle, et la navigation des robots autonomes peuvent être grandement améliorées par la disponibilité d'un champ de vue très étendu.

Cependant, plusieurs approches existantes pour obtenir des images avec large champ de vision, tel que l'utilisation : d'images multiples, et d'objets spéciaux comme les miroirs convexes [8].

Dans ce contexte, la stéréoscopie est de loin la méthode de prédilection pour résoudre le problème de la reconstruction tridimensionnelle. Inspirée du processus de vision biologique, elle utilise deux ou plusieurs images prises depuis des points de vue différents, afin d'en extraire le relief. C'est un domaine de recherche en vision par ordinateur toujours actif [2]. Généralement, pour aboutir à la reconstruction 3D d'une scène par la stéréovision il est nécessaire de résoudre deux problèmes. Il s'agit:

- i. En fixant une primitive dans une des images, rechercher son homologue dans l'autre image. C'est le problème de la *mise en correspondance*.
- ii. Pour tous les couples appariés calculer la position 3D du point de la scène réelle. C'est le problème de la *reconstruction*.

1.3.4 Quelques applications de la vision artificielle:

Les applications de la vision artificielle sont multiples. En règle générale, tout ce qui peut être vu par un humain peut aussi l'être par un système de vision artificielle. Le contraire n'est pas nécessairement vrai. Nous citons quelques unes dans ce qui suit :

- Effectuer des opérations à distance : le but généralement poursuivi est alors de construire un robot remplaçant au mieux l'homme pour des tâches répétitives : convoyage, surveillance, nettoyage industriel, suivi de routes, où devant évoluer dans un milieu hostile et/ou dangereux : mines, milieu sous-marin [9], espace, sites nucléaires contaminés etc.
- L'imagerie médicale est également un domaine en plein essor, les domaines d'application principaux étant l'aide au diagnostic et l'assistance aux opérations chirurgicales, étude des chromosomes [10], de tumeurs, de lésions osseuses, etc., par le traitement d'images d'électrophorèses, de scanner, résonance magnétiques, d'échographie, etc.
- L'inspection d'images géologiques : reconstruction et étude des strates par l'analyse d'images sismiques [11].
- La télédétection : cartographie, interprétation automatique des images fournies par satellite [12].
- L'étude du mouvement : suivi de cible [13], estimation du déplacement [14], etc.
- Le problème de la reconnaissance d'objets est également un problème d'actualité, pour des applications robotiques, militaires ou civiles. Différentes solutions peuvent être utilisées, nous citons la reconstruction d'objet à partir des points 3D [15] [16].

1.4 Modélisation de la caméra :

L'acquisition d'images est réalisée par l'intermédiaire d'une caméra dont le signal est numérisé. Une caméra comprend un système optique composé de lentilles qui forme une projection bidimensionnelle d'une scène de l'environnement observée sur un plan image photo-sensible (ou rétine).

Chaque caméra délivre, à cadence ou séquence vidéo, une copie de l'image optique reçue par sa rétine électronique, qui est divisée, en abscisse et ordonnée, en un certain nombre de points appelés pixels.

1.4.1 modéliser une caméra:

La modélisation d'une caméra est basée sur la définition d'un ensemble de paramètres qui se rapproche le plus possible du comportement physique du capteur à l'aide d'un modèle géométrique, connu comme étant le modèle sténopé (en anglais : pinhole). Le modèle géométrique sténopé est constitué d'un plan, appelé plan rétinien, dans lequel l'image se forme à l'aide d'une projection perspective. Un point O_c est placé à une distance fixe du plan image. Ce point est appelé le centre optique (ou point focal) et il est défini de sorte que les rayons de lumière reflétés par les objets passent par ce point, formant une image perspective de la scène dans le plan rétinien. A ce moment, deux considérations doivent être prises en compte. La première est que le point focal est placé devant le plan rétinien, dans ce cas l'image obtenue est une projection inverse de la scène (figure 1.4). La deuxième considération, est que le point focal est placé derrière le plan image, cependant, l'image obtenue est une projection de la scène (figure 1.5).

Cette section décrit un modèle de capteur avec un centre optique placé derrière le plan rétinien. Toutefois, les deux configurations sont modélisées avec les mêmes paramètres à un signe près sur la distance focale [7].

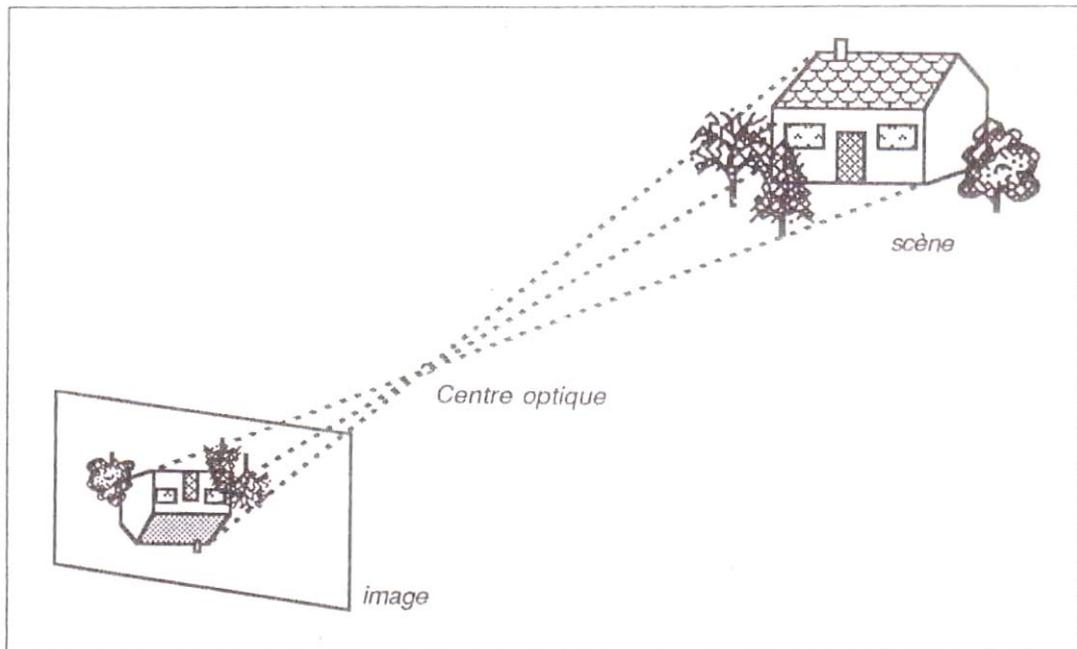


Figure 1.4 : Acquisition d'une scène par caméra. Le centre optique est situé entre la scène et le plan image

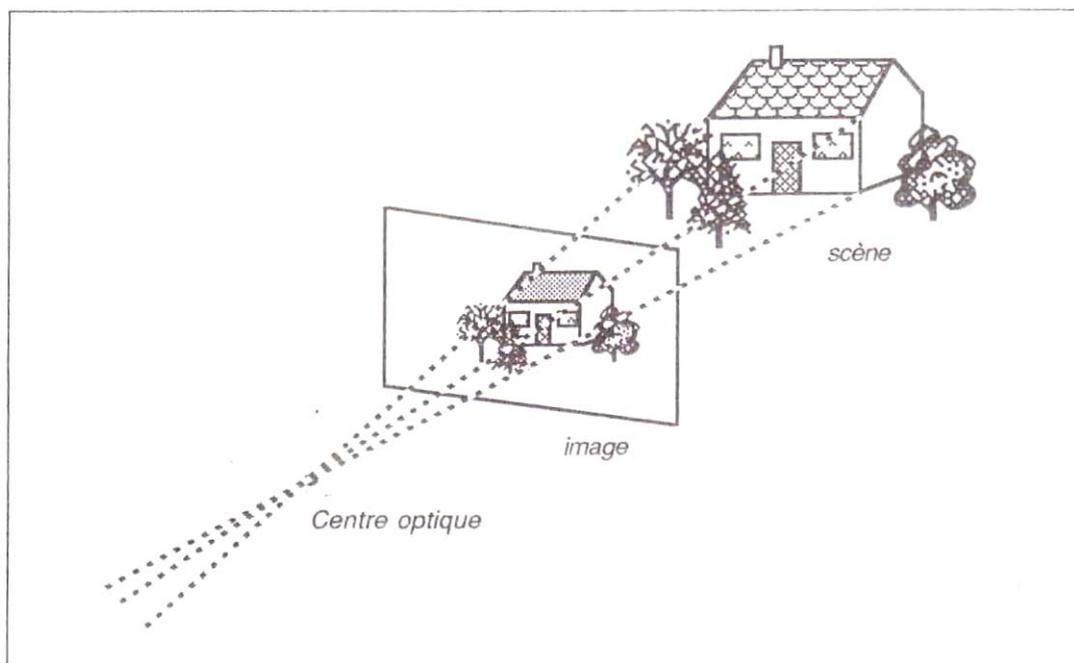


Figure 1.5 : Acquisition d'une scène par caméra. Le centre optique est représenté en arrière du plan image

Il existe deux ensembles de paramètres. Tout d'abord, il y a l'ensemble des paramètres intrinsèques qui modélisent la géométrie interne et les caractéristiques optiques du capteur. Le second ensemble contient les paramètres extrinsèques qui lient le repère du capteur au repère associé à la scène où se trouvent les objets de référence utiles pour la calibration.

L'ensemble des paramètres intrinsèques contient :

- La distance focale f , qui correspond à la distance (en mm) qui sépare le plan rétinien (plan image) et le point focal O_c .
- Les paramètres de conversion \mathbf{k}_u et \mathbf{k}_v , qui représentent l'ajustement horizontal et vertical qui permettent de passer d'un repère du plan rétinien (exprimé en mm) au repère image (exprimé en pixel).
- La position $(\mathbf{u}_0, \mathbf{v}_0)$ du point principal (le centre de l'image) définie comme étant le projeté du centre optique O_c sur le plan image, exprimée en pixel.

L'ensemble des paramètres extrinsèques définit la relation entre le repère caméra localisé sur le centre optique avec le repère du monde localisé dans la scène. Cette relation est un mouvement rigide, elle est définie par la position et l'orientation du repère de la caméra par rapport au repère du monde. La position est donnée par un vecteur translation 3x1 nommé \mathbf{T} et une matrice rotation 3x3 nommée \mathbf{R} .

En résumé, onze paramètres sont utilisés pour modéliser un capteur caméra :

- Six paramètres extrinsèques : les trois composantes du vecteur \mathbf{T} , les trois angles de rotation (α, β, γ) ,
- Cinq paramètres intrinsèques : f , \mathbf{k}_u , \mathbf{k}_v , \mathbf{u}_0 et \mathbf{v}_0 .

1.4.2 Paramètres extrinsèques:

Les points de la scène sont référencés dans le repère du monde, ayant comme origine O_w et comme axes (X_w, Y_w, Z_w) . Soient un point P, dont les coordonnées sont exprimées dans le repère du monde (P_{xw}, P_{yw}, P_{zw}) et un second repère, ayant pour origine O_c et comme axes (X_c, Y_c, Z_c) , placé sur le centre optique de la caméra comme le montre la figure 1.6.

Note : Le point P a pour coordonnées $(P_{xw}, P_{yw}, P_{zw})^t$ dans le repère O_w (repère monde) et $(P_{xc}, P_{yc}, P_{zc})^t$ dans le repère O_c (repère caméra).

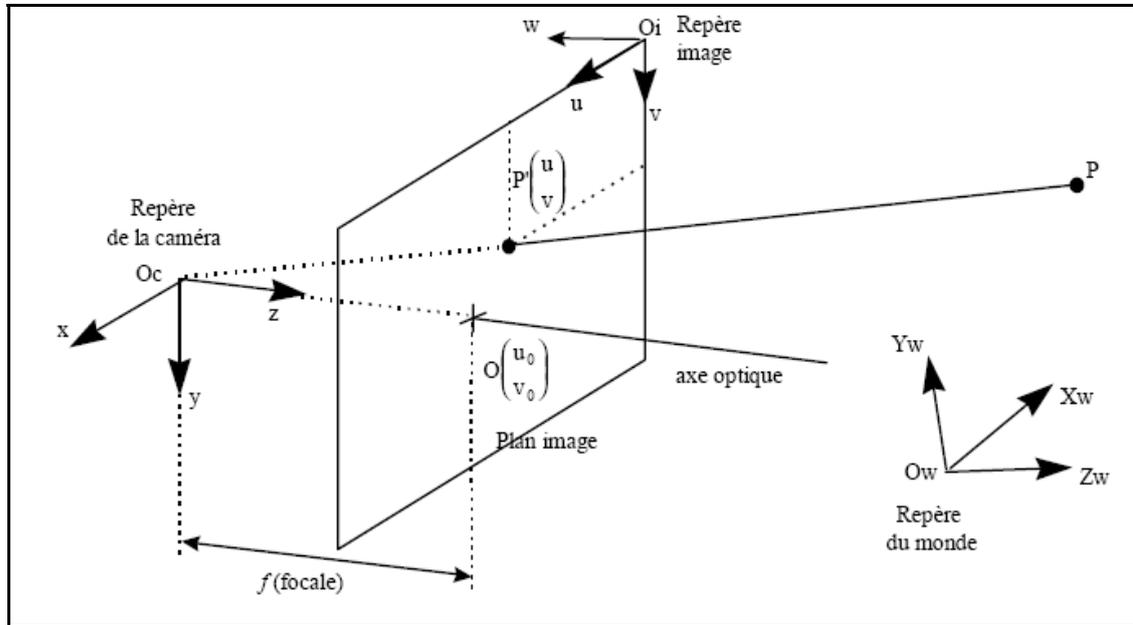


Figure1.6 : Représentation géométrique des repères utilisés pour la détermination du modèle de la caméra.

La relation géométrique qui existe entre ces deux repères permet de définir quels sont les paramètres extrinsèques du modèle [17], de sorte que ces paramètres déterminent la position et l'orientation du repère de la caméra par rapport au repère du monde. La position du point focal O_c par rapport à O_w est donnée par le vecteur translation T de la forme :

$$T = \begin{bmatrix} tx \\ ty \\ tz \end{bmatrix} \dots\dots\dots 1.1$$

L'orientation des axes de la caméra par rapport au repère du monde est donnée par la matrice de rotation R . Cette matrice peut être obtenue par le produit matriciel de trois matrices de rotation simple. Une rotation α autour d'un axe Δ sera exprimée par $Rot(\Delta, \alpha)$.

Comme le montre les équations suivantes, trois matrices de rotation simple peuvent être exprimées (figure 1.7) [18].

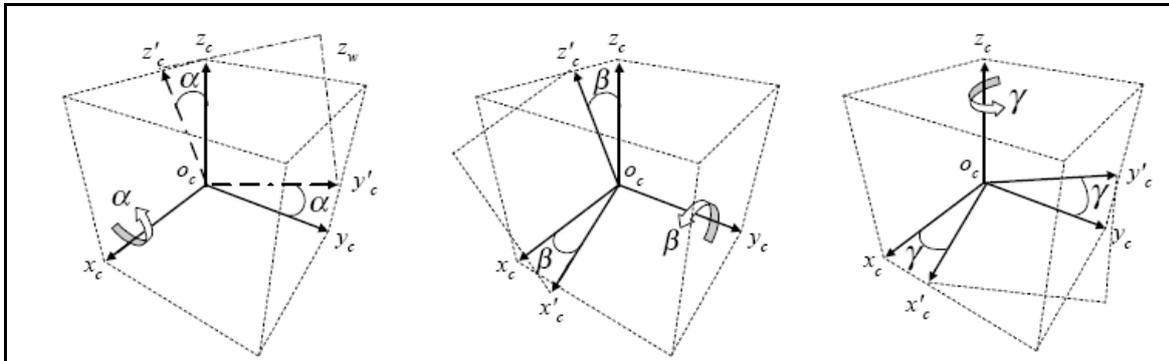


Figure 1.7 : Les trois rotations simples.

$$x'_c = x_c$$

$$y'_c = y_c \cos \alpha - z_c \sin \alpha$$

$$z'_c = y_c \sin \alpha + z_c \cos \alpha$$

$$x'_c = x_c \cos \beta + z_c \sin \beta$$

$$y'_c = y_c$$

$$z'_c = -x_c \sin \beta + z_c \cos \beta$$

$$x'_c = x_c \cos \gamma - y_c \sin \gamma$$

$$y'_c = y_c \sin \gamma + x_c \cos \gamma$$

$$z'_c = z_c$$

$$Rot(X, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \dots\dots 1.2$$

$$Rot(Y, \beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \dots\dots 1.3$$

$$Rot(Z, \gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots 1.4$$

Avec :

$$R = Rot(Z, \gamma) \cdot Rot(Y, \beta) \cdot Rot(X, \alpha)$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \cdot \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos \gamma \cos \beta & \cos \gamma \sin \beta \sin \alpha - \sin \gamma \cos \alpha & \cos \gamma \sin \beta \cos \alpha + \sin \gamma \sin \alpha \\ \sin \gamma \cos \beta & \sin \gamma \sin \beta \sin \alpha + \cos \gamma \cos \alpha & \sin \gamma \sin \beta \cos \alpha - \cos \gamma \sin \alpha \\ -\sin \beta & \cos \beta \sin \alpha & \cos \beta \cos \alpha \end{bmatrix} \dots\dots\dots 1.5$$

De manière à simplifier la notation de la matrice R , il est plus intéressant d'écrire cette matrice à l'aide de ses trois vecteurs d'orientation r_i , où chaque vecteur d'orientation possède trois composantes.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \dots\dots\dots 1.6$$

r_1 , r_2 et r_3 sont des vecteurs unitaires orthogonaux représentant l'orientation du repère de la caméra par rapport au repère du monde.

Donc, les deux repères sont en relation par l'équation suivante,

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \dots\dots\dots 1.7$$

De ce fait, pour un point P , de coordonnées $(P_{xw}, P_{yw}, P_{zw})^t$ dans le repère du monde, nous pouvons calculer ses coordonnées $(P_{xc}, P_{yc}, P_{zc})^t$ dans le repère lié à la caméra.

$$\begin{bmatrix} P_{xc} \\ P_{yc} \\ P_{zc} \end{bmatrix} = R \cdot \begin{bmatrix} P_{xw} \\ P_{yw} \\ P_{zw} \end{bmatrix} + T \dots\dots\dots 1.8$$

La relation (1.8) est habituellement exprimée par une seule matrice avec des coordonnées homogènes, comme le montre l'équation,

$$\begin{bmatrix} P_c \\ 1 \end{bmatrix} = A \cdot \begin{bmatrix} P_w \\ 1 \end{bmatrix} \dots\dots\dots 1.9$$

$$A = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \dots\dots\dots 1.10$$

Cette matrice représente une transformation rigide : trois rotations et trois translations. Elle représente les paramètres extrinsèques. La matrice inverse est donnée par l'expression suivante :

$$A^{-1} = \begin{bmatrix} R^t & -R^t T \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \dots\dots\dots 1.11$$

En résumé, six paramètres extrinsèques sont nécessaires pour modéliser la position et l'orientation de la caméra par rapport au repère du monde. Les six paramètres sont : les trois composantes du vecteur translation $T = (t_x, t_y, t_z)^t$ et les trois angles (α, β, γ) qui permettent de définir la matrice de rotation R .

1.4.3 Paramètres intrinsèques :

Dans cette partie, nous allons nous intéresser à la modélisation des paramètres intrinsèques au capteur caméra. De ce fait, nous allons nous focaliser sur la détermination des paramètres qui modélisent la géométrie interne et les caractéristiques optiques du capteur. Le modèle intrinsèque de la caméra permettra d'effectuer la relation entre un point 3D et son projeté 2D observable dans le plan image. Les coordonnées du point 3D doivent être exprimées dans le repère du capteur, et les coordonnées 2D du point projeté seront exprimées dans le repère image, en pixel.

1.4.3.1 La projection perspective :

Soit un point O dans le plan image, appelé point principal, et soit une droite perpendiculaire au plan image passant par O , appelée axe optique. Soit un point O_c placé sur l'axe optique à une distance f du plan image. Le point O_c est le centre de projection et f est la distance focale. Comme nous l'avons exprimé précédemment, nous plaçons le centre de projection derrière le plan image (figure 1.8).

Un point P se projette sur le plan image le long d'une droite passant par P et O_c . Comme nous l'avons déjà précisé, les coordonnées du point P sont exprimées dans le repère attaché à la caméra [17] : le plan $x-y$ de ce repère est parallèle au plan image et l'axe z est confondu avec l'axe optique. L'origine de ce repère se trouve en O_c . Les coordonnées du point P dans ce repère sont $(P_{xc}, P_{yc}, P_{zc})^t$. Dans ce même repère, selon le théorème de Talas les coordonnées du point projeté sont :

$$x' = f \cdot \frac{P_{xc}}{P_{zc}}$$

$$y' = f \cdot \frac{P_{yc}}{P_{zc}}$$

$$z' = f$$

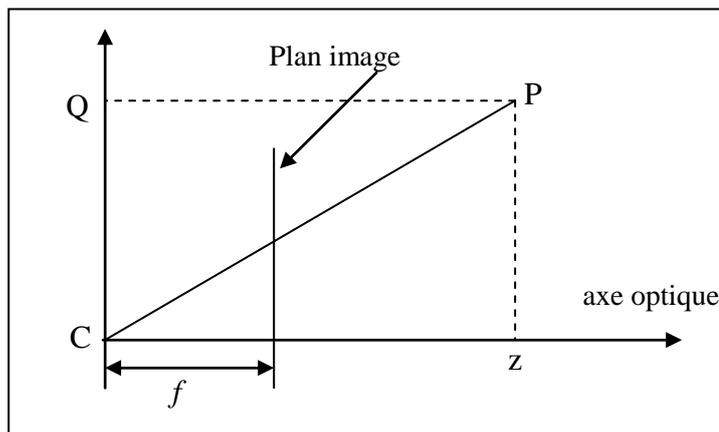


Figure1.8 : projection perspective cas du plan vertical

En utilisant les coordonnées homogènes de $P' : (P_{xc}, P_{yc}, P_{zc})^t$, et en écrivant cette transformation sous forme matricielle, on obtient la relation :

$$\begin{bmatrix} sx' \\ sy' \\ sz' \\ s \end{bmatrix} = PP \cdot \begin{bmatrix} P_{xc} \\ P_{yc} \\ P_{zc} \\ 1 \end{bmatrix}$$

Avec :

$$PP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \dots\dots\dots 1.12$$

Les coordonnées cartésiennes de P' sont : sx'/s , sy'/s et sz'/s

1.4.3.2 Transformation caméra/image :

Les points images sont mesurés en pixel dans un repère bi-dimensionnel $u-v$ associé à l'image [19]. Afin de pouvoir écrire la matrice de transformation du repère caméra au repère image, nous devons introduire les paramètres suivants k_u , le facteur d'échelle horizontal (pixels/mm),

- k_v , le facteur d'échelle vertical (pixels/mm).
- u_0 , v_0 et w_0 sont les coordonnées de O_c , centre optique, dans le repère image (mesurées en pixel).

La transformation du repère caméra au repère image s'écrit (pour le point P'),

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \\ w_0 \end{bmatrix} \dots\dots\dots 1.13$$

C'est une transformation affine représentant un changement d'échelle et une translation. La composante w étant toujours nulle, on peut ignorer la troisième ligne et écrire cette transformation sous la forme suivante :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} k_u & 0 \\ 0 & k_v \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \dots\dots\dots 1.14$$

Remarque : u_0 et v_0 sont les coordonnées, dans le plan image, du point principal (le centre de l'image) donné par l'intersection entre l'axe optique de la caméra et le plan image. Le point est considéré comme étant l'origine du plan image.

La transformation (1.14) peut s'écrire sous forme matricielle par :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \dots\dots\dots 1.15$$

Ou bien encore :

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = K \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \dots\dots\dots 1.16$$

Avec :

$$K = \begin{bmatrix} k_u & 0 & 0 & u_0 \\ 0 & k_v & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots 1.17$$

Cette transformation représente une application linéaire de l'espace projectif vers le plan projectif.

En multipliant les matrices K et PP (transformation affine et projection perspective), nous pouvons écrire les équations du modèle géométrique de la

caméra, soit la relation entre les coordonnées caméra $(P_{xc}, P_{yc}, P_{zc})^t$ du point P et ses coordonnées image (u, v) ,

$$\begin{cases} u = k_u \cdot f \cdot \frac{P_{xc}}{P_{zc}} + u_0 \\ v = k_v \cdot f \cdot \frac{P_{yc}}{P_{zc}} + v_0 \end{cases} \dots\dots\dots 1.18a, b$$

En effet, le produit $K.PP$ est,

$$\begin{bmatrix} k_u & 0 & u_0 / f & 0 \\ 0 & k_v & v_0 / f & 0 \\ 0 & 0 & 1 / f & 0 \end{bmatrix}$$

En multipliant tous les coefficients de la matrice par f (ce qui ne modifie rien puisque les coordonnées homogènes sont définies à un facteur multiplicatif près) on obtient :

$$I_c = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \dots\dots\dots 1.19$$

Avec :

$$\begin{cases} \alpha_u = f \cdot k_u \\ \alpha_v = f \cdot k_v \end{cases} \dots\dots\dots 1.20a, b$$

I_c est une application linéaire de l'espace projectif vers le plan projectif exprimant la transformation perspective :

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = I_c \cdot \begin{bmatrix} P_{xc} \\ P_{yc} \\ P_{zc} \\ 1 \end{bmatrix} \dots\dots\dots 1.21$$

Ce modèle comporte quatre paramètres : α_u, α_v, u_0 et v_0 . Ce sont ces paramètres qui vont être estimés lors de la phase de calibration. Donc, il n'y a plus onze mais dix paramètres à déterminer.

Remarque : la distance focale ne peut être calculée explicitement car elle n'apparaît plus dans le modèle. De plus, l'utilisation de paramètres tel que k_u et k_v montre que lorsqu'on calibre une caméra, on s'intéresse à la modélisation de la caméra et du

convertisseur analogique/digital. Si on change l'un des deux, il faut recalibrer le système [20].

1.4.4 Le modèle complet sans distorsion :

De manière à exprimer la relation entre la métrique des points 3D référencés dans le repère du monde et leurs points 2D, résultats de la projection dans le plan image, exprimés en pixel, les équations matricielles vont être étendues. Cette relation est habituellement exprimée par le produit de deux matrices. La première matrice exprime l'intégralité des paramètres intrinsèques du capteur caméra, et la seconde exprime l'intégralité des paramètres extrinsèques. De (1.9) et (1.21), on en déduit,

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} Px_w \\ Py_w \\ Pz_w \\ 1 \end{bmatrix} \dots\dots\dots 1.22$$

On obtient donc les équations,

$$\begin{cases} u = \alpha_u \cdot \frac{r_{11}Px_w + r_{12}Py_w + r_{13}Pz_w + t_x}{r_{31}Px_w + r_{32}Py_w + r_{33}Pz_w + t_z} \\ v = \alpha_v \cdot \frac{r_{21}Px_w + r_{22}Py_w + r_{23}Pz_w + t_y}{r_{31}Px_w + r_{32}Py_w + r_{33}Pz_w + t_z} \end{cases} \dots\dots\dots 1.23a,b$$

Nous avons donc un système linéaire qui modélise le capteur. Le principe pour calibrer une caméra, c'est-à-dire déterminer les paramètres du modèle, consiste à placer devant la caméra une mire (un objet étalon). Une mire est un ensemble de points dont les coordonnées sont parfaitement connues dans un repère de la mire (confondu avec le repère du monde) qui est différent du repère de la caméra, voir Figure 1.9. Chaque point de la mire se projette dans l'image et on mesure ses coordonnées dans le repère image (l'annexe B propose les méthodes de calibrage existe et une liste de différents types de mire). La transformation mire/image correspond donc au modèle de (1.22).

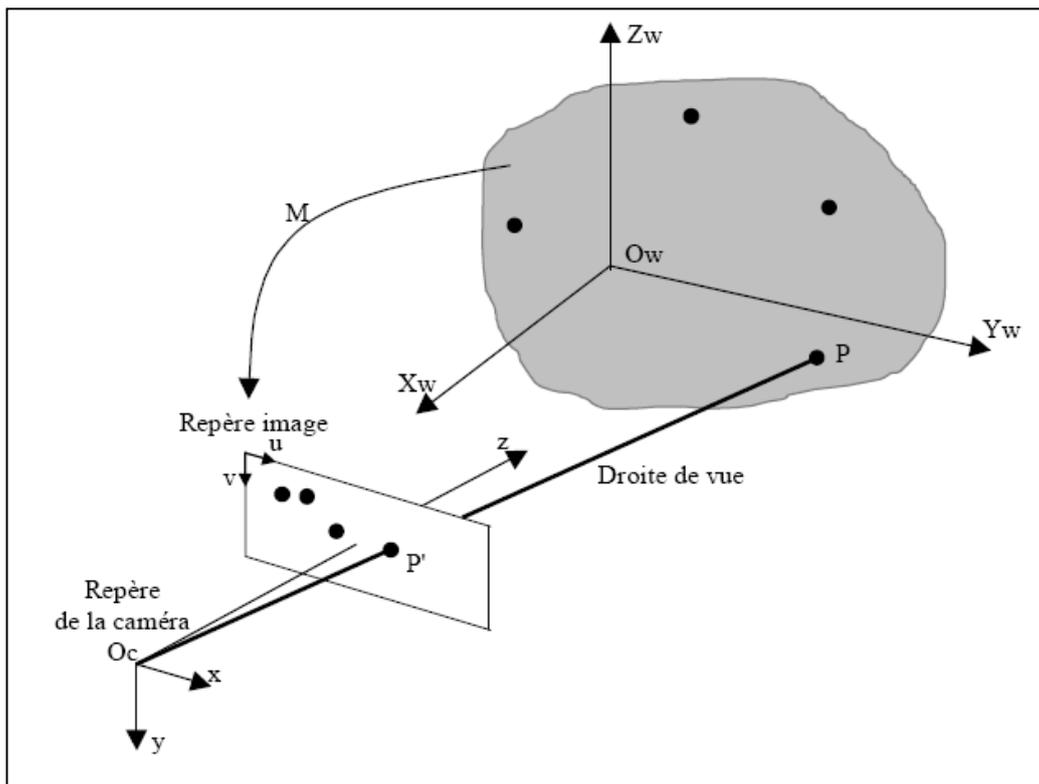


Figure 1.9 : Principe de calibration d'une caméra. Un point P, dont les coordonnées sont exprimées dans le repère mire, se projette en P', dont les coordonnées sont exprimées dans le repère image.

Il est maintenant possible d'écrire la transformation mire/image sous la forme d'une matrice 3x4 :

$$M = I_c \times A$$

$$M = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_2 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{bmatrix} \dots\dots\dots 1.24$$

La matrice **M**, appelée matrice de projection perspective (Figure 1.9), s'écrit de façon la plus développée sous la forme :

$$M = \begin{bmatrix} \cos\beta\cos\gamma\alpha_u - \sin\beta u_0 & (\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma)\alpha_u + \sin\alpha\cos\beta u_0 & (\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma)\alpha_u + \cos\alpha\cos\beta u_0 & t_x\alpha_u + t_z u_0 \\ \cos\beta\cos\gamma\alpha_v - \sin\beta v_0 & (\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma)\alpha_v + \sin\alpha\cos\beta v_0 & (\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma)\alpha_v + \cos\alpha\cos\beta v_0 & t_y\alpha_v + t_z v_0 \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta & t_z \end{bmatrix} \quad 1.25$$

Ce résultat, très connu, est de la forme matricielle suivante :

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} P_{x_w} \\ P_{y_w} \\ P_{z_w} \\ 1 \end{bmatrix} \dots\dots\dots 1.26$$

Dans cette formule, P_{x_w} , P_{y_w} et P_{z_w} sont les coordonnées d'un point P de la mire dans le repère de la mire (monde). Cette matrice peut également s'écrire sous la forme plus simple :

$$M = \begin{bmatrix} m_1 & m_{14} \\ m_2 & m_{24} \\ m_3 & m_{34} \end{bmatrix} \dots\dots\dots 1.27$$

Avec m_i représentatif d'un vecteur ligne à 3 composantes :

$$m_i = [m_{i1} \quad m_{i2} \quad m_{i3}] \dots\dots\dots 1.27, (i=1,2,3).$$

Par identification des équations (1.25), (1.26) , nous trouvons les différents coefficients qui composent la matrice de projection perspective qui sont:

$$\left\{ \begin{array}{l} m_{11} = \cos\beta\cos\gamma\alpha_u - \sin\beta u_0 \\ m_{12} = (\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma)\alpha_u + \sin\alpha\cos\beta u_0 \\ m_{13} = (\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma)\alpha_u + \cos\alpha\cos\beta u_0 \\ m_{14} = t_x\alpha_u + t_z u_0 \\ m_{21} = \cos\beta\sin\gamma\alpha_v - \sin\beta v_0 \\ m_{22} = (\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma)\alpha_v + \sin\alpha\cos\beta v_0 \\ m_{23} = (\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma)\alpha_v + \cos\alpha\cos\beta v_0 \\ m_{24} = t_y\alpha_v + t_z v_0 \\ m_{31} = -\sin\beta \\ m_{32} = \sin\alpha\cos\beta \\ m_{33} = \cos\alpha\cos\beta \\ m_{34} = t_z \end{array} \right. \dots\dots\dots 1.28$$

1.4.5 Relation géométrique entre les images:

La géométrie épipolaire est intrinsèque à tout système de deux caméras quel que soit le modèle utilisé pour ces caméras. Cette géométrie décrit une relation géométrique entre deux images prises sous deux angles de vue différents. La précision de l'estimation de la géométrie épipolaire est très importante parce qu'elle conditionne la précision des algorithmes de mise en correspondance entre points d'un couple d'images stéréoscopiques, ces algorithmes reposant souvent sur la connaissance a priori de cette géométrie.

Considérons deux caméras qui observent une même scène, ce qui est par exemple le cas de la stéréovision. Appelons les deux images produites par ces deux caméras image gauche et image droite (figure 1.10).

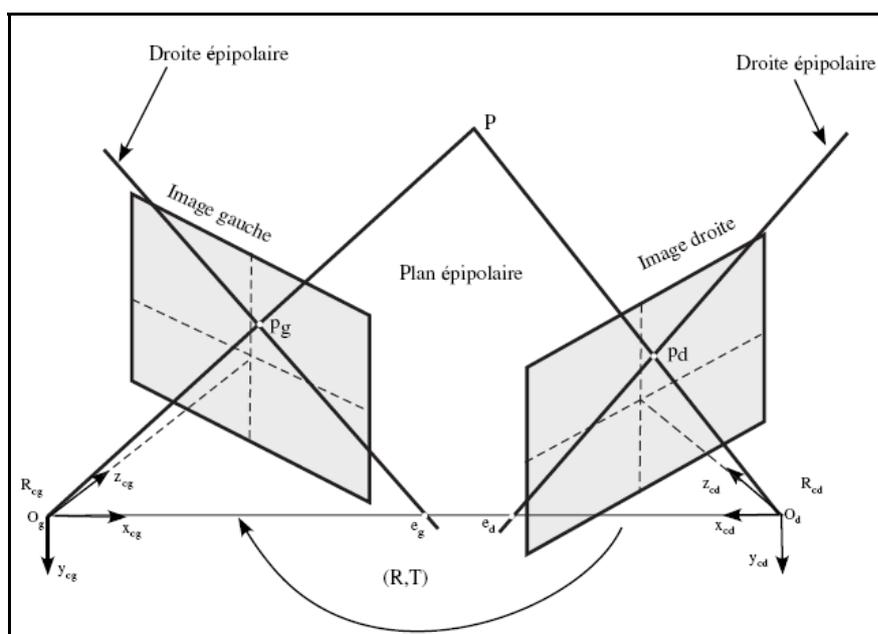


Figure 1.10 : Relation géométrique entre une paire d'images : géométrie épipolaire

Il est clair (figure 1.10) que le point P_g de l'image gauche représente la projection de tous les points de l'espace appartenant à la droite $O_g P$ [21]. Les correspondants possibles de P_g dans l'image droite sont contraints de se trouver sur une droite, appelée droite épipolaire de P_g . Cette dernière n'est autre que la projection de la droite $O_g P$ dans l'image droite.

Toutes les droites épipolaires de l'image droite passent par e , intersection de la droite $O_g O_d$ avec le plan de l'image droite. e_d est appelé épipole de l'image droite par rapport à l'image gauche.

De manière symétrique, l'épipole de l'image gauche par rapport à l'image droite est défini comme étant l'intersection de $O_g O_d$ avec le plan de l'image gauche. En d'autres termes, un point P de l'espace définit avec les deux centres de projection O_g et O_d un plan appelé plan épipolaire. Ce dernier coupe les deux plans images en deux droites, dit les droites épipolaires. Tous les plans épipolaires passent par la droite $O_g O_d$ et donc par les épipoles e_g et e_d . Cette relation entre deux images est appelée *géométrie épipolaire*. Elle est d'une grande importance. En particulier elle donne une contrainte sur la mise en correspondance, (plus de détails dans chapitre3).

1.4.6 Rectification d'images :

La rectification d'une paire d'images stéréoscopiques permet de se ramener à une géométrie épipolaire simple où les droites épipolaires sont parallèles à l'axe horizontal passant par les deux centres optiques. Ceci permet de réduire la recherche du point correspondant de l'image droite à une droite horizontale de l'image gauche située à la même cordonnée. Dans cette configuration, les deux épipoles e_g et e_d sont projetés à l'infini. Lorsqu'on procède à un étalonnage des deux caméras, la rectification des images est simple [22]. Par contre, cette rectification devient ardue lorsque les positions des deux caméras ne sont pas connues [30]. La figure 1.11, indique le principe d'une rectification des images.

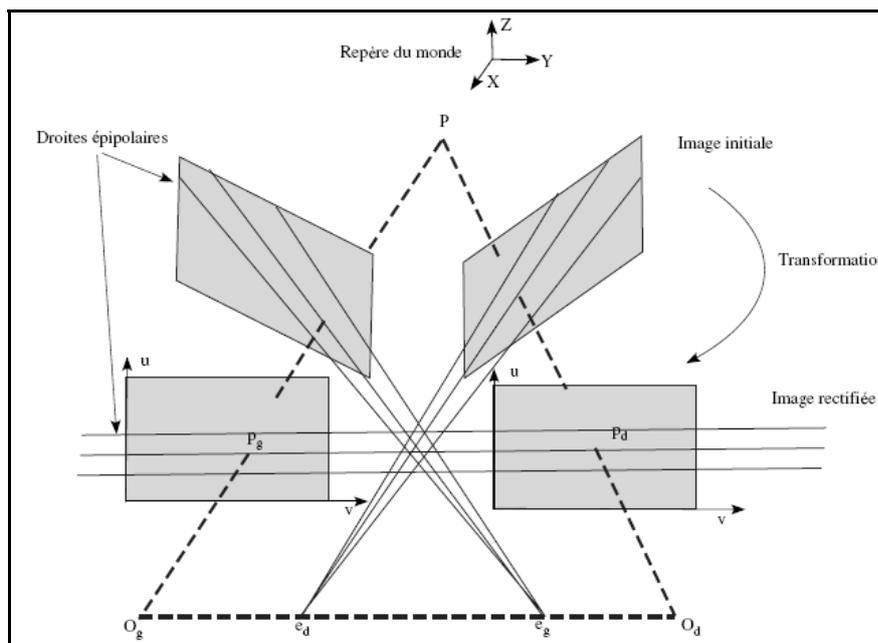


Figure 1.11: Rectification d'images.

1.5 Modélisation de l'environnement :

La construction de la carte de l'environnement, ou cartographie de l'environnement, peut se diviser en trois processus [23] :

- Mémorisation de la carte.
- Localisation dans la carte.
- Recherche d'un chemin dans la carte.

Si le dernier problème est relativement découplé des deux autres, les premiers sont indissociables. C'est le problème de la poule et de l'oeuf : sans carte on ne peut pas se localiser, mais sans connaître sa position on ne peut pas construire de carte. Cette relation est repérée en anglais par le terme SLAM (Simultaneous Localization And Mapping).

Nous pouvons diviser les approches de la cartographie en deux grandes familles :

Les cartes *métriques* placent dans un plan muni d'un repère absolu les obstacles repérés par leur position et leur occupation de l'espace. Au contraire, les cartes appelées *topologiques* repèrent les lieux par la façon dont on peut naviguer entre eux. L'information de position ou de distance peut aussi être ajoutée, mais elle est alors secondaire, la principale étant la connexité. La figure 1.12 représente cette différence [24].

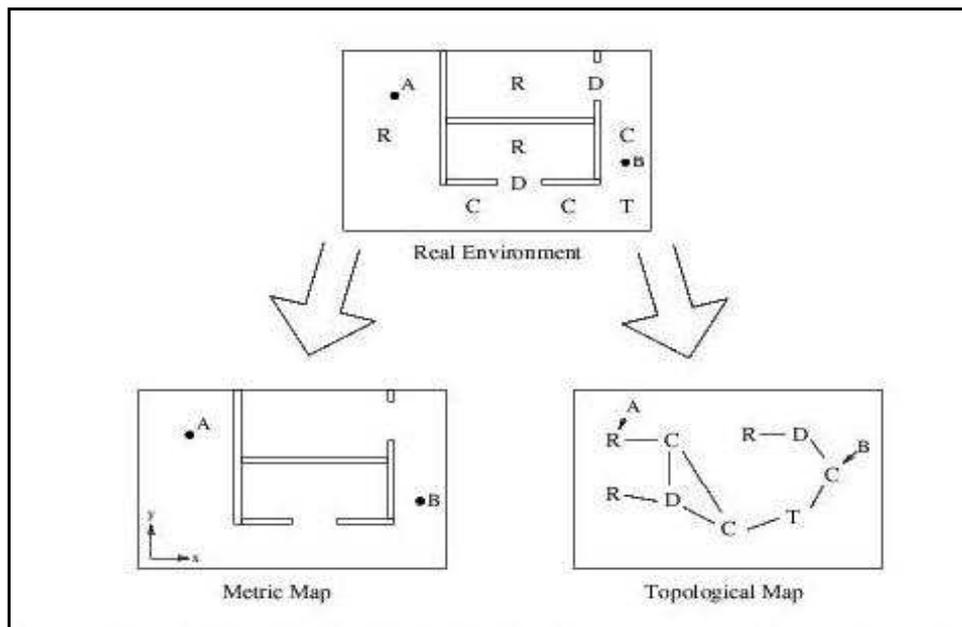


Figure 1.12: Illustration de la distinction classique entre les cartes métriques et les cartes topologiques.

Dans le cadre métrique, les positions des objets sont calculées et représentées dans un repère commun. Deux positions, A et B, sont représentées dans la carte par leurs coordonnées dans ce repère. Dans les cartes topologiques, les lieux sont stockés avec leurs relations spatiales. Deux positions A et B dans l'environnement peuvent être reconnues comme faisant partie des lieux R et C. Cela permet d'inférer que la position B peut être atteinte à partir de A en passant par C et T.

1.5.1 Carte métrique :

Les cartes métriques, qui sont à la fois les plus naturelles et les plus utilisables par l'homme, ont fait l'objet de la majorité des travaux dans le domaine de la robotique. La construction d'une carte métrique comprend deux étapes essentielles : à chaque pas de temps, il convient

- i. d'estimer la position du robot ;
- ii. d'associer à cette position les observations disponibles (obstacles etc.).

Cependant, l'approche la plus courante dans les cartes métriques est d'utiliser une grille d'occupation [25] [26], c'est à dire une matrice qui stocke la probabilité de présence d'un obstacle et qui est mise à jour grâce à la règle de Bayes. Avec un tel modèle, il est possible d'appliquer des algorithmes pour générer une trajectoire qui conduit le robot dans l'espace, de (x_1, θ_1) position courante, au (x_2, θ_2) le but.

1.5.2 Carte topologique :

Le principe des cartes *topologiques* est de construire un graphe qui va représenter l'environnement. Les lieux sont alors représentés par des noeuds, chaque noeud correspond à un endroit caractéristique (carrefour entre couloirs, entrées dans les espaces ouverts etc.), que nous appellerons *lieu*. Chaque lieu devra être décrit par un ensemble des caractéristiques propres, qui permettront au robot mobile de le reconnaître ; leurs positions relatives sont stockées dans les arcs. En général les noeuds conservent alors l'information des capteurs, alors que les arcs stockent les informations de l'odomètre. La figure 1.13 présente une carte topologique d'un environnement naturel.

Le problème le plus difficile est de savoir si un lieu correspond à un noeud déjà dans la carte ou s'il n'a jamais été visité. S'il n'y a pas de *perceptual aliasing*⁽¹⁾ [27], l'ajout de noeud dans la carte est trivial.

En effet, si l'information des capteurs correspond à un noeud déjà présent, alors la position est connue parfaitement, sinon on sait que l'on est dans un nouveau lieu, et il suffit de le rajouter.

Dans le cas de la navigation topologique, le robot n'a nul besoin de connaître sa position précise par rapport à un repère du monde. Il doit seulement déterminer une localisation qualitative « *Je suis dans le lieu A* » et, au mieux, une estimation de sa position relative par rapport à un repère lié au lieu dans lequel il se trouve.

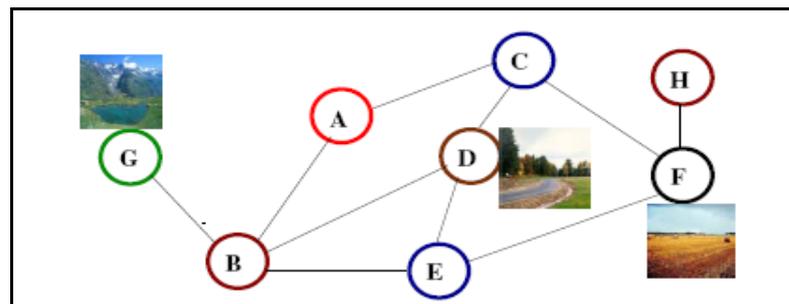


Figure 1.13: Carte topologique. Chaque noeud contient la description d'un lieu distinctif ; les arêtes décrivent la trajectoire reliant deux nœuds

¹ *perceptual aliasing*, faux perception: c'est à dire que deux lieux différents peuvent donner la même mesure dans deux situations différentes (le même lieu pourrait être interprété comme deux places différentes).

1.6 Conclusion :

Les différentes méthodes de perception sont généralement comparées sur les caractéristiques suivantes :

- Vitesse d'acquisition,
- Densité des données acquises,
- Précision obtenue sur les résultats,
- Type des résultats : coordonnées *3D* avec ou sans informations de type photométrique pour le cas d'une caméra.

Comme nous l'avons montré dans ce chapitre, le choix d'un système d'acquisition *3D* dépend généralement de l'application considérée et de la qualité de l'information demandée. Rappelons que la finalité de ce travail sera utilisée pour la navigation du robot mobile ATRV2 équipé d'un banc stéréoscopique, donc nous nous intéresserons dans la suite de cette mémoire au traitement d'une paire d'images acquise par un banc stéréoscopique calibré (stéréovision).

Nous avons présenté aussi la modélisation géométrique de la caméra, capteur essentiellement utilisé dans la vision artificielle, en particulier dans la perception de l'environnement des robots mobiles, la modélisation de ce dernier a été étudié en utilisant le modèle sténopé.

La dernière section a été consacrée à la modélisation de l'environnement; deux approches ont été exposées les cartes métriques et cartes topologiques.

CHAPITRE 2

SEGMENTATION D'IMAGES & CHOIX DES PRIMITIVES

Dans une application de la vision en robotique, le temps réel est indispensable, c'est-à-dire, être en mesure d'actualiser suffisamment la commande envoyée aux actionneurs du robot. Le traitement de l'image brute nécessite un temps de calcul très important, en plus l'image contient des informations inutiles pour notre application (navigation des robots mobiles). Pour cela on fait appel à la segmentation, qui nous permet d'avoir une représentation appropriée des données.

Dans la première partie de ce chapitre, on montre l'importance du choix de l'indice image ou primitive à extraire, ainsi que les différents types de primitive utilisés avec leurs avantages et inconvénients. Nous justifions notre choix par comparaison avec les autres primitives images les plus fréquemment employées.

La deuxième partie, sera consacrée à la segmentation d'image, nous détaillerons les différentes étapes qui permettent d'aboutir à partir d'une image brute à une image de segments de droite, à savoir, l'extraction de contours, le chaînage et l'approximation polygonale.

2.1 Choix de primitives :

2.1.1 Introduction:

Les indices visuels sont des objets extraits de l'image qui contiennent de manière concise une information importante de la scène observée. En général ces indices sont soit des contours, soit des régions de l'image. Contours et régions sont complémentaires car on définit qualitativement les régions comme les zones de l'image homogènes au sens d'un certain critère [28], tandis que les contours sont les zones de transition entre les régions homogènes.

Les propriétés requises par les indices visuels dépendent de l'application et sont en général un sous-ensemble de liste ci-dessous [29]:

1. **Compacts** : la représentation de l'image doit être aussi concise que possible pour réduire la complexité des algorithmes des phases ultérieurs.
2. **Intrinsèques** : les indices visuels doivent correspondre à la projection dans l'image d'objets physiques. En particulier, ils doivent être invariants par changement de point de vue.
3. **Robustes** : la représentation doit être peu sensible aux petites variations d'intensité dans l'image provoquées par des bruits divers (acquisition, digitalisation, etc.).
4. **Discriminants** : les indices visuels doivent posséder des propriétés qui permettent de les discriminer, afin de faciliter la phase de mise en correspondance de deux descriptions.
5. **Précis** : la position des indices visuels doit pouvoir se calculer avec précision car la qualité de la localisation des objets physiques en dépend.
6. **Denses** : la densité des indices visuels doit être suffisante pour représenter tous les objets intéressants de la scène.

Dans la littérature, il existe plusieurs variétés d'indices visuels (ou primitives) telles que :

- Des points (pixels, points d'intérêt, points caractéristiques, etc.)
- Des contours (segments de droites, arcs de cercle, etc.)
- Des régions.

2.1.2 Points d'intérêt :

Les points d'intérêts correspondent généralement à un changement bidimensionnel du signal d'intensité comme par exemple les coins, les jonctions etc. De nombreuses méthodes pour l'extraction de tels points sont proposées dans la littérature. Malgré leur diversité, elles peuvent être classées en deux grandes catégories:

- La première extrait les points d'intérêt à partir de primitives plus évoluées comme les contours (*Contour based methods*).
- La seconde catégorie, propose d'extraire les points directement à partir du signal d'intensité selon une approche différentielle (*Signal-based methods*).

Ces points doivent pouvoir être appariés facilement dans l'autre image. Ils ont un voisinage particulier, facile à identifier sans ambiguïté par une mesure de corrélation. En fait, les points d'intérêts seuls ne sont pas d'un grand apport pour une reconstruction 3D. Ils sont généralement utilisés dans le processus d'étalonnage du capteur pour trouver le modèle de la caméra (estimation de la matrice fondamentale) : dans une paire d'images stéréoscopiques, les points d'intérêt peuvent servir à créer la liste d'appariements initiale.

Le résultat de reconstruction 3D en utilisant les points d'intérêts comme élément de base est une pauvre présentation de la scène, car ces derniers ne présente qu'une quantité assez faible d'information de la scène. Notons que c'est une primitive qui a été utilisée pour la navigation des robots mobiles dans des environnements très bien structurés [27] [30]. Nous pouvons résumer les avantages liés à l'utilisation de cette primitive dans cinq points qui sont :

1. La plus facile et rapide à extraire.
2. La plus stable car les points d'intérêt sont invariants vis-à-vis : du changement d'échelle, rotation, point de vue et de variation de luminosité.
3. La primitive qui peut être utilisé quel que soit la complexité et la nature de l'environnement (d'intérieur ou d'extérieur).
4. La phase de l'appariement est beaucoup plus rapide quand le nombre des candidats à apparier est réduit.
5. Les points d'intérêt sont bien localisés sur les images (avec une précision théorique inférieure au pixel).

2.1.3 La primitive région :

L'objectif de la segmentation d'une image en régions est de partitionner l'image en zones correspondant à des entités de la scène observée [28]. Une région est définie comme étant un ensemble de points connexes qui vérifient un critère d'homogénéité. De nombreuses propriétés sont attachées à cet ensemble de points, comme la longueur des frontières, la surface, le centre de gravité, l'intensité moyenne, etc.

Du fait qu'elles sont extraites à partir de propriétés photométriques, les régions sont sensibles aux variations d'intensité lumineuse. Ainsi, dans un couple d'images stéréoscopiques, les régions peuvent avoir des formes et des tailles différentes. La

mise en correspondance de régions est donc une mise en correspondance d'un ensemble de pixels. Dans la littérature, la mise en correspondance de régions est bien moins utilisée que l'appariement exploitant d'autres primitives géométriques. La segmentation en régions est consommatrice en temps de calcul [28]. Par ailleurs, les primitives de type région couvrent tous les points de l'image et, en appariant les régions, on apparie tous les points de l'image, ce qui permet d'avoir une reconstruction 3D dense (tous les points des images seront reconstruits dans l'espace 3D). [31]

L'utilisation des indices visuels de type région ne vérifie pas suffisamment les propriétés citées ci-dessus surtout la deuxième et la cinquième propriété, c'est-à-dire l'obtention de régions intrinsèques et précises. Ces pour ces raisons la primitive régions est moins utilisée dans la stéréovision [29] [32]. En effet elle est plus adaptée à la vision monoculaire, principalement, dans l'interprétation de la scène sans passer par le calcul de la troisième dimension, généralement on exploite l'information couleur [33] [34].

2.1.4 Les points contours :

Les avantages, qui rendent la mise en correspondance de primitives basées sur l'information de type contour très intéressante, sont très nombreux. Tout d'abord, ils conservent une grande partie du contenu sémantique de l'image. De plus, le temps de calcul des algorithmes de mise en correspondance exploitant ce type de primitives est réduit, à cause du nombre réduit de points de contour contenus dans l'image [21]. Ce dernier avantage est très appréciable pour des applications qui exigent des traitements en temps réel. Un autre avantage de l'utilisation des primitives extraites de contour réside dans la précision des mesures que l'on peut effectuer sur les contours; ce qui apporte de la fiabilité aux applications, par exemple dans le domaine de la navigation des robots mobiles. En plus, les contours sont robustes aux variations géométriques et photométriques.

2.1.5 Les segments de droite :

Les segments sont plus discriminants que les points de contour pris séparément. Ils permettent de réduire plus la complexité de l'algorithme de mise en correspondance en réduisant le nombre d'appariements à réaliser, car le nombre de segments est toujours inférieur au nombre de points. De plus, les attributs

géométriques mesurés sur les segments de contour sont beaucoup plus riches en information que ceux mesurables sur des points isolés. Enfin, la précision sur la mesure de la position et de l'orientation d'un segment est souvent beaucoup plus grande que celle liée à des mesures sur des points isolés ou points contours.

La réalisation d'une tâche de navigation ou de localisation visuelle en robotique mobile nécessite d'une part un traitement en temps réel, et d'autre part une représentation assez dense de la scène observée. Notons aussi que pour une telle tâche une représentation filaire des objets suffit. Pour ces raisons et d'autres, la primitive segments de droite semblent la primitive la plus séduisante, car elle nous permet de faire un compromis entre toutes ces contraintes.

Ceci explique le grand nombre de méthodes de stéréovision fondées sur l'utilisation de ce type de primitives dans la navigation des robots mobiles, à titre d'exemple nous citons les travaux de Buffa, Faugeras et Zhang dans [35].

En résumé :

Inconvénients :

1. L'utilisation de segments de droite nous permet une reconstruction filaire des objets de la scène.
2. Pour un même objet 3D, la projection sur les deux images n'ont jointre pas forcément le même nombre de segment dans la paire d'images stéréoscopique.

Avantages :

1. Le nombre de segments de droite dans une image est beaucoup plus inférieur que le nombre de points.
2. Un segment dépend faiblement de la variation d'éclairage et par conséquent, de l'angle de prise de vue des caméras donc les segments résultant de l'extraction de deux images sont relativement similaires.
3. Les segments contiennent beaucoup d'information ou attributs géométriques exemple : longueur, orientation, point milieu, etc.

2.1.6 Discussion :

Les types d'indices visuels que nous avons étudiés ci-dessus permettent de représenter les informations contenues dans une image d'une façon très compacte et très structurée. Cette réduction des données est très utile lorsqu'on applique les algorithmes de mise en correspondance.

Pour la vision en robotique, il est nécessaire pour le traitement de haut niveau tel que la mise en correspondance et la reconstruction 3D, d'extraire des deux images un ensemble d'indices visuels permettant une meilleure représentation du relief de l'environnement. Le choix de l'une des primitives citée ci-dessus, peut être principalement justifié par les arguments suivants :

- Type de l'environnement : pour la navigation des robots mobiles, les segments sont mieux utilisés, quand l'environnement est structuré (milieu d'intérieur ou urbain). Par contre la navigation dans un environnement naturel nécessite l'utilisation de primitives région.
- La richesse de la carte 3D : la richesse de la carte 3D dépend directement de la quantité d'information à l'entrée du processus de reconstruction, et donc du type de primitives choisi.

En ce qui nous concerne, nous avons choisi comme élément de base « segments de droite », et ceci, dans le but de faire un compromis entre le temps de calcul (application en temps réel), l'architecture embarquée (espace mémoire) et une reconstruction tridimensionnelle assez riche (type d'application).

2.2 Segmentation d'image :

2.2.1 Introduction :

La segmentation de l'image est à la base de tout système de vision. Elle a pour but de détecter et de caractériser par des attributs les zones d'intérêt de l'image. Il s'agit essentiellement des contours et des régions. Dans une image représentant une scène réelle avec un certain nombre d'entités, les contours correspondent généralement aux frontières et arêtes des objets, et les régions à leurs surfaces.

Nous pouvons classer les méthodes de segmentation en deux grandes familles [32]:

- Segmentation par les régions
- Segmentation par points contours

Dans ce présent travail, nous nous intéressons à la segmentation par points contours.

2.2.2 L'extraction des contours

2.2.2.1 Définition :

Les contours correspondent généralement à des changements brusques de propriétés physiques ou géométriques de l'image perçue ; autrement dit le contour correspond à la frontière de l'objet donc aux discontinuités de l'intensité lumineuse.

Il est caractérisé par [36]:

- **La continuité** des points contours d'un objet : Un contour est représenté par une chaîne connexe de points sur l'image.
- **Epaisseur** d'un pixel: les contours représentent les frontières des objets. Il est important d'évaluer avec précision la position du contour de l'objet et de la représenter par un seul point qui correspond à la zone de variation maximale de la luminance.
- **La direction** du contour est orthogonale à celle du gradient. Le contour correspond à la frontière de l'objet et la variation de luminance est perpendiculaire au contour sur l'image.

Il existe plusieurs approches pour l'extraction de contours, nous pouvons les regrouper en deux types d'approche: les approches classiques et les approches analytiques[37].

2.2.2.2 Les approches classiques :

Cette section présente une vue générale sur l'ensemble de méthodes qui ont eu historiquement une grande importance en traitement des images. Bien que certaines soient encore régulièrement employées, nous pouvons classer les approches classiques en deux grands familles : gradient et Laplacien.

- Opérateurs du premier ordre (Approche gradient) :

Soit $I(x, y)$ une image quelconque et $f(x, y)$ la fonction représentant l'intensité lumineuse, on définit le gradient dit aussi approche dérivée première de cette image par :

$$G_x(x, y) = \frac{\partial I(x, y)}{\partial x} \dots\dots\dots 2.1$$

$$G_y(x, y) = \frac{\partial I(x, y)}{\partial y} \dots\dots\dots 2.2$$

On dispose donc de la valeur de gradient en tout point de l'image soit la fonction G. On définit aussi le module et la direction de vecteur gradient par :

$$G = \sqrt{(G_x^2 + G_y^2)} \dots\dots\dots 2.3$$

$$\cong \max(G_x, G_y)$$

$$\cong |G_x| + |G_y|$$

$$\phi = \arctan\left(\frac{G_y}{G_x}\right) \dots\dots\dots 2.4$$

La norme (le module) du gradient est donné par :

$$N(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \dots\dots\dots 2.5$$

Avec :

$G(x, y)$ Le gradient de I au point (x, y) donc $G(x, y) = (G_x(x, y), G_y(x, y))^t$

L'extraction des points de contour s'effectue par plusieurs manières, nous citons [32] :

- i. Par la sélection des points dont la norme du gradient est élevée.
- ii. Extraction des extrema locaux du gradient.
- iii. Seuillage par hystérésis des extrema.

Dans ce contexte, nous trouvons les travaux de plusieurs chercheurs tels que Robert, Prewitt, Sobel, Kirsh etc.

L'équation (2.6) présente les masques de Robert, et (2.7) présente celui de Prewitt et Sobel avec, pour $c=1$, on a les masques de Prewitt et pour $c=2$ les masques de Sobel.

$$H_x = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \dots\dots\dots 2.6$$

$$H_x = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{bmatrix} \dots\dots\dots 2.7$$

- Opérateurs de second ordre (Approche Laplacien)

Dans le cas d'une approche dérivée seconde, on calcule la valeur du Laplacien en chaque point de l'image, soit la fonction Δ . Les points de contour sont localisés au passage par zéro de Laplacien (M est un point de contour si $\Delta M = 0$). Mais généralement l'approximation du Laplacien est assez bruitée donc on détecte les points de contour où ils changent de signe [37].

On définit le Laplacien d'une image $I(x,y)$ par :

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \dots\dots\dots 2.8$$

L'extraction du passage par zéro de Laplacien s'effectue classiquement en trois étapes :

1. Détermination d'une image de polarité : il s'agit de calculer une image I_p telle que :

$$I_p(M) = 0 \quad \text{si } \Delta M > 0$$

$$I_p(M) = 1 \quad \text{si } \Delta M \leq 0$$

2. Détection de passage par zéro : on calcule une image I_z telle que :

$$I_z(M) = 1 \text{ si } M \text{ correspond à une transition 0-1 ou 1-0}$$

dans I_p

$$I_z(M) = 0 \text{ si non}$$

3. Seuillage des passages par zéro : consiste à éliminer les passages par zéro de faible normes de gradient, cela peut s'effectuer par un algorithme de seuillage quelconque, cette étape nécessite le calcul de la norme du gradient au points de passage par zéro.

Un des premiers algorithmes de détection de contours consiste à calculer le Laplacien par convolution de l'image avec un des masques présenté par les équations (2.9), (2.10), (2.11).

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \dots\dots\dots 2.9$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \dots\dots\dots 2.10$$

$$\begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 \end{bmatrix} \dots\dots\dots 2.11$$

2.2.2.3 Les approches analytiques :

Nous allons voir maintenant une approche qui a permis une meilleure compréhension des conditions d'une bonne détection de contours et qui a ainsi conduit à des détecteurs de très bonne qualité [37]. Plus particulièrement, nous trouvons le filtre de Canny, Deriche, Shen et Castan. Dans ce qui suit, nous nous limitons uniquement au filtre de Deriche car c'est celui utilisé dans notre travail.

- Deriche :

L'approche développée par Deriche [36] a consisté en la recherche de l'opérateur optimal sous forme de filtre à Réponse Impulsionnelle Infinie (RII) ($W = \text{inf}$). Il en résulte une équation différentielle admettant comme solution l'opérateur suivant (pour plus de détails le lecteur peut consulter [36] [37]):

$$f(x) = \frac{s}{w} e^{-\alpha|x|} \sin(wx) \dots \dots \dots 2.12$$

Valeur de *alpha* :

Plus α est choisi grand, plus le filtre favorise la localisation au détriment de l'unicité. Ce filtre détecte la moindre variation. Pour α petit, il y a peu de réponse (Moins sensible au bruit) mais une perte en localisation. En pratique, il est d'usage d'utiliser $\alpha = 1$.

2.2.3 Chaînage :

Dans la plus part des applications, la description matricielle des contours fournis par la détection de contours n'est pas suffisante. Il est alors nécessaire de passer à une description sous forme de listes chaînées de contours.

L'objectif du chaînage de contours est de passer d'une description matricielle des contours à une description sous forme de listes de contours chaînées. Cette étape qui semble aisée pose en fait des problèmes difficiles [32] :

- Le temps de calcul : cette étape n'étant que transitoire, il faut éviter des temps de calcul trop prohibitifs;
- La représentation des relations entre les chaînes : les relations entre les chaînes doivent être gérées sans perte d'information par rapport à l'information matricielle ;
- La structuration de l'information : il faut bien structurer l'information pour que les manipulations de données effectuées par la suite soient aisées.

P. Garnesson et G. Giraudon [38] ont mis au point un algorithme de chaînage opérant dans le cas 2D qui résout ces problèmes en permettant :

- Un chaînage rapide en une seule lecture de l'image contours;
- Une gestion efficace des liens de parenté;
- La possibilité d'un stockage étendu pour divers types information (niveau de gris, pixel, amplitude du gradient, etc.).
- La possibilité d'éliminer des chaînes;
- La minimisation du nombre de chaînes;

L'algorithme de chaînage se divise en trois étapes:

- i. Création de la chaîne
- ii. Phase de fusion
- iii. Phase d'élimination suivant un critère de longueur

Le détail de chaque étape de l'algorithme se trouve dans [38].

2.2.4 Segmentation de contours :

Les contours peuvent être partitionnés dans des segments de courbes qui ont une description analytique connue telles des lignes droites et des coniques. Dans ce qui suit nous nous intéressons uniquement aux algorithmes qui nous permettent de passer d'une image contours à une liste de segments de droite.

2.2.4.1 Transformée de Hough :

Il est possible de grouper des points de contour colinéaires sans passer par l'étape de chaînage. On peut utiliser pour cela une méthode qui consiste à faire correspondre à des points de contour de l'image des points d'un espace de paramètres. Cette technique s'appelle «transformée de Hough», du nom de son inventeur. Conceptuellement elle peut s'étendre pour détecter des courbes de forme quelconque. Voici le principe général de cette transformation [32]:

Soit l'équation d'une droite (figure 2.1) $x \sin \phi + y \cos \phi = d$

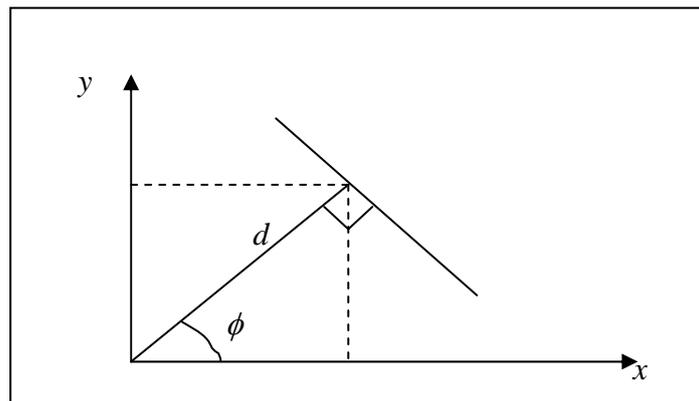


Figure 2.1 : Représentation d'une droite dans l'espace de Hough

Considérons maintenant un point de contour du plan image. Ce point est caractérisé par sa position dans l'image, $[x_i, y_i]$ et par son orientation ϕ_i . On peut donc définir une droite qui passe par ce point et qui a la direction du contour :

$$x_i \sin \phi_i + y_i \cos \phi_i = d_i \dots \dots \dots 2.13$$

A chaque point-contour du plan image correspond donc un point dans le plan $\phi - d$. Soient n points-contour colinéaires. On aura : $\phi_1 = \dots = \phi_n$ et $d_1 = \dots = d_n$. A tous ces points colinéaires il correspond le même point dans le plan des paramètres

(figure 2.2). Autrement dit, une droite significative dans l'image contours se caractérise par un pic dans le plan des paramètres.

Afin d'utiliser pratiquement cette technique nous devons discrétiser le plan des paramètres en $N \times M$ cellules. A chaque point-contour du plan image correspond donc une cellule du plan $\phi - d$. De plus on associe un compteur à chaque cellule. Chaque fois qu'une cellule reçoit un « vote » de l'image on incrémente son compteur. Des points de contour colinéaires voteront pour la même cellule, le compteur correspondant contiendra par conséquent une valeur élevée par rapport au compteur d'une cellule qui ne correspond pas à des points de contour colinéaires.

La détection de droite dans une image par la transformée de Hough est l'approche adoptée par Djekoune dans [39].

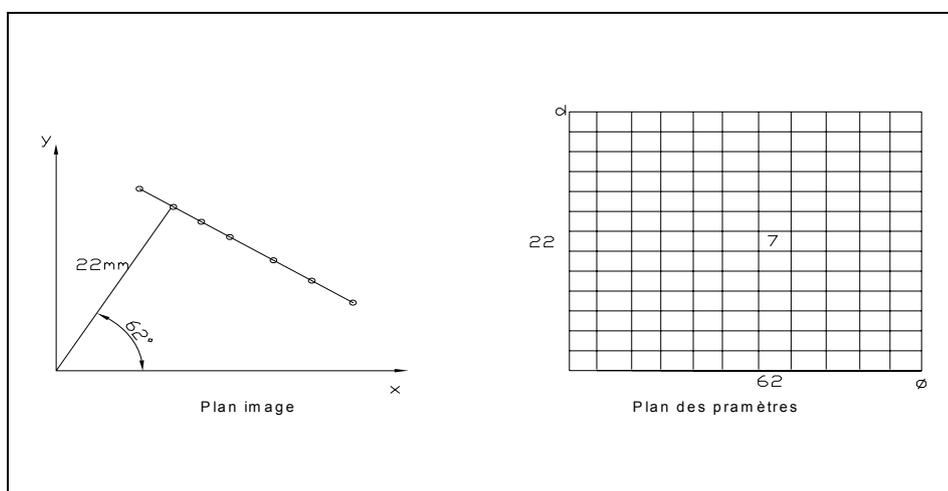


Figure 2.2 : Détection de droites avec la transformée de Hough.

Problèmes liés à la transformée de Hough

Le principal problème rencontré lorsque on utilise la transformée de Hough est le pas de discrétisation du plan des paramètres. En effet un pas trop petit risque de rendre l'exécution trop lente, de même que certains segments peuvent ne pas être détectés à cause de la dispersion de leurs votes dans des cellules voisines. A l'inverse, un pas trop grand peut conduire à la fusion des votes correspondant en réalité à des segments dont les paramètres sont proches. Il y a alors apparition de fausses détections. Il faut donc réaliser un compromis entre ces deux effets.

Un autre inconvénient de cette transformation est sa demande importante de l'espace mémoire, plus elle est très coûteuse en matière de temps de calcul.

2.2.4.2 L'approximation polygonale :

a. Pourquoi ?

Passer d'une image brute à ses chaînes de contours permet une première compression de l'information. Typiquement, on passe de 512x512 valeurs d'intensité lumineuse, soit plus d'un quart de million de valeurs, à quelque centaines de chaînes comportant en tout moins de dix mille points (soit moins de vingt mille coordonnées) . Le taux de compression est donc voisin de 1/12. De plus on a réalisé une structuration de l'image qui fait ressortir l'information pertinente pour notre application [29].

Une compression supplémentaire et une nouvelle structuration sont obtenues par une approximation polygonale de chaque chaîne de contours. On exploite toutes les portions de contours quasi-linéaires :

- On comprime les données, puisqu'on ne conserve que les extrémités d'un segment, soit quatre coordonnées, typiquement il ne reste que quelques centaines de segments ;
- Manipuler des formes géométriques simples

b. Approximation polygonale par découpage récursif :

Le principe de l'algorithme de découpages récursifs [40] est de relier le premier et le dernier point de la chaîne, P_1 , P_N respectivement, par un segment de droite, et de calculer la distance maximale noter e_m qui sépare la chaîne de contours du segment de droite, qui correspond au point q_1 (figure 2.3). Si cette distance est inférieure à un seuil e , on a terminé (la chaîne P_1 , P_N est approximé par la droite $[P_1P_N]$); sinon on découpe la chaîne initiale en deux sous chaînes au point où la distance e_m est atteinte, et on réitère l'algorithme précédent indépendamment sur chacune des deux nouvelles chaînes.

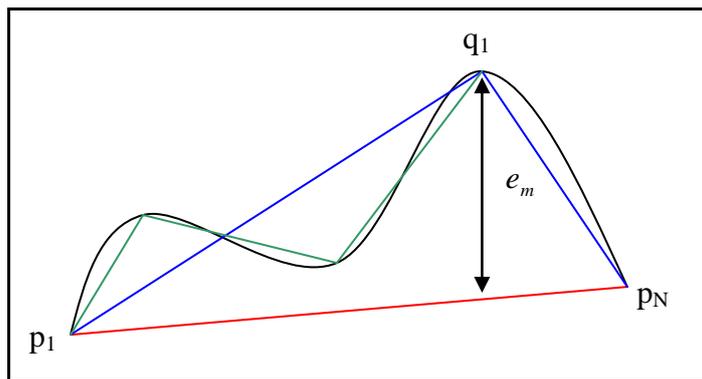


Figure 2.3 : Principe de l'approximation polygonale par découpage récursif

Calcul de la distance e_m

Soit l'exemple de la figure 2.4:

$$A = (x_2 - x_1, y_2 - y_1)$$

$$B = (x_3 - x_1, y_3 - y_1)$$

$$A \cdot B = |A| \cdot |B| \cos \theta$$

$$\theta = \arccos \left(\frac{A \cdot B}{|A| \cdot |B|} \right)$$

$$e_m = |B| \cdot \sin \theta$$

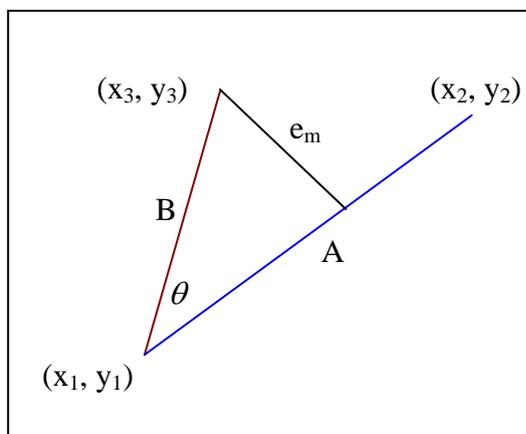


Figure 2.4 : Calcul de la distance e_m

Algorithme de découpage récursif

Soient P_i, P_j les deux extrémités de la chaîne C_i

Début

 Pour toutes les chaînes C_i faire :

 Parcourir les points de P_{i+1} à P_{j-1} pour calculer l'écart maximal à la droite $[P_i, P_j]$ noté e_{\max}

 Si $e_{\max} < \text{seuil}$

 La chaîne C_i est bien approximé par la droite $[P_i, P_j]$

 Si non

 Soit le point P_k , tel que $i < k < j$, P_k correspond à l'écart e_{\max}

 Découpage (P_i, P_k)

 Inscrire P_k comme nouveau sommet de l'approximation

 Découpage (P_k, P_j)

 Fin si

Fin

Remarque :

Pour une chaîne fermée on choisit les deux points les plus éloignés de la courbe: P_1 et P_N pour construire le vecteur p_1p_N ; on divise la courbe fermée en deux parties sur lesquelles on applique récursivement la procédure de division précédente.

c. Approximation polygonale par fusion :

Le principe des méthodes par fusion [41] consiste à parcourir la chaîne du premier point jusqu'au dernier (figure 2.5). Au fur et à mesure du parcours, les points rencontrés sont fusionnés en un même segment tant que l'ensemble de ces points vérifie un certain critère. Le dernier point vérifiant le critère correspond à l'extrémité du segment. On recommence ce processus à partir de ce point.

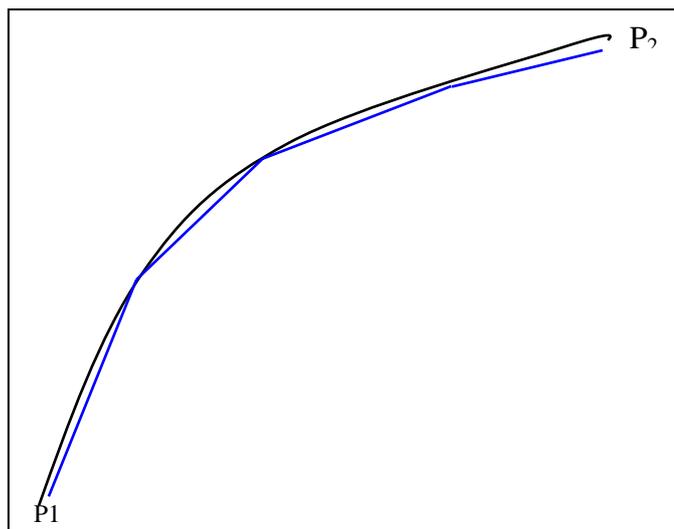


Figure 2.5 : Principe de l'approximation polygonale par fusion

Algorithme de fusion

Soient P_1, P_2 les deux extrémités de la chaîne C_i

Début

Pour toutes les chaînes C_i faire :

Parcourir linéairement la chaîne C_i de P_1 à P_2 , lors de parcours, on mesure l'écart entre le point P_i et le segment $[P_1 P_i]$ noté e

Si $e > \text{seuil}$

Soit le point P_k , tel que $i < k < j$, P_k correspondant à l'écart e

La portion $P_1 P_k$ de chaîne C_i est approximé par la droite $[P_1 P_k]$

Inscrire P_k comme nouveau sommet de l'approximation (nouveau point de départ)

Fusion (P_k, P_2)

Fin si

Fin

La figure 2.6 présente le résultat de chaînage d'une scène de bureau sur laquelle on applique les deux algorithmes vus ci-dessus de l'approximation polygonale par découpage récursive et par fusion, les résultats sont sur la figure 2.7 et 2.8 respectivement



Figure 2.6 : Les chaînes initiales d'une scène de bureau.

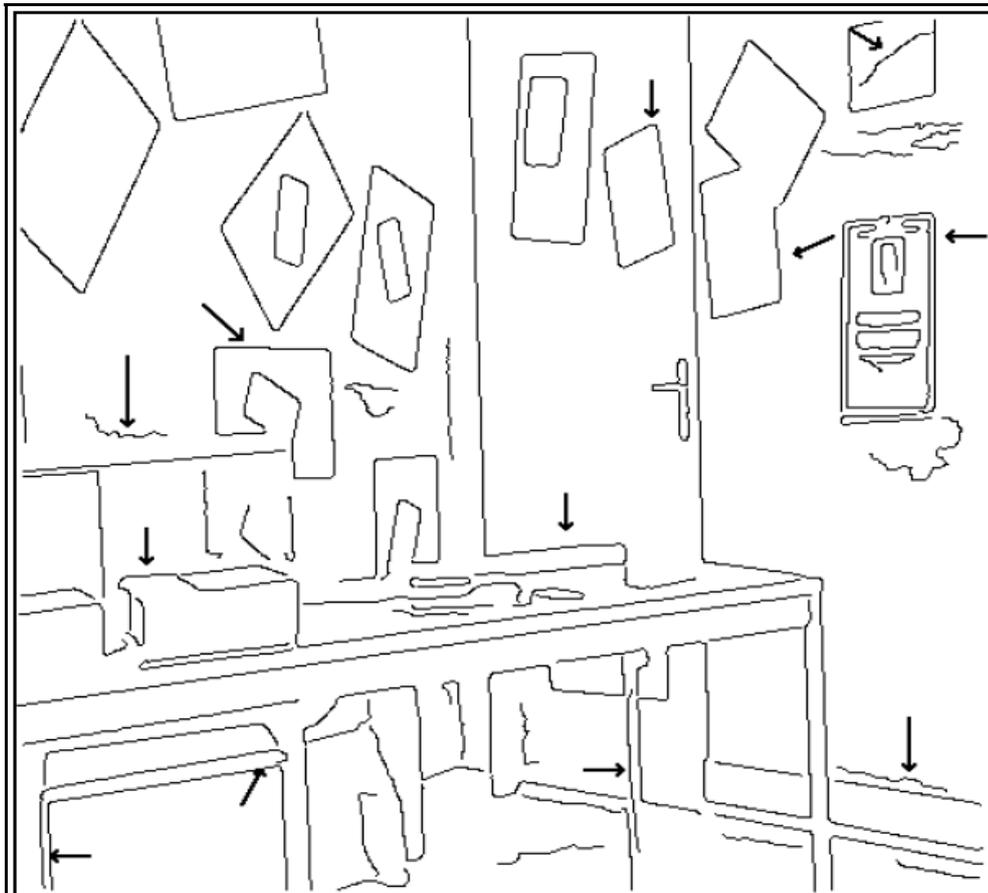


Figure 2.7: Approximation avec la méthode de découpage récursive l'écart $e_m = 3$

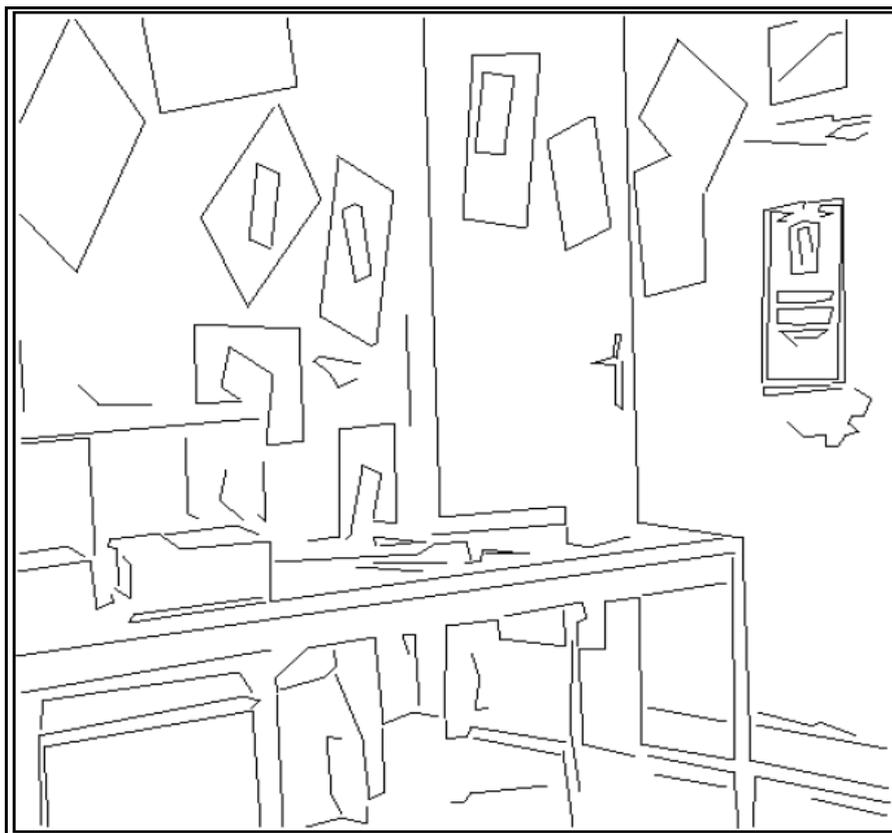


Figure 2.8 : Approximation avec la méthode de fusion selon l'écart $e_m = 3$.

2.2.4.3 Discussion :

Ces deux méthodes ont été largement utilisées. Deux raisons justifient cet « engouement », d'une part ils sont simples à mettre en oeuvre et d'autre part permettent de contrôler l'écart maximal entre la chaîne et les segments.

D'un point de vue complexité, l'algorithme de découpage récursif est plus coûteux que les méthodes par fusion, aussi bien du point de vue espace mémoire (l'algorithme de découpage récursif oblige à stocker les n points de la chaîne); que de point de vue complexité (sa nature récursive nécessite une pile dont la taille est bornée par $2 \cdot \log 2(n)$ [42] [43]).

Dans certains travaux, nous trouvons que la méthode de découpage est suivie par une étape de fusion. Dans cette optique, l'approche de Pavlidis dans [43], est intéressante, il fait coopérer trois étapes: une étape de découpage, une étape de fusion et une étape d'ajustement des points de coupure. Cependant, et bien évidemment le temps de calcul et mémoire utilisée sera plus importante.

2.3 Conclusion :

La segmentation d'images constitue un élément clé entre traitement et analyse de l'image. Elle permet de passer du support de l'image à une liste d'objets décrits en terme de primitives ayant des attributs géométriques.

Le type de primitive image utilisée : segments de droite, dont nous avons justifié le choix par comparaison avec les autres primitives images les plus fréquemment employées.

Dans la deuxième partie de chapitre, nous avons donné une vue générale des deux approches de segmentation d'image en contours, à savoir les approches classique et celle dit analytiques.

Nous avons exposé, par la suite, les techniques de segmentation des contours appelés souvent traitement de bas niveau. Rappelons que dans notre travail nous avons adopté : pour la phase d'extraction de contours l'algorithme de Deriche, l'algorithme de Giraudon pour la phase de chaînage, et l'algorithme de l'approximation polygonale par fusion pour de l'approximation des chaînes par des segments de droite.

CHAPITRE 3

LA MISE EN CORRESPONDANCE

Dans ce chapitre, nous traiterons le problème crucial dans la vision stéréoscopique qui est l'appariement d'images. Après une brève définition, nous parlerons des difficultés liées à l'appariement, puis les contraintes souvent utilisées pour les surmonter. Ce chapitre décrit les méthodes d'appariement à base des primitives et celle dites les méthodes surfaciques.

A la fin de ce chapitre, nous détaillerons la méthode d'appariement développée en utilisant la primitive segment de droite.

3.1 Introduction :

La mise en correspondance est le problème capital dans le domaine de la vision stéréoscopique. En effet, c'est l'une des étapes les plus importantes, car la qualité de la reconstruction 3D dépend directement de la qualité de l'appariement.

C'est pour cette raison que beaucoup de chercheurs se sont intéressés à ce sujet durant les deux dernières décennies. En effet, malgré ces efforts, la mise en correspondance demeure un exercice difficile et un sujet de recherche toujours ouvert [44]. Nous pouvons dire et sans douter qu'il n'existe pas actuellement une méthode de mise en correspondance robuste, précise et suffisamment générale pour qu'elle puisse s'appliquer à une paire d'images indépendamment du contenu. Cependant le choix d'une méthode d'appariement dépend fortement du type de l'image (images aériennes, scène d'intérieur, scène d'extérieur, image médicale, etc.), et le type de primitive choisi.

3.2 Définition :

Considérons deux images 1(ou droite D) et 2 (ou gauche G) d'une scène, la mise en correspondance ou l'appariement s'agit de déterminer, pour un élément de l'image D, l'élément qui lui correspond dans l'image G. C'est un processus

intermédiaire entre les processus dit de *haut niveaux*: reconstruction, reconnaissance, etc., et ceux de *bas niveaux* : extraction d'indices.

En effet, le calcul de l'appariement stéréoscopique, en général, repose sur l'introduction de la mesure de similarité (ou ressemblance) entre une primitive de l'image gauche et une primitive de l'image droite.

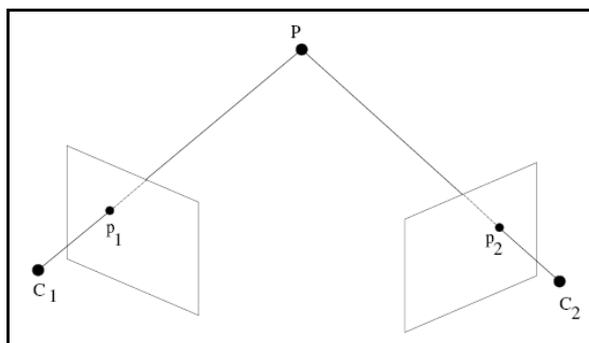


Figure 3.1 : Deux pixels P_1 et P_2 correspondants.

3.3 Problèmes liés à la mise en correspondance :

La mise en correspondance est, en général, ambiguë à cause des phénomènes physiques qui existent dans la nature de la scène elle-même ou bien, à des problèmes dans la chaîne d'acquisition; dans ce qui suit nous présentons les problèmes les plus importantes qui rendent l'appariement un problème difficile à résoudre.

3.3.1 Le bruit de l'image :

Le premier élément perturbateur est le bruit [22], il est l'accumulation de plusieurs sources d'erreurs, par exemple l'erreur provient des éléments optiques et électroniques utilisés lors de l'acquisition, donc il peut être due à l'échantillonnage (on a seulement les niveaux de gris sur les points des coordonnées entières), de la quantification (les niveaux de gris est généralement représenté sur 8 bits), etc.

3.3.2 Variation photométrique :

Une seconde catégorie d'éléments pouvant nuire à la mise en correspondance est liée à l'éclairage. Entre deux vues, les conditions d'éclairage ne sont pas identiques, surtout si les deux images ne sont pas prises aux même temps (cas de mono-caméra). De plus, les surfaces éclairées des objets ne réfléchissent pas

uniformément la lumière reçue lorsqu'ils ne sont pas lambertiennes. Un autre facteur important est l'ombre des objets qui dépend directement de l'éclairage de la scène.

3.3.3 Distorsion perspective :

L'image d'un objet peut être géométriquement déformée dans l'une des images, ce problème est important quand l'angle entre deux vues est grand.

3.3.4 Répétition ou absence de textures :

Dans une image des textures semblables peuvent exister (image texturée), ce qui engendre des ambiguïtés pour l'appariement.

Il existe aussi des images avec des zones uniformes (image non texturée), il est très difficile de déterminer quel point dans une image doit être apparié avec un point dans l'autre image, car ils sont tous localement identiques [22].

3.3.5 Les occultations :

Les principaux problèmes de la mise en correspondance proviennent des occultations, c'est-à-dire, des zones que l'on aperçoit dans une image mais qui sont cachées dans la seconde ; autrement dit, un point dans une image n'apparaît pas forcément dans l'autre ; soit parce qu'il est caché par des obstacles, ou par ce que sa projection sur l'autre image est en dehors du support de l'image. La figure 3.2 montre ce problème [45].



Figure 3.2 : Exemple du problème d'occultation dans une paire d'images stéréoscopique

3.4 Les contraintes de la mise en correspondance :

3.4.1 Contrainte Lambertienne :

Les variations de l'intensité lumineuse reçues par le capteur (caméra) permettent de percevoir les objets de la scène. La puissance de cette intensité reçue est directement proportionnelle à la puissance émise par la scène à l'endroit où se projette l'objet observé. Pour la mise en correspondance, il est important que les intensités reçues pour un point 3D de la scène soient identiques dans toutes les images. Pour que les images d'un point 3D aient la même intensité, la surface de l'objet étudié doit satisfaire la contrainte lambertienne [46]. Cette contrainte suppose que les surfaces doivent, en chaque point, être lambertiennes, c'est-à-dire que la luminosité réfléchie ne dépend pas de la position d'observation. Ceci est rarement vrai. Dans la pratique, nous appliquons la contrainte quasi lambertienne, qui dit que les correspondants d'un point doivent avoir un niveau de gris, ou une composante de couleur, borné autour du niveau de gris, ou de composante de couleur, du point initial. Les bornes de cet intervalle dépendent des conditions de prise des images.

3.4.2 Contrainte épipolaire :

Cette contrainte réduit considérablement l'espace de recherche des correspondants, de l'image entière (recherche bidimensionnelle) à la droite épipolaire (recherche monodimensionnelle). Un point donné appartenant à une ligne épipolaire Ep_1 (figure 3.3) trouve son homologue sur la ligne épipolaire correspondante Ep_2 dans l'autre image. Rappelons que les lignes épipolaires sont définies comme étant les droites d'intersection du plan épipolaire C_1PC_2 avec les plans images [47].

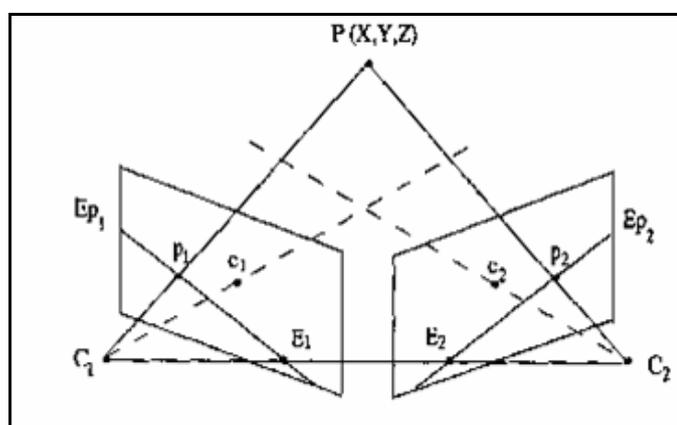


Figure 3.3 : Géométrie de la stéréovision binoculaire

3.4.3 Contrainte d'orientation :

La contrainte d'orientation s'applique surtout sur les primitives images de type points de contour (l'orientation de la droite tangente en ce point) et segments de droite [32].

3.4.4 Contrainte d'ordre :

D'une façon simple, nous pouvons décrire la contrainte d'ordre comme suit : soit P_1, P_2 deux points de l'espace et soit p_1, p_2 et p_1', p_2' leurs projections respectives dans l'image de gauche et de droite. Si le point p_1 se trouve à gauche d'un point p_2 dans l'image droite, [32] alors cette relation mutuelle existe dans l'autre image : p_1' est à gauche de p_2' (figure 3.4).

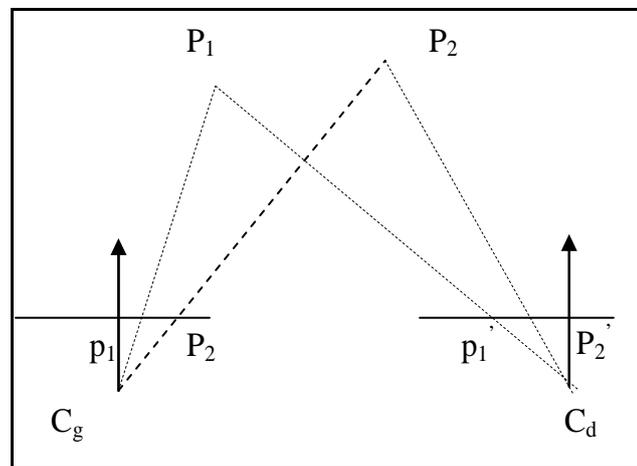


Figure 3.4: Contrainte d'ordre.

La seule configuration pour laquelle la contrainte de l'ordre n'est pas satisfaite, est le cas de surfaces d'objets fortement inclinées par rapport au plan des images. Cette situation est caractérisée par un ordre différent des points images sur les droites épipolaires. La figure 3.5 illustre bien ce cas.

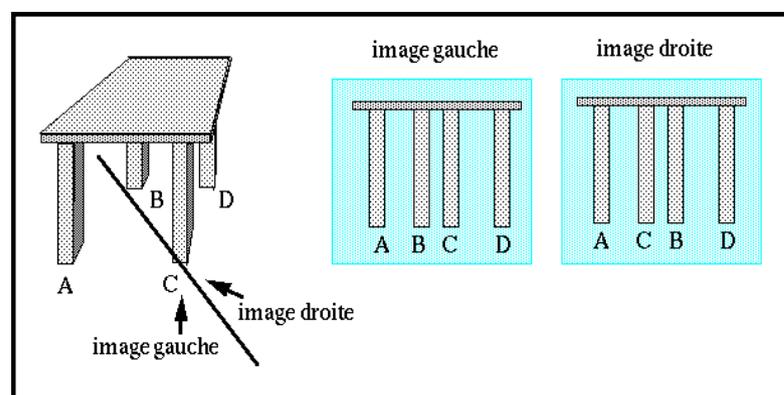


Figure 3.5 : Cas où la contrainte d'ordre n'est pas respectée.

3.4.5 Contrainte d'unicité :

La contrainte d'unicité garantit qu'il y a au plus un appariement pour chaque pixel, et elle autorise l'existence de non appariement. Autrement dit la contrainte d'unicité impose qu'une primitive de l'image gauche a au plus une correspondante dans l'image droite. Cela suppose qu'une primitive est une projection d'une unique entité 3D [48].

$$\sum_{g=0}^{g=N-1} M_{g,d} = 0,1 \quad \text{et} \quad \sum_{d=0}^{d=N-1} M_{d,g} = 0,1 \dots \dots \dots 3.1$$

Cette règle est mise en défaut dans le cas où plusieurs points 3D sont alignés (figure 3.5). On peut justifier partiellement la contrainte d'unicité en supposant l'opacité des points P_i , ce qui conduit à ne considérer lors d'un alignement avec le centre optique que le point physique le plus proche de la caméra ; c'est-à-dire P_2 dans l'exemple de la figure 3.6.

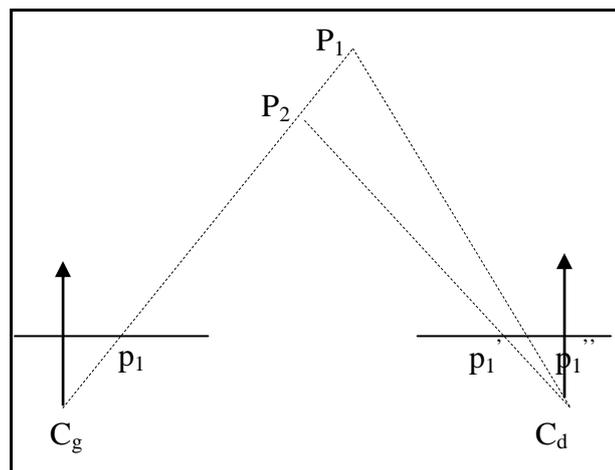


Figure 3.6 : Cas où la contrainte d'unicité n'est pas respectée.

3.4.6 Contrainte de continuité :

Plaçons nous dans le cas où la scène est formée d'objets physiques dont on peut approximer la structure spatiale par des ensembles de primitives géométriques correspondant aux indices visuels extraits de chaque image. Si on suppose que la surface des objets est suffisamment régulière, et que la densité des primitives géométriques est suffisamment grande, les objets seront représentés par des sous-ensembles de primitives voisines dans l'espace que l'on peut parcourir de manière régulière (presque continue)[48].

La contrainte de continuité repose sur ce principe : une courbe continue de l'espace se projette en des courbes continues dans les deux images. Les points

appartenant à un contour de l'image de gauche ont été mis en correspondance avec des points de l'image de droite qui n'appartiennent pas nécessairement au même contour. Ce type d'erreur peut être simplement corrigé en imposant aux points de l'image de droite d'appartenir au contour possèdent déjà une majorité de points qui ont été mis en correspondance avec les points d'un contour de l'image de gauche.

3.4.7 Contrainte de voisinage :

La contrainte de voisinage prend en compte la configuration relative des points d'intérêt. Ainsi, à chaque point P_j dans une image sont associés les p points d'intérêt les plus proches. Un point d'une image est alors représenté par son vecteur d'invariants (ou vecteurs d'attribut) et ces p points voisins [51]:

$$(\vec{v}_j, P_{j,1}, \dots, P_{j,p})$$

Si le vecteur de caractéristiques \vec{v}_j issu de l'image est apparié avec le vecteur \vec{v}_k d'une seconde image, tout ou partie des p points d'intérêt voisins de P_j doivent correspondre à des voisins de P_k . Ceci est illustré par la figure 3.6 et s'écrit formellement :

$$\vec{v}_{j,m} \text{ et } \vec{v}_{k,n} \text{ correspondent pour quelques } (m,n)$$

Ou

$$\vec{v}_{j,m} \text{ vecteur de } P_{j,m} \text{ et } P_{j,m} \text{ un des } p\text{-voisins de } P_j$$

$$\text{Et } \vec{v}_{k,n} \text{ vecteur de } P_{k,n} \text{ et } P_{k,n} \text{ un des } q\text{-voisins de } P_k$$

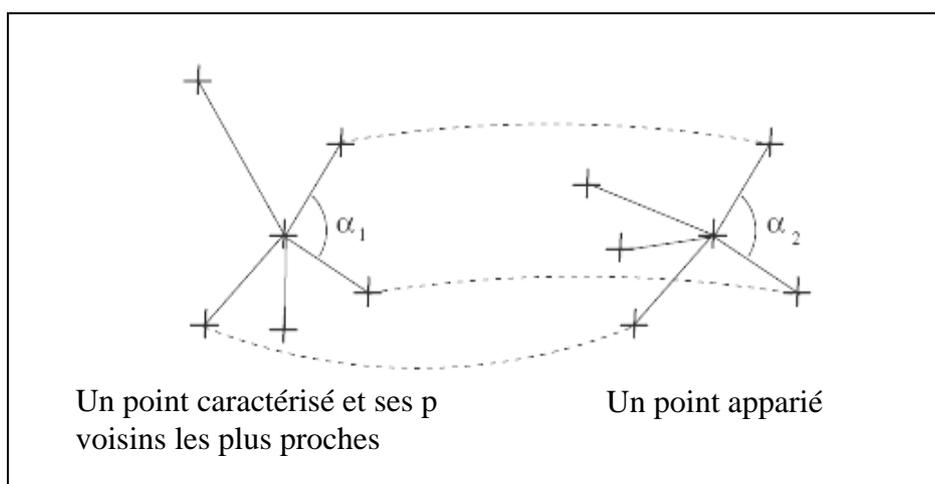


Figure 3.7 : Contrainte de voisinage

Dans les travaux de [51] la contrainte de voisinage est ajoutée au processus de mise en correspondance en stockant les p voisins les plus proches d'un point. Durant la phase d'appariement, il est imposé qu'au moins 50% de ces voisins se correspondent pour que la correspondance soit validée.

Dans le cadre de notre travail la contrainte de voisinage est exploitée dans l'algorithme de mise en correspondance des segments de droite ; nous détaillons plus loin cette exploitation.

3.5 Approches principales d'appariement :

3.5.1 Méthodes surfaciques ou dense appariement (dense stereo matching) :

Les méthodes surfaciques cherchant à mettre en correspondance les deux images pixel à pixel, sans une phase d'extraction particulière. En effet, ces méthodes exploitent la constatation que deux images sont localement semblables autour des points appariés (section), puisque la projection de la scène 3D sur l'image aux voisinages de ces points est similaire, d'où la dénomination surfacique [47].

De manière générale, ces méthodes s'appuient pour la recherche d'une correspondance sur l'appariement par corrélation, qui se déroule de la façon suivante (figure 3.8). Si l'on cherche le correspondant d'un point m de l'image G , on définit tout d'abord une zone de recherche dans l'image D . On utilise alors une fenêtre fixe de corrélation F_1 dans l'image G centrée sur le point m , ainsi qu'une fenêtre glissante F_2 dans l'image D qui va parcourir la zone de recherche. Pour chaque position de F_2 un score de corrélation $C(x, y)$ est calculé entre F_1 et F_2 . L'ensemble des scores calculés sur la zone de recherche définit la surface des scores de corrélation associée au point m . le point m' choisi sera alors celui pour lequel le score de similarité sera le plus grand possible.

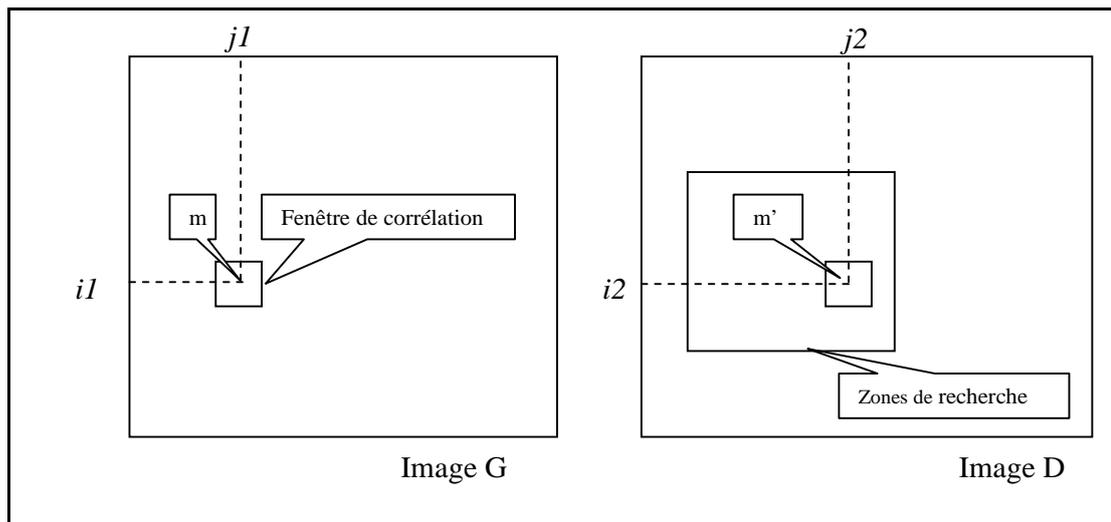


Figure 3.8 : recherche d'un correspondance par corrélation

Plusieurs expressions de C sont disponibles dans la littérature [22]. En voici deux qui donnent de très bons résultats:

$$C(x, y, d) = 1 - \frac{\sum_{i,j} I_1(x+i, y+j) \cdot I_2(x+i+d, y+j)}{\sqrt{\sum_{i,j} I_1^2(x+i, y+j)} \cdot \sqrt{\sum_{i,j} I_2^2(x+i+d, y+j)}} \dots\dots\dots 3.2$$

$$C(x, y, d) = \frac{\sum_{i,j} \left[\left(I_1(x+i, y+j) - I_1(x, y) \right) - \left(I_2(x+i+d, y+j) - I_2(x+d, y) \right) \right]^2}{\sqrt{\sum_{i,j} \left(I_1(x+i, y+j) - \bar{I}_1(x, y) \right)^2} \cdot \sqrt{\sum_{i,j} \left(I_2(x+i+d, y+j) - \bar{I}_2(x+d, y) \right)^2}} \dots\dots\dots 3.3$$

De nombreux travaux de reconstruction à base des méthodes surfaciques aient été menés à bien, par exemple, les modèles numériques de terrain calculés à partir de photographies aériennes ou spatiales sont réalisés couramment et avec un grand succès [22].

Bien que les méthodes surfaciques ont l'avantage, par rapport aux méthodes à base de primitives, de produire une représentation 3D dense de la scène, le temps de calcul important pris par le CPU, reste l'un des principaux problèmes de l'appariement dense et par conséquent, il est impossible d'utiliser ces méthodes pour des applications en temps réel. De plus les méthodes d'appariement par corrélations

ne s'adaptent pas à la stéréovision car elles dépendent de l'intensité lumineuse de l'image, attribut qui varie selon l'angle de prise de vue.

3.5.2 Méthodes à base de primitives (feature-based stereo matching)

Comme nous avons montré dans le chapitre précédent, la primitive est une représentation fidèle et utile du contenu d'une image. Les primitives ont en général les propriétés suivantes : unicité, répétitivité et sens physique. Dans le contexte de la stéréoscopie, le but des approches d'appariement par primitives est d'obtenir des correspondances correctes, même en présence des différents bruits cités précédemment. Dans le cas où l'on dispose de données de plus haut niveau extraites des images, telles que des points de contour ou des segments de droite, le problème de la mise en correspondance est légèrement différent. En effet, les données de type primitives sont moins nombreuses et plus riches en information que des données de type pixel sélectionnées dans les images. Ces primitives sont plus fiables et possèdent en particulier des attributs géométriques qui peuvent être utilisés pour l'appariement.

Dans le domaine de la vision par ordinateur, et plus particulièrement dans la vision robotique, nous trouvons souvent des approches d'appariement à base de primitives.

Le choix de la méthode de mise en correspondance dépend de la primitive choisie, autrement dit « comment apparier » dépend de « quoi apparier » [7]. Cependant, nous citons:

- Appariements de points contours (edge-based stereo matching)
- Appariements de régions (region-based stereo matching)
- Appariements de primitives géométriques (feature-based stereo matching)

Dans la littérature, on trouve plusieurs approches de mise en correspondance utilisées pour l'appariement des primitives, nous citerons dans ce qui suit les plus souvent utilisées :

3.5.2.1 Mise en correspondance hiérarchique :

La méthode mise en correspondance hiérarchique consiste à construire deux pyramides d'images [32]. Chaque pyramide contient l'image initiale de taille $N \times M$ ainsi que des images de tailles $\frac{N}{2} \times \frac{M}{2}$, $\frac{N}{4} \times \frac{M}{4}$, $\frac{N}{8} \times \frac{M}{8}$ et $\frac{N}{16} \times \frac{M}{16}$.

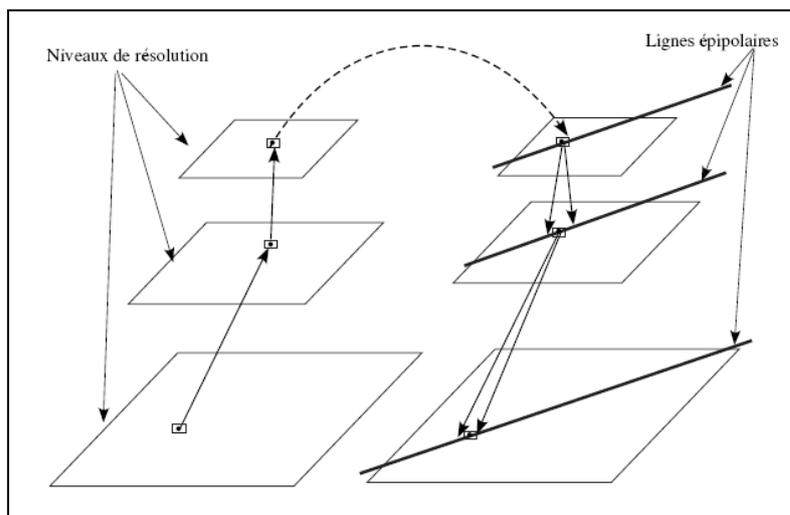


Figure 3.9 : Algorithme de mise en correspondance hiérarchique

Chaque pyramide a donc, dans ce cas, cinq niveaux de précision du plus fin (ou haute résolution) au plus grossier (ou basse résolution). La construction de la pyramide est très simple : on réduit une image en calculant la moyenne des pixels dans un carré de deux pixels sur deux pixels. Ces quatre pixels sont remplacés par un seul pixel (la moyenne) dans l'image de résolution inférieure, et ainsi de suite. On peut remplacer ce calcul de moyenne par des techniques plus sophistiquées.

Soit un point p de l'image de gauche haute résolution, pour chaque image de la pyramide de gauche on calcule successivement les coordonnées de ce point, jusqu'à l'image de basse résolution [48]. On s'intéresse ensuite à la recherche d'une correspondance au niveau de résolution le plus faible. On utilise le produit de corrélation comme mesure de ressemblance.

La prochaine étape consiste à parcourir la pyramide droite de l'image basse résolution vers l'image haute résolution pour trouver le point p' correspondant à p .

3.5.2.2 Mise en correspondance par programmation dynamique :

La programmation dynamique est une technique utile pour appairer deux séquences tout en respectant l'ordre des éléments à l'intérieur de chaque séquence [44]. En effet, si on suppose que la scène observée contient peu de surfaces transparentes, l'ordre des projections est le même dans les deux images : Soient par exemple les séquences {a1, a2, a3, a4} et {b1, b2, b3, b4}. Si l'élément a2 est en correspondance avec l'élément b2 alors les éléments à droite de a2 ne pourront être appariés qu'avec les éléments à droite de b2. L'algorithme cherche alors le meilleur appariement entre les deux séquences parmi tous les appariements possibles. Pour trouver ce meilleur appariement, on ajoute une autre contrainte qui est la contrainte d'unicité. Au final, on calcule, pour toutes les combinaisons d'appariements valides, une mesure de ressemblance et on conserve la combinaison de séquence qui obtient le plus grand score de mise en correspondance.

3.5.2.3 Mise en correspondance par relaxation :

Le problème de mise en correspondance peut être aussi vu comme un problème d'étiquetage : aux primitives d'une image, on associe les primitives de l'autre image [51].

On est alors amené à faire coopérer les primitives d'une image entre eux en mettant en oeuvre des relations découlant des contraintes géométriques et où physiques. La mise en correspondance de ces points dépend du point considéré ainsi que de son voisinage.

L'algorithme se déroule en deux phases [44]:

- Un ensemble d'étiquettes (candidats à l'appariement) est sélectionné pour chaque point et une mesure de confiance est ensuite associée à chaque étiquette ;
- Les différentes étiquettes (avec leurs confiances) possibles pour chaque point sont comparées avec celles des points voisins. Cette comparaison est faite sur la base des relations entre les points voisins. Les étiquettes ainsi que leurs confiances sont ensuite modifiées de façon itérative pour réduire les incohérences d'étiquetage entre points voisins.

Le terme de relaxation vient du fait que, pour chaque itération, on cherche une combinaison entre les étiquettes qui minimise un critère calculé sur l'ensemble des étiquettes.

On obtient ainsi, pour chaque itération, la meilleure mise en correspondance possible en prenant en compte l'ensemble des points candidats à ce niveau d'itération.

3.5.2.4 Mise en correspondance par isomorphisme de graphe :

Le problème de mise en correspondance peut être aussi vu comme un problème d'isomorphisme de graphes. On effectue l'ensemble de primitives extraites d'une image peut constituer l'ensemble des nœuds d'un graphe. Des relations entre ces primitives constitueront alors les arcs (ou arêtes) de ce graphe. C'est l'approche qui a été suivie par Skordas dans [51].

Dans le cadre de cette méthode les primitives à mettre en correspondance sont des segments de droite et les relations entre ces segments sont :

- i. à gauche de,
- ii. à droite de,
- iii. Colinéaire avec,
- iv. A une jonction commune avec.

Chaque image est donc décrite par un réseau de segments de droite et de relations entre ces segments. Le problème d'appariement est le problème de trouver le meilleur appariement entre les nœuds et les arcs de ces deux graphes.

3.5.2.5 Mise en correspondance par invariants :

Le principe de ces méthodes repose sur le fait que chaque point réel s'étant projeté dans les plans images possède des caractéristiques invariantes (par exemple l'intensité lumineuse) [53]. On peut ainsi caractériser chaque point de chaque image par une liste d'invariants et, par l'intermédiaire des contraintes citées plus haut, on cherche à mettre en correspondance les points qui offrent des listes d'invariants similaires.

Comme, la méthode par invariant repose sur les attributs associés à la primitive extraite de deux images, elle est souvent utilisée pour l'appariement des primitives géométriques ou des régions, car ils sont dotés de plusieurs attributs invariants par

exemple la longueur dans le cas des segments; la forme, centre de gravité, surface, etc. dans le cas des régions, ou une forme géométrique quelconque.

3.6 L'appariement des segments de droite (approche adoptée) :

3.6.1 Pourquoi appairier des segments :

Pourquoi appairier des segments de droite au lieu d'autres indices visuels ? La réponse a été donnée dans la partie choix de primitives du chapitre précédent, et elle peut être résumée en quatre points:

- Réduire la complexité en réduisant le nombre d'appariements à réaliser. Il y a toujours moins de segments que de points.
- Prendre en compte de manière explicite la continuité des contours. Lorsque deux points sont appariés, il est très probable que leurs voisins le soient également.
- Les attributs géométriques mesurés sur les segments de contour sont plus riches et donc plus discriminants que ceux mesurable sur les points. Ils peuvent donc contraindre d'avantage l'appariement.
- La position et l'orientation d'un segment se mesure généralement avec plus de précision que la position d'un point contour. La reconstruction 3D n'en est que plus précise.

3.6.2 Principe de l'algorithme :

Le principe de notre algorithme s'intègre dans la méthode d'appariement par relaxation, son principe est le suivant :

On établit, pour l'initialisation de l'algorithme, tous les appariements potentiels entre les primitives (segments de droite) qui sont à l'intérieur de la fenêtre de recherche F_i et on attribue à chaque couple (S_g, S_d) une valeur de vraisemblance Q_{ij} . L'étape qui suit permet de changer ces valeurs.

En se basant sur le fait que deux appariements peuvent s'exclure mutuellement (contrainte d'unicité), on peut effectuer plusieurs itérations en cherchant les « réactions mutuelles » pendant lesquels Q_{ij} change. Il est modifié de la façon suivante :

- Si les appariements confirmant (S_g, S_d) ont une bonne vraisemblance : Q_{ij} augmente.
- Si les appariements infirmant (S_g, S_d) ont une mauvaise vraisemblance : Q_{ij} diminue.

Au bout d'un certain nombre d'itérations, l'algorithme s'arrête et les valeurs de vraisemblance sont converties en vrai ou faux pour l'interprétation finale.

3.6.3 La fonction d'appariement :

Nous disposons dans les images G et D de deux ensembles de segments notés respectivement $\zeta = \{S_i, i = 1, \dots, m\}$ et $\zeta' = \{S'_j, j = 1, \dots, n\}$. On définit une fonction d'appariement F de la manière suivante [29] :

$$F \text{ de } \zeta \times \zeta' \xrightarrow{F} \{0,1\}$$

Tel que l'ensemble $F^{-1}(1)$ forme l'ensemble des couples de segments appariés.

La fonction F doit apparier tous les segments homologues, c'est-à-dire tous les segments qui sont l'image d'un même segment 3D de la scène. Sa qualité se mesure

- Au nombre de segments appariés, qui doit être maximum.
- Au pourcentage d'erreurs, qui doit être minimum.

3.6.4 Difficultés :

Le problème est un peu plus compliqué que pour des points, car la fragilité de l'algorithme de segmentation, plus exactement l'extraction de segments de droite, fait qu'une courbe de la scène peut être segmentée de façon différente entre les deux images droite et gauche (figure 3.10), c'est un problème que nous avons rencontré avec une paire d'images.

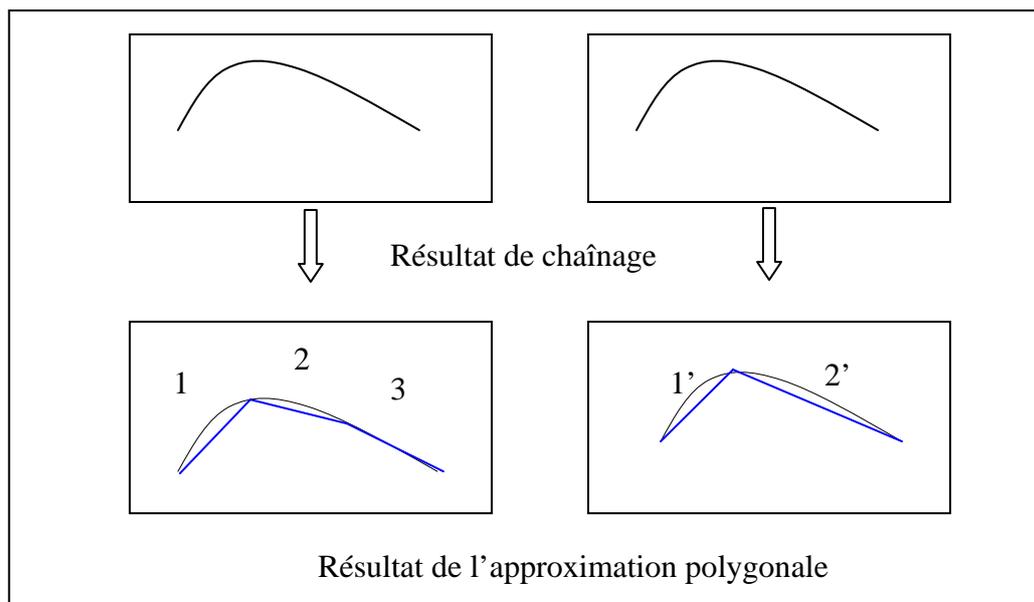


Figure 3.10 : Différence de segmentation entre image gauche et droite

3.6.5 Calcul de voisinage :

Nous définissons le voisinage d'un segment de droite de la manière suivante [29]:

Le voisinage v_j d'un segment S_j est l'ensemble $\{S_k\}$ des segments qui intersectent une cellule commune avec S_j . Cette liste (les segments voisins), ainsi calculée est ajoutée comme attribut local à la description de chaque segment.

Sachant que nous avons utilisé des cellules de forme rectangulaire C_i pour structurer l'image. Ces cellules forment une partition régulière de l'image, et donc nous pouvons calculer pour chaque segment la liste des cellules que celui-ci traverse.

La figure 3.11 explicite le calcul des voisinages, on donne dans la légende de la figure la liste des voisinages de chaque segment.

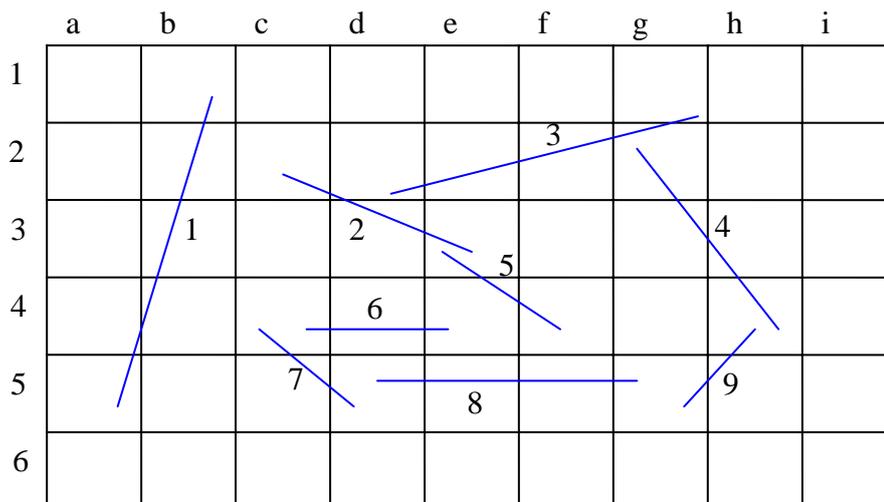


Figure 3.11 : Calcul des voisinages : $\nu(1) = \emptyset$; $\nu(2) = \{3,5\}$;
 $\nu(3) = \{2,4\}$; $\nu(4) = \{3,9\}$; $\nu(5) = \{2,6\}$; $\nu(6) = \{5,7\}$; $\nu(7) = \{6,8\}$; $\nu(8) = \{7,9\}$.

3.6.6 Algorithme d'appariement des segments de droite :

Avant tout rappelons que les données en sortie de la procédure d'extraction des segments de droite d'une paire d'images (approximation polygonale) détaillée dans le chapitre précédant, se présente sous forme de deux listes de segments SDD, SDG liste de segments extraits de l'image droite et gauche respectivement. Chaque segment est stocké dans un accumulateur avec ces attributs suivants:

- i. **Indice** : Un entier qui caractérise sa position dans la liste de segments de droite.
- ii. **Is** : Sa longueur
- iii. **θ** : Son orientation dans le repère de l'image.
- iv. **(I1, J1), (I2, J2)** : Les coordonnées de ses extrémités.
- v. **vois [NSV]** : Liste de NSV (nombre de segments voisins) segments voisins au segment S_i dans on connaît leur attribues locaux.

Pour aboutir à un appariement le plus exact que possible et par conséquent une construction 3D fiable, nous avons utilisé un algorithme inspiré des travaux de Ayache dans [29] qui exploite les contraintes suivantes :

- Calcul de la fenêtre de recherche :

Cette contrainte remplace la contrainte épipolaire. Elle définit pour chaque segment S_i d'une image une fenêtre de recherche $F(S_i)$ de taille dynamique sur l'autre image, ces dimensions dépendent de la taille du segment courant S_i et de la valeur du seuil T_{fen} (une valeur donnée par l'utilisateur). Les segments de droite qui appartiennent à cette fenêtre sont des candidats potentiels pour la mise en correspondance. La figure 3.12 montre bien la forme de la fenêtre de recherche $F(S_i)$.

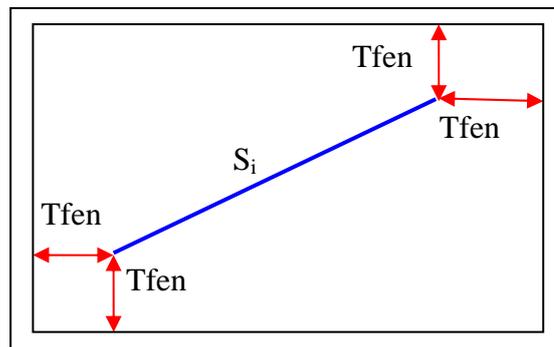


Figure 3.12 : Construction de la fenêtre de recherche.

- Contrainte d'orientation :

Deux segments sont dits d'orientation similaire si la différence de leur orientation est inférieure à un certain seuil (*seuil_teta*).

- Contrainte de longueur :

Deux segments sont dits de longueur similaire si la différence de leur longueur est inférieure à un certain seuil (*seuil_long*).

- Contrainte de distance :

Deux segments sont dits de distance similaire si la différence de leur milieu est inférieure à un certain seuil (*seuil_dis*).

Pour chaque segment d'une image, nous aurons un ensemble de couple de segments potentiels. Nous affectons à chaque couple de segments potentiels un coefficient « Q » qui met en œuvre une comparaison des attributs de ce couple, et qui sont la longueur, l'orientation et la distance reliant les deux points milieux des segments du couple. La valeur de ce coefficient est calculée comme suit :

Pour un couple de segments (S_i, S_j) ,

$$Q_{ij} = \frac{long|S_i - S_j|}{seuil_long} + \frac{orient|S_i - S_j|}{seuil_teta} + \frac{dist|S_i - S_j|}{seuil_dis} \dots\dots\dots 3.4$$

$$Sim_{ij} = \frac{1}{Q_{ij}} \dots\dots\dots 3.5$$

Avec :

$long|S_i - S_j|$ Différence de longueur entre le segment S_i de l'image droite et le segment potentiel S_j de l'image gauche.

$orient|S_i - S_j|$ Différence d'orientation entre le segment S_i de l'image droite et le segment potentiel S_j de l'image gauche.

$dist|S_i - S_j|$ Différence de la distance entre les deux points milieux des segments S_i de l'image droite et le segment potentiel S_j de l'image gauche.

$Seuil_long$: l'erreur permise sur la valeur de la longueur du segment.

$Seuil_teta$: l'erreur permise sur la valeur de l'orientation du segment.

$Seuil_dist$: l'erreur permise sur la valeur de la distance entre les deux segments homologues.

En fin, nous retenons que les couples de segments potentiels dont la valeur du coefficient « Q » est minimale (la similitude « Sim » est maximal), les autres couples des segments potentiels (avec un coefficient Q supérieur à Q_{min}) seront rejetés.

Le segment S_k dont la valeur du coefficient Q est minimale est le segment homologue du segment S_i .

Cette procédure d'appariement permet d'obtenir pour chaque segment d'une image, zéro ou un correspondant dans l'autre image.

Ce processus est appliqué deux fois en inversant les rôles respectifs des deux images de la paire stéréoscopique. On dispose alors de deux listes de segments appariés résultats de l'appariement gauche- droite et droite- gauche. Mais ces deux listes ne sont pas cohérentes.

- Contrainte de compatibilité :

Étant donné un appariement (S_1, S_2) :

Si S_2 est apparié à S_1 (appariement gauche-droite) et si S_3 est apparié à S_2 (appariement droite-gauche) avec S_3 voisins à S_1 , alors (S_1, S_2) est validé (figure 3.13.a).

Si non (S_1, S_2) est rejeté (figure 3.13.b).

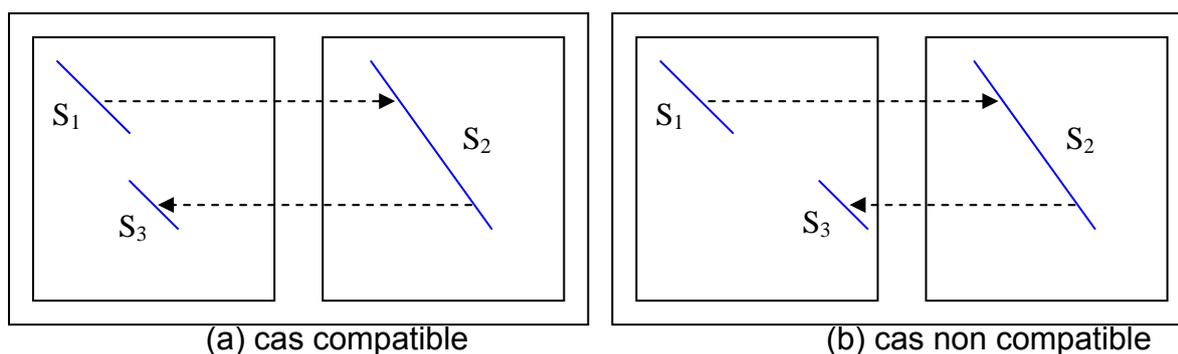


Figure 3.13: Contrainte de compatibilité entre les appariements gauche-droite et droite-gauche

Soulignons ici que les appariements gauche-droite et droite-gauche résultent de deux processus d'appariement totalement indépendants. Le test de la contrainte de compatibilité mutuelle des résultats obtenus offre donc une garantie supplémentaire de fiabilité des appariements finalement retenus.

- Contrainte d'unicité :

Cette contrainte impose que chaque primitive d'une image ait un seul correspondant dans l'autre image. Ceci interdit donc les recouvrements entre segments de droite (figure 3.14).

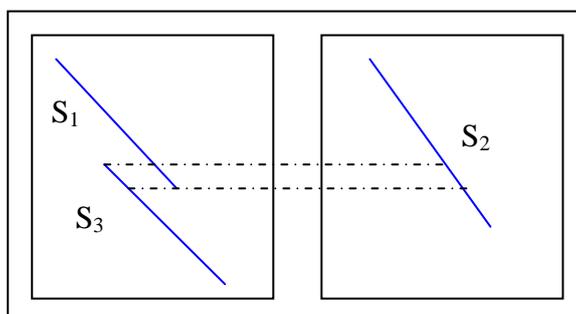


Figure 3.14 : Cas de recouvrement entre deux segments de droite.

Dans le cas d'un recouvrement entre deux segments de droite, nous supprimons celui dont l'appariement est de qualité la moins bonne.

3.7 Conclusion :

Dans ce chapitre nous avons formalisé le problème de la mise en correspondance stéréoscopique. Nous avons présenté les différents problèmes qui rendent l'appariement un exercice difficile à résoudre. Par la suite nous avons donné une série de contraintes permettant de surmonter ces difficultés.

Un survol sur les méthodes de la mise en correspondance rencontrées dans la littérature est aussi présentée.

Nous avons montré dans ce chapitre que les contraintes locales ne suffisent pas à résoudre le problème de l'appariement. Dans ce cas de figure l'utilisation des contraintes physiques dites globales, tel que la contrainte de voisinage, et la contrainte d'unicité, s'avère nécessaire. Nous avons, aussi détaillé l'algorithme d'appariement développé dans notre travail.

Le chapitre suivant utilisera les résultats de l'appariement pour aboutir à une reconstruction tridimensionnelle de la scène.

CHAPITRE 4

LA RECONSTRUCTION TRIDIMENSIONNELLE

Dans ce chapitre nous nous intéressons à la reconstruction 3D (la profondeur) à partir de deux images acquises par un banc stéréoscopique calibré.

Nous donnerons un aperçu général sur les approches mono-image et multi-images existantes dans la littérature pour remonter à cette information 3D. Ensuite, nous ciblerons les méthodes de reconstruction stéréoscopique. Nous détaillerons à la fin de ce chapitre la méthode adoptée dans nos travaux.

4.1 Introduction :

La vision 3D à laquelle nous nous intéresserons est donc définie par la capacité de reconstituer une scène ou un objet, ou par la capacité d'agir ou de se mouvoir dans une scène 3D. Notons que cette scène 3D pourra être "réelle" (la navigation d'un robot mobile dans un milieu d'intérieur ou d'extérieur) ou "virtuelle" (domaine de la réalité virtuelle, visite virtuelle d'un musée). Dans les deux cas, avoir une vision en relief nécessite des informations concernant les distances relatives des points de l'objet ou de la scène. Ces informations peuvent être prises en compte de différentes manières. Dans ce chapitre nous nous limiterons à la reconstruction de l'environnement par stéréovision qui permet de calculer les informations 3D à partir de paires d'images stéréoscopiques.

Le problème de reconstruction 3D d'une scène à partir d'une séquence d'images a suscité beaucoup d'attention ces dernières années. Les approches existantes peuvent être classées en différentes catégories suivant le nombre d'images utilisées (2, 3, ou n images), suivant que la caméra est étalonnée ou non (i.e. les paramètres intrinsèques de la caméra sont connus ou non), suivant le modèle de caméra utilisé (affine ou perspectif), suivant le type de primitives considérées (points, droites, ellipses, contours, etc.), et suivant le type de reconstruction obtenue : projective, affine ou euclidienne.

Les résultats à la sortie de la reconstruction seront une entrée du module de planification de trajectoire d'un robot mobile doté d'un banc stéréoscopique.

4.2 Approches de reconstruction :

Il existe diverses méthodes en vision par ordinateur permettant de remonter à une représentation 3D de l'environnement. On distingue principalement deux grandes classes qui sont les approches mono-image et multi-images.

4.2.1 Approches mono-image :

Appelé aussi vision monoculaire, ces approches n'utilisent qu'une seule image. Pour remonter à l'information 3D d'une scène, des informations complémentaires sont alors nécessaires, comme par exemple [5] :

- i. L'intensité lumineuse de l'image : la structure 3D d'un objet peut être déterminée en s'appuyant sur les variations de luminosité de l'image de ce objet. Ces variations dépendent non seulement de l'éclairage de la surface 3D, mais aussi de la forme et de la nature de cette surface. Si les positions des sources lumineuses ainsi que les propriétés réfléchives de la surface observée sont connues, on peut alors extraire des informations tridimensionnelles sur ce l'objet [48][54].

Les contraintes que l'on impose sur l'environnement sont principalement :

- La surface est lisse ;
- Les propriétés réfléchives sont uniformes sur toute la surface de l'objet, la surface étant souvent supposée lambertienne.
- Les sources lumineuses sont ponctuelles.

Les approches de reconstruction par l'intensité lumineuses sont malheureusement très sensibles au bruit et leur mise en œuvre est délicate. De plus les contraintes qu'elles introduisent pour assurer une solution unique sont rarement vérifiées dans le cas de scènes réelles.

- ii. Le couplage caméra éclairage structuré : cette méthode repose sur l'utilisation d'un éclairage structuré (généralement un laser) et d'une caméra. La profondeur des points de la scène visé peut être obtenue en fusionnant les données de la caméra et le laser. Cette façon de faire est connue sous le nom vision active [55] [56].

- iii. Le flux dans l'image: cette approche est basée sur la mesure de la quantité du flux dans l'image. La présence du flux est due au décalage du plan image par rapport au plan de mise au point (changement de la distance focale). Cette approche, dite de «depth from defocus », est donc un moyen d'obtenir une reconstruction 3D en tout point de l'image.

4.2.2 Approches multi-images :

Contrairement à la classe précédente, ces approches font appel à au moins deux images pour déterminer la structure 3D d'une scène. On distingue deux catégories d'approches qui diffèrent de par la structure même du système de vision utilisé [57]:

- i. Les approches multi-capteurs (stéréovision passive) : comme il est déjà montrée dans le premier chapitre, la vision stéréoscopique permet de reconstruire la structure 3D d'une scène en l'observant depuis plusieurs points de vue différents avec un système à plusieurs caméras, généralement on parle de la *stéréovision binoculaire* dans le cas de deux caméras et de la *stéréovision trinoculaire* dans le cas de trois caméras. Dans la section 4.3, nous donnerons quelques méthodes de reconstruction basée sur la stéréovision binoculaire, des travaux s'y sont intéressés [57].
- ii. Les approches mono-capteurs (caméra mobile) : L'objectif de cette classe de techniques est de réaliser la reconstruction d'une scène à partir d'un certain nombre d'images acquise par une caméra embarquée sur un robot mobile parcourant un environnement statique ou éventuellement contenant des objets en mouvement. En effet, la reconstruction se fait à partir de deux images acquises à des instants différents.

4.2.3 Discussion :

Toutes les techniques de la vision monoculaire font appel à des contraintes strictes d'éclairage ou de géométrie. Par ailleurs ces techniques, sont assez instables, de ce fait, elles ne peuvent constituer un système de vision 3D robuste et fiable pour un robot mobile [5].

La stéréovision semble être très bien adaptée à la perception de l'environnement, et par conséquent une meilleure reconstruction de ce dernier, surtout dans le domaine de la navigation des robots mobiles. Cependant, elle n'a pas les mêmes performances lorsqu'il s'agit d'analyser une zone bien précise de l'environnement en vue d'une reconnaissance par exemple.

4.3 Méthodes de reconstruction 3D stéréoscopique :

L'étape de reconstruction tridimensionnelle consiste à calculer la géométrie de la scène observée par le système de vision. Cette phase de reconstruction utilise les résultats des algorithmes de mise en correspondance. Cependant, on doit, pour chaque paire de primitives (points, segments, régions, etc.) mise en correspondance entre les images stéréoscopiques, trouver dans l'espace la position de la primitive 3D correspondante.

En vision stéréoscopique, différentes approches permettant la reconstruction tridimensionnelle, à partir de deux images, ont été suggérées. Ces approches sont liées au type d'étalonnage utilisé. Le calcul de la position d'un point 3D à partir de ses deux projections dans les deux images se fait par les méthodes dites de triangulation. Cette dernière consiste à estimer l'intersection dans l'espace des deux droites de vue passant par les deux projections dans les deux images et les centres de projection modélisant les caméras.

Plusieurs types d'étalonnage existent et conduisent à des reconstructions à différents niveaux d'information. Pour obtenir une reconstruction euclidienne, c'est-à-dire une reconstruction qui contient des informations métriques sur la scène, il faut avoir réalisé un étalonnage euclidien ou métrique. Si tous les paramètres d'étalonnage ne peuvent être déterminés, la structure 3D reconstruite est moins riche en information. Dans le cas d'un étalonnage partiel, on peut distinguer l'étalonnage affine et l'étalonnage projectif.

En résumé, le type de reconstruction qu'on peut alors effectuer dépend du type de calibrage dont on dispose. Nous pouvons distinguer les cas suivants [53] :

- i. Si aucun calibrage n'est disponible, autrement dit, ni les paramètres intrinsèques et extrinsèques des deux caméras, ni la transformation rigide entre ces dernières est connue ; il faut alors estimer la matrice

fondamentale. A partir de cette matrice nous pouvons effectuer une reconstruction projective.

- ii. Les paramètres intrinsèques et extrinsèques des deux caméras sont connus. On dispose donc de deux matrices de projection, mais la transformation rigide entre les deux caméras n'est pas connue. Il faut alors estimer la matrice essentielle à partir de laquelle on peut extraire la transformation rigide entre les deux caméras à un facteur d'échelle près. Dans ce cas on obtient une reconstruction euclidienne dans le repère de l'une ou l'autre des deux caméras.
- iii. Le capteur stéréoscopique est calibré et on dispose alors de paramètres intrinsèques et extrinsèques de chaque caméra, ainsi que de la transformation rigide entre les deux caméras, dans ce cas on obtient une reconstruction euclidienne dans le repère de calibrage.

4.3.1 Reconstruction par la distance minimale entre deux rayons optiques :

Le problème de la triangulation consiste en la détermination de l'intersection des deux rayons de projection. Pourtant, en présence de bruit dans les coordonnées des points de l'image, les rayons de projection ne se coupent généralement pas.

La méthode classique consiste à choisir le point minimisant la somme des distances aux rayons de projection non incidents il s'agit du point milieu de la perpendiculaire commune aux rayons [58] (figure 4.1).

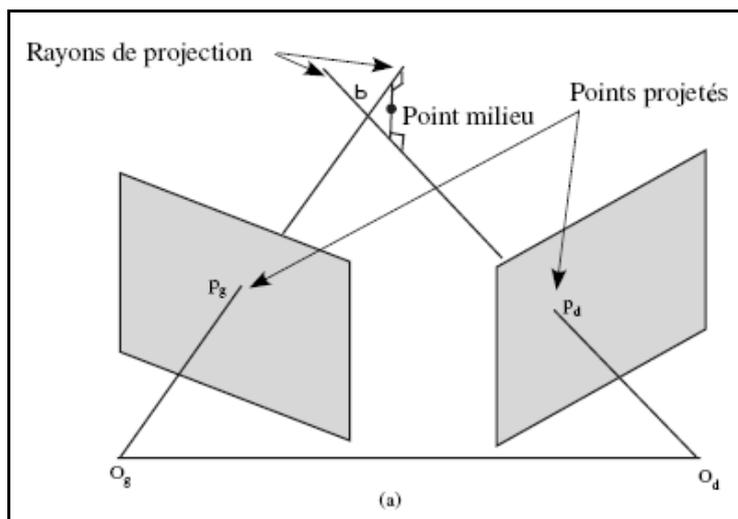


Figure 4.1 : Principe de la reconstruction par la distance minimale entre deux rayons optiques

4.3.2 Reconstruction par triangulation :

La phase de la reconstruction 3D est un simple calcul de triangulation pour deux caméras parfaitement alignées.

La profondeur Z est obtenue en fonction de trois paramètres qui sont :

- La disparité $d = u - v$ (le déplacement ou la distance entre la primitive gauche et droite appariées),
- La focale f
- La ligne de base B (distance entre les centres de projection des deux caméras).

L'expression de Z est donnée par [59]:

$$\frac{b + d(u, v)}{Z - f} = \frac{b}{Z} \dots\dots\dots 4.1$$

On déduit de l'expression (4.1), les coordonnées du point 3D $P(X, Y, Z)$, correspondant à la paire de points $P_g(u_g, v_g)$ et $P_d(u_d, v_d)$, respectivement dans les images gauche et droite :

$$X = \frac{b \cdot u}{d(u, v)} \dots\dots\dots 4.2$$

$$Y = \frac{b \cdot v}{d(u, v)} \dots\dots\dots 4.3$$

$$Z = \frac{b \cdot f}{d(u, v)} \dots\dots\dots 4.4$$

Remarque :

Dans le cas où les caméras ne sont pas parallèles, une phase de rectification (expliquée dans chapitre 1) est nécessaire pour exprimer les points 3D calculés dans le repère du capteur.

4.3.3 Reconstruction en déterminant la projection des points 3D sur les deux plans images :

Nous allons maintenant présenter la méthode de reconstruction qui utilise la projection des points de la scène sur les deux images plus les matrices de projections de chaque caméra, c'est la méthode que nous décrivons plus en détails que les méthodes précédentes puisque c'est celle que nous avons utilisée.

Rappelons que calibrer une caméra, c'est déterminer la transformation qui permet de passer de la position d'un point quelconque $P(X,Y,Z)$ de la scène tridimensionnelle à la position de sa projection $P_g(u_g, v_g)$ sur le plan image gauche, et $P_d(u_d, v_d)$ sa projection sur le plan image droite [22] (figure 4.2).

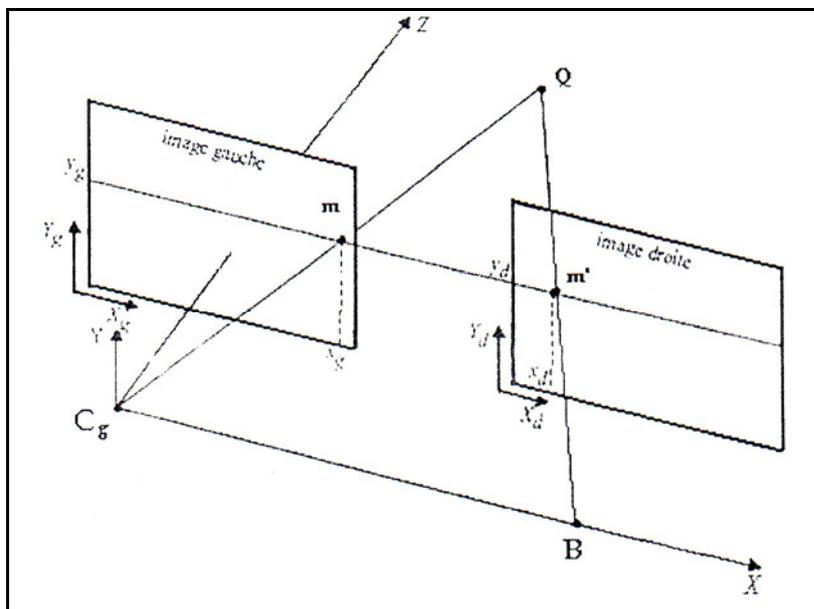


Figure 4.2 : Reconstruction en déterminant la projection des points 3D sur les deux plans images

Si les points sont exprimés en coordonnées homogènes, la transformation perspective M_g et M_d de la caméra gauche et droite respectivement s'écrivent sous la forme linéaire suivante :

$$P_g = M_g \times P \dots \dots \dots 4.5$$

$$P_d = M_d \times P \dots \dots \dots 4.6$$

Avec :

P : Point de la scène tridimensionnelle, $P(X, Y, Z)$ s'écrit $(X, Y, Z, 1)$

P_g : Projection de P sur le plan image gauche, $P_g(u_g, v_g)$ s'écrit $(\mu u_g, \mu v_g, \mu)$

P_d : Projection de P sur le plan image droite, $P_d(u_d, v_d)$ s'écrit $(\mu u_d, \mu v_d, \mu)$

Les deux matrices M_g et M_d (matrice de projection de la caméra gauche et droite respectivement) possèdent 3 lignes et 4 colonnes. La calibration des caméras consiste donc à déterminer les douze coefficients de chaque matrice. Pour complément, nous donnons en Annexe B le détail pour le calcul de ces coefficients.

M_g et M_d s'écrivent comme suit:

$$M_g = \begin{bmatrix} m_{11}^g & m_{12}^g & m_{13}^g & m_{14}^g \\ m_{21}^g & m_{22}^g & m_{23}^g & m_{24}^g \\ m_{31}^g & m_{32}^g & m_{33}^g & m_{34}^g \end{bmatrix} \quad M_d = \begin{bmatrix} m_{11}^d & m_{12}^d & m_{13}^d & m_{14}^d \\ m_{21}^d & m_{22}^d & m_{23}^d & m_{24}^d \\ m_{31}^d & m_{32}^d & m_{33}^d & m_{34}^d \end{bmatrix}$$

Des équations (4.5) et (4.6) on obtiendra:

$$\begin{pmatrix} \mu u_g \\ \mu v_g \\ \mu \end{pmatrix} = M_g \times \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \dots \dots \dots 4.7$$

$$\begin{pmatrix} \mu u_d \\ \mu v_d \\ \mu \end{pmatrix} = M_d \times \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \dots \dots \dots 4.8$$

Donc les équations de reconstruction s'écrivent sous la forme :

$$\left\{ \begin{array}{l} u_g = \frac{m_{11}^g X + m_{12}^g Y + m_{13}^g Z + m_{14}^g}{m_{31}^g X + m_{32}^g Y + m_{33}^g Z + m_{34}^g} \\ v_g = \frac{m_{21}^g X + m_{22}^g Y + m_{23}^g Z + m_{24}^g}{m_{31}^g X + m_{32}^g Y + m_{33}^g Z + m_{34}^g} \\ u_d = \frac{m_{11}^d X + m_{12}^d Y + m_{13}^d Z + m_{14}^d}{m_{31}^d X + m_{32}^d Y + m_{33}^d Z + m_{34}^d} \\ v_d = \frac{m_{21}^d X + m_{22}^d Y + m_{23}^d Z + m_{24}^d}{m_{31}^d X + m_{32}^d Y + m_{33}^d Z + m_{34}^d} \end{array} \right. \dots\dots\dots 4.9$$

Les coordonnées (X, Y, Z) du point P reconstruit dans le repère de calibrage, se calculent en résolvant ce système de quatre équations linéaires. Le système (4.3) peut s'écrire sous forme :

$$\begin{bmatrix} m_{11}^g - u_g m_{31}^g & m_{12}^g - u_g m_{32}^g & m_{13}^g - u_g m_{33}^g \\ m_{21}^g - v_g m_{31}^g & m_{22}^g - v_g m_{32}^g & m_{23}^g - v_g m_{33}^g \\ m_{11}^d - u_d m_{31}^d & m_{12}^d - u_d m_{32}^d & m_{13}^d - u_d m_{33}^d \\ m_{21}^d - v_d m_{31}^d & m_{22}^d - v_d m_{32}^d & m_{23}^d - v_d m_{33}^d \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} m_{34}^g u_g - m_{14}^g \\ m_{34}^g v_g - m_{24}^g \\ m_{34}^d u_d - m_{14}^d \\ m_{34}^d v_d - m_{24}^d \end{bmatrix} \dots\dots\dots 4.10$$

Qui peut être noter de manière plus concise par :

$$A \times P = b. \dots\dots\dots 4.11$$

Ce système peut être résolu aux moindres carrés par la méthode de la pseudo inverse, c'est-à-dire :

$$P = (A' \times A)^{-1} \times A' \times b. \dots\dots\dots 4.12$$

Sous réserve que $A' A$ soit inversible.

La méthode de reconstruction linéaire à l'avantage de conduire à une méthode de calcul non itérative, facile à mettre en œuvre, c'est pour ça elle est souvent utilisée dans la vision en robotique.

4.4 Conclusion :

Nous venons de présenter dans ce chapitre les principales approches permettant de déterminer la structure 3D d'une scène. Les approches utilisant une seule image de la scène se sont avérées peu fiables et sensibles au bruit, leur champ d'application est restreint.

Les approches stéréoscopiques quant à elles se sont avérées beaucoup plus prometteuses tant en précision qu'en fiabilité. La principale difficulté rencontrée par ces approches est la phase de mise en correspondance d'indices bidimensionnels entre les deux images, qui implique l'emploi d'algorithmes complexes.

CHAPITRE 5

RESULTATS EXPERIMENTAUX

Dans ce chapitre, nous présenterons les résultats intermédiaires et finaux des algorithmes implémentés à savoir, l'extraction de contours, chaînage, l'approximation polygonale, la mise en correspondance, et en fin la reconstruction tridimensionnelle. Ces résultats sont issus des méthodes de reconstruction 3D ainsi que des algorithmes de traitement d'images présentés dans les chapitres précédents.

5.1 Introduction :

Pour l'application de la vision par ordinateur, dans la robotique mobile, il est essentiel de développer des outils permettant une reconstruction 3D suffisamment riche et précise du contenu de l'image, pour donner au robot mobile une capacité de mouvoir en sécurité, et accomplir un certain nombre de tâches.

Dans cette optique, nous avons adopté une approche pour la reconstruction tridimensionnelle on utilisant la stéréovision. Cela nous a amené à passer par plusieurs étapes de calcul qui ont été développés dans les chapitres précédentes, nous la résumer en six tâches qui sont:

- Choix et extraction des primitives;
 - L'extraction de contours;
 - Chaînage;
 - L'approximation des chaînes de contours par des segments de droite;
- La mise en correspondance de primitives (segments de droite);
- La reconstruction tridimensionnelle du relief de la scène observée.

Les programmes implémentés ont été testés sur un grand nombre d'images de différentes types et tailles (256×256 et 512×512), afin de valider les différentes étapes citées plus haut.

Le logiciel faisant la reconstruction du relief est un programme écrit en langage C++ Builder sous l'environnement Windows, intégrant tous les algorithmes et procédures développés (l'ouverture d'une image, ouverture d'une paire d'images, l'extraction de contour, chaînage, mise en correspondance etc..). Son interface conviviale permet l'interaction avec l'utilisateur à travers un ensemble de menus, de boutons et de boîtes de dialogue qui permettent le choix et la variation des paramètres techniques liés à l'application.

5.2 L'interface graphique du logiciel développé :

Le logiciel développé permet de réaliser les différentes étapes de calcul cité ci-dessus. La figure 5.1 montre la fenêtre principale (l'interface graphique) de ce logiciel.

Le menu principal contient tous les algorithmes implémentés. Des raccourcis vers les opérations souvent utilisées sont réalisées.

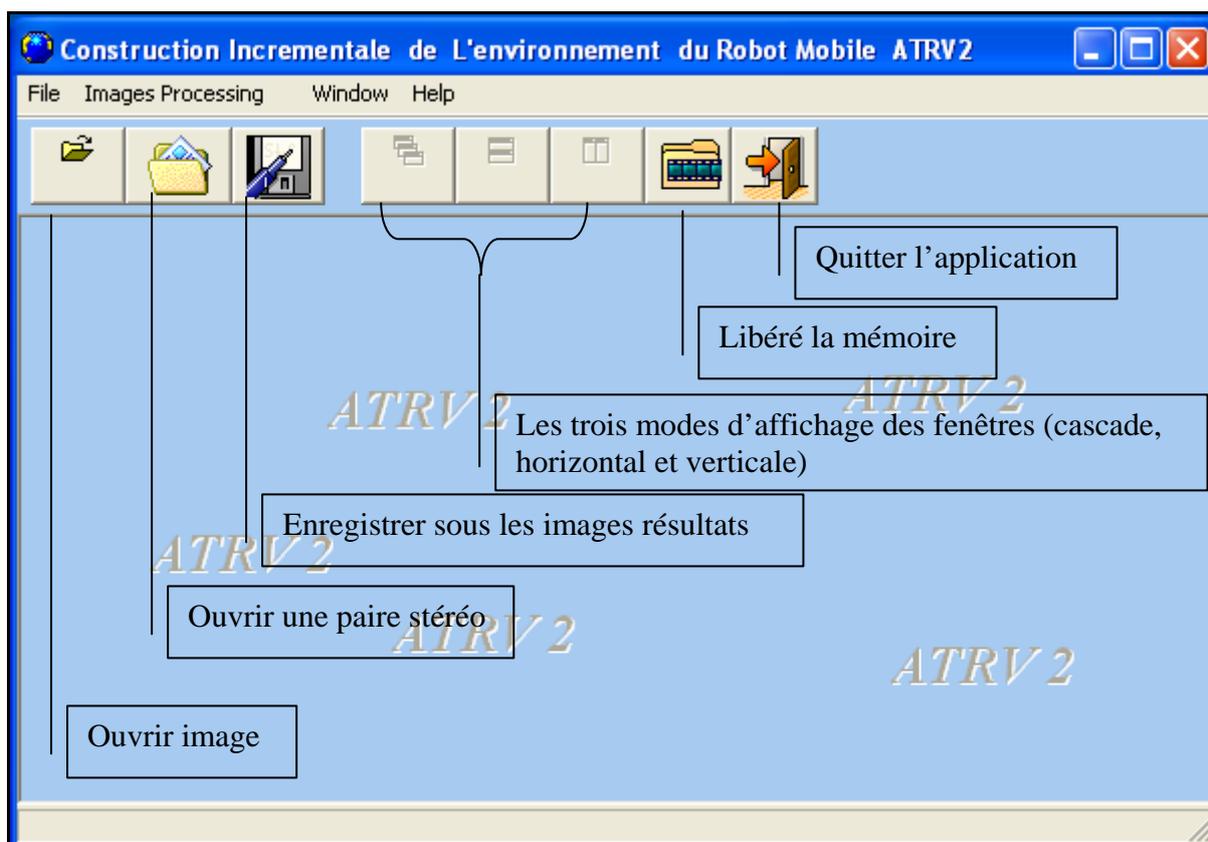


Figure 5.1 : Fenêtre principale du logiciel

Pour l'ouverture d'une image ou d'une paire d'images deux boîtes de dialogue s'affichent, qui sont montrées sur la figure 5.2 et 5.3 respectivement.

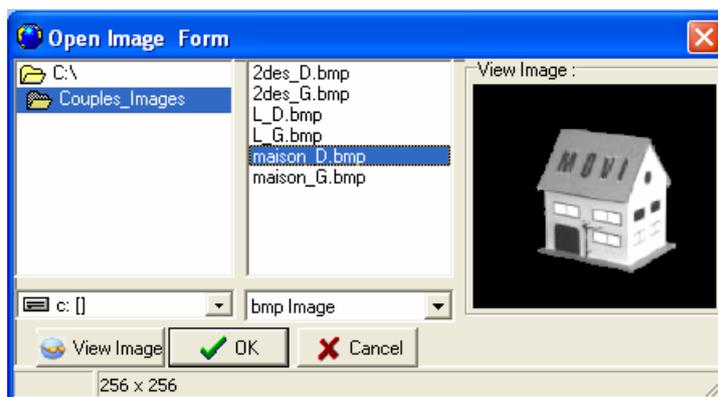


Figure 5.2 : boîte de dialogue pour l'ouverture d'une image

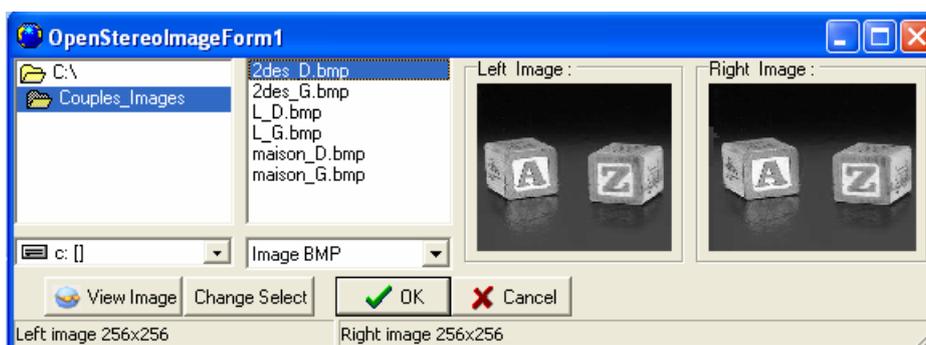


Figure 5.3 : boîte de dialogue pour l'ouverture d'une paire stéréo d'images

Dans les deux cas, le bouton « View Image » permet de voir l'image à sa taille réelle. « OK » pour valider l'ouverture de l'image ou les images sélectionnées. « Cancel » pour fermer la boîte de dialogue. Dans le cas de l'ouverture d'une paire stéréo le bouton « Change Slect » permet de basculer entre le sélectionnemement de l'image droite ou gauche.

Chaque fois qu'une image est sélectionnée sa taille est affichée sur la barre d'état de la boîte de dialogue.

5.3 Choix et extraction de la primitive :

La première et la plus importante de ces tâches pour une chaîne de reconstruction stéréoscopique est le choix de primitives à apparier. Rappelons que

notre choix est porté sur les segments de droite, la justification de ce choix est bien expliquée dans le chapitre 2.

Nous développons dans ce paragraphe, l'ensemble des traitements bas niveau nécessaire pour aboutir à une image de segments de droite.

5.3.1 Extraction de contours :

Les contours de chaque image sont extraits par l'opérateur de Deriche (section 2.2.2.3), l'algorithme de ce dernier admet en entrée une image en niveau de gris et la valeur du paramètre α , il génère en sortie une image contours binaires.

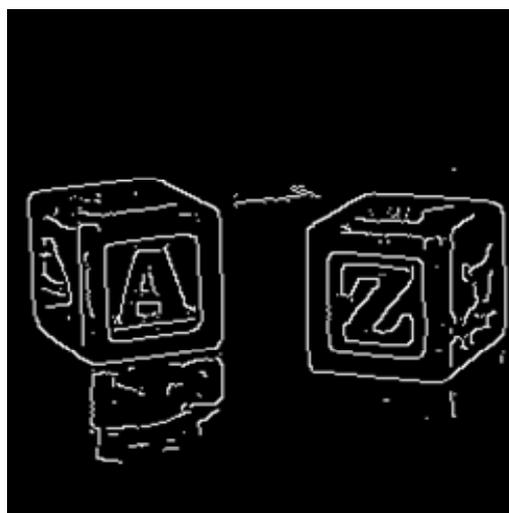
Dans ce qui suit, nous montrons l'influence ou l'impact de la valeur du paramètre α sur le résultat de la détection.



Figure 5.4 : Image originale « 2dés » en niveau de gris



(a) $\alpha = 0.1$



(b) $\alpha = 0.5$

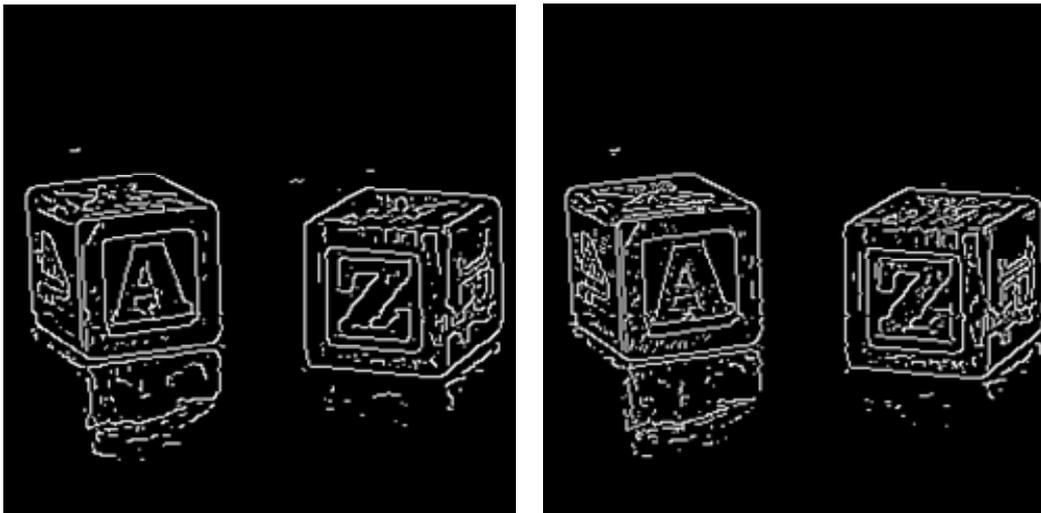
(c) $\alpha = 1$ (d) $\alpha = 1.5$

Figure 5.5 : Extraction de contours pour différentes valeurs de α . Cas de l'image « 2dés »

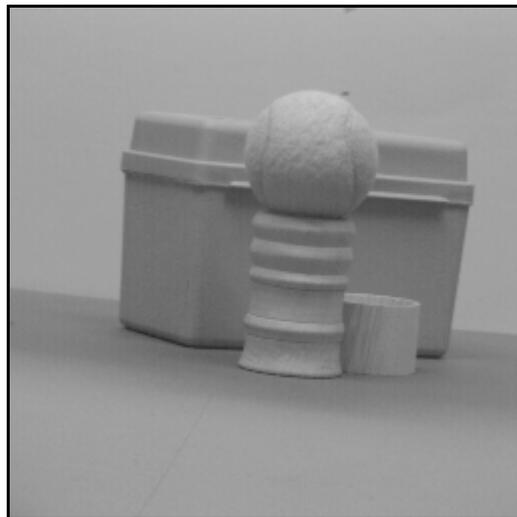
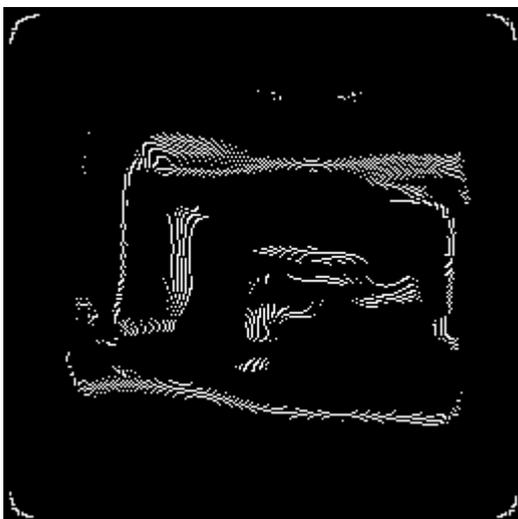
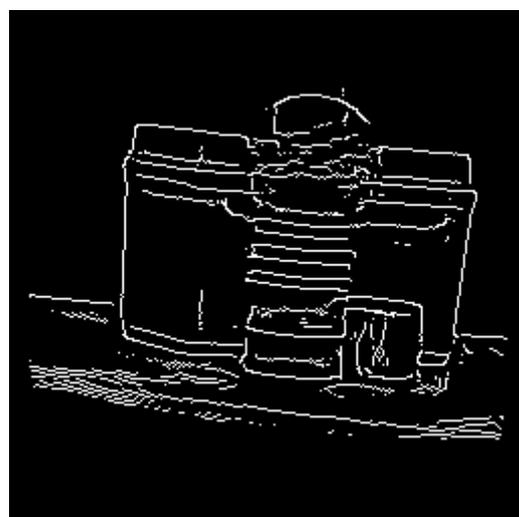


Figure 5.6 : Image originale « ball » en niveau de gris

(a) $\alpha = 0.1$ (b) $\alpha = 0.5$

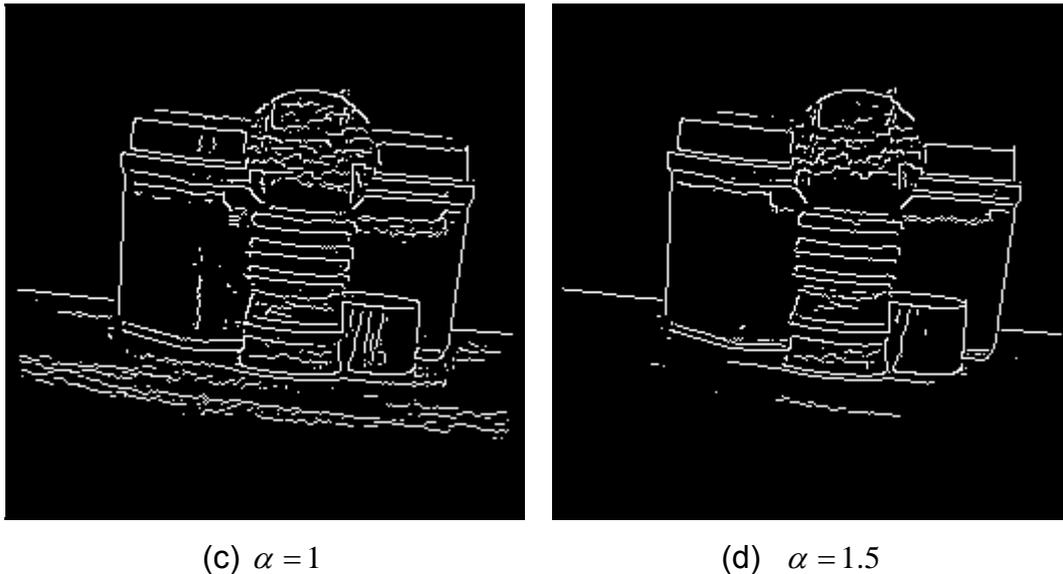


Figure 5.7 : Extraction de contours pour différentes valeurs de α . Cas de l'image « ball »

Discussion

Le paramètre α définit la largeur du filtre, donc le compromis entre la détection et la localisation. Plus α est grand, plus on localise précisément le contour, ainsi que le filtre détecte la moindre variation. Plus α est petit, plus on détecte facilement la présence des contours, cependant il y a peu de réponse (moins sensible au bruit) mais une perte en localisation.

Dans la suite de notre travail, nous avons fixé α à 0.5 pour que le filtre soit moins sensible aux petites variations de l'intensité lumineuse qui ne correspondent pas généralement aux frontières des objets.

5.3.2 Chaînage des contours :

Le chaînage des contours consiste à construire une représentation des contours sous forme de liste de chaînes. Ce chaînage est nécessaire pour, d'une part, réduire et compresser les données manipulées et pour, d'autre part, faciliter les manipulations ultérieures des primitives contours. Nous avons utilisé l'algorithme de Giraudon (section 2.2.3).

Cet algorithme est appliqué sur l'image contours, le résultat de ce dernier est une liste de chaînes de contours. Lorsqu'on exécute cette étape, une fenêtre de dialogue permet d'introduire la valeur d'un paramètre nome « Long », il s'agit d'un

paramètre d'ajustement de chaîne, donc, il permet d'annuler toutes les chaînes dont la longueur est inférieure à la valeur saisie (en pixels).

Le résultat de chaînage est mémorisé sous forme de vecteur structuré, où chaque chaîne est identifiée par un indice « i » (la position de la chaîne dans l'image), sa longueur, et les coordonnées image (I, j) de tous les points de chaque chaîne.

La figure 5.8 montre l'interface graphique réalisée pour cette étape, où chaque chaîne est tracée par une couleur différente. Si l'utilisateur sélectionne une chaîne dans la liste à droite, elle sera tracée en rouge et toutes les autres chaînes en noir.

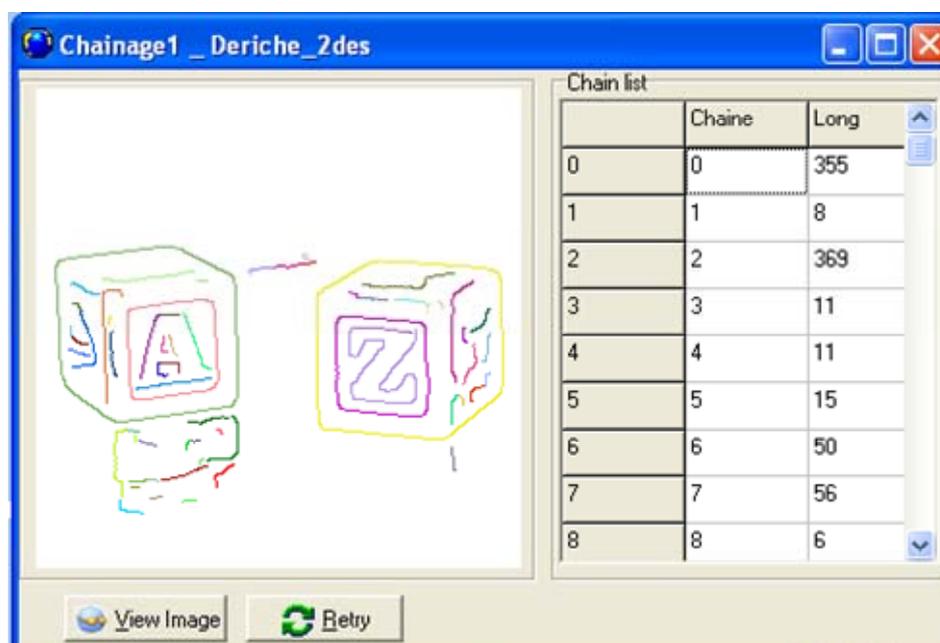
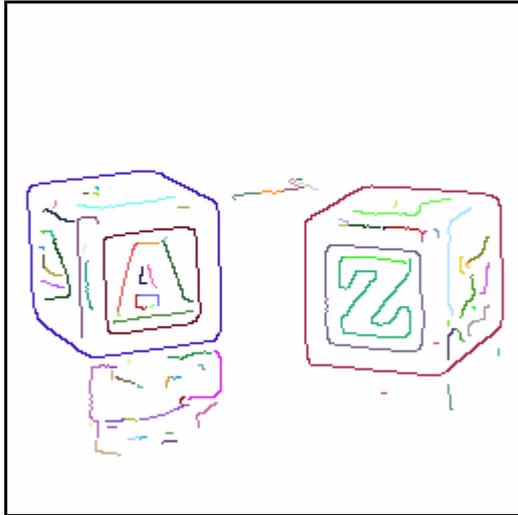
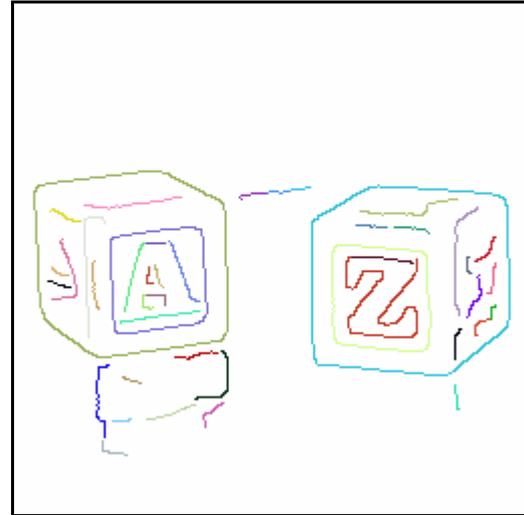


Figure 5.8: Interface de l'algorithme de chaînage. Résultat pour $Long = 5$,
Nombre de chaînes = 64.

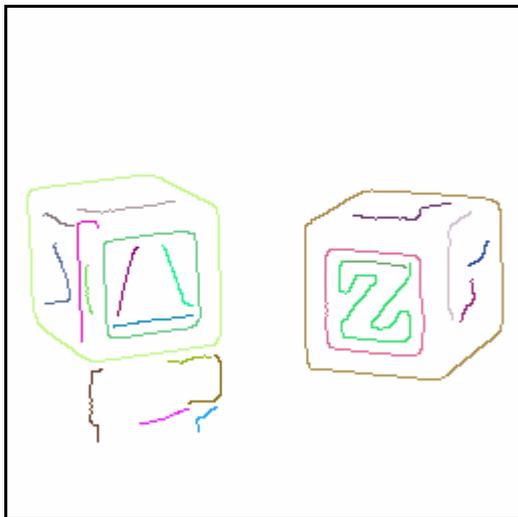
Pour voir l'effet du paramètre d'ajustement de longueur «Long», on applique l'algorithme de chaînage sur l'image contours de la figure 5.5 et 5.7 pour différentes valeurs de « Long ». Les résultats obtenus, sont illustrés dans les figures 5.9 et 5.10 pour les images « 2dés » et « ball » respectivement.



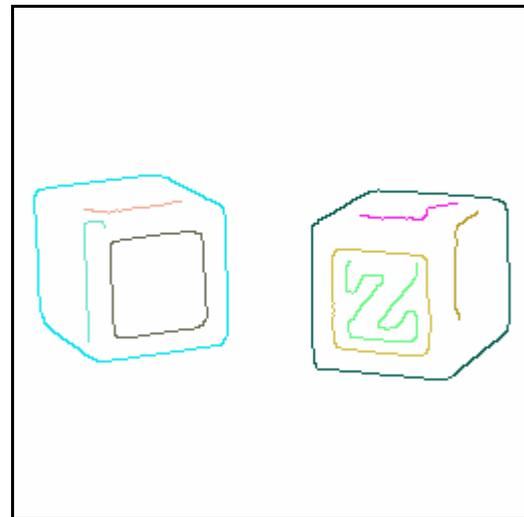
(a) $Long = 3$, Nombre de chaînes = 91.



(b) $Long = 10$ Nombre de chaînes = 45



(c) $Long = 20$, Nombre de chaînes = 23



(d) $Long = 50$, Nombre de chaînes = 9

Figure 5.9: Résultats de l'algorithme de chaînage pour différentes valeurs de « Long » appliqué sur l'image contours « 2dés »

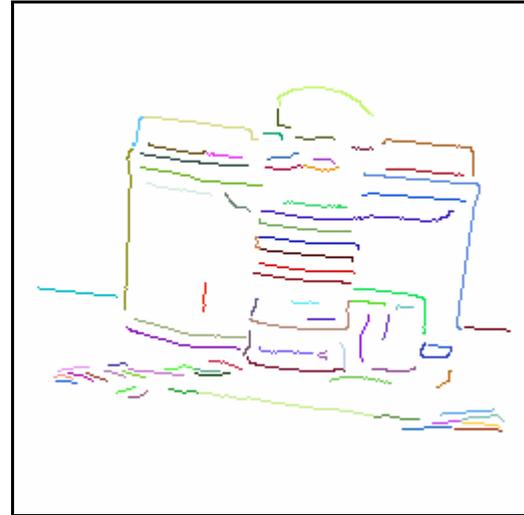
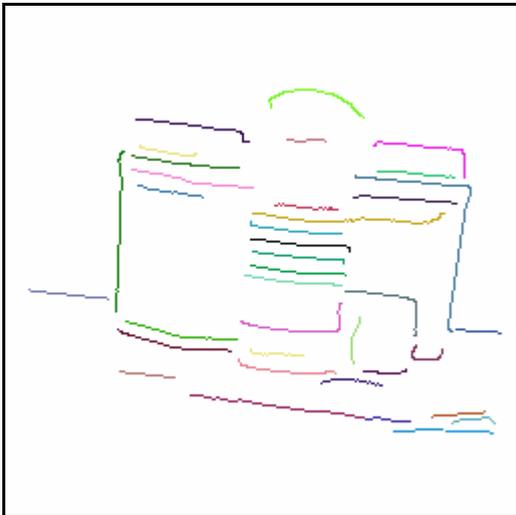
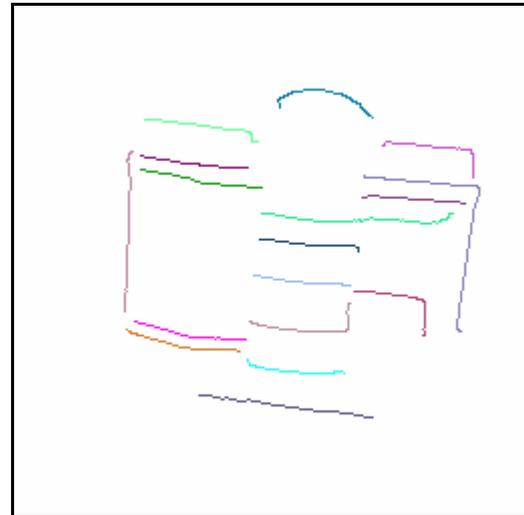
(a) $Long = 3$, Nombre de chaînes = 208.(b) $Long = 10$ Nombre de chaînes = 78(c) $Long = 20$, Nombre de chaînes = 38(d) $Long = 50$, Nombre de chaînes = 17

Figure 5.10: Résultats de l'algorithme de chaînage pour différentes valeurs de « Long » appliqué sur l'image contours « ball »

Discussion

D'après les résultats ci-dessus, c'est trivial de remarquer que l'augmentation de « Long » résulte moins de chaîne, par conséquent on risque, pour des valeurs élevées de ce paramètre, de perdre des informations importantes dans l'image. Donc il est préférable que sa valeur ne soit pas très grande. Dans la suite de notre travail « Long » est fixé à 5 pixels.

5.3.3 Approximation polygonale :

La dernière étape pour aboutir à une image de segments de droite, est la phase de l'approximation polygonale. Nous avons implémenté l'algorithme de l'approximation polygonal par fusion (section 2.2.4.2. b).

La figure 5.11 présente la boite de dialogue permettant de saisir les paramètres nécessaires pour cette étape, il s'agit de la distance maximale autorise entre la chaîne et le segment (l'erreur autorisé), la résolution en X et Y de la grille.

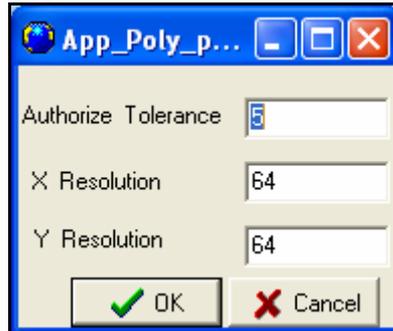


Figure 5.11: Boite de dialogue pour la saisie des paramètres de l'approximation polygonale

Le résultat obtenu après exécution est une liste de segments de droite où chaque segment est stocké dans un vecteur structuré « SD » avec les attributs suivants:

- vi. **Indice** : Un entier qui caractérise sa position dans la liste de segments de droite.
- vii. **Is** : Sa longueur
- viii. **θ** : Son orientation dans le repère de l'image.
- ix. **(I1, J1), (I2, J2)** : Les coordonnées de ses extrémités.
- x. **vois [NSV]** : Liste de NSV (nombre de segments voisins) segments voisins au segment S_i dont on connaît leurs attributs locaux.

Pour l'affichage de toutes ces informations, nous avons réalisé l'interface illustrée par la figure 5.12, on visualise sur ce dernier deux listes; sur la première «Segments Liste» sur la quelle on affiche tous les segments de droite dans l'image avec leurs attributs géométriques (l'indice i , longueur Long, orientation Orent, et les coordonnées des deux extrémités $((I_1, J_1)(I_2, J_2))$). Sur la deuxième liste on affiche l'ensemble des segments voisins au segment sélectionné dans la première liste.

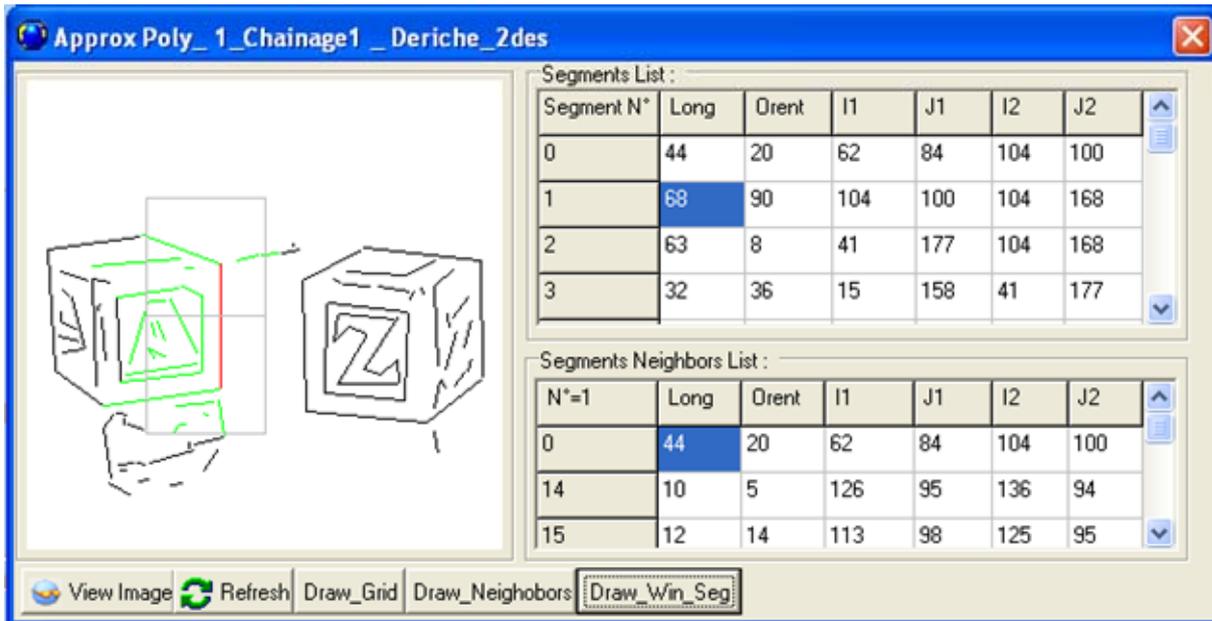


Figure 5.12 : L'interface de l'algorithme de l'approximation polygonale.

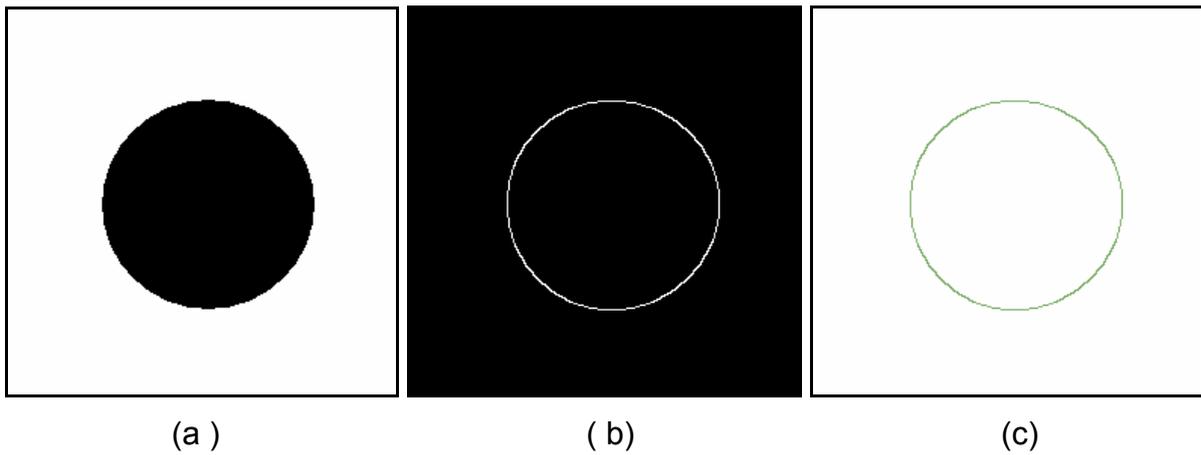
Résultat pour $e_m = 5$, X résolution= 64, Y résolution= 64.

Nombre de segments = 94.

Le bouton « Draw_Neighbors » permet de donner la couleur vert à tous les segments voisins au segment sélectionné dans la liste des segments. «Draw_Grid» permet le tracé de la grille de l'image en cellules régulières de taille donner par l'utilisateur (X et Y résolution). «Draw_Win_Seg» permet le tracé des cellules que le segment sélectionné les traverse (figure 5.12). « View_Image » permet de visualiser l'image à sa taille réelle. La sélection d'un segment voisin (dans la liste des segments voisins) permet de lui donner une couleur bleu dans l'image.

- Le paramètre e_m :

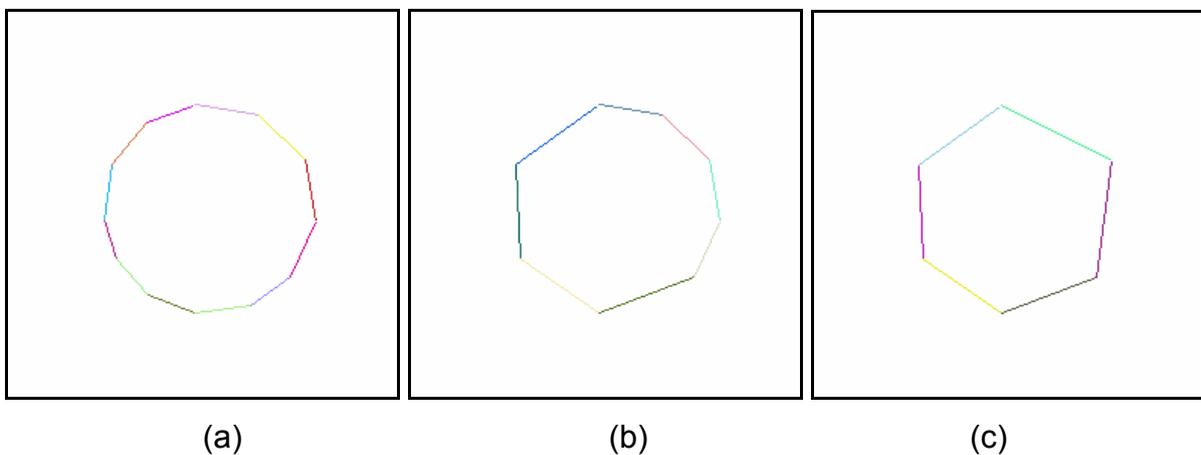
Pour bien montrer l'effet du paramètre « distance autorise e_m » sur le résultat de l'approximation (la qualité l'image résultat), nous appliquons l'algorithme sur une image particulière soit un disque (la figure 5.13, et figure 5.14).



(a) (b) (c)

Figure 5.13 : Résultats de traitements de l'image « disque » :

(a) : image originale, (b) : résultat d'extraction de contour, (c) : résultat du chaînage



(a) (b) (c)

Figure 5.14 : Résultats de l'approximation polygonale pour différentes valeurs de e_m :

(a) : $e_m = 5 \text{ pixels}$, nombre de segments =12 ;

(b) : $e_m = 10 \text{ pixels}$, nombre de segments =8 ;

(c) : $e_m = 20 \text{ pixels}$, nombre de segments =6.

Dans ce qui suit nous présentons d'autres résultats issue de l'algorithme de l'approximation polygonale implémenté. La figure 5.15 présente l'image originale et le résultat des deux traitements extraction de contour et chaînage. Les résultats de l'approximation polygonale pour différentes valeurs du paramètre e_m sont montrés sur la figure 5.16.

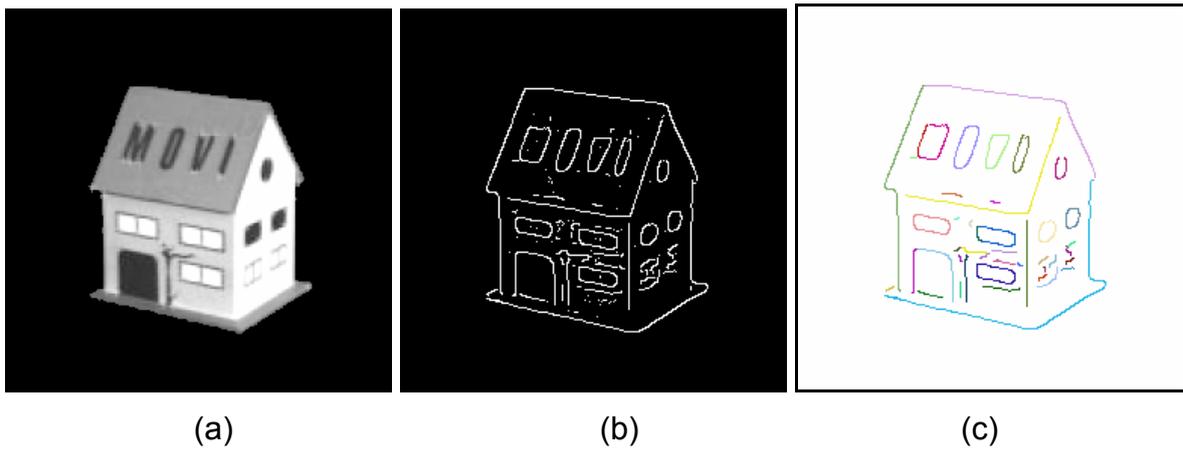


Figure 5.15 : Résultats de traitements de l'image «maison» :

(a) : image originale, (b) : résultat d'extraction de contour, (c) : résultat du chaînage

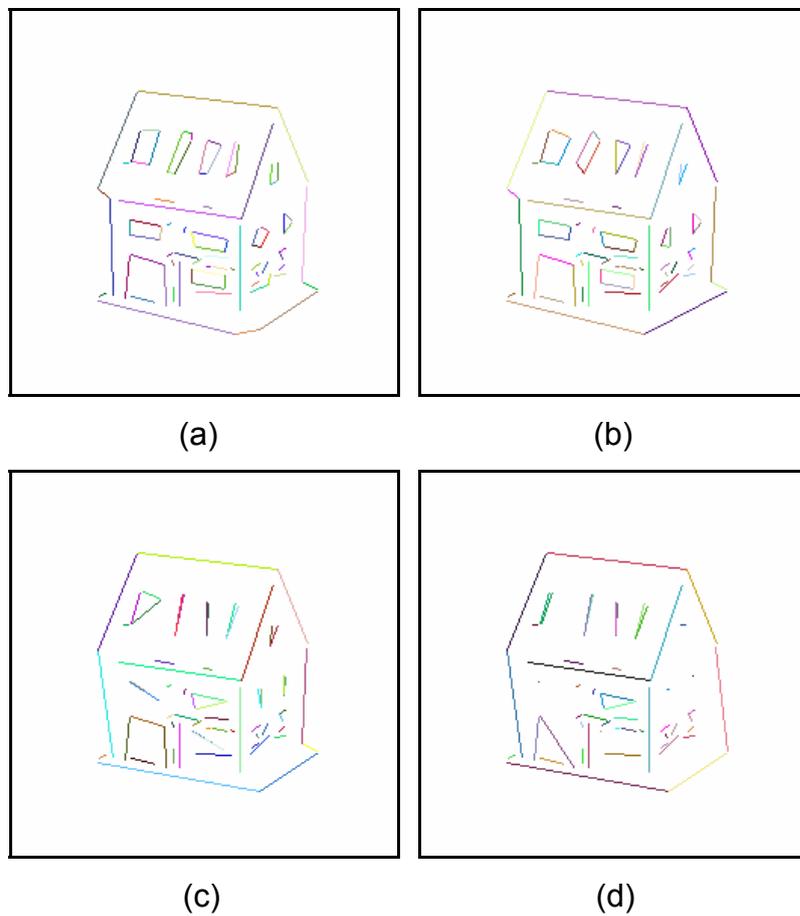


Figure 5.16 : Résultats de l'approximation polygonale pour différentes valeurs de e_m :

(a) : $e_m = 3 \text{ pixels}$, nombre de segments =82 ;

(b) : $e_m = 5 \text{ pixels}$, nombre de segments =75 ;

(c) : $e_m = 10 \text{ pixels}$, nombre de segments =63.

(c) : $e_m = 15 \text{ pixels}$, nombre de segments =55.

Dans le tableau 5.A, nous résumons les résultats obtenus (nombre de segments de droite résultant) pour les deux couples d'image « 2dés » et « maison »

e_m (en pixels) images	3	5	10	15
2dés_G	109	94	88	80
2dés_D	114	97	91	83
Maison_G	82	75	63	55
Maison_D	79	72	60	52

Tableau 5.A : Résumer des résultats obtenu de l'algorithme de l'approximation polygonale pour différentes valeurs de e_m

Discussion :

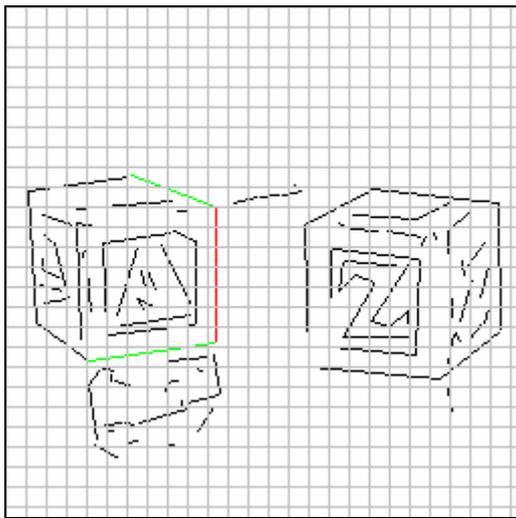
D'après les résultats obtenus de l'approximation polygonale pour différentes valeurs du paramètre e_m (tableau 5.A) , nous remarquons que l'augmentation de la valeur e_m minimise le nombre de segments de droite résultant. En effet cette augmentation ne permet pas un meilleur suivi de points de contour, par conséquent on risque de perdre l'information de l'image contours (forme des objets), pour cela la valeur de e_m doit être faible.

▪ Calcul de voisinage :

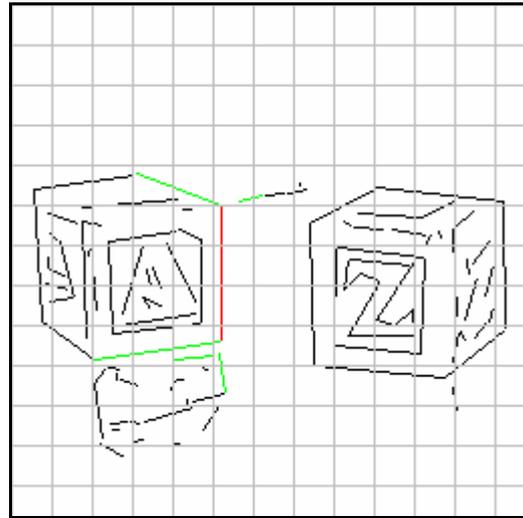
Comme il a été expliqué dans la section 3.6.5 le voisinage d'un segment S_j est l'ensemble des segments qui intersectent une cellule commune avec S_j . Sachant que nous avons utilisé des cellules de forme rectangulaire de taille dépend des valeur saisie dans la boite de dialogue de la figure 5.11 (résolution en X et Y).

Nous signalons que, la résolution de la grille influent directement sur le calcul de voisinage, soient à titre d'exemple les résultats (figure 5.17) obtenus pour les valeurs suivante :

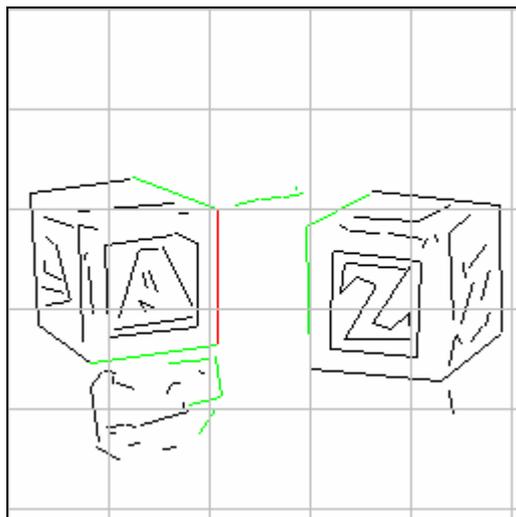
- $e_m = 5$, X résolution = 10, Y résolution = 10
- $e_m = 5$, X résolution = 20, Y résolution = 20
- $e_m = 5$, X résolution = 50, Y résolution = 50
- $e_m = 5$, X résolution = 64, Y résolution = 64



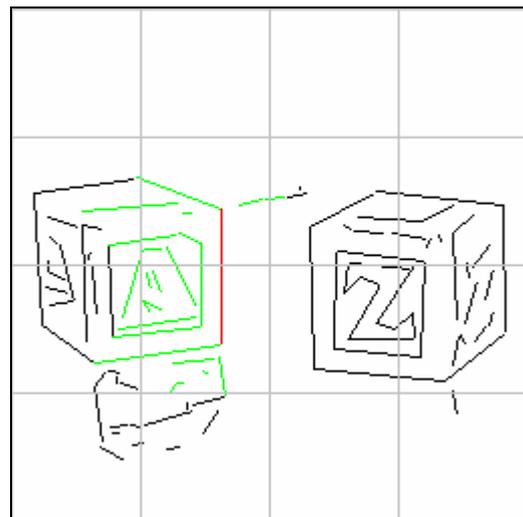
(a) nombre de segments voisins =2



(b) nombre de segments voisins =5



(c) nombre de segments voisins =13



(d) nombre de segments voisins =25

Figure 5.17: Résultats de calcul de voisinage pour différentes valeurs de résolution de la grille.

5.4 Mise en correspondance :

La phase d'appariement a pour but de trouver pour chaque segment de droite de l'image gauche son homologue dans l'image droite (section 3.6). Pour mesurer la qualité de l'appariement nous utiliserons la formule 5.1 dit taux de réussite, pour un meilleur appariement le quotient de la formule doit être très proche de un.

$$\frac{\text{nombre_de_paires}}{\text{nombre_de_primitives_de_image}} \cong 1 \dots \dots \dots 5.1$$

L'exécution de l'algorithme de mise en correspondance nécessite un ensemble de paramètres qui sont :

- *Tfen* : c'est un entier (en pixels) qui détermine la taille de la fenêtre de recherche en fonction de la longueur et l'orientation du segment sélectionné c'est-à-dire la fenêtre qui contient les candidats potentiels à l'appariement; c'est très important de bien le fixer car la valeur attribuée à ce dernier influe directement sur la qualité du résultat de l'appariement, ainsi que le temps de calcul.
- *seuil_dis*, *seuil_long*, *seuil_teta*: sont les seuils autorisés pour que deux segments soient mites en correspondance, il s'agit respectivement : de la distance autorisée entre les deux points milieux des segments homologues, la différence de longueur, et différence d'orientation autorisée.

Une boîte de dialogue (figure 5.18), permet à l'utilisateur de saisir les valeurs des différents paramètres de la mise en correspondance citée ci-dessus. Après la validation des données par le bouton « OK », la fenêtre de la figure 5.19 apparaît, elle comporte le résultat de l'algorithme de cette phase.



Figure 5.18 : Boîte de dialogue de mise en correspondance.

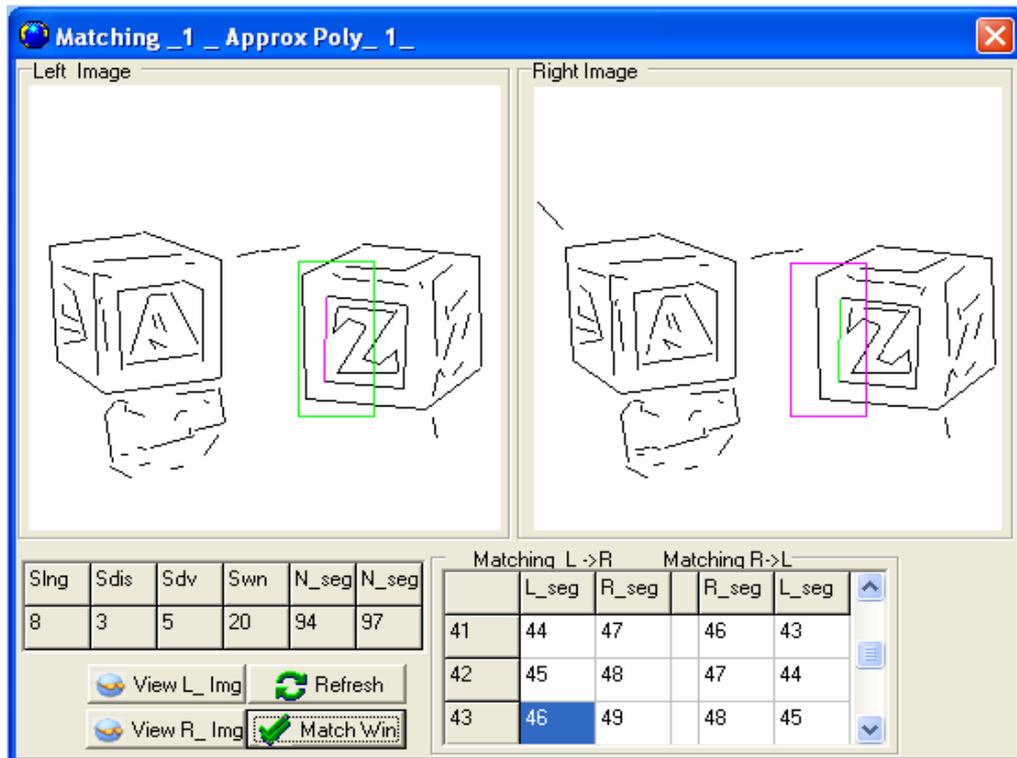


Figure 5.19: L'interface de l'algorithme de mise en correspondance.

Sur cette interface, nous avons quatre boutons dont deux nommés «View_L_Img» «View_R_Img» pour visualiser l'image gauche et droite respectivement à sa taille réelle. «Refresh» pour initialiser l'état des deux images. Le bouton «Match_Win» sert à afficher la fenêtre de recherche. Sur la liste on affiche séparément tous les segments homologues résultat de l'appariement gauche-droite (matching $L \rightarrow R$), et droite-gauche (matching $R \rightarrow L$). Pour voir le segment sur l'image, il suffit de le sélectionner dans la liste, ce dernier sera affiché par une couleur différente (rose, vert), (Figure 5.18).

- L'impacte du paramètre T_{fen} :

Dans ce qui suit, nous présentons quelque résultats de l'appariement de la paire image «2dés_G, 2dés_D» et «maison_G, maison_D» pour différentes tailles de la fenêtre de recherche.

Premier cas : on applique l'algorithme de l'appariement pour les paramètres suivants :

- $T_{fen} = 5$;
- $seuil_dis = 3\text{ pixel}$;
- $seuil_long = 5\text{ pixels}$
- $seuil_teta = 10^\circ$.

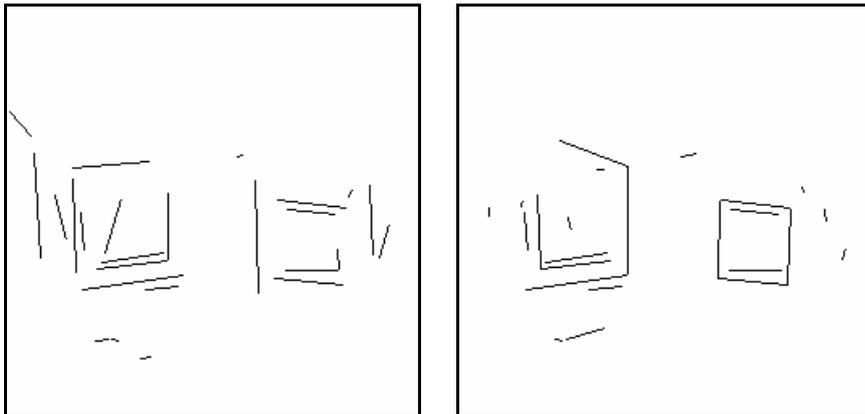


Figure 5.20 : Résultat de l'appariement de la paire stéréo « 2dés » pour $T_{fen} = 5$

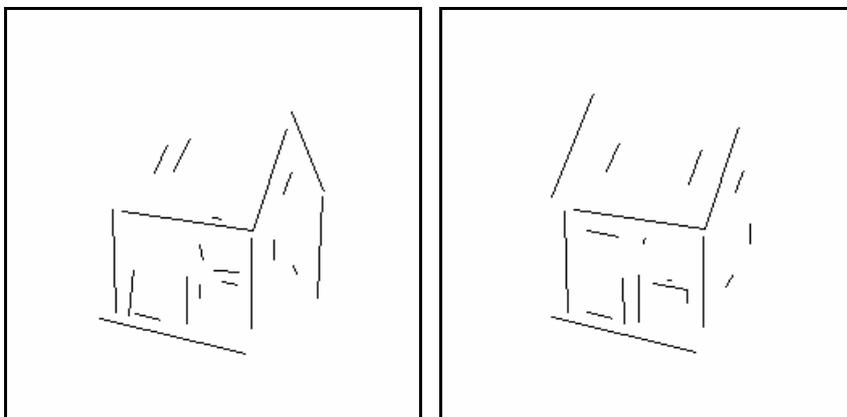


Figure 5.21 : Résultat de l'appariement de la paire stéréo « maison » pour $T_{fen} = 5$

Deuxième cas : on applique l'algorithme de l'appariement pour les paramètres suivants :

- $T_{fen} = 20$;
- $seuil_dis = 3\text{ pixel}$;
- $seuil_long = 5\text{ pixels}$
- $seuil_teta = 10^\circ$.

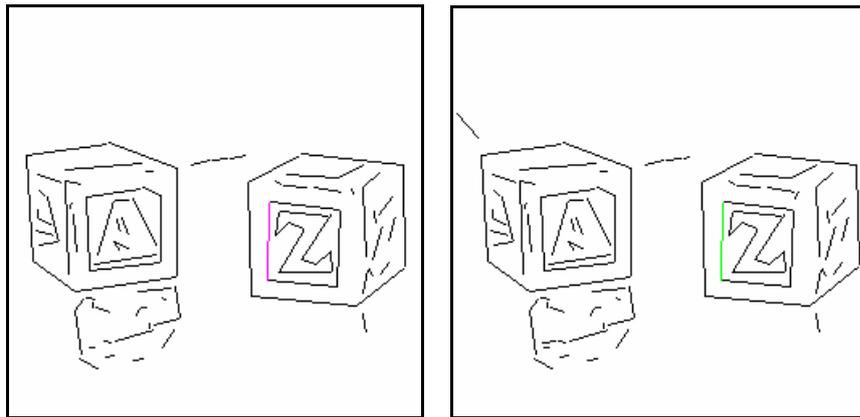


Figure 5.22 : Résultat de l'appariement de la paire stéréo « 2dés » pour
 $T_{fen} = 20$

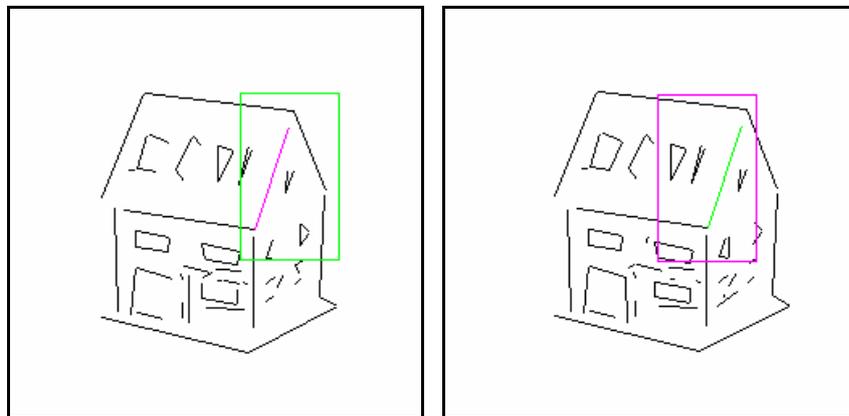


Figure 5.23 : Résultat de l'appariement de la paire stéréo « maison » pour
 $T_{fen} = 20$

Troisième cas : on applique l'algorithme de l'appariement pour les paramètres suivants :

- $T_{fen} = 40$;
- $seuil_dis = 3\text{pixel}$;
- $seuil_long = 5\text{pixels}$
- $seuil_teta = 10^\circ$.

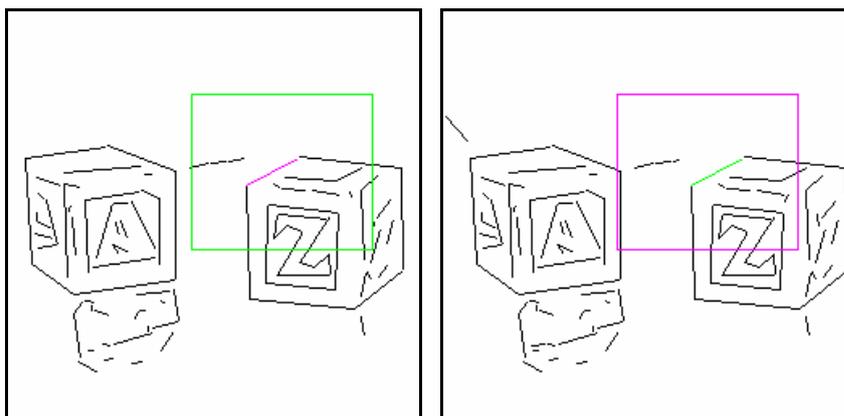


Figure 5.24 : Résultat de l'appariement de la paire stéréo « 2dés » pour
 $T_{fen} = 40$

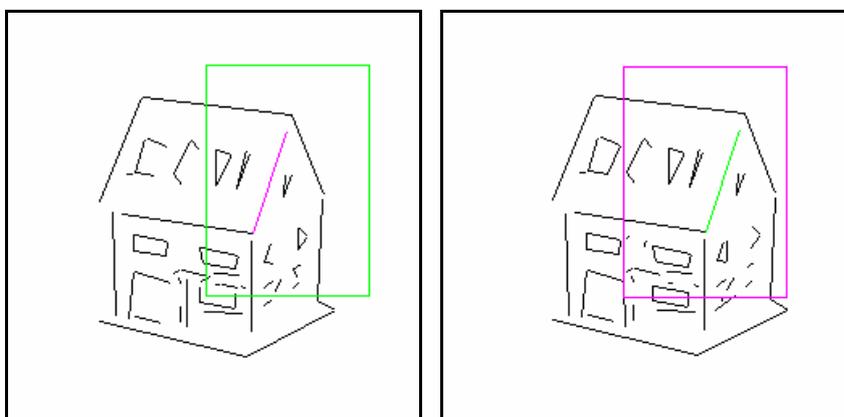


Figure 5.25 : Résultat de l'appariement de la paire stéréo « maison » pour
 $T_{fen} = 40$

Discussion :

Remarquons que la taille de la fenêtre de recherche, (donc la valeur du paramètre T_{fen}) a une grande influence sur le résultat de l'appariement, pour une meilleure illustration de cette influence, nous donnons dans le tableau 5.B ci-dessous le taux de réussite de l'appariement pour différentes valeurs du paramètre T_{fen} .

Signalons aussi que pour des valeurs importantes de T_{fen} , le résultat de l'appariement est presque le même (résultat de la figure 22, 24 pour la paire 2dés; figure 23, 25 pour la paire maison), mais avec un temps de calcul plus élevé, (car pour des grande taille de la fenêtre de recherche les candidats potentiels à

l'appariement est plus important), cependant, il faut trouver la bonne valeur de ce dernier pour un compromis entre la qualité de l'appariement et le temps de calcul.

	2dés	maison
Nbre. de segments gauches	94	75
Nbre. de segments droites	97	72
Nbre. de segments appariés pour $T_{fen} = 5$	25	19
Taux de réussite ($T_{fen} = 5$)	25.77 %	25.33 %
Nbre. de segments appariés pour $T_{fen} = 10$	91	32
Taux de réussite ($T_{fen} = 10$)	93.81 %	42.66 %
Nbre. de segments appariés pour $T_{fen} = 20$	89	60
Taux de réussite ($T_{fen} = 20$)	91.75 %	80.00 %
Nbre. de segments appariés pour $T_{fen} = 30$	89	61
Taux de réussite ($T_{fen} = 30$)	93.81	81.33 %

Tableau 5.A : Les résultats expérimentaux obtenu on appliquons l'algorithme d'appariement sur deux paires d'images stéréoscopiques.

5.5 Reconstruction 3D de la scène :

Nous avons implémenté une méthode de reconstruction euclidienne (section 4.3.3) qui permet de calculer les segments 3D, à partir de leur projection dans la paire stéréoscopique.

La phase de reconstruction implémentée nécessite d'introduire les différents paramètres extrinsèques et intrinsèques de chaque caméra du banc stéréoscopique. Cependant, une fenêtre de dialogue (figure 5.26) permettant à utilisateur de saisir ces paramètres apparaît lors de l'exécution de la phase de reconstruction.

Left Camera Parametres :		Right Camera Parametres :	
Focal	1,0	Focal	1,0
Image discretisation		Image discretisation	
au	963,394727	au	965,382788
av	1382,027466	av	1385,102661
Centre Image		Centre Image	
u0	223,394424	u0	249,500900
v0	282,252075	v0	257,570740
Translation Vector		Translation Vector	
tx	-900,015503	tx	-805,351013
ty	146,275742	ty	140,578391
tz	-900,394897	tz	-1002,109009
Rotation matrix		Rotation matrix	
R11	0,677717	R11	0,6E9617
R12	-0,015955	R12	0,002074
R13	0,74542	R13	0,743604
R21	-0,012746	R21	-0,017822
R22	0,993373	R22	0,9E93754
R23	0,033031	R23	0,013237
R31	-0,741604	R31	-0,743394
R32	-0,031606	R32	-0,022103
R33	0,677093	R33	0,6E93489

Figure 5.26 : Boite de dialogue pour la saisie des paramètres du banc stéréoscopique

Les segments 3D reconstruits sont stockés dans un vecteur structure, tel que chaque segment présente les attributs suivants:

- **Indice** : Un entier qui caractérise sa position dans la liste de segments de droites 3D
- $(x_1, y_1, z_1), (x_2, y_2, z_2)$: Les coordonnées 3D de ses extrémités.

L'interface graphique réalisé pour cette étape est illustrée dans la figure 5.27, sur la quelle nous affichons les trois perspectives (vue de face, de profile, et de haut). En plus, une liste qui contient tous les informations liées aux segments 3D. Nous affichons aussi un aperçu de la paire d'images à reconstruire, un double clique sur l'une d'image pour la voir à sa taille réelle. La sélection d'un segment dans la liste permet de le voir avec une couleur différente (rouge) dans les trois vu et sur l'aperçu de la paire stéréo.

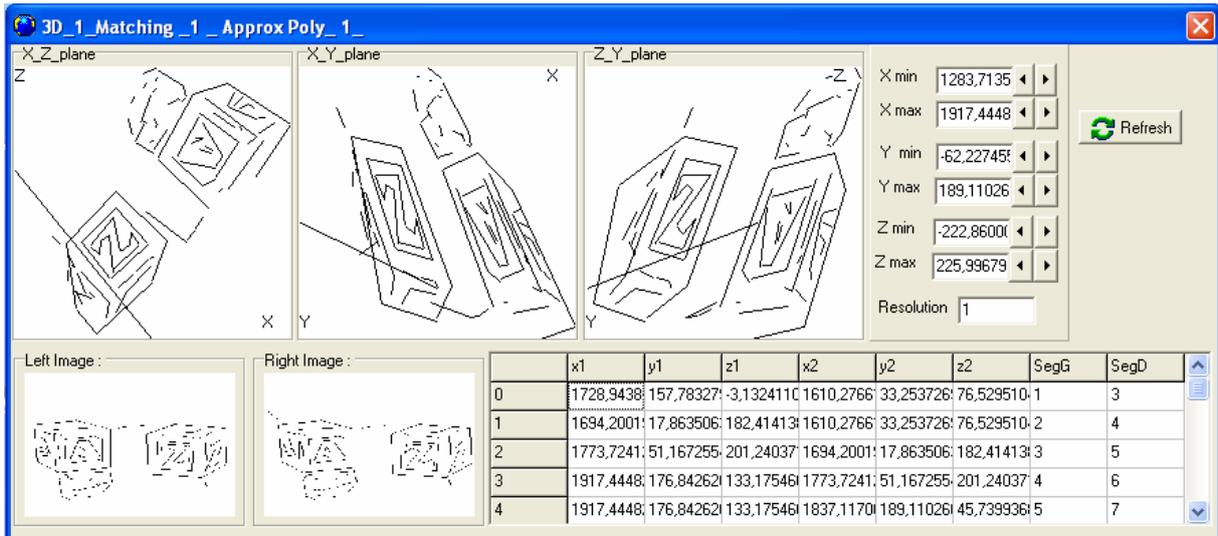


Figure 5.27 : L'interface de la phase de reconstruction 3D. Cas de la paire stéréo « Cub ».

La phase de reconstruction 3D, était aussi programmée en utilisant un autre langage de programmation (Matlab), dans le but de mieux voir les segments 3D reconstruits. Nous présentons ci-dessous les résultats obtenus pour les deux paires stéréoscopiques calibrées.

Exemple 1: reconstruction de la paire stéréoscopique « 2dés » :



(a) image gauche

(b) image droite

Figure 5.28 : La paire stéréoscopique «2dés» originale en niveau de gris.

Premier cas : on applique l'algorithme de la reconstruction 3D pour le résultat de l'appariement de la figure 5.20 ($T_{fen} = 5$) :

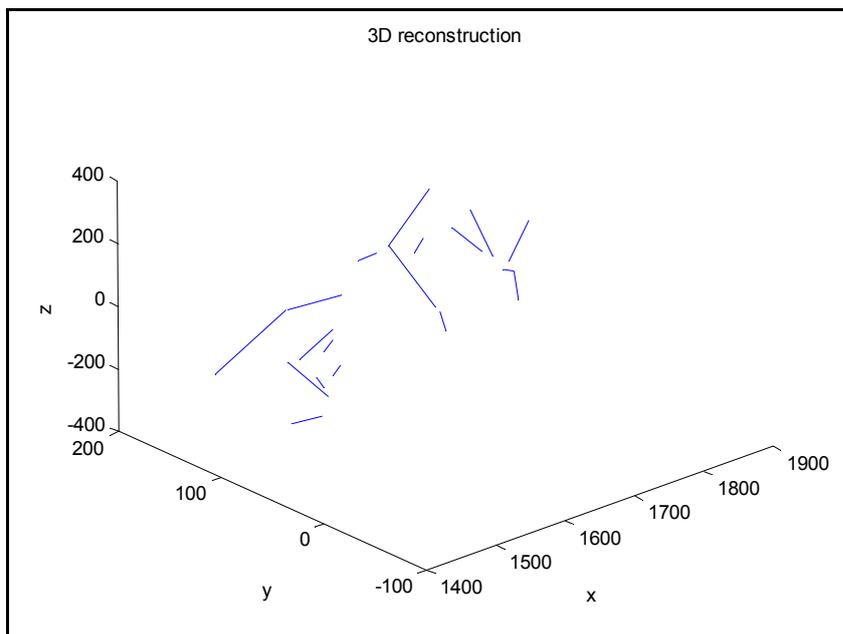


Figure 5.29: Résultat de reconstruction de la paire stéréo « 2dés », le premier cas.

Deuxième cas : on applique l'algorithme de la reconstruction 3D pour le résultat de l'appariement de la figure 5.22 ($T_{fen} = 20$) :

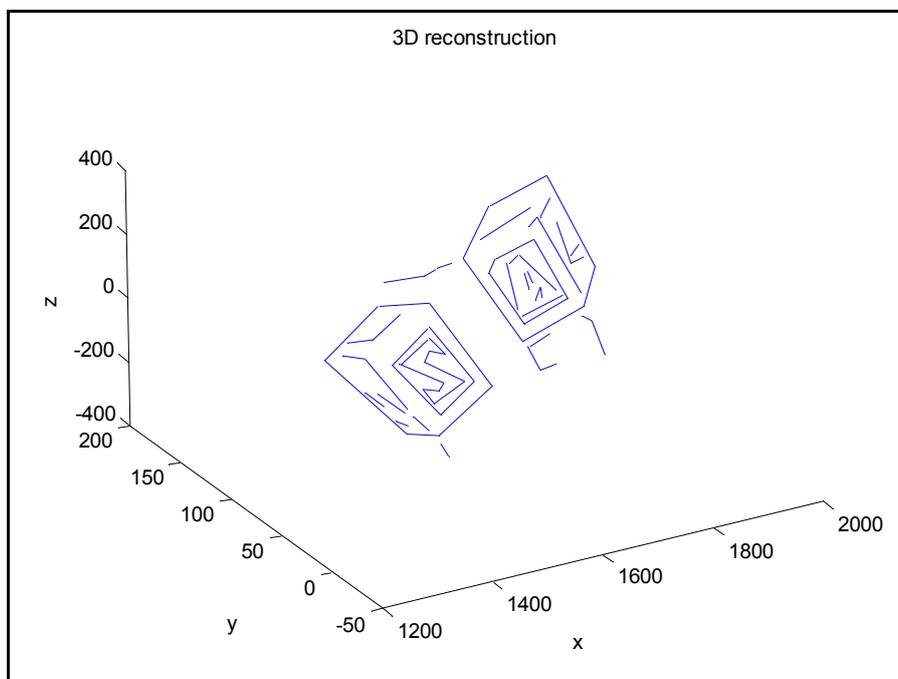
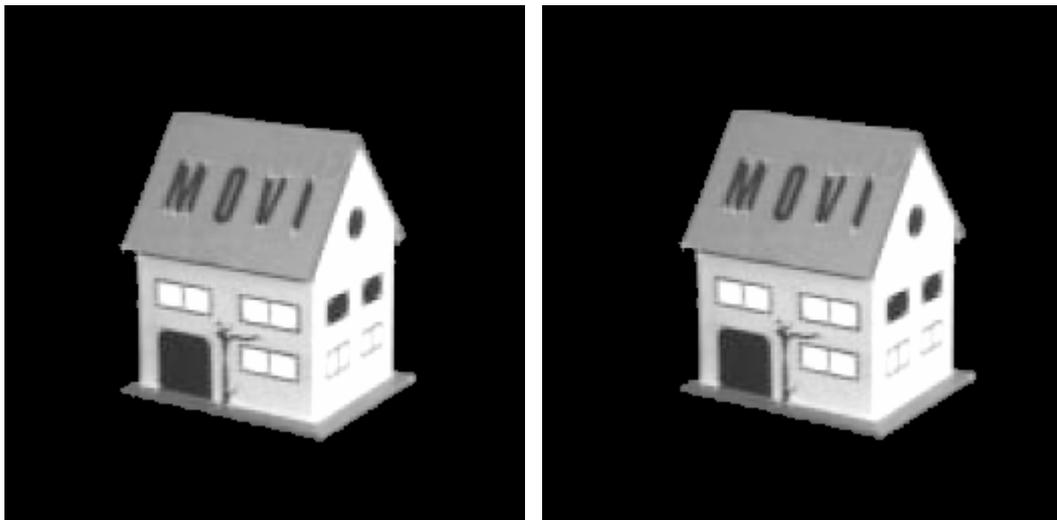


Figure 5.30: Résultat de reconstruction de la paire stéréo « 2dés », deuxième cas

Exemple 2 : reconstruction de la paire stéréoscopique «maison» :



(a) image gauche

(b) image droite

Figure 5.31 : La paire stéréoscopique «maison» originale en niveau de gris.

Premier cas : on applique l'algorithme de la reconstruction 3D pour le résultat de l'appariement de la figure 5.21 ($T_{fen} = 5$) :

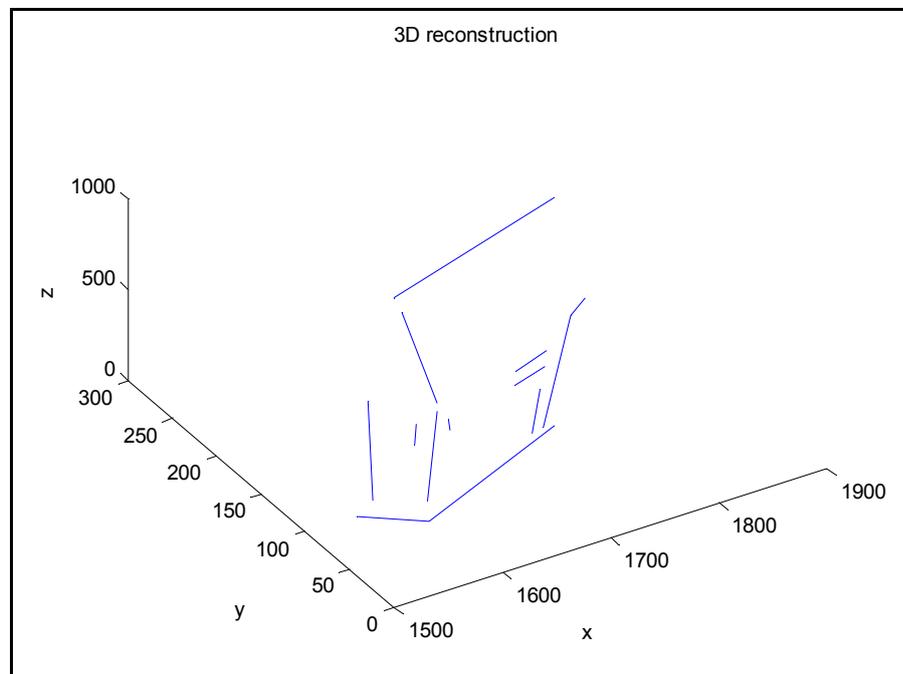


Figure 5.32: Résultat de reconstruction de la paire stéréo «maison», premier cas.

Deuxième cas : on applique l'algorithme de la reconstruction 3D pour le résultat de l'appariement de la figure 5.23 ($T_{fen} = 20$) :

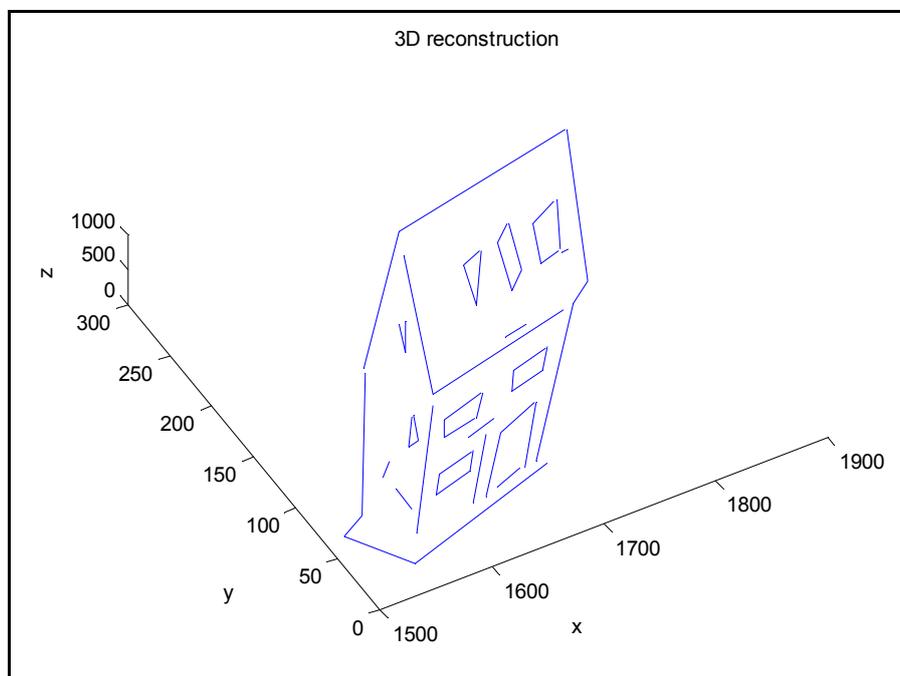


Figure 5.33: Résultat de reconstruction de la paire stéréo « Maison ».deuxième cas

Discussion :

D'après les résultats de cette phase de calcul, il est remarquable que la qualité de la reconstruction 3D est liée directement à la qualité de l'appariement, en effet la précision et la richesse de la carte 3D reconstruite dépend de la quantité et la précision des résultats fournies par l'algorithme de mise en correspondance, ainsi que la précision des paramètres intrinsèques et extrinsèques du système d'acquisition d'images (la précision de la phase de calibrage du banc stéréoscopique).

5.6 Conclusion :

Les résultats expérimentaux montrent qu'on obtient des très bons résultats des algorithmes de traitement bas niveau (l'extraction de contour, chaînage, et l'approximation polygonale), et ce la tel que soit la complexité de la scène observée. Le bon choix des paramètres de l'approximation polygonale « e_m » permet un meilleur suivi des points contours, et par conséquent, nous conservons la totalité de l'information de l'image contours.

La méthode de l'appariement des segments de droite adoptée dans notre travail a donné de bons résultats et se révèle être robuste et efficace, à condition de faire un bon choix de l'ensemble des paramètres de cette étape.

La reconstruction 3D, exploite les couples de primitive (segments de droite) stéréo correspondance, donc la qualité de cette dernière dépend directement de la qualité de l'appariement. Bien entendu, des erreurs de reconstruction persistent encore qui proviennent notamment des imprécisions qui affectent les paramètres géométriques des caméras issus d'un processus de calibrage.

Comme nous avons expliqué, les choix des paramètres introduits dans les différentes étapes de calcul ont une grande influence sur les qualités des résultats, le problème c'est que nous ne pouvons pas faire un choix unique indépendamment du contenu de l'image, donc ces paramètres sont fixés d'une manière expérimentale.

Signalons aussi, que la reconstruction à base de l'indice visuelle segments de droite n'est pas applicable à tout sorte d'image, comme par exemple les images satellitaire ou médicale. Mais elle est bien adaptée à des scènes structurée ou semi structurée comme les milieux d'intérieur ou urbain.

CONCLUSION GENERALE ET PERSPECTIVES

Le travail présenté dans cette mémoire a porté sur l'implémentation d'un module pour la reconstruction tridimensionnelle de l'environnement. Ce problème a été abordé dans le cadre d'une observation stéréoscopique. Le résultat de cet travail consiste à réaliser un ensemble d'outils pour obtenir un système de modélisation 3D du relief automatique. Ce processus nécessite plusieurs étapes, qui étaient étudiées durant cette mémoire; chacune a fait l'objet d'un chapitre du manuscrit.

Dans un premier temps, nous nous sommes attachés à une introduction à la vision artificielle dans laquelle s'intègre ce projet, nous avons aussi donné un aperçu sur les principaux domaines d'applications de cette dernière. La modélisation du capteur (caméra) modèle sténopé, est présentée avec les équations permettant de passer d'un point dans le repère du monde au point correspondant dans le repère image.

Dans le chapitre 2, nous avons présenté en premier lieu la première phase de tout système de reconstruction du relief, il s'agit de déterminer l'indice visuel à utiliser, notre choix s'est porté sur un élément descriptif essentiel des objets de la scène, c'est le segment de droite extraite à partir de l'image contours.

En deuxième lieu, nous avons développé les différentes étapes permettant d'aboutir à image sous forme de liste de segments de droite. A savoir, la segmentation d'images en contours, suivi d'un chaînage pour manipuler des listes de chaînes, l'approximation polygonale pour aboutir à une description plus significative des chaînes.

Après le traitement de bas niveau sur chaque image de la paire stéréo, nous cherchons à trouver pour chaque primitive (segment de droite) de l'image gauche son homologue dans l'image droite. Notre méthode de mise en correspondance, dite «appariement par relaxation», s'appuie sur le calcul, pour chaque couple (S_g, S_d) , une valeur de vraisemblance Q_{ij} on utilisant les attributs géométriques des segments.

Pour augmenter la fiabilité d'appariement, nous avons utilisé d'autres contraintes tel que l'unicité et la compatibilité qui s'appuient sur le calcul du graphe de voisinage.

La dernière phase dans notre boucle de traitement, consiste à calculer les segments de droite 3D, pour cela nous avons implémenté une méthode de reconstruction en déterminant la projection des points 3D sur les deux plans images.

Enfin, nous souhaitons terminer en évoquant des travaux que nous pourrions mener dans le futur sur la thématique présentée dans ce projet. Tout d'abord, des améliorations devraient être apportées au logiciel que nous avons développé, par exemple ajouter un module qui permet le calibrage des caméras pour augmenter la performance d'utilisation.

Nous avons validé notre algorithme de reconstruction tridimensionnelle par des tests sur des images déjà calibrées. Il serait plus intéressant de pouvoir effectuer le calibrage du banc stéréoscopique monté sur le robot mobile ATRV2. Par conséquent, faire le traitement et la reconstruction des images acquises par ce dernier, et donc l'intégration des algorithmes implémentés sur le robot mobile, pour lui permettre d'exploiter les informations fournies par le banc stéréoscopique en plus des informations des ultrasons utilisées actuellement au sein de l'équipe, et cela pour atteindre le but de doter le robot mobile ATRV2 par un système de perception multi sensorielle.

APPENDICE A

LE ROBOT MOBILE *ATRV2*

Cet appendice a été rédigé à partir du document de Monsieur Djekoune Oualid [60] de l'équipe de recherche «Vision artificielle et robotique» du centre de recherche CDTA.

A.1 Présentation robot mobile *ATRV2* :

Le robot mobile *ATRV2* (All Terrain Robot vehicle 2) est une plate forme expérimentale construite par la société iRobot (figure A.1), peut se déplacer un terrain accidenté et possédant quatre roues motrices différentielles commandées par quatre moteurs indépendants. La paire de roues du même côté roule avec une même vitesse et sont commandées indépendamment de celles du cote opposé. Si les deux paires de roues roulent avec une même vitesse, le robot mobile se déplace dans le même sens de déplacement des roues, mais s'ils roulent avec des vitesses opposées, le robot va tourner sur lui-même. Et si ces deux paires roulent avec des vitesses différentes (un coté roule plus vite que l'autre), le robot suivra une courbe du côté roulant avec la vitesse la plus faible.

Le *ATRV2* est contrôlé par le système « rFLEX » de la société iRobot fournissant des contrôles standardisés et une interface utilisateur facile à utiliser et conviviale permettant l'interaction avec l'utilisateur à travers un ensemble de menus permettent la gestion des capteurs, moteurs, et autres systèmes embarqués. Ce système est considéré comme le système nerveux central du robot *ATRV2* où les câbles, nœuds et hubs son facile à configurer et son réseau est très optimisé pour lier les capteurs embarqués et les actionneurs à son ordinateur du traitement central (cerveau) (figure A.2).

Le contrôle software se fait par Mobility, logiciel distribué, pour la construction de programmes de contrôle d'un seul ou plusieurs robot de la famille iRobot. Il consiste en :

- Un ensemble d'outils logiciel ;
- Un modèle objet pour le logiciel du robot mobile ;
- Un ensemble de modules de contrôle de base du robot mobile ;
- Une structure de classes orientées objet pour simplifier le développement du code.



Figure A.1 : Le robot mobile ATRV2

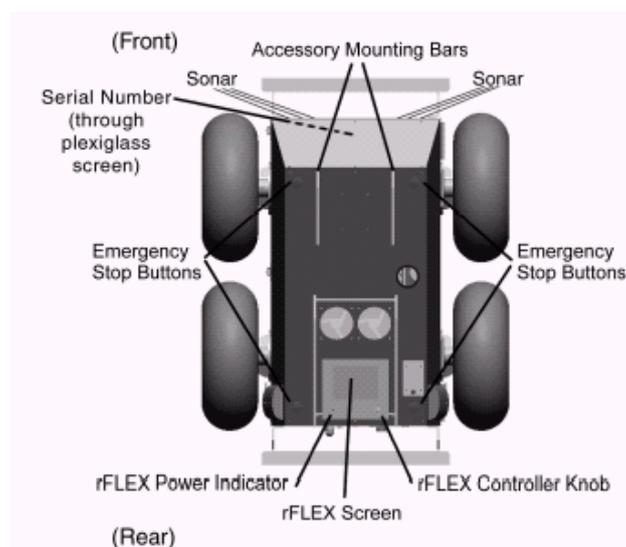


Figure A.2 : vue de haut du robot mobile ATRV2

A.2 Architecture Matérielle du robot Mobile ATRV2 :

Chaque roue du robot mobile *ATRV2* est munie d'un odomètre qui sert à mesurer son déplacement angulaire et à calculer le déplacement et l'orientation effectués par le robot. Malheureusement, ce type de capteur n'est pas toujours précis et donne des informations parfois erronées dans les cas suivants: la rugosité de la surface du sol, les ondulations des roues, et la perte d'adhérence des roues, etc.

Les roues motrices du robot mobile permettent le déplacement de ce dernier ainsi que le contrôle de sa direction en fonction de la différence des vitesses linéaires de chacun de ses côtés (côté gauche et côté droit) (figure A.3).

ATRV2 possède également une ceinture de douze capteurs télémétriques à ultrasons, six sur sa face avant, deux sur sa face arrière et deux sur chaque côté. Ils sont utilisés principalement pour la détection et l'évitement des obstacles se trouvant à proximité du robot mobile. Avec les six capteurs sur sa face avant, le robot mobile *ATRV2* peut facilement décider avec quelle direction il peut tourner pour éviter les obstacles rencontrés. Chacun de ces capteurs reçoit des échos des objets se trouvant à $\approx \pm 15$ degrés de la ligne médiane du capteur et jusqu'à une distance de dix mètres.

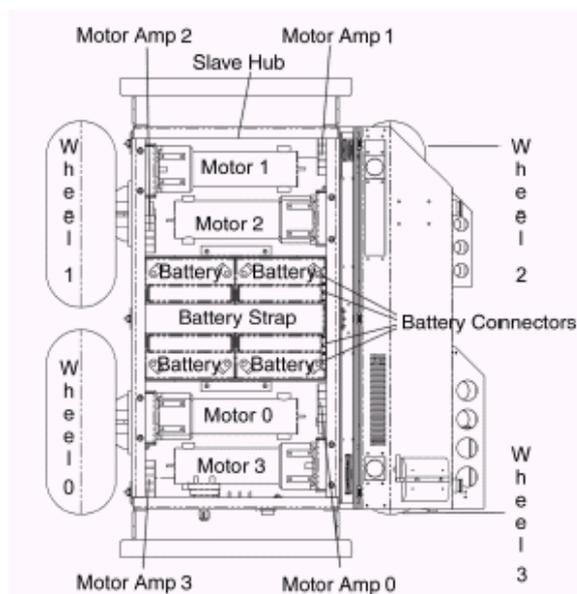


Figure A.3 : Vue de l'intérieur du robot mobile *ATRV2*

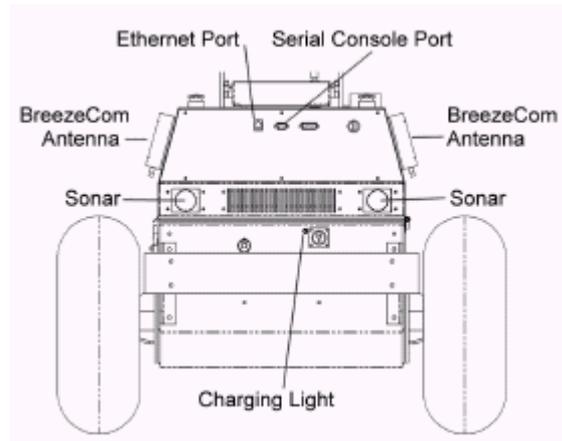


Figure A.4 : vue arrière du robot mobile *ATRV2*

Sur le robot *ATRV2*, le système de vision est constitué par une tête stéréo constituée de caméras couleurs montées sur une tourelle Pan-Tilt. Ces caméras permettent d'obtenir des images couleur de haute qualité (résolution maximale 1024 x768).

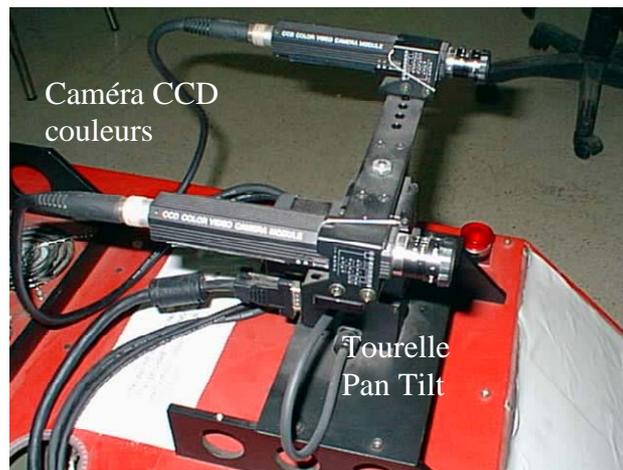


Figure A. 5: le banc stéréoscopique.

A.3 Architecture Logicielle du robot Mobile *ATRV2* :

Le logiciel « Mobility » utilisé basé sur CORBA, collecte les données en provenance des capteurs embarquées, les interprètes et les exécute. En utilisant CORBA tous les capteurs et actionneurs peuvent être vu comme des ressources Internet. C'est une librairie composée de programmes écrits en C et C++ contrôlant le hardware du robot mobile *ATRV2*.

La connexion avec les différents capteurs embarqués se fait en exécutant le programme serveur qui réagit réciproquement avec le capteur. Ce dernier, se connectera à son tour au name server. Le capteur peut être donc connecté et contrôlé par un simple programme écrit en C incluant les fonctions et procédures fournis avec Mobility.

A.4 Géométrie du robot mobile ATRV2 :

La géométrie du robot est illustrée dans la figure A.6, il peut être représenté par rectangle de taille $105 \times 80 \times 65$. Cette représentation sera utile lors de la définition de l'espace nécessaire pour son déplacement.

Le robot a un poids de $120Kg$, y compris les batteries. Il peut transporter une charge utile de $100Kg$ à une vitesse maximale de $1.5m/s$. Les rotations faire avancer ou reculer le robot, tourner sur lui-même ou suivre des courbes de déplacement.

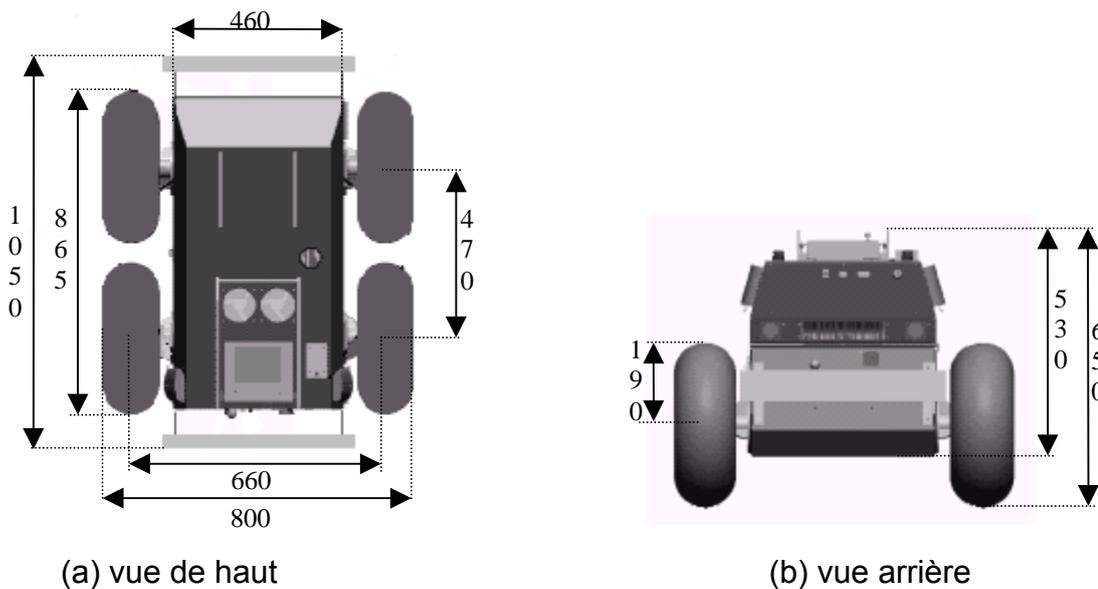


Figure A.6 : La géométrie du robot mobile ATRV2

A.5 Capacité de mouvement du robot mobile ATRV2 :

Le mouvement est crée par la rotation des quatre roues motrices du robot. Les deux paires de roues correspondant aux coté gauche et droit du robot possèdent une vitesse de rotation instantanée $\omega_d(t)$ et $\omega_g(t)$.

Nous pouvons expliquer le comportement du robot mobile pour les cas présentés par la figureA.7. Dans le cas des vitesses $v_d = v_g$, illustré par la figure

A.6a, le déplacement est rectiligne est égal à v_d (ou v_g). La vitesse de rotation du robot Ω est nulle. Si $v_d < v_g$, figure A.6b, le déplacement est un virage à droite, c'est-à-dire, avec une vitesse angulaire $\Omega < 0$. Dans le cas contraire, présenté par la figure A.6c, $v_d > v_g$, on obtient un déplacement avec un virage à gauche qui correspond à une vitesse angulaire $\Omega > 0$. Pour le cas spécial, où $v_d = -v_g$, le déplacement sera nul, c'est-à-dire, $v = 0$, et on aura une rotation pure sur place avec une valeur $\Omega \neq 0$. Ce dernier cas est présenté par la figure A.6d pour $\Omega < 0$.

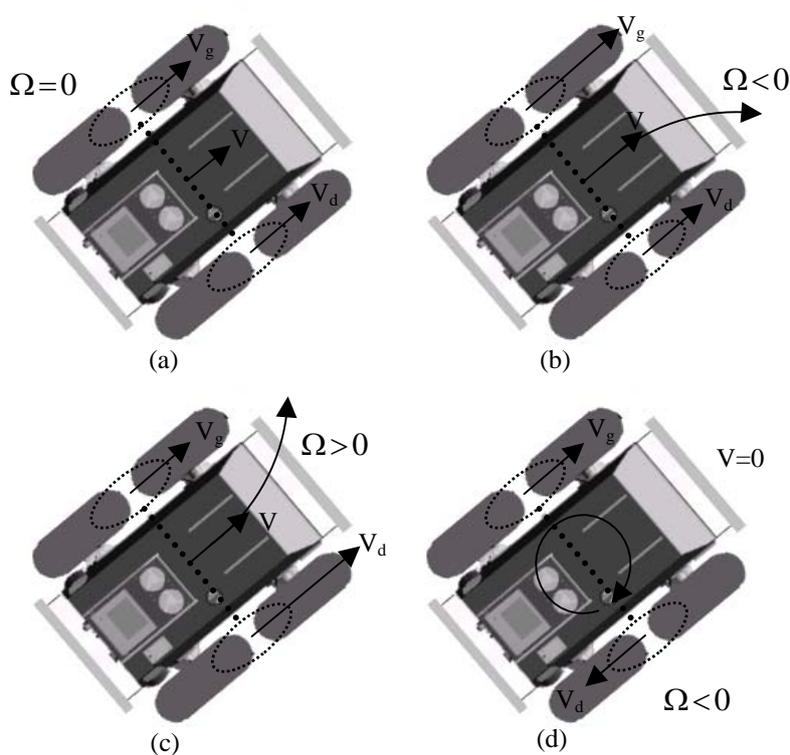


Figure A.7 : Quatre différents types de mouvement réalisable par le robot mobile ATRV2.

APPENDICE B

CALIBRAGE DU CAPTEUR (caméra)

Cet appendice a été rédigé à partir du document de Monsieur *CAUCHOIS Cyril* [17] de Groupe de Recherche sur l'Analyse et la Commande des SYstèmes (GRACSY). Université de Picardie Jules Verne - AMIENS.

La reconstruction 3D des primitives de l'environnement impose la détermination préalable des transformations géométriques liant le plan image à l'espace tridimensionnel. En effet, les mesures de la structure 3D des objets de l'environnement perçue par la caméra sont exprimées dans le repère de référence (R_{monde}) qui est différent du repère de projection ($R_{caméra}$). Il est ainsi nécessaire de déterminer la transformation rigide liant le repère de la scène au repère de la caméra.

B.1 Principe de calibrage

Le principe de l'étalonnage est d'utiliser une grille (mire) ou tout autre objet, pour lesquels les positions de points marqués sur l'objet, appelés points d'étalonnage (ou de contrôle), sont connues et dont on observe la position dans les images. Ces points d'étalonnage sont sélectionnés dans des zones particulières de l'image (coins de la mire, intersections de droites, points de la mire), ou peuvent être toute autre primitive extraite des images. À partir de cette observation et connaissant les points de l'espace à l'origine des points projetés dans l'image, on peut en déduire les paramètres de la transformation qui amène un point appartenant à la mire vers le point correspondant dans l'image [61].

L'étalonnage est donc un processus qui permet d'estimer les paramètres d'un modèle de caméra. Plus précisément, le problème de l'étalonnage peut être présenté comme suit : étant donné un ensemble de points de coordonnées 3D (X, Y, Z) connues, il s'agit de déterminer les paramètres de la projection associée au capteur

pour que leurs projections correspondent au mieux aux points correspondants extraits des images. La fonction de projection peut être décomposée en deux transformations distinctes : la première est la transformation entre le repère de l'espace 3D et le système de coordonnées de la caméra; ce déplacement 3D qui englobe une rotation et une translation forme ce qu'on appelle les paramètres extrinsèques du capteur. La seconde est la transformation entre les coordonnées des points 3D, exprimées dans le repère attaché à la caméra et les points 2D dans le plan image de la caméra.

B.2 Méthodes d'étalonnage :

Ce paragraphe présente une synthèse non exhaustive des principales méthodes d'étalonnage mises au point dans le cadre de travaux portant sur la vision par ordinateur.

B.2.1 Etalonnage selon Tsai :

Cette méthode est devenue une référence depuis de nombreuses années. Effectivement, un grand nombre de méthodes en reprennent les fondements et son modèle de caméra.

Roger Tsai utilise un déplacement du plan d'étalonnage et modélise la formation de l'image en déterminant des facteurs d'échelles qui permettent le passage entre le plan image et l'image échantillonnée. Salvi [63] a renommé cette technique « étalonnage à deux étapes ». La première étape utilise une modélisation simplifiée de la caméra, tandis que la deuxième raffine l'étalonnage en estimant un coefficient qui exprime la distorsion radiale du système optique. Dans cette méthode, la mire utilisée est le damier repris dans un grand nombre d'autres approches. Le plan d'étalonnage doit se déplacer de façon parallèle à sa position initiale dans le but d'utiliser une contrainte de parallélisme.

B.2.2 Etalonnage selon Faugeras et Toscani (La contrainte $\|m_3\| = 1$)

Cette méthode d'étalonnage est parmi les méthodes les plus utilisées [63]. Toscani a enveloppé sa procédure d'étalonnage d'une caméra en utilisant un formalisme mathématique couramment employé et d'implémentation immédiate. Par l'utilisation d'une mire d'étalonnage et par l'exploitation des équations de la projection

perspective linéaire du modèle sténopé, Toscani extrait les paramètres intrinsèques et extrinsèques par la résolution d'un système à $2n$ équations, n étant le nombre de points non coplanaires appartenant à la mire d'étalonnage (n doit être supérieur à 6). Une contrainte physique est adoptée pour résoudre ce système d'équations. Elle repose sur des contraintes d'orthogonalité de la matrice de rotation qui ne subit pas l'influence des paramètres intrinsèques. Cette contrainte est appelée contrainte de Faugeras-Toscani (la contrainte $\|m_3\| = 1$)

$$\left(\begin{array}{cccccccccccc} x^i & y^i & z^i & 1 & 0 & 0 & 0 & 0 & -u_i x^i & -u_i y^i & -u_i z^i & -u_i \\ 0 & 0 & 0 & 0 & x^i & y^i & z^i & 1 & -v_i x^i & -v_i y^i & -v_i z^i & -v_i \\ \vdots & & & & & & & & & & & \\ \vdots & & & & & & & & & & & \end{array} \right) \begin{pmatrix} M_{11} \\ M_{12} \\ M_{13} \\ M_{14} \\ M_{21} \\ M_{22} \\ M_{23} \\ M_{24} \\ M_{31} \\ M_{32} \\ M_{33} \\ M_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \dots B.1$$

B.2.3 Etalonnage selon Zhang :

Cette méthode est basée sur la détection d'une mire d'étalonnage représentant un damier dont les caractéristiques sont connues et mesurées. L'utilisation d'une mire plane simplifie sa fabrication et permet une simplification du modèle à adopter car la coordonnée Z du point dans l'espace est annulée en liant le repère du monde au plan. Zhang [65], [66] a proposé une nouvelle technique flexible pour étalonner facilement une caméra.

La technique utilise une mire d'étalonnage vue à partir d'au moins trois points de vue d'orientations différentes. La caméra et la mire peuvent être déplacées librement sans avoir besoin de connaître le mouvement. La distorsion radiale de l'objectif est modélisée. Le processus d'étalonnage proposé se compose d'une recherche d'une solution initiale, suivie d'une optimisation non linéaire basée sur le critère du maximum de vraisemblance. Cette méthode reprend l'approche proposée par Tsai, mais est d'une utilisation plus souple. En effet, le fait de pouvoir utiliser simplement un plan déplacé d'une image à l'autre sans que le mouvement ait besoin d'être connu lui donne une grande flexibilité.

B.3 Les mires de calibration :

Afin de calibrer une caméra, il est nécessaire d'utiliser une mire en trois dimensions dont les formes la caractérisant sont parfaitement répertoriées et mesurées dans un repère spécifique. Dans la majorité des cas étudiés, ces mires sont constituées de plusieurs plans comportant des formes géométriques simples afin de faciliter la reconnaissance de ces formes dans le plan image de la caméra. La mise en correspondance des objets de la mire et de leur projection respective dans le plan image permet de résoudre l'équation B.1, et de caractériser les paramètres de la caméra.

La qualité des mires utilisées et la précision d'extraction des mesures sont directement proportionnelles aux résultats de calibration. Dans la très large bibliographie qui existe sur la calibration, nous avons répertorié quelques mires caractéristiques qui sont utilisées lors de ce processus. Nous avons regroupé ces différentes mires en trois ensembles différents.

B.3.1 Les mires à motifs rectangulaires :

La mire utilisée par R. Y. Tsai est une feuille à lettrage installée sur un bloc métallique pour la rigidité. L'ensemble est posé sur une table à déplacement micrométrique avec une précision de 0.1 mm. Cette mire comporte 16 carrés.

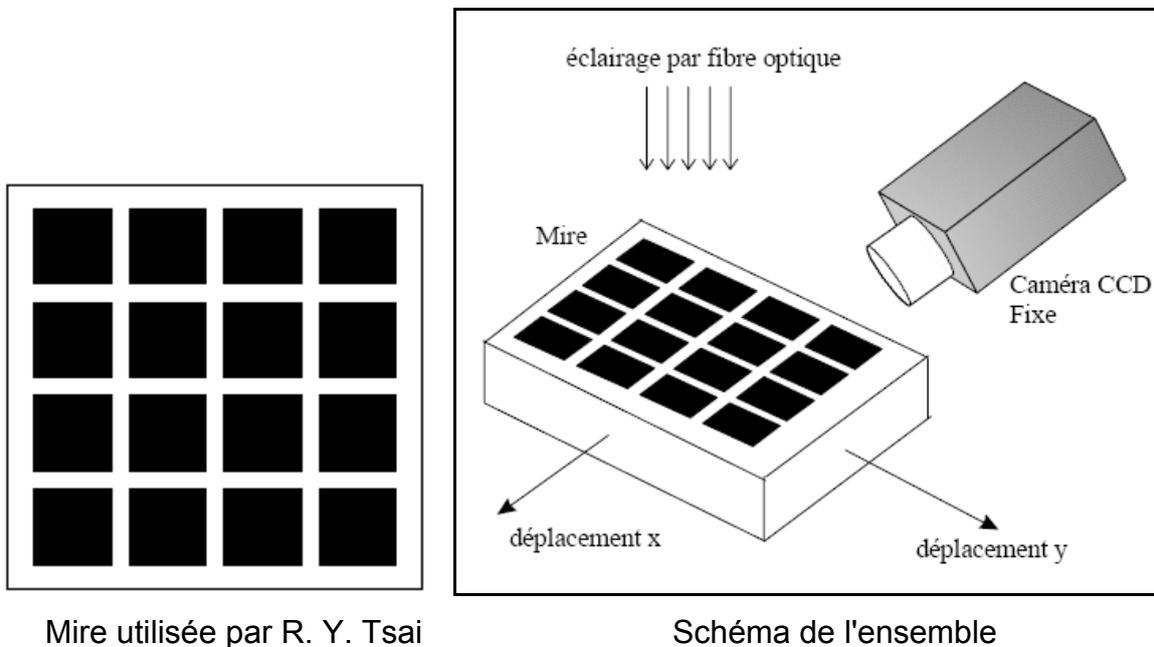


Figure B.1 : La mire utilisée par R. Y. Tsai et le schéma de l'acquisition.

La mire utilisée à l'INRIA

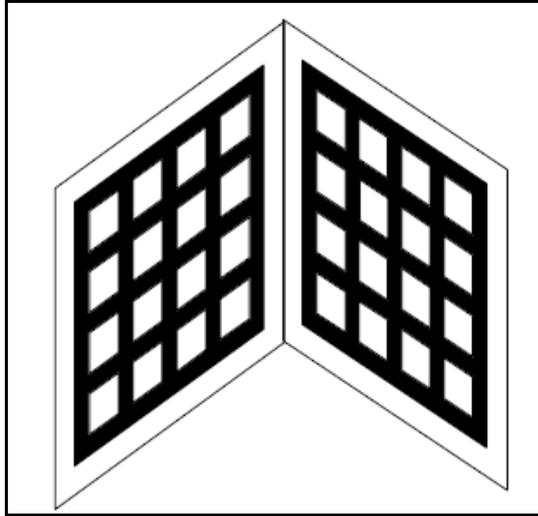


Figure B.2 : La mire utilisée à l'INRIA.

B.3.2 Les mires à motifs circulaires :

La mire est composée de 36 disques noirs (6 lignes) de 2mm de diamètre, séparés de 5 mm avec une précision de $\pm 1\mu m$.

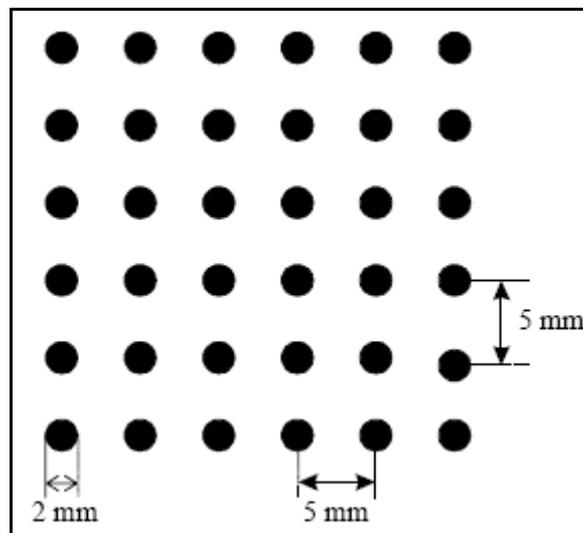


Figure B.3 : La mire à motifs circulaires

Dans autres travaux la mire utilisée est une mire métallique bi-plan composée de 7x6 ellipses par plan (Figure B.4).

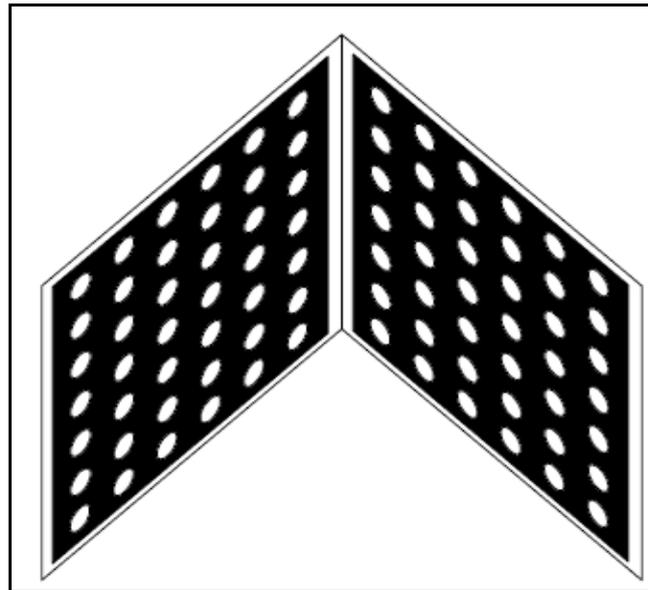


Figure B.4 : La mire bi-plan à motifs circulaires

B.3.3 Les autres types de mires :

Nous trouvons aussi d'autre forme mire de calibration, tel que celle de la figure B.5 qui est constituée d'un quadrillage 6x6 dont les intersections n'existent pas. Le trait a une épaisseur de 3 points et chaque carré mesure 1 pouce (figure B.5a).

La mire de la figure B.5b, a été utilisée pour mesurer les phénomènes de distorsions. La mire de la figure B.5c a été utilisée pour calibrer des caméras utilisant des lentilles à forte distorsion.

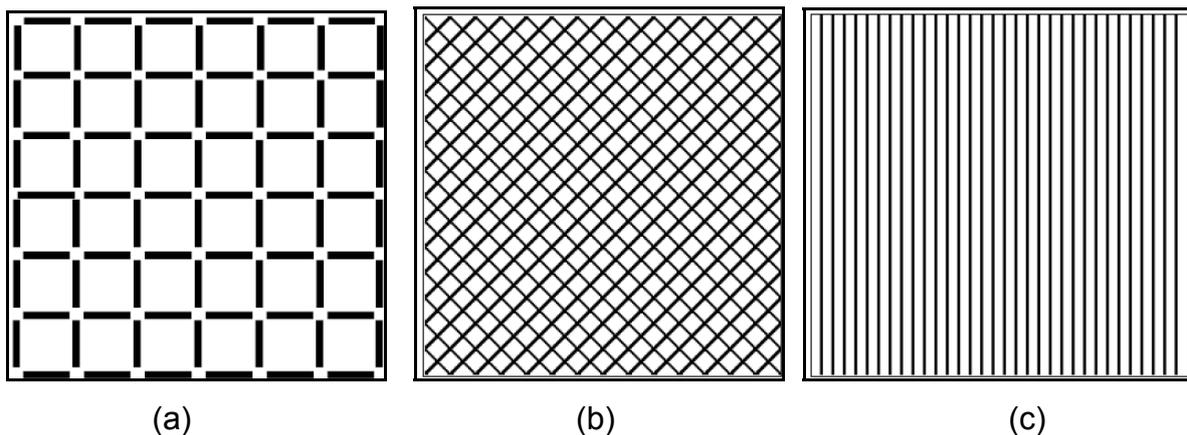


Figure B.5 : Quelques autres formes des mires

REFERENCES

1. D. Marr « Cooperative computation of stereo disparity », science vol 194, N° 4262, 15 October 1976.
2. A. M. Hernandez, «Vision dynamique pour la navigation d'un robot mobile », thèse doctorat de l'Institut National Polytechnique de Toulouse, 2004.
3. S. Merkle, R. yates, «Computer vision algorithms for autonomous mobile robot map building and path planning», international conference on system sciences IEEE, page 292-301, 1998.
4. M. Khatib «Contrôle du mouvement d'un robot mobile par retour sensoriel», thèse doctorat, université de Toulouse France, Décembre 1996.
5. S. BOUKIR «Reconstruction 3D d'un environnement statique par vision active», thèse doctorat de l'université de Rennes I, Octobre 1993.
6. R. HORAUD, « La Vision Artificielle », conférence international en traitement de signal, volume 4 - n° 3 - 1987
7. B.W. Dautcourt «Perception de la distance par mise en correspondance de régions entre des images stéréoscopiques » thèse doctorat de l'institut national polytechnique de Lorraine, mars 1988
8. L. Delahoche, C. Pégard, M. Mouaddib, P. Vasseur « incremental map building for mobile robot navigation in an indoor environment », international conference en robotics & automation IEEE, May 1998.
9. A. Montanvert, J.M. Chassery, et C. Charles «décomposition de formes complexes appliquées à un problème de trajectographie sous-maine » Actes du congrès MARI-87, pages 411-418, vol. 2 Paris, 1987.
10. P. Bonton, S. Gidel, L. Grouche, G.Rongier «Tri chromosomique par traitement d'images» du congrès MARI-87, pages 326-333, vol. 2 Paris, 1987.

11. N. Keskes, A. Boulanouar, et O.D. Faugeras «Application of image analysis technique to seismic data» International Conference on Acoustic Signal and Speech Processing (ICASSP), pages 855-857, Paris, 1982.
12. N. PAPANODITIS «Reconstruction tridimensionnelle de paysages péri-urbains en imagerie stéréoscopique satellitale haute résolution» thèse de doctorat en science janvier 1998.
13. M. Hernandez «Vision dynamique pour la navigation d'un robot mobile» thèse de doctorat, l'Institut National Polytechnique de Toulouse 2004.
14. G.Toscani. «Système de calibration optique et perception du mouvement en vision». Thèse de doctorat, université de paris-sud Orsay 1987.
15. M. Busque «Reconnaissance d'objets à partir de points 3D obtenus par appariement stéréoscopique pour la localisation absolue d'un robot mobile», thèse grade de maîtres sciences, Faculté des études supérieures de l'Université Laval, février 2001
16. M. Busque «Reconnaissance d'objets dans des scènes tridimensionnelles» Rapport de stage, septembre 1998.
17. C. Cyril, «Calibration du capteur de vision omnidirectionnelle: SYCLOP », Diplôme d'Etudes Approfondies, Université de Picardie – Jules Verne, Août 1998.
18. M. Bénallal «Système de calibration de caméra localisation de forme polyédrique par vision monoculaire», thèses doctorat de l'école des mines de Paris, novembre 2002.
19. O. Aït-Aider, «Localisation référencée modèle d'un robot mobile d'intérieur» thèses doctorat de l'universite d'Evry, Décembre 2002.
20. F. Chaumtte, P. Rives « Réalisation et calibration d'un système expérimental de vision compose d'une caméra mobile embarquée sur un robot manipulateurs», rapport de recherche IRISA N°994, mars 1989.
21. A. Far «Analyse multi-images. application à l'extraction contrôlée d'indices images et à la détermination de descriptions scéniques», thèses doctorat de Strasbourg I, Décembre 2005.
22. A. Hassak «Génération des scènes tridimensionnelles à partir d'un banc stéréoscopique» thèse magister de l'école militaire polytechnique d'Alger 2005.

23. J. M. D. Filliat «Global localization and topological map-learning for robot navigation», thèse doctorat université de Picardie Jules Verne, 2002.
24. D. Filliat and J. A. Meyer. «Map-based navigation in mobile robots» a review of map learning and path planning strategies, 2002.
25. M. Elfes «High resolution maps from wide angular sensors» ICRA -85, p116-121 IEEE Press 1985.
26. Thrun, «Learning metric topological maps for indoor mobile robot navigation» Artificial Intelligence, p 21-71. Mars 1999.
27. M. M. Ben Khelifa, M. Bouchouicha « Du traitement de l'information visuelle à la planification de trajectoires application à la robotique mobile d'assistance», rapport de stage, laboratoire SIS/AI université de Toulon et du Var, 2003.
28. Nikhil R.Pal & Sankar K.Pal «A review on image segmentation techniques» Pattern Recognition, vol. 26, pages 1277 – 1294 September 1993.
29. Nicholas Ayache «Construction et Fusion de Représentation Visuelles 3D Application à la robotique Mobile», thèse doctorat en sciences, université de Paris-Sud, 1988.
30. M. J. Stephens, R. J. Blisset « Outdoor vehicle navigation using passive 3D vision», Plessey Research and technology, ch 2752 page 556-562; IEEE 1989.
31. H.A. Beyer « Accurate Calibration of CCD-Caméras », International Conference on Computer Vision and Pattern Recognition, Urbana-Chmpaing, Illinois, USA, page 96 -101, 1992.
32. Radu Horaud, Olivier Monga « vision par ordinateur outils fondamentaux » Editions Hermés , Paris 1993.
33. [H. Bulata](#) «Modélisation d'un environnement structure et localisation sur amers pour la navigation d'un robot mobile autonome», thèse Doctorat, Université Paul Sabatier, Toulouse, Mai 1996.
34. Juan Gabriel Avina Cervantes « Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi structuré » thèse Doctorat de l'Institut National Polytechnique de Toulouse, février 2005.
35. M. Buffa, O. Faugeras et Z. Zhang « A stereovision-Based navigation system for a mobile robot » Rapport de recherche INRIA N° 1895.

36. Rachid Deriche « Techniques d'extraction de contours », Support de cour, INRIA Sophia-Antipolis, 2004.
37. I. Bloch, Y. Cousseau, «Le traitement des images », polycopie du cours ANIM tome 2 ; Paris, septembre 2004.
38. G. Giraudon, P. Garnesson « chaînage efficace de contour » INRIA research report 605, February 1987.
39. O. Djekoune, M. Benkhelif, N. Zenati et K. Achour « application de la transformée de Hough dans l'appariement des images de scènes » p.343-358, CARI 2000.
40. M. Xie, P. Rives, «Estimation des segments 2D : un algorithme robuste», publication interne IRISA n°429, Septembre 1988.
41. Philippe Garnesson, et Gérard Giraudon « polygonal approximation overview and perspectives» INRIA rapport de recherche N° 1621. Juin 1991.
42. Augustin Lux, «Algorithmique et contrôle en vision par ordinateur», Thèse de docteur en sciences Informatique, Université de Grenoble, 20 septembre 1985.
43. T. Pavlidis, «Structural Pattern Recognition », Springer-Verlag, Berlin Heidelberg New York, pp. 147-211, 1979.
44. C. RABAUD «mise en correspondance des images stéréoscopiques», support de cour, Laboratoire d'Informatique de Robotique et de Micro-électronique, France 2002.
45. Zhong-Dan LAN, «Méthodes robustes en vision : Application aux appariements Visuels», thèse de doctorat de l'institut national polytechnique de Grenoble, Mai 1997.
46. F. Devernay «vision stéréoscopique et propriétés différentielles des surfaces», thèse doctorat, école polytechnique Palaiseau France 1997.
47. Christophe GARCIA, Saïda BOUAKAZ, Denis VANDORPE, «Réseau de Hopfield, un outil pour la mise en correspondance», Rapport de recherche Université Lyon I, Traitement du Signal - Volume 12 - n° 2, 1995.
48. FRANKOT R., CHELLAPPAR R. « A method for enforcing integrability in shape from shading algorithms », Proc. Of 1st Int. Conf. On computer vision, Londres, pp 118 – 127, 1987.

49. K. Ghanem , M.C Batouche & B. Boufama « Nouvelle Méthode de mise en Correspondance Dense pour la Résolution du problème de l'Occlusion dans le domaine de la Réalité Augmentée », Equipe vision et infographie, Laboratoire Lire, Université Mentouri .Constantine. Algérie. 2006.
50. Cordelia SCHMID, «Appariement d'images par invariants locaux de niveaux de gris application a l'indexation d'une base d'objets », thèse de doctorat de l'institut national polytechnique de Grenoble, juillet 1996.
51. Nikom Suvonvorn, «Mise en correspondance d'images pour l'analyse du mouvement et la stéréovision », thèse doctorat, université de Paris-Sud, 18 décembre 2006.
52. T. Skordas, « mise en correspondance et reconstruction stéréo utilisant une description structurelle des images », thèse doctorat, institut national polytechnique de Grenoble. Octobre 1988.
53. E. Trucco, A. Verri, «Introductory techniques for 3D computer vision», prentice hall, 1998.
54. A.P Pentland « Local shading analysis » IEEE Trans. On PAMI, vol 6 pp. 170 – 187 mars 1984.
55. Groupe Perception, Analyse d'Images et commande « Contribution d'un éclairage structuré à l'asservissement visuel », rapport de recherche INRIA, Octobre 2002.
56. J. Pagès, C. Collewet, F. Chaumette, J. salvi «Couplage asservissement visuel et lumière structurée pour la réalisation d'une tâche de positionnement plan à plan», University of Girona, et Université de Beaulieu France, 2000.
57. P. Lanvin « Suivi multicapteurs d'objet 3D par filtrage non-linéaire : une approche dense », Université du Buisson France 1999.
58. T. Q. Phong, R. Horaud, A. Yassine, P.D. Tao « Object pose from 2D to 3D point and line correspondences », International of computer vision, 1995.
59. M. Gaetan «stereo vision for mobile robotics », search report, Micro-Engineering laboratory, Swiss Federal institute of technology of Lausanne, Swiss 2003.
60. O. Djekoune «Le robot mobile ATRV2 : définitions et cinématique », rapport d'activité CDTA N°1 , 2002.
61. DORNAIKA « contributions a l'intégration vision /robotique : calibrage, localisation et asservissement », thèse doctorat de l'institut national polytechnique de Grenoble, 25 Septembre 1995.

62. H.A. Beyer « Accurate Calibration of CCD-Cameras », International Conference on Computer Vision and Pattern Recognition, Urbana-Champaign, Illinois, USA, page 96 -101, 1992.
63. J. Salvi, X. Armangué, and J. Batlle. « A comparative review of camera calibrating methods with accuracy evaluation» *Pattern Recognition*, 35(1), pages 1617–1635, 2002.
64. G. Toscani. «Système de calibration optique et perception du mouvement en vision artificielle », Thèse de doctorat, Université d'Orsay, Décembre 1987.
65. Z. Zhang. « Flexible camera calibration by viewing a plane from unknown orientations». In Proceedings of International Conference on Computer Vision, Corfu, Greece, pages 666–673, Septembre 1999.
66. Z. Zhang. « A flexible new technique for camera calibration » *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), pages 1330–1334, 2000.