



HAL
open science

Reconstruction de surfaces tridimensionnelles en vision par ordinateur

Changsheng Zhao

► **To cite this version:**

Changsheng Zhao. Reconstruction de surfaces tridimensionnelles en vision par ordinateur. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1993. Français. NNT : . tel-00343706

HAL Id: tel-00343706

<https://theses.hal.science/tel-00343706>

Submitted on 2 Dec 2008

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

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

TU 20162

THESE

présentée par

ChangSheng ZHAO

pour obtenir le grade de DOCTEUR

de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

(Arrêté ministériel du 30 Mars 1992)

Spécialité: **Informatique**

**RECONSTRUCTION DE SURFACES
TRIDIMENSIONNELLES
EN VISION PAR ORDINATEUR**

Date de soutenance : 21 Décembre 1993

Composition du jury :

Président : **Pierre-Jean LAURENT**
Rapporteurs : **Serge CASTAN**
Michel DHOME
Examineurs : **Marie-Odile BERGER**
Roger MOHR

Thèse préparée au sein du laboratoire LIFIA - IMAG - INRIA Rhône - Alpes
sous la direction de Roger MOHR

Remerciements

Je tiens à exprimer mes remerciements et ma plus grande gratitude :

à **Roger Mohr** qui m'a accueilli dans son laboratoire et qui m'a guidé pendant ces années de travail. Sans les nombreux échanges que nous avons eus, ce travail n'aurait pas vu le jour.

à Monsieur **Pierre-Jean Laurent**, professeur à l'Université Joseph Fourier de Grenoble, Monsieur **Serge Castan**, professeur à l'Université Paul Sabatier de Toulouse, Monsieur **Michel Dhome**, directeur de recherche au CNRS, Mademoiselle **Marie-Odile Berger**, chargée de recherche à l'INRIA, qui m'ont fait l'honneur de faire partie du jury de soutenance.

aux stagiaires **Serge Plossu** et **Gautier Koscielny** qui m'ont apporté des idées dans ce travail.

à **Françoise Veillon**, **Radu Horaud**, **Long Quan**, **Boubakeur Boufama**, **Pascal Brand**, **Patrick Gros**, et tous les membres du projet **MOVI** qui, plus que des collègues, ont d'abord su être des amis.

à **Danièle Herzog** et **Lionel Lecorre** pour leurs aides.

à **Françoise Nayrolles**, **Henri Saya** pour leurs aides du réseaux d'informatique.

à tous les membres de ma famille qui ont su me soutenir et m'encourager pendant mes années d'étude.

Sommaire

Introduction	15
1 Poursuite et correction de contours courbes	21
1.1 Introduction	22
1.2 Corrélation	23
1.2.1 Corrélation de deux signaux	23
1.2.2 Corrélation de la vision par ordinateur	26
1.3 L'approche pyramidale de mise en correspondance	27
1.3.1 Pyramide et multi-résolution	27
1.3.2 Principe	29
1.3.3 Construction des pyramides	29
1.3.4 Calcul d'images et extraction de contours d'une pyramide	31
1.3.5 Mise en correspondance de points	32
1.3.6 Contrainte épipolaire	34
1.3.7 Résultats expérimentaux	36
1.4 Les raccordements et coupures de contours courbes	51
1.4.1 Mise en correspondance de contours	51
1.4.2 Détection et élimination des mises en correspondance chaîne à chaîne peu fiables	53
1.4.3 Détection et élimination des chaînes de contours non productives	54
1.4.4 Raccordement des chaînes de contours	54
1.4.5 Coupure des chaînes de contours	55
1.4.6 Une approche globale	55
1.4.7 Modélisation par les graphes	56
1.4.8 Principe	62
1.4.9 Description des règles de transformations de graphes	64
1.4.10 Choix des règles	69
1.4.11 Traitement d'une séquence par application des règles	70

1.4.12	Inclusion de graphes	72
1.4.13	Affinités entre chaînes de contours	75
1.4.14	Résultats expérimentaux	77
1.5	Perspectives et discussion	85
1.6	Conclusion	85
2	Reconstruction des surfaces B-splines	87
2.1	Introduction	87
2.2	Définitions et notations préliminaires	89
2.2.1	Courbes et surfaces B-splines uniformes de degré d	89
2.2.2	Contour occultant	91
2.2.3	Paramétrisation épipolaire d'une surface spatio-temporelle	93
2.3	Calibrage d'une caméra en mouvement	97
2.3.1	Le modèle de caméra	97
2.3.2	Les équations	97
2.4	Modélisation mathématique	99
2.4.1	Equations élémentaires de reconstruction de surfaces	101
2.4.2	Système d'équations non linéaires pour une séquence d'images	104
2.5	Résolution du système	104
2.5.1	Energie et moindres carrés	105
2.5.2	Energie et moindres carrés avec régularisation	105
2.6	Implémentation	107
2.7	Résultats expérimentaux	109
2.7.1	Données synthétiques	109
2.7.2	Données réelles	127
2.8	Perspectives et discussion	141
2.9	Conclusion	142
3	Raccordement des surfaces B-splines	143
3.1	Introduction	143
3.2	Notre approche	144
3.3	Le maillage	145
3.3.1	Grappe	146
3.3.2	Structure et algorithme	147
3.4	L'intersection de deux surfaces	147
3.4.1	Critères de détection	148
3.4.2	Algorithme des zones communes de deux surfaces	149
3.5	Traitement des zones extraites	151
3.5.1	Changement de repère	151
3.5.2	Interpolation d'un ensemble des points 3D	152
3.5.3	Réduction de la surface	152
3.6	Le raccordement	153
3.6.1	Difficulté du raccordement	154
3.6.2	La fusion des points	154
3.6.3	Algorithme	155
3.7	Résultats expérimentaux	158
3.7.1	Données synthétiques	158

3.7.2	Données réelles	158
3.8	Perspectives et discussion	164
3.9	Conclusion	164
Conclusion		167
A Géométrie différentielle des courbes et des surfaces		171
A.1	Les courbes	171
A.2	Première et deuxième formes fondamentales	172
B Liste des règles de transformations de graphes		175
B.1	Conventions de représentation	175
B.2	Règles de fermeture	176
B.3	Règles de coupure	184
C Interpolation de points 3D par surfaces B-splines		187
C.1	Energie et moindres carrés	187
C.2	Energie et moindres carrés avec régularisation	189
C.2.1	Régularisateur de courbure gaussienne	190
C.2.2	Régularisateur de courbure moyenne	190
C.2.3	Régularisateur de variation quadratique	190
C.2.4	Régularisateur simplifié	191

Liste des Figures

0.1	Un silo à riz typique en Chine avec ouverture sur le toit	16
0.2	Différentes images extraites d'une séquence où l'objet à reconstruire est une carafe. Nous pouvons suivre les déplacements successifs de gauche à droite et de haut en bas. Notre carafe est posée sur deux cubes dont les sommets sont traités comme des points de référence et ainsi la scène dispose d'un repère absolu. La scène contient des papiers noirs et blancs pour calibrer la caméra en mouvement.	18
1.1	Deux signaux (a) non ressemblants. (b) ressemblants. (c) inversés. . . .	25
1.2	Structure pyramidale d'une image	28
1.3	Schéma de calcul de la pyramide 4 Points	30
1.4	Schéma de calcul de la pyramide Passe-Bas	30
1.5	Schéma de calcul de la pyramide Gaussienne	31
1.6	Le déplacement du point M est donné par celui du plus proche voisin N	33
1.7	Contrainte épipolaire	35
1.8	Bande épipolaire	35
1.9	Deux images à apparier de taille 512×512	37
1.10	Deux pyramides gaussiennes	38
1.11	Deux pyramides gaussiennes	39
1.12	L'information de mise en correspondance (pyramide 4 Points, corrélation de Laplacien).	40
1.13	L'information de mise en correspondance (pyramide Passe-Bas, corrélation de gradient).	41
1.14	L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien).	42
1.15	Les points indiqués sont tous les points des contours de la première image qui ont un correspondant dans la deuxième image.	44
1.16	L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien, seul un point sur trois a été pris en compte).	45

1.17	Deux images à appairer de taille 512×256	46
1.18	L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien).	46
1.19	L'information de mise en correspondance (pyramide Gaussienne, corrélation normalisée de Laplacien).	47
1.20	Deux images à appairer de taille 512×512 avec les lignes épipolaires . .	48
1.21	L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien).	49
1.22	Deux images à appairer de taille 512×512 avec les lignes épipolaires . .	50
1.23	L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien).	50
1.24	Cas de contours coupés	52
1.25	Cas de contours se segmentant de façon différente	52
1.26	Cas d'erreurs de mise en correspondance de points sur les contours . . .	53
1.27	Graphe engendré par une chaîne de contour	59
1.28	Exemples de chaînes de contenus sémantiques faibles	60
1.29	Suppression d'un arc de liaison dans un graphe spatio-temporel	62
1.30	Jonction de sommets dans un graphe spatio-temporel	63
1.31	Disjonction de sommets dans un graphe spatio-temporel	63
1.32	Représentation schématique de la règle	69
1.33	Liste des motifs primaires	73
1.34	Ordre des motifs primaires	74
1.35	Critères de raccordement de chaînes de contours	76
1.36	Un cas de direction du gradient variant sur la même chaîne	77
1.37	Une séquence de 6 images initiales de contours	78
1.38	Une séquence de 6 images finales de contours après le traitement	79
1.39	Deuxième image initiale de contours	80
1.40	Deuxième image finale de contours après le traitement	81
1.41	Sixième image initiale de contours	82
1.42	Sixième image finale de contours après le traitement	83
2.1	Une surface S et le <i>contour occultant</i> relativement au point O	92
2.2	Contours perçus par la caméra en mouvement	93
2.3	Formation d'une surface <i>spatio-temporelle</i> dans un repère absolu	94
2.4	Surface <i>spatio-temporelle</i> : passage au continu	94
2.5	Paramétrisation de la surface <i>spatio-temporelle</i>	95
2.6	Paramétrisation épipolaire	96
2.7	Le modèle géométrique d'une caméra	98
2.8	Le modèle sténopé d'une caméra	102
2.9	Propriété d'un contour occultant	103
2.10	Reconstruction par <i>triangulation</i> à l'aide de deux caméras. Ces deux caméras sont calibrées, les matrices de transformation du repère absolu dans le repère de la caméra A_1 et A_2 sont donc connues. Nous pouvons déterminer la matrice de transformation entre le repère caméra gauche et le repère caméra droite A_s par $A_2 \cdot A_1^{-1}$	108
2.11	Une séquence de huit images d'une demi-sphère avec les lignes épipolaires	111
2.12	Partie d'une sphère observée lors du mouvement de la caméra	112

2.13	Surfaces B-splines reconstruites correspondant à la partie d'une sphère observée avec les points de contrôle	112
2.14	Superposition des surfaces B-splines reconstruites et de la partie d'une sphère observée	113
2.15	Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) des surfaces B-splines reconstruites	113
2.16	Surfaces B-splines reconstruites correspondant à la partie d'une sphère observée avec un bruit blanc gaussien	114
2.17	Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) des surfaces B-splines reconstruites avec un bruit blanc gaussien	114
2.18	Courbes d'intersection après le découpage; résultats sur les données parfaites (à gauche); résultats sur les données bruitées (à droite)	115
2.19	Superposition des surfaces B-splines reconstruites d'un segment et du segment réel dans l'espace	116
2.20	Superposition des surfaces B-splines reconstruites d'un segment et du segment réel dans l'espace avec un bruit blanc gaussien	117
2.21	Points de contrôle des mêmes surfaces B-splines avec les différents nombres d'intervalles choisis sur les axes s et t	118
2.22	Partie d'un cylindre observée par la caméra	119
2.23	Surfaces B-splines reconstruites d'un cylindre	120
2.24	Partie d'un hyperboloïde observée lors du mouvement de la caméra	120
2.25	Superposition des surfaces B-splines reconstruites et de la partie d'un hyperboloïde observée	121
2.26	Deux surfaces B-splines reconstruites d'un hyperboloïde avec deux différents mouvements de la caméra	121
2.27	Incertitude des points de contrôle des surfaces B-splines reconstruites d'un segment	122
2.28	Incertitude des points de contrôle des surfaces B-splines reconstruites d'une sphère	123
2.29	Contour d'occultation de l'objet composé par un ellipsoïde et un hyperboloïde	124
2.30	Surfaces B-splines reconstruites lors du mouvement de la caméra	124
2.31	Courbes d'intersection obtenues après le découpage	125
2.32	Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) des surfaces B-splines reconstruites de l'objet composé par un ellipsoïde et un hyperboloïde	125
2.33	Mouvement de la caméra autour de deux cylindres parallèles	126
2.34	Superposition des surfaces B-splines reconstruites et des portions observées de deux cylindres lors du mouvement de la caméra	126
2.35	Courbes d'intersection obtenues après le découpage	127
2.36	Une nouvelle paramétrisation de la surface. Le paramètre t est déterminé à partir de la distance tridimensionnelle entre deux contours consécutifs. l_i présente la distance de deux contours entre l'instant t_i et t_{i+1}	128
2.37	Courbes d'intersection obtenues après le découpage	129
2.38	Première (à gauche) et dernière (à droite) image d'une séquence d'images de la carafe	130
2.39	Surfaces B-splines reconstruites avec le cube transparent	131

2.40	Une vue des surfaces B-splines reconstruites sans le cube transparent . .	131
2.41	Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) de surfaces B-splines reconstruites de notre carafe	132
2.42	Superposition de deux graphes de la courbure gaussienne des surfaces B-splines reconstruites de notre carafe	133
2.43	Première (à gauche) et dernière (à droite) image d'une séquence d'images d'un tube	134
2.44	Surfaces B-splines reconstruites avec le cube transparent	134
2.45	Une vue des surfaces B-splines reconstruites sans le cube transparent . .	135
2.46	Zoom dans la zone de rupture du tube	136
2.47	Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) des surfaces B-splines reconstruites de notre tube	136
2.48	Différentes images extraites d'une séquence où l'objet à reconstruire est une carafe. Le mouvement est une rotation. Nous pouvons suivre les déplacements successifs de gauche à droite et de haut en bas	137
2.49	Ensemble de surfaces B-splines reconstituant la surface de notre carafe .	138
2.50	Une vue de dessus d'un ensemble de surfaces B-splines reconstituant la surface de notre carafe	139
3.1	a) Une portion observée d'un hyperboloïde lors du mouvement de la caméra (à gauche). b) Deux surfaces B-splines reconstruites d'un hyperboloïde avec deux mouvements différents de la caméra (à droite).	144
3.2	Maillage à huit sommets adjacents (à gauche) et à six adjacents (à droite) pour neuf points d'une surface projetée sur un plan. On remarquera l'intersection des arêtes dans le premier cas.	146
3.3	Deux surfaces d'une hyperboloïde qui se superposent.	148
3.4	Méthode de projection d'une surface par morceaux. Ici, la surface sera projetée en trois fois. d_i représente la direction moyenne de i -ième sous-surface.	149
3.5	Zone commune extraite de deux surfaces. On distingue encore l'appartenance des points à l'une et à l'autre surface	151
3.6	Deux surfaces privées de leur zone commune. Les points cerclés sont les points possédant l'estampille	154
3.7	Ajout de la zone à raccorder. Les différents problèmes énumérés plus tôt y sont exposés.	155
3.8	Fusion à effectuer entre deux zones.	156
3.9	Exemple de raccordement : les traits en pointillés représentent les futures liaisons.	156
3.10	Résultat de la fusion.	157
3.11	Première surface (à gauche) et seconde surface (à droite) reconstruites à partir d'un hyperboloïde.	158
3.12	Superposition de deux surfaces.	159
3.13	Extraction des zones communes de deux surfaces.	159
3.14	Fusion de deux surfaces d'un hyperboloïde.	160
3.15	Première (à gauche) et dernière (à droite) image d'une séquence d'images de la carafe.	160

3.16 Première surface (à gauche) et seconde surface (à droite) des surfaces reconstruites à partir de notre carafe.	161
3.17 Superposition de deux surfaces.	162
3.18 Extraction de deux surfaces.	162
3.19 Fusion de deux surfaces.	163
A.1 Paramétrisation d'une courbe	172

Liste des Tables

1.1	Table des modifications des chaînes de contours	84
2.1	Certaines coordonnées d'une partie d'un segment reconstruit 3D et des erreurs	116
2.2	Certaines coordonnées d'une partie d'un segment reconstruit 3D et des erreurs avec un bruit blanc gaussien	117
2.3	Erreurs sur les diamètres de notre carafe reconstruite (en <i>mm</i>)	133

Introduction

J'introduirai cette partie en citant une anecdote singulière qui s'est déroulée lorsque j'étais petit enfant.

Une grande partie de mon enfance en Chine s'est déroulée à la campagne.

Je me souviens de grandes promenades à travers les champs et les rizières encore embrumés et dont la beauté me ravissait et me comblait de joie.

Je me rappelle en particulier la présence à proximité du village de mes grands-parents d'une étrange construction qui ne datait certainement pas de l'époque impériale. C'était un silo à riz comme il en existait et existe encore maintenant par milliers dans la Chine rurale. Ce silo avait la forme d'une tour cylindrique et me paraissait immense en raison de ma petite taille. Un jour, voulant satisfaire ma curiosité d'enfant, je décidais de le voir de plus près et de faire le tour de ce bâtiment (voir Figure 0.1).

Mais, je fus stupéfait car malgré mon déplacement autour du silo, la vue que j'en avais à chaque instant était toujours identique et seul le paysage autour se déplaçait !

Et le soir avant de m'endormir, plusieurs questions se bousculèrent dans ma tête. Pourquoi quand j'avais tourné autour du silo, la vue qui s'offrait à mes yeux était identique? Comment savoir si ce que je voyais à chaque instant était bien la même chose? Quelle était la largeur réelle de la tour? Comment la mesurer?

La nuit porta conseil et le lendemain je retrouvais la construction qui m'avait intriguée.

Je décidais de marquer le mur du silo d'un trait de craie, puis tournant légèrement autour, je m'aperçus, grâce à la marque imprimée, que même si ce que je voyais était semblable, les bords du silo eux étaient différents de ceux de départ.

De même, pour connaître la largeur du silo, j'essayais d'embrasser avec les bras ce dernier mais en vain à cause de sa largeur.

J'utilisais alors la largeur de mes bras comme mesure. Je commençais par marquer la position de ma main gauche sur le silo, puis celle de ma main droite. En plaçant alors ma main gauche sur cette dernière marque, j'embrassais à nouveau le silo de mes bras et marquais à nouveau la position de ma main droite. Ainsi de suite jusqu'à ce que j'eus fait le tour du silo.

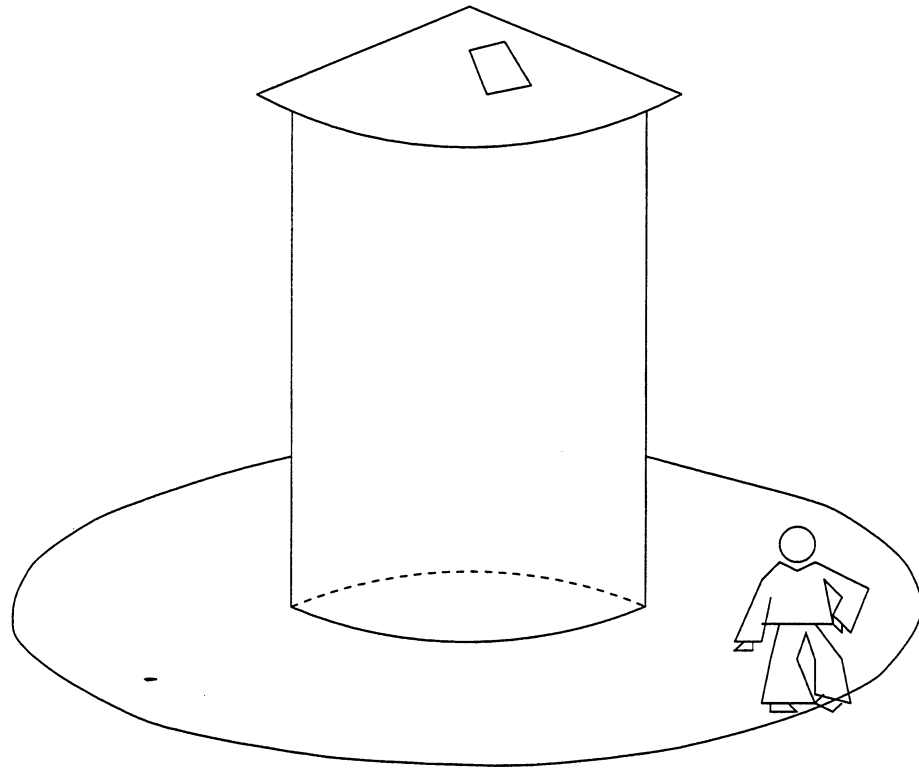


FIG. 0.1 - Un silo à riz typique en Chine avec ouverture sur le toit

Ainsi je connaissais de façon approximative la longueur du tour du silo en l'ayant parcouru morceau par morceau.

L'enfant que j'étais a grandi et les questions naïves que je me posais jadis ont trouvé dans la vision par ordinateur des réponses scientifiques. Et aujourd'hui, je travaille avec les images fournies par les caméras, alors qu'hier encore j'utilisais mes yeux pour observer un objet.

Une image en vision est simplement une matrice dont chaque élément possède une certaine intensité lumineuse. J'utilise ainsi les informations bidimensionnelles extraites de ces images, en particulier les bords des objets que nous appelons "contours occultants". Les contours sont des ensembles de points qui sont caractérisés par une variation rapide de l'intensité dans l'image.

Si un parallèle peut être fait, les bords du silo que j'observais et qui se projetaient sur ma rétine sont de tels contours. Donc les contours occultants d'un objet se modifient avec le déplacement même si avec un objet régulier tel un cylindre, ils paraissent identiques.

Plus rigoureusement, un contour d'occultation est un ensemble de points d'image dont la ligne de vue est tangente à la surface de l'objet observé. L'étude de Vaillant [Vai 89] montre qu'il est possible de les détecter si au moins trois observations réalisées de points de vue suffisamment proches sont disponibles.

En suivant ces contours occultants dans une séquence d'images, nous pouvons ainsi reconstruire la surface tridimensionnelle de l'objet. Ainsi, si nous voulons reconstruire la surface de l'objet globalement, il faudra le faire petit à petit de la même façon que

j'avais mesuré la circonférence du silo dans mon enfance.

Le cadre général de cette thèse est l'étude théorique de trois problèmes principaux suivants : la poursuite de contours courbes dans une séquence d'images, la représentation et la reconstruction de surfaces tridimensionnelles d'objets non polyédriques. Notre travail a été plus spécialement consacré à la reconstruction de surfaces tridimensionnelles en utilisant des surfaces B-splines, à partir de l'observation du mouvement de contours occultants dans une séquence d'images (voir Figure 0.2).

Dans ce mémoire, nous admettons que les contours occultants sont déjà détectés dans l'image [Vai 89, Vai 90a]. Nous voulons démontrer que la reconstruction des surfaces tridimensionnelles d'un objet non polyédrique observé est possible par une caméra en mouvement, sauf que certaines parties de l'objet ne peuvent pas être reconstruites : des parties planes, concaves.

La première partie de cette thèse est consacrée à l'étude théorique du suivi de contours courbes dans une séquence d'images : un problème pratiquement incontournable de la reconstruction des surfaces, à partir des contours occultants, est la détermination des correspondances entre ces contours occultants d'images successives.

Nous proposons un algorithme pyramidal du suivi de contours courbes basé sur des primitives de type "points de contour" en utilisant une technique de corrélation. Cette approche est maintenant classique. La contribution de cette recherche est la comparaison de différentes corrélations, ainsi que la comparaison de plusieurs constructions de pyramide. Dans cette partie, nous allons montrer comment appliquer ces deux concepts au problème spécifique du suivi de contours courbes dans une séquence d'images. Après l'étape de mise en correspondance des chaînes de contours, nous développons une approche originale utilisant une technique de manipulation de graphes spatio-temporels pour éliminer les erreurs de segmentation, et des erreurs dues au bruit dans les images. Cette méthode consiste à rechercher un état du système cohérent à partir de l'ensemble des informations d'appariement point à point d'une séquence. Les résultats expérimentaux montrent que cette approche est bien adaptée à notre problème.

D'autre part, afin de faciliter notre reconstruction de surfaces, nous présentons des méthodes de traitement de contours : lissage par les courbes B-splines cubiques régularisées, raccordement et coupure.

La seconde partie qui constitue le cœur de la thèse, est consacrée à l'étude de la reconstruction de surfaces courbes tridimensionnelles à partir d'un nombre fini d'observations effectuées par une caméra en mouvement.

Cette recherche se situe dans la suite des travaux de Giblin [Gib 87], Cipolla [Bla 89, Cip 89, Cip 92], Vaillant [Vai 90a, Vai 90b], et Arbogast [Arb 91a] qui permettent de calculer aux points observés de l'objet non polyédrique les courbures gaussiennes, moyennes, et normales, ainsi que les courbures et les directions principales. La contribution originale de notre travail est de caractériser les propriétés locales de la surface à reconstruire, ainsi que reconstruire la surface globale, ce qui n'avait encore jamais été fait à notre connaissance. Nous avons reformulé le problème de reconstruction de surfaces sous la forme d'un problème d'optimisation en utilisant les surfaces B-splines comme modélisation de la surface d'un objet, et cette façon de procéder a permis d'aboutir à des résultats quantitatifs non encore égalés.

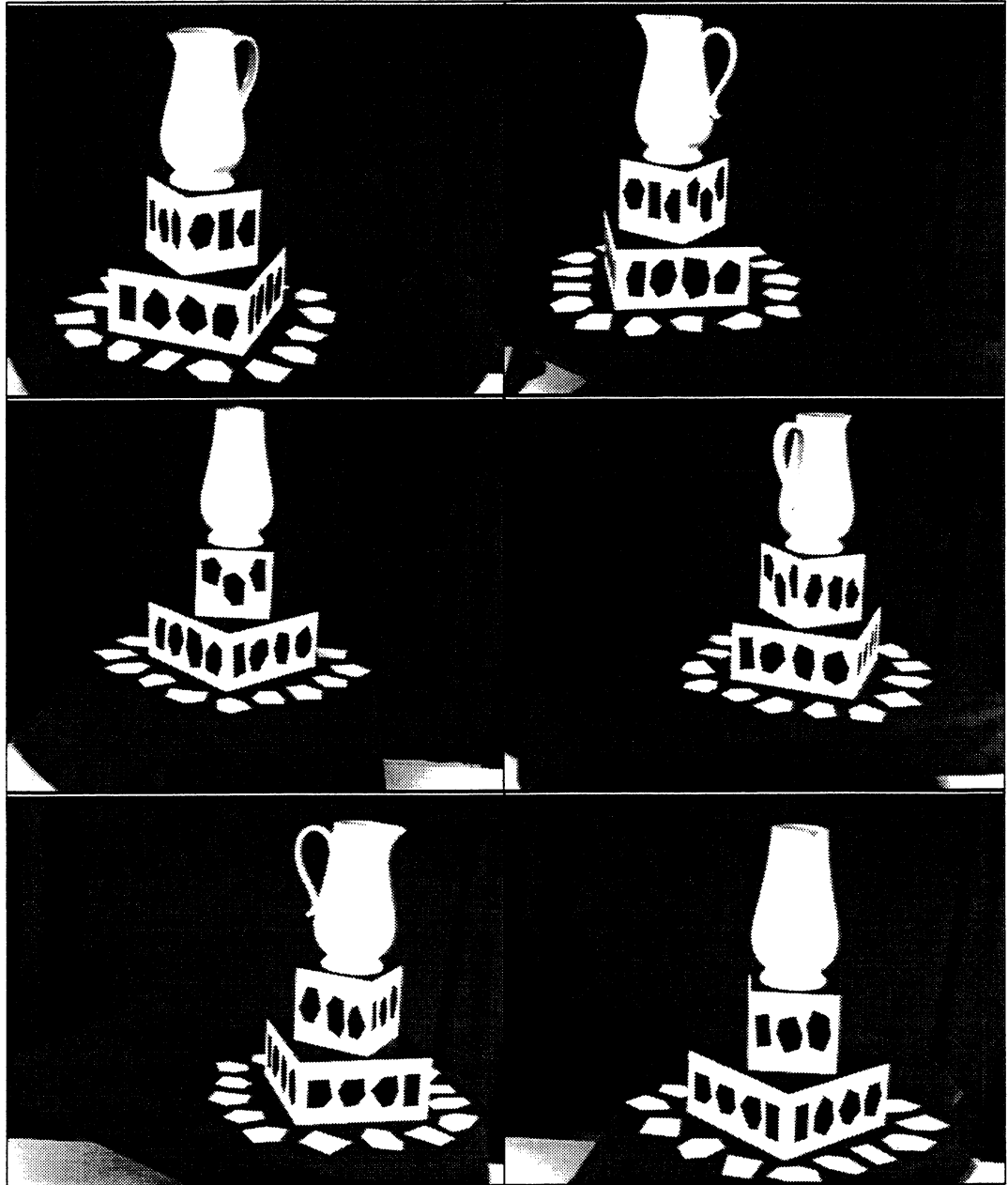


FIG. 0.2 - Différentes images extraites d'une séquence où l'objet à reconstruire est une carafe. Nous pouvons suivre les déplacements successifs de gauche à droite et de haut en bas. Notre carafe est posée sur deux cubes dont les sommets sont traités comme des points de référence et ainsi la scène dispose d'un repère absolu. La scène contient des papiers noirs et blancs pour calibrer la caméra en mouvement.

La reconstruction de surfaces tridimensionnelles exige des méthodes de calibrage de la caméra, d'une part très précises, et d'autre part faciles à mettre en œuvre et stables puisque nous avons une centaine d'images à calibrer lors de la reconstruction des surfaces. Dans cette partie, nous proposons une nouvelle méthode basée sur les points de référence de la scène, seuls cinq points sont supposés connus et sont utilisés comme repère relatif. La connaissance du mouvement de la caméra est représentée par une matrice de projection perspective. Nous allons voir comment établir trois équations élémentaires à partir d'un point d'un contour d'occultation en utilisant la matrice de projection perspective. En itérant ce processus sur un ensemble de points extraits de contours, nous obtenons un système d'équations non linéaires dont la résolution donnera des points de contrôle des surfaces B-splines. Notre méthode de reconstruction introduit directement un terme de régularisation sur les surfaces tridimensionnelles qui permet d'ajuster facilement la forme des surfaces dans le cas où il y a du bruit dans les images. Ce terme permet de minimiser les variations de l'orientation de la surface locale pour avoir une surface lissée.

Finalement, les surfaces tridimensionnelles de l'objet sont reconstruites morceau par morceau. L'approche proposée s'applique dans le cas d'un mouvement quelconque d'une caméra calibrée, avec des surfaces C^2 par morceaux au moins. Des résultats expérimentaux sur des données synthétiques et des données réelles sont présentés.

La dernière partie est consacrée au problème du raccordement des surfaces B-splines reconstruites par la méthode proposée dans la précédente partie. Ce problème a déjà été posé dans les domaines de la Conception Assistée par Ordinateur et de la Conception et Fabrication Assistée par Ordinateur [Per 92] qui utilisent notamment les modèles surfaciques pour la création de formes. En général, la conception d'une surface complexe est réalisée "par morceaux", construits indépendamment les uns des autres ou non.

Ici, dans un contexte de vision par ordinateur, nous disposons d'un ensemble de surfaces B-splines possédant des zones communes. Nous allons fusionner ces surfaces dont une ou plusieurs parties se superposent. D'une part, la paramétrisation de surfaces B-splines est incapable de décrire la surface nouvelle qui sera créée. Il faut donc chercher une nouvelle représentation discrète et intermédiaire qui s'adapte à notre problème. D'autre part, en chacun des points discretisés de surfaces B-splines les équations paramétrées nous donnent des informations géométriques telles que le rayon de courbure et la normale aux points. Cependant, ces points ne sont pas structurés dans l'espace. Donc nous allons choisir une structure simple et efficace "maillage triangulaire" qui nous permet facilement non seulement d'établir les relations d'adjacence entre les points de surfaces, mais aussi de manipuler les informations géométriques et topologiques des points de surfaces.

Chaque zone commune à deux surfaces fournit un ensemble de points tridimensionnels qui est interpolé par une surface B-spline uniforme et bicubique qui va être fusionnée avec les deux surfaces initiales afin d'avoir une surface globale. Plus particulièrement, notre méthode d'interpolation de surface aux moindres carrés consiste à résoudre un système d'équations linéaires déterminant les points de contrôle de surfaces B-splines.

Finalement, un nouvel algorithme de raccordement de différentes surfaces B-splines est proposé. Des résultats expérimentaux sur des données synthétiques et des données réelles sont présentés.

Trois appendices fournissent quelques détails supplémentaires propres à faciliter la lecture de ce mémoire :

- Le premier présente quelques notions essentielles de la géométrie différentielle des courbes et des surfaces.
- Le second donne une liste de règles de transformations de graphes que nous avons utilisées dans le processus de la correction des anomalies de chaînes de contours d'images.
- Le dernier présente une méthode d'interpolation de surfaces par les surfaces B-splines uniformes et bicubiques étant donné un ensemble de points tridimensionnels dans l'espace.

Chapitre 1

Poursuite et correction de contours courbes

Dans ce chapitre, nous allons présenter un algorithme pyramidal de poursuite de contours courbes basé sur des primitives de type “points de contour” en utilisant une technique de corrélation. La mise en correspondance s’effectue en plusieurs étapes correspondant aux différentes résolutions de pyramides d’images appariées.

Cette approche est maintenant classique. La contribution de ce travail est la comparaison de différentes corrélations, ainsi que la comparaison de plusieurs constructions de pyramide. Nous appliquons ces deux concepts au problème de la poursuite de contours courbes dans une séquence d’images. Dans ce cas la réponse de l’appariement doit être globale. Nous pouvons donc nous permettre quelques erreurs locales, mais en revanche, il faut utiliser les appariements pour compenser les erreurs de segmentation.

Dans cette recherche, nous étudions la corrélation simple, la corrélation normalisée, et la corrélation absolue sur les différents types d’image : images de niveaux de gris, images de gradient et images de Laplacien en utilisant les contours de Deriche [Der 87]. Nous comparons également les résultats obtenus en utilisant plusieurs sortes de pyramides : Pyramide 4 Points, Pyramide Basse-Bas, Pyramide Gaussienne.

Après cette étape de mise en correspondance des chaînes de contours, nous nous intéressons à l’élimination des erreurs produites par les étapes initiales du processus de vision. Nous proposons une approche originale basée sur une technique de manipulation de graphes spatio-temporels.

Des résultats expérimentaux sont présentés pour les différents types d’images.

1.1 Introduction

La reconstruction des surfaces tridimensionnelles et l'analyse du mouvement d'une scène à partir d'une séquence d'images sont deux objectifs importants en vision par ordinateur. Un problème pratiquement incontournable est la détermination des correspondances entre les primitives¹ d'images successives.

Le choix des primitives d'images à mettre en correspondance est crucial. En pratique, nous utilisons trois sortes de primitives : point, droite et région ; Moravec [Mor 77] a développé un système de vision pour guider un robot dans un environnement contenant des obstacles. Il extrait des points d'intérêt, puis effectue la mise en correspondance entre deux images successives en utilisant une technique de corrélation. Deriche [Der 90b] choisit comme points d'intérêt les maxima locaux de courbure des contours. Sa méthode est assez robuste car ces points sont relativement stables. L'étude de Crowley [Cro 88] utilise comme primitives d'images des segments de contours. En utilisant le filtre de Kalman [May 79], il obtient des résultats satisfaisants pour le suivi de contours. Deriche [Der 90a] utilise une technique similaire en changeant la représentation du segment. Dans le cadre d'images de l'environnement routier, Koller [Kol 90], à Karlsruhe, a développé également une méthode de poursuite de contours courbes basée sur l'identification de taches² à des résolutions de plus en plus fines ; ce type de primitive semble bien adapté à ce type d'images.

Notre recherche a pour but de reconstruire les surfaces tridimensionnelles [Zha 93b, Zha 93c] à partir de l'observation des contours occultants dans une séquence d'images. Notre méthode de suivi de contours courbes quelconques est fondée sur un algorithme pyramidal basé sur les points de contours en utilisant une technique de corrélation. Nous nous sommes inspirés du travail de P. Anandan [Ana 89] qui utilise une corrélation avec des fenêtres 5×5 pondérées par des poids gaussiens. La mesure de corrélation utilisée est la différence quadratique. Le travail de Meygret [Mey 89] en stéréovision montre que l'approche pyramidale est appropriée pour la mise en correspondance des contours.

Dans le cadre de la poursuite de contours, nous nous sommes placés dans le cas où la caméra et la scène sont en mouvement relatif. En fait, la caméra est déplacée petit à petit pour fournir une séquence d'images de la scène. Nous supposons que le mouvement de la caméra par rapport à la scène est suffisamment lent pour que les changements successifs soient relativement faibles, typiquement, ce déplacement pourrait être supérieur, et cela augmenterait de façon proportionnelle le temps du calcul.

Ce chapitre est organisé de la manière suivante :

Dans la première partie, nous introduisons deux notions nécessaires : la corrélation de deux signaux et la multi-résolution. Nous commentons plusieurs approches dans ces deux domaines.

Dans la deuxième partie, nous décrivons d'abord brièvement la construction des trois sortes de pyramides les plus souvent utilisées, et de leurs correspondantes laplaciennes. Nous présentons ensuite en détail notre approche. Nous proposons également une méthode de traitement des contours. Ce traitement (raccordement et coupure de contours) est rendu nécessaire par les erreurs rencontrées tout au long du processus de

¹indices fiables d'une image.

²zones d'une image présentant une faible variation d'intensité lumineuse.

vision : bruit électronique dû à la caméra, fautes de segmentation de contours, etc.

Dans la troisième partie, nous nous intéressons à la correction des anomalies dans les images de chaînes de contours. La mise en correspondance de points nous apporte des informations essentielles pour la localisation de ces erreurs. Nous montrons comment les utiliser par une approche originale basée sur une technique de transformation de graphes.

Finalement, nous présentons des résultats à partir de différents types de scènes qui montrent l'efficacité de notre approche pyramidale. Nous montrons également des résultats concernant la localisation des erreurs de contours. Nous discutons enfin des perspectives de notre approche.

1.2 Corrélation

De nombreuses méthodes de suivi de contours ont été développées, basées sur des techniques de corrélation, de relaxation, de prédiction et vérification d'hypothèses, de programmation dynamique ou de recuit simulé. Une étude détaillée de ces méthodes a été faite par Ayache [Aya 89]. Parmi celles-ci, la corrélation a été une des premières techniques à donner de bons résultats. Elle consiste à prendre une fenêtre centrée autour d'un point de contour dans la première image et à trouver, dans la deuxième image, le point dont la position réalise la meilleure superposition.

1.2.1 Corrélation de deux signaux

La corrélation est une technique souvent utilisée dans le domaine du traitement du signal. Dans cette section, nous introduisons deux notions, *énergie d'erreur* et *coefficient de corrélation*, pour mesurer le niveau de ressemblance entre deux signaux.

Soient deux signaux continus unidimensionnels $f(x)$ et $g(x)$. Etant donné un réel a positif, l'énergie d'erreur E est définie par :

$$E = \int_{-\infty}^{+\infty} (f(x) - ag(x))^2 dx \quad (1.1)$$

Le problème se pose de la manière suivante : comment choisir la valeur a pour que l'énergie E soit la plus petite possible ?

En dérivant la formule (1.1), nous trouvons :

$$\frac{\partial E}{\partial a} = 2 \int_{-\infty}^{+\infty} (f(x) - ag(x))(-g(x)) dx$$

Il nous faut :

$$\frac{\partial E}{\partial a} = 0$$

Nous obtenons donc :

$$a = \frac{\int_{-\infty}^{+\infty} f(x)g(x) dx}{\int_{-\infty}^{+\infty} g^2(x) dx} \quad (1.2)$$

La formule (1.1) peut s'écrire sous la forme suivante en remplaçant a par sa valeur :

$$E = \int_{-\infty}^{+\infty} \left(f(x) - g(x) \frac{\int_{-\infty}^{+\infty} f(x)g(x) dx}{\int_{-\infty}^{+\infty} g^2(x) dx} \right)^2 dx \quad (1.3)$$

En développant l'expression au carré, nous obtenons ;

$$E = \int_{-\infty}^{+\infty} f^2(x) dx - \frac{\left(\int_{-\infty}^{+\infty} f(x)g(x) dx \right)^2}{\int_{-\infty}^{+\infty} g^2(x) dx} \quad (1.4)$$

Si nous posons :

$$C_{fg} = \frac{\int_{-\infty}^{+\infty} f(x)g(x) dx}{\left(\int_{-\infty}^{+\infty} f^2(x) dx \int_{-\infty}^{+\infty} g^2(x) dx \right)^{\frac{1}{2}}} \quad (1.5)$$

Nous appelons C_{fg} le *coefficient de corrélation* des signaux $f(x)$ et $g(x)$.
D'après l'inégalité de Schwartz :

$$\left(\int_{-\infty}^{+\infty} f(x)g(x) dx \right)^2 \leq \int_{-\infty}^{+\infty} f^2(x) dx \int_{-\infty}^{+\infty} g^2(x) dx$$

Nous avons cette inégalité :

$$\left| \frac{\int_{-\infty}^{+\infty} f(x)g(x) dx}{\left(\int_{-\infty}^{+\infty} f^2(x) dx \int_{-\infty}^{+\infty} g^2(x) dx \right)^{\frac{1}{2}}} \right| \leq 1 \quad (1.6)$$

Nous déduisons $|C_{fg}| \leq 1$.

Finalement, la formule (1.4) peut s'écrire sous la forme ci-dessous en remplaçant C_{fg} par sa valeur :

$$1 - C_{fg}^2 = \frac{E}{\int_{-\infty}^{+\infty} f^2(x) dx} \quad (1.7)$$

Si le signal $f(x)$ ressemble à $g(x)$, alors nous avons la relation suivante :

$$\left| \int_{-\infty}^{+\infty} f(x)g(x) dx \right| \simeq \int_{-\infty}^{+\infty} f^2(x) dx \simeq \int_{-\infty}^{+\infty} g^2(x) dx \quad (1.8)$$

Selon les formules (1.5) et (1.8), si les deux signaux $f(x)$, $g(x)$ se ressemblent, alors nous déduisons que le coefficient de corrélation C_{fg} est égal à 1 ou -1, en cumulant ces résultats avec la formule (1.7), nous obtenons l'énergie d'erreur $E = 0$. Dans ce cas, nous disons que les deux signaux $f(x)$, $g(x)$ sont *corrélés*.

D'après la formule (1.5), la valeur de C_{fg} est fonction de l'intégrale du produit des signaux $f(x)$ et $g(x)$. Il y a trois cas de signaux unidimensionnels : le cas (a) $|C_{fg}| < 1$, le cas (b) $C_{fg} = 1$, le cas (c) $C_{fg} = -1$ (voir Figure 1.1).

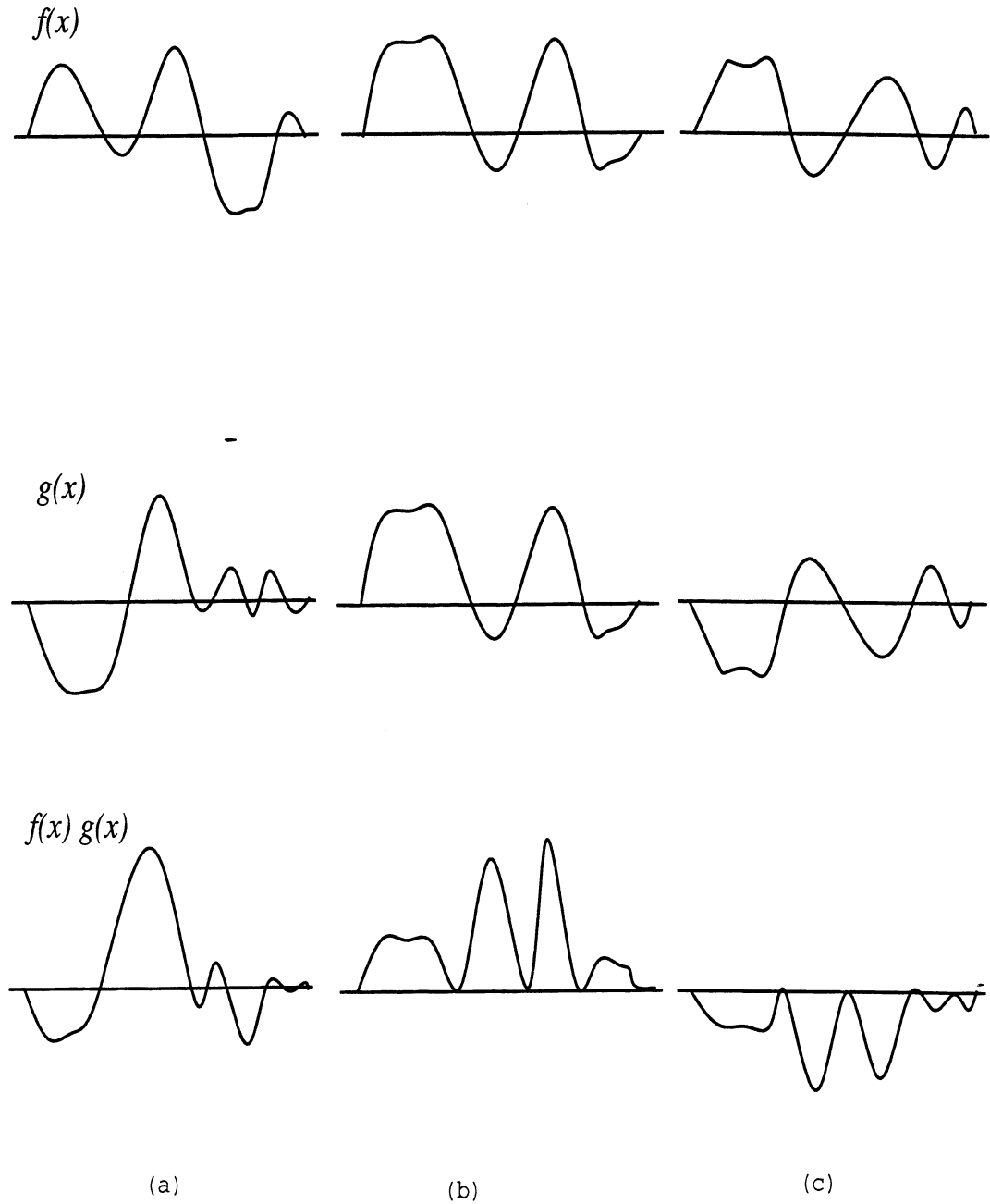


FIG. 1.1 - Deux signaux (a) non ressemblants. (b) ressemblants. (c) inversés.

Dans le cas des signaux discrets bidimensionnels, les domaines de $f(x, y)$ et de $g(x, y)$ sont bornés. Le domaine de l'intégrale par rapport à x est $[-K, +K]$ et le domaine de l'intégrale par rapport à y est $[-L, +L]$. Nous avons la formule suivante :

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (f(x, y) - ag(x, y))^2 dx dy = \sum_{x=-K}^{x=+K} \sum_{y=-L}^{y=+L} (f(x, y) - ag(x, y))^2$$

Donc le coefficient de corrélation C_{fg} est défini facilement par la formule qui suit selon la formule (1.5) :

$$C_{fg} = \frac{\sum_{x=-K}^{x=+K} \sum_{y=-L}^{y=+L} f(x, y)g(x, y)}{\left(\sum_{x=-K}^{x=+K} \sum_{y=-L}^{y=+L} f^2(x, y) \sum_{x=-K}^{x=+K} \sum_{y=-L}^{y=+L} g^2(x, y) \right)^{\frac{1}{2}}} \quad (1.9)$$

1.2.2 Corrélation de la vision par ordinateur

Dans le cadre de la corrélation en vision par ordinateur, il existe différentes formulations. Nous présentons quelques unes de ces techniques et les façons dont elles ont été mises en œuvre pour résoudre le problème difficile de la mise en correspondance de points dans les images à apparier.

Supposons que les points d'intérêt soient définis par $I_1(i, j)$ et $I_2(i, j)$ au pixel (i, j) dans les deux images. Nous pouvons définir c le coefficient de corrélation de $I_1(i, j)$ et $I_2(i, j)$ par :

– Corrélation normalisée [Law 82] :

$$c = \frac{\sum_i \sum_j I_1(i, j)I_2(i, j)}{\sqrt{\sum_i \sum_j I_1(i, j)^2} \sqrt{\sum_i \sum_j I_2(i, j)^2}} \quad (1.10)$$

– Corrélation normalisée de Moravec [Mor 77] :

$$c = \frac{2 \sum_i \sum_j I_1(i, j)I_2(i, j)}{\sum_i \sum_j I_1(i, j)^2 + \sum_i \sum_j I_2(i, j)^2} \quad (1.11)$$

– Corrélation normalisée par valeur absolue de différences [Law 82] :

$$c = 1.0 - \frac{\sum_i \sum_j |I_1(i, j) - I_2(i, j)|}{\sum_i \sum_j I_1(i, j) + \sum_i \sum_j I_2(i, j)} \quad (1.12)$$

A partir de ces trois types de corrélation, l'idée est de minimiser un terme de telle façon que $c \simeq 1.0$. Nous remarquons que la formule (1.9) est identique à (1.10) dans un cas discret.

Hannah [Han 84] a développé un système de stéréovision en implémentant un algorithme très voisin de celui de Moravec. La mise en correspondance s'effectue toujours sur des points d'intérêt extraits en utilisant une structure pyramidale.

Fua [Fua 90] a proposé un algorithme de stéréovision parallèle en utilisant un autre type de corrélation normalisée donné par :

$$s = \max(0, 1 - c) \quad (1.13)$$

$$c = \frac{\sum_i \sum_j (I_1(x + i, y + j) - I_2(x + dx + i, y + dy + j))^2}{\sqrt{(\sum_i \sum_j I_1(x + i, y + j)^2)(\sum_i \sum_j I_2(x + dx + i, y + dy + j)^2)}} \quad (1.14)$$

où I_1 et I_2 sont deux intensités dans les images gauche et droite, dx et dy représentent le déplacement le long de la ligne épipolaire.

1.3 L'approche pyramidale de mise en correspondance

L'utilisation de la technique de corrélation permet une modélisation simple de la ressemblance entre primitives de deux images successives d'une séquence. La fenêtre de recherche de la corrélation doit obligatoirement être plus grande que le déplacement des contours entre deux images successives. En pratique, sa taille a été ajustée en fonction des expérimentations.

Dans toutes les techniques de corrélation, le but est de trouver des valeurs extrémales pour une fonction donnée. L'avantage de cette technique est qu'elle consiste en un appariement global à partir des informations d'appariement local. En revanche, ce gain se paie par une complexité de calcul supérieure.

Selon les travaux de Moravec [Mor 77] et ceux de Lawton [Law 82], les méthodes de corrélation classique sont sensibles au bruit et sont mises en difficulté en présence de structures périodiques. La multi-résolution apporte cependant une solution à ces problèmes. Dans notre étude, nous nous placerons dans le cadre plus général de la poursuite de contours courbes, nous rapprochant ainsi des travaux de P. Anandan [Ana 89] relatifs à une séquence d'images.

1.3.1 Pyramide et multi-résolution

Les scènes réelles que nous voulons analyser contiennent, *a priori*, des objets de tailles différentes. Le choix d'une échelle correcte pour déceler les caractéristiques de l'image devient critique lorsque nous cherchons à trouver une interprétation complète des objets de la scène.

En vision par ordinateur, une image est une matrice d'intensités lumineuses. A partir de cette image nous pouvons construire une série de nouvelles images par diminution progressive de la résolution. A chaque étape l'image est filtrée et échantillonnée [Bal 82]. Cette série d'images est appelée *Pyramide d'une image* (voir Figure 1.2). Cette structure présente deux intérêts: d'abord, elle permet d'importantes réductions de la taille de l'image et du temps d'exécution des algorithmes. Ensuite, les petites structures d'une image disparaissent aux niveaux supérieurs, ne laissant ainsi apparaître que les structures les plus stables et les plus importantes, facilitant ainsi la mise en correspondance.

L'approche multi-résolution présente l'avantage de réduire considérablement les temps de calculs par rapport à la méthode de corrélation classique. De plus, l'estimation à basse résolution est très robuste car le bruit est réduit et les distorsions au niveau du pixel sont lissées.

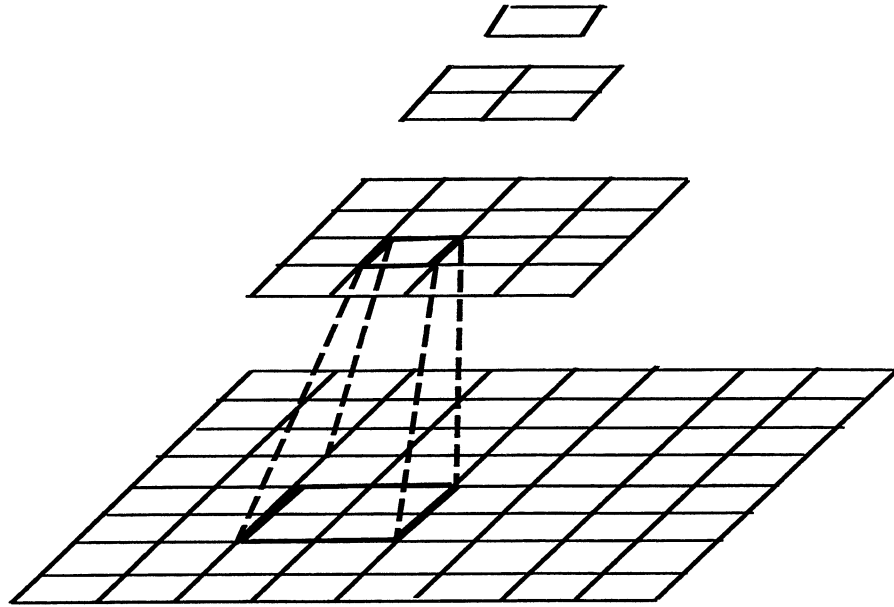


FIG. 1.2 - Structure pyramidale d'une image

P. Anandan [Ana 89] a utilisé une pyramide laplacienne proposée par Burt [Bur 81]. Il a implémenté un algorithme pyramidal pour estimer le déplacement des contours en utilisant une corrélation classique. La mise en correspondance s'effectue sur les points de contours. L'idée est de minimiser la somme des différences carrées pondérées par les poids de Gauss entre les pixels appariés. Supposons que la taille de la fenêtre de corrélation centrée autour d'un pixel $M = (i, j)$ soit $F1$, et celle de la fenêtre de recherche soit $F2$. Le déplacement $\vec{d} = (k_i, l_j)$ de ce pixel entre la première image et la deuxième image est donné par :

$$\vec{d} = (k_i, l_j) = \left\{ \Delta \in F2 \text{ qui minimise } \sum_{\delta \in F1} (I1(M) - I2(M + \delta + \Delta))^2 \times P(\delta) \right\} \quad (1.15)$$

où $I1$ et $I2$ sont des valeurs de Laplacien dans leur image respective. $P(\delta)$ est la pondération de Gauss du pixel $M + \delta$. Pour chaque pixel de la première image, la meilleure correspondance est le pixel pour lequel la somme est minimale. Les résultats qu'il obtient sur les contours rectilignes sont très précis.

A Karlsruhe, l'équipe de H. H. Nagel [Kol 90] a développé une approche hiérarchique basée sur les primitives taches à différentes résolutions. L'idée est d'apparier les taches plutôt que les contours en utilisant des régions comme indices d'image. Les résultats présentés dans cet article montrent la robustesse de leur approche pyramidale.

Meygret [Mey 90] a implémenté un algorithme de stéréovision pyramidal basé sur des primitives de type "points de chaîne de contour". La mise en correspondance est réalisée au moyen d'une fonction de similarité symétrique entre les images gauche et droite, satisfaisant la contrainte d'unicité. Il a utilisé la pyramide 4 Points dont la résolution est calculée par une simple moyenne.

1.3.2 Principe

Notre algorithme pyramidal est basé sur les points de contours en utilisant une corrélation simple. La mise en correspondance s'effectue en plusieurs étapes correspondant aux différentes résolutions. La méthodologie de mise en correspondance est la suivante :

- construction d'une pyramide.
- calcul d'images et extraction de contours d'une pyramide de résolution de plus en plus faible.
- en partant du niveau de plus faible résolution, estimation par corrélation du mouvement des points de contours de l'image.
- propagation de cette estimation vers les niveaux inférieurs de plus haute résolution.
- sélection de points appariés.
- détermination définitive des arcs ou parties de courbes en correspondance.

Notre algorithme de suivi de contours utilise cinq modules. L'entrée est une séquence d'images de contours qui a déjà été traitée avec notre logiciel FEAGRO³. Après l'évaluation de la procédure de corrélation, nous produisons une séquence d'images de contours avec les coordonnées du déplacement des points de contours entre deux images appariées. Puis nous effectuons des raccordements et découpages sur les contours pour réduire les erreurs dues au bruit dans les images. Enfin nous visualisons les résultats du suivi de contours dans cette séquence d'images.

1.3.3 Construction des pyramides

Plusieurs techniques de construction de pyramides sont en concurrence pour ce processus. Ce sont les suivantes : pyramide 4 Points, pyramide Passe-Bas, pyramide Gaussienne.

Pyramide 4 Points

C'est la plus simple des pyramides que nous avons utilisée au cours de notre étude. La résolution de l'image est réduite par une consolidation [Mey 90]. Cette technique consiste à calculer une simple moyenne de 4 points pour obtenir un point de l'image en résolution inférieure. Sa description est donnée par la figure suivante (voir Figure 1.3) : où $P(x, y)$ est le niveau de gris du pixel (x, y) . Les L_i représentent les différents niveaux de la pyramide 4 Points avec i variant de 0 à 5.

Le Laplacien de cette pyramide est calculé à partir de la pyramide en appliquant à chaque niveau le filtre de dérivée seconde, symétrique, de taille 3×3 , par exemple :

$$f = \begin{pmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (1.16)$$

³FEATure GROUping : a été développé par Thomas Skordas, Radu Horaud, et Françoise Veillon au LIFIA.

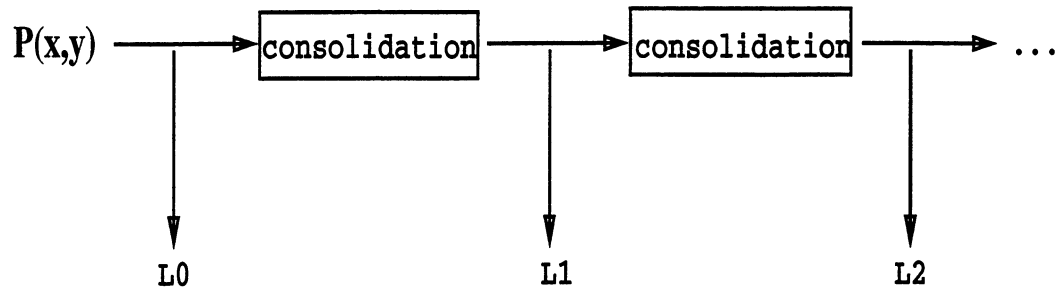


FIG. 1.3 - Schéma de calcul de la pyramide 4 Points

Pyramide Passe-Bas

Cette pyramide nous a été fournie par A. Chéhikian et J.L. Crowley [Cro 90]. La technique consiste à convoluer en cascade l'image initiale avec des filtres gaussiens discrets de tailles différentes. Pour le calcul d'une telle pyramide, l'algorithme est très simple et efficace au niveau du temps de calcul. Sa description est donnée par la figure suivante (voir Figure 1.4): $P(x,y)$ est le niveau de gris du pixel (x,y) , s_2 représente la phase d'échantillonnage. Les L_i représentent les différents niveaux de la pyramide Passe-Bas avec i variant de 0 à 15. Le filtre g_0 est égal à $[1\ 2\ 1]$ et le filtre g_1 est égal à $[1\ 4\ 6\ 4\ 1]$.

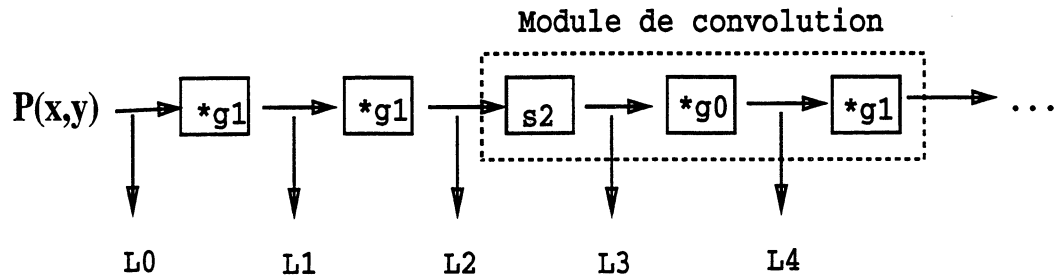


FIG. 1.4 - Schéma de calcul de la pyramide Passe-Bas

Le Laplacien de la pyramide Passe-Bas est calculé de la même façon que celui de la pyramide 4 Points.

Pyramide Gaussienne

Cette pyramide est lissée par une gaussienne [Mar 80]. La construction se fait en convoluant l'image initiale avec une fonction du type :

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.17)$$

et en l'échantillonnant à chaque niveau de pyramide. Sa description est présentée en Figure 1.5: $P(x,y)$ est le niveau de gris du pixel (x,y) . f est une fonction de Gauss, s_2 représente la phase d'échantillonnage. Les L_i représentent les différents niveaux de la pyramide gaussienne avec i variant de 0 à 5.

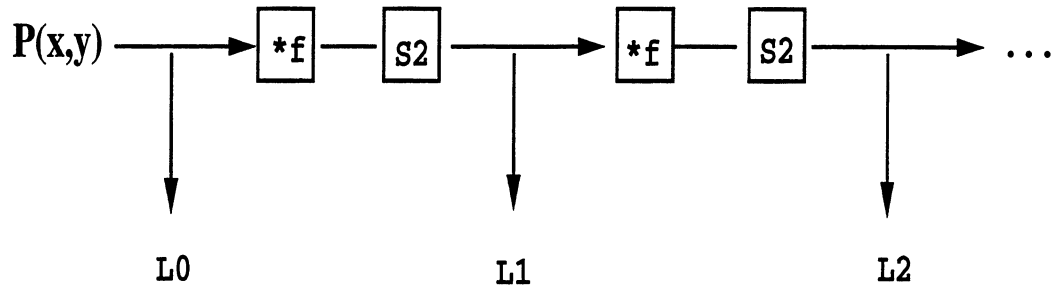


FIG. 1.5 - Schéma de calcul de la pyramide Gaussienne

Le Laplacien de la pyramide gaussienne est calculé par une différence de deux gaussiennes telles que le rapport de leurs paramètres σ_1 et σ_2 soit égal à environ 1.6.

1.3.4 Calcul d'images et extraction de contours d'une pyramide

En vision par ordinateur, une image est une matrice d'intensités lumineuses. Les caractéristiques d'une image sont les variations de l'intensité lumineuse. Deux méthodes d'extraction de contours sont bien connues : soit en opérant à partir de l'image du gradient, soit à partir de celle du Laplacien.

Les deux composantes du vecteur *gradient* d'un signal bidimensionnel $P(x, y)$ sont données par :

$$\nabla P(x, y) = G_x(x, y) \vec{x} + G_y(x, y) \vec{y}$$

où

$$\begin{aligned} G_x(x, y) &= \frac{\partial P(x, y)}{\partial x} \\ G_y(x, y) &= \frac{\partial P(x, y)}{\partial y} \end{aligned} \quad (1.18)$$

Ce vecteur permet de mettre en évidence les variations d'intensité lumineuse de l'image traitée. A chaque niveau de résolution de la pyramide, la dérivée première est calculée dans les deux directions x et y . Ceci nous permet d'associer à chaque point de l'image le module et la direction du gradient. Les maxima locaux du module de gradient, nommés points de contraste, constituent les contours de l'image. L'obtention des contours dans notre algorithme comprend un filtrage préliminaire, l'extraction par la méthode du gradient développé par Deriche [Der 87], puis le seuillage par hystérésis en utilisant les critères de Canny [Can 86], finalement le chaînage des points est réalisé par la procédure de Giraudon [Gir 87].

Le Laplacien d'un signal bidimensionnel est donné par :

$$\text{Laplacien}(x, y) = \frac{\partial^2 P(x, y)}{\partial x^2} + \frac{\partial^2 P(x, y)}{\partial y^2}$$

En pratique, il peut être approximé par différents filtres discrets.

La méthode d'extraction de contours par le gradient consiste à chercher les extrêmes qui localisent les zones de transition entre des régions de niveau de gris différent, et

celle par le Laplacien consiste à chercher les passages à zéro qui se situent au milieu de la zone de transition séparant les régions de luminosités différentes.

Pour une image initiale, nous calculons une pyramide d'images, une pyramide de gradients, une pyramide de Laplaciens et une pyramide de contours.

1.3.5 Mise en correspondance de points

Lors de la construction des pyramides, les contours sont détectés à l'aide du gradient. La mise en correspondance des points de contours se fait ensuite à des résolutions de plus en plus fines.

Initialisation du déplacement des contours

Tout d'abord, nous estimons les déplacements de points de contours au niveau de résolution le plus fort par la corrélation suivante :

Soit $M = (i, j)$ un point quelconque d'un contour de la première image. Nous supposons que la taille de la fenêtre de corrélation centrée autour de M est $F1$, et que celle de la fenêtre de recherche est $F2$. Le déplacement $\vec{d} = (k_i, l_j)$ de ce point entre les deux images appariées est donné par la valeur minimale de la formule suivante :

$$\vec{d} = (k_i, l_j) = \left\{ \Delta \in F2 \text{ qui minimise } \sum_{\delta \in F1} (I1(M) - I2(M + \delta + \Delta))^2 \right\} \quad (1.19)$$

où $I1$ et $I2$ sont les valeurs de Laplaciens des deux images.

Propagation du déplacement des contours

Le déplacement obtenu à une résolution donnée est utilisé pour préciser la zone de recherche d'un correspondant à une résolution plus fine. Autrement dit, nous faisons une propagation du déplacement de points de contours du niveau plus grossier au niveau plus fin.

Soit $M = (i, j)$ un point de contour de la première image à la résolution inférieure, le déplacement $\vec{t} = (k_i, l_j)$ de ce point est estimé en cherchant le plus proche voisin qui a été mis en correspondance à la résolution supérieure (voir Figure 1.6).

Le calcul du déplacement exact de ce point M se fait alors par recherche de la valeur minimale de la corrélation :

$$\vec{d} = (k_i, l_j) = \left\{ \Delta \in F2 \text{ qui minimise } \sum_{\delta \in F1} (I1(M) - I2(M + \delta + t + \Delta))^2 \right\} \quad (1.20)$$

où $F1$, $F2$, $I1$, $I2$ sont les mêmes que précédemment. Donc la nouvelle estimation du déplacement est $\vec{t} + \vec{d}$.

L'estimation du déplacement des points de contours est ensuite propagée dans la pyramide du niveau supérieur au niveau inférieur, et ainsi de suite de façon récursive jusqu'au niveau le plus fin.

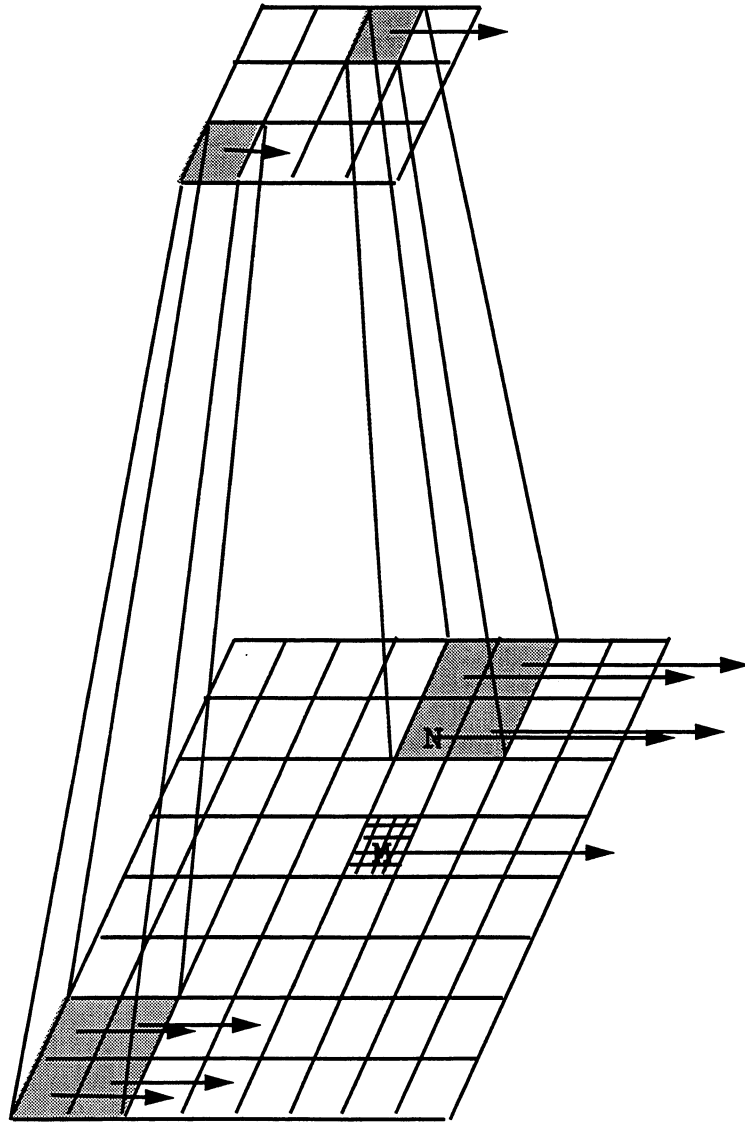


FIG. 1.6 - Le déplacement du point M est donné par celui du plus proche voisin N

Phase de sélection des points appariés

Après avoir achevé la propagation du déplacement, nous nous trouvons au niveau le plus fin de la pyramide. Pour tous les points de contours de la première image, nous avons les estimations du déplacement entre les deux images appariées. En pratique, la position des points de contours est un peu bruitée, de par la discrétisation du plan image et de par les erreurs de détection de contours. Il est donc indispensable d'identifier les fausses estimations sur les déplacements de contours. Nous utilisons pour cela une procédure qui permet de supprimer les appariements incorrects des points de contours de l'image. La sélection se fait de la manière suivante :

Pour un point $M = (i, j)$ de contour de la première image, nous avons son déplacement $\vec{d} = (k, l)$. Si le point dont les coordonnées sont $(i + k, j + l)$ ne se trouve sur aucun contour de la deuxième image, alors c'est un appariement incorrect. Nous disons que ce point n'a pas de correspondant dans la deuxième image. Cette brutale validation d'appariements de points nous permet de donner une bonne mise en correspondance de contours.

Accélération de corrélation

En pratique, le temps de calcul du processus de corrélation est assez coûteux. Le problème se pose très naturellement : comment pouvons nous accélérer le processus de corrélation ? D'après les formules (1.19) et (1.20), deux paramètres doivent être déterminés. L'un est la taille de la fenêtre de corrélation, l'autre est le nombre de pixels que nous prenons sur les contours. Dans notre expérimentation, la taille de la fenêtre de corrélation est 5×5 . Nous pouvons prendre un point de contour sur k , par exemple 1 point sur 2.

1.3.6 Contrainte épipolaire

Il existe entre les deux images d'une paire stéréoscopique des relations géométriques très fortes. Soit P un point de l'espace, les projections de P sur les deux images sont respectivement p_1, p_2 . La contrainte épipolaire est la seule véritable contrainte géométrique du problème de stéréovision. Cette contrainte spécifie que le correspondant p_2 dans la deuxième image d'un point p_1 dans la première image se trouve sur la droite épipolaire D_{12} associée à p_1 (voir Figure 1.7).

Corrélation épipolaire

Cette contrainte présente les deux intérêts suivants : d'abord, elle accélère la recherche du correspondant puisqu'elle permet de réduire le nombre de candidats possibles. En effet la recherche d'un correspondant ne s'effectue plus dans un espace à deux dimensions, mais dans un espace à une dimension : la droite épipolaire.

Les formules (1.19) et (1.20) deviennent dans ce cas les suivantes :

$$\vec{d} = (k_i, l_j) = \left\{ \Delta \in D_{12} \text{ qui minimise } \sum_{\delta \in F_1} (I_1(M) - I_2(M + \delta + \Delta))^2 \right\} \quad (1.21)$$

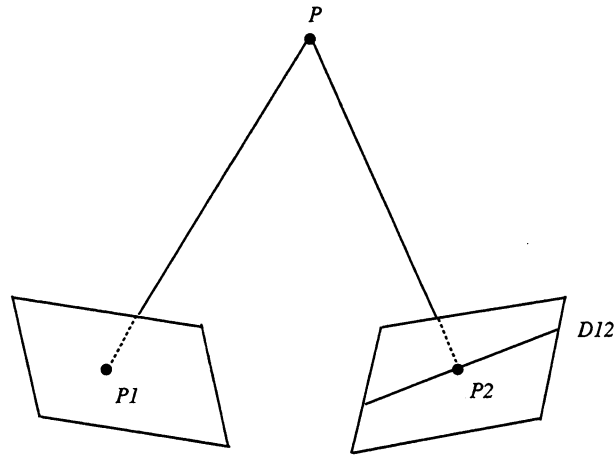


FIG. 1.7 - Contrainte épipolaire

$$\vec{d} = (k_i, l_j) = \left\{ \Delta \in D12 \text{ qui minimise } \sum_{\delta \in F1} (I1(M) - I2(M + \delta + t + \Delta))^2 \right\} \quad (1.22)$$

$F1$, $I1$, $I2$ étant définis précédemment, $D12$ est la droite épipolaire associée au point M .

En pratique, la géométrie épipolaire est imprécise. Chaque droite épipolaire $D12$ définit une bande épipolaire contenant le point correspondant. La corrélation doit donc être faite sur cette bande (voir Figure 1.8). La taille $m \times n$ de la bande épipolaire a été ajustée en fonction des expérimentations.

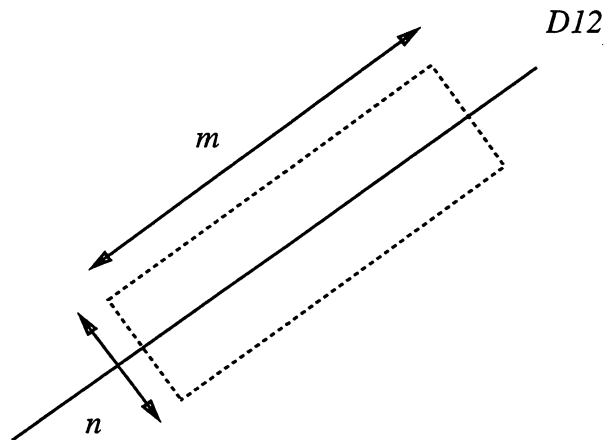


FIG. 1.8 - Bande épipolaire

Matrice essentielle et calcul des lignes épipolaires

Un algorithme simple de reconstruction d'une scène à l'aide de 8 points suivis dans deux images projectives est développé par Longuet-Higgins [LH 81] et Tsai [Tsa 84]. La

matrice essentielle joue un rôle très important à ce stade. Soient $\{x_i\}_{i=1\dots q}$ un ensemble de points d'intérêt en coordonnées homogènes dans la première image, $\{y_i\}_{i=1\dots q}$ l'ensemble de leurs correspondances en coordonnées homogènes dans la deuxième image, $E = (e_{ki})_{k=1\dots 3, i=1\dots 3}$ la matrice essentielle entre ces deux images. D'après la contrainte de coplanarité [Fau 89], [Hua 89], [May 90], [Fau 92a], et [Fau 92b], nous avons la relation suivante :

$$\forall i, x_i^t E y_i = 0 \quad (1.23)$$

Etant donné un point x_i dans la première image, il est alors très facile de calculer sa ligne épipolaire dans la deuxième image à l'aide de la matrice essentielle E . L'équation de la droite épipolaire est donnée par $E x_i = 0$.

Nous remarquons que la matrice E est une matrice de type 3×3 , possédant 9 éléments définis à un facteur multiplicatif près. Nous pouvons fixer le neuvième élément de la matrice, e_{33} , par exemple à 1. Finalement, E a 8 degrés de liberté. Il nous faut donc au moins 8 points pour calculer E car chaque point de l'image fournit une équation (1.23), et E est de rang 2.

En pratique, un système de 8 équations linéaires ne donne pas de bons résultats à cause des coordonnées bruitées des points de l'image. Lors de notre expérimentation, nous avons utilisé une méthode non linéaire développée par Quan [Moh 92] qui cherche à minimiser la formule suivante :

$$\min\{d(x_i^t, E y_i)^2 + d(y_i^t, E^t x_i)^2\} \quad (1.24)$$

où d est la distance euclidienne du plan.

1.3.7 Résultats expérimentaux

Dans ce paragraphe, nous montrons les résultats de mise en correspondance point à point de notre algorithme pyramidal sur différentes pyramides et différents types de scènes : des scènes d'intérieur et d'extérieur.

Images pyramidales

La Figure 1.9 présente deux images à apparier de taille 512×512 .

La Figure 1.10 présente deux pyramides gaussiennes de 4 niveaux de deux images à apparier.

Les contours d'images sont obtenus par le logiciel de traitement d'images FEAGRO dont nous disposons dans notre équipe.

La Figure 1.11 présente deux pyramides gaussiennes de 4 niveaux de contours de deux images à apparier.

Corrélation simple

Au cours de cette recherche, nous avons étudié la corrélation simple, la corrélation normalisée et la corrélation absolue sur les différents types d'images : images de niveau

FIG. 1.9 - Deux images à apparier de taille 512×512

de gris, images de gradient, images de Laplacien. La taille de la fenêtre de corrélation est de 5×5 .

Pour la scène de la tasse, la taille de la fenêtre de recherche de corrélation simple est de 3×3 au niveau 4 de la pyramide.

Nous testons notre algorithme sur les trois pyramides suivantes : pyramide 4 Points, pyramide Passe-Bas, pyramide Gaussienne.

- Une scène d'intérieur

1. Pyramide 4 Points.

La Figure 1.12 présente les résultats des mises en correspondance point à point de contours d'une portion d'image en effectuant la corrélation simple sur les images de Laplacien.

Nous avons superposé deux images de contours, et dessiné des flèches entre les points correspondants des deux images. Nous remarquons facilement que certaines flèches sont complètement erronées.

Cette solution est performante quant à sa vitesse de calcul. En revanche elle présente sur le plan théorique un défaut de repliement de spectre et ses résultats expérimentaux sont médiocres.

2. Pyramide Passe-Bas.

La Figure 1.13 présente les résultats des mises en correspondance point à point de contours d'une portion d'image en effectuant la corrélation simple sur les images de gradient.

Les résultats de cette méthode sont meilleurs que ceux de la précédente.

3. Pyramide Gaussienne



FIG. 1.10 - Deux pyramides gaussiennes

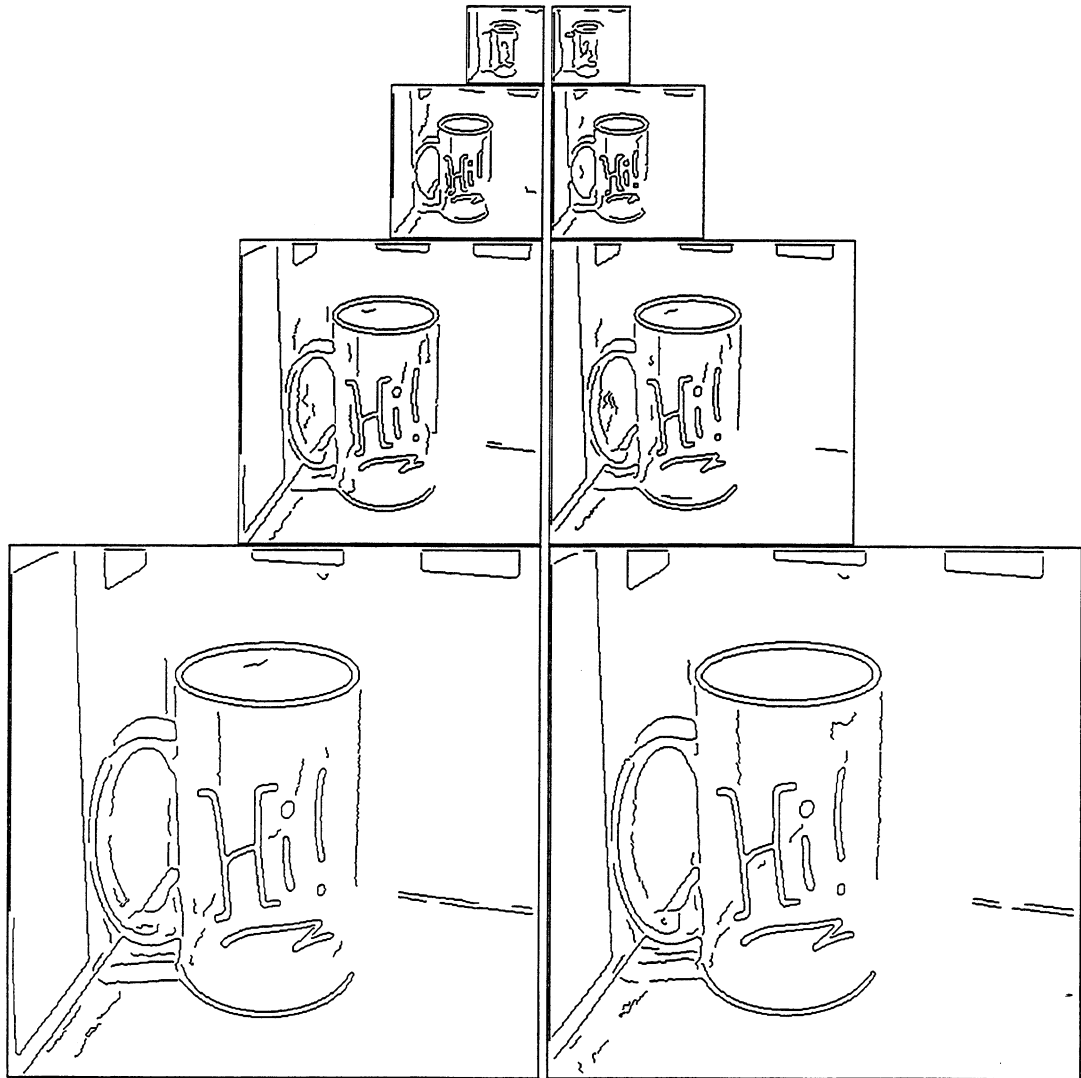


FIG. 1.11 - Deux pyramides gaussiennes

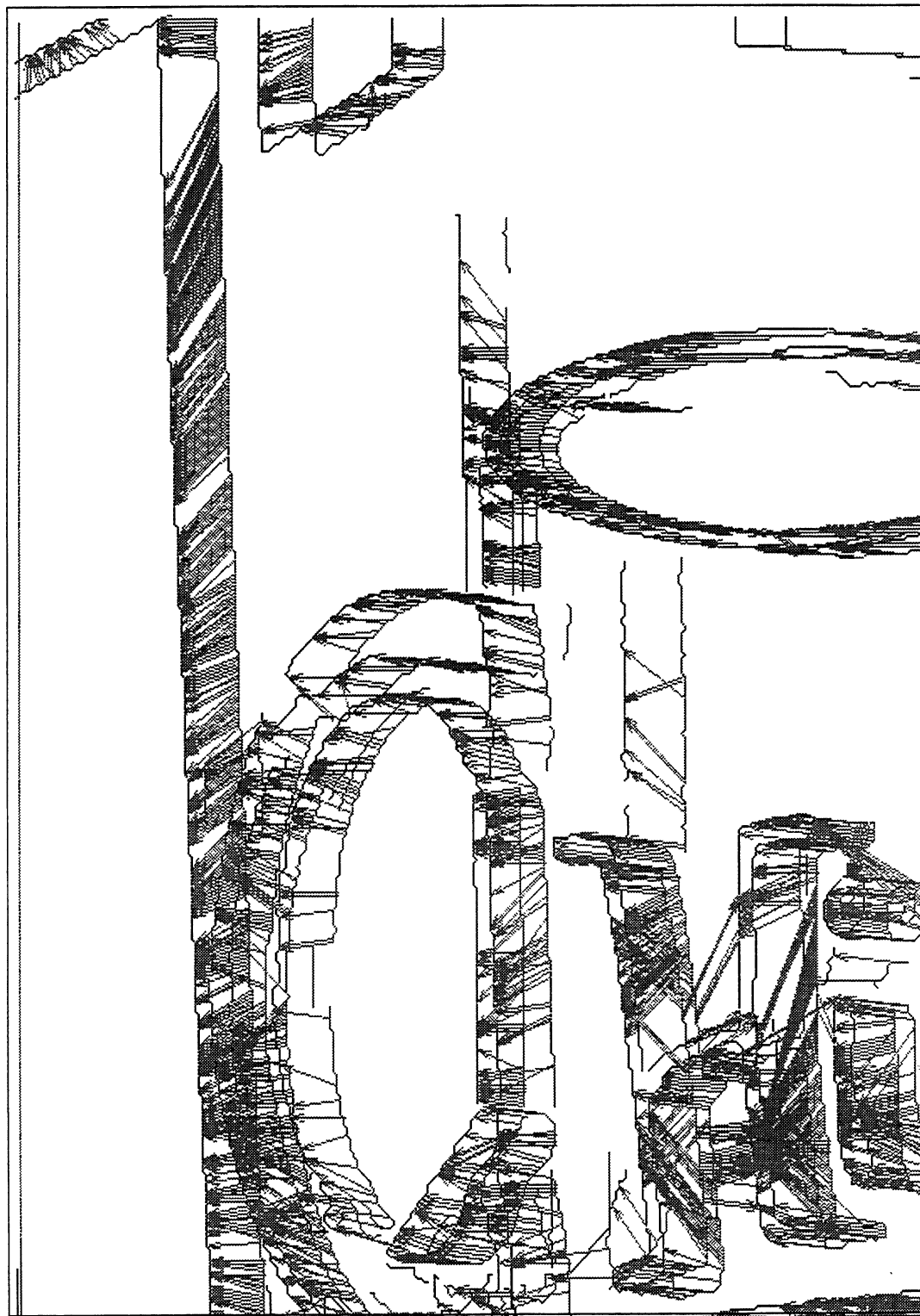


FIG. 1.12 - L'information de mise en correspondance (pyramide 4 Points, corrélation de Laplacien).

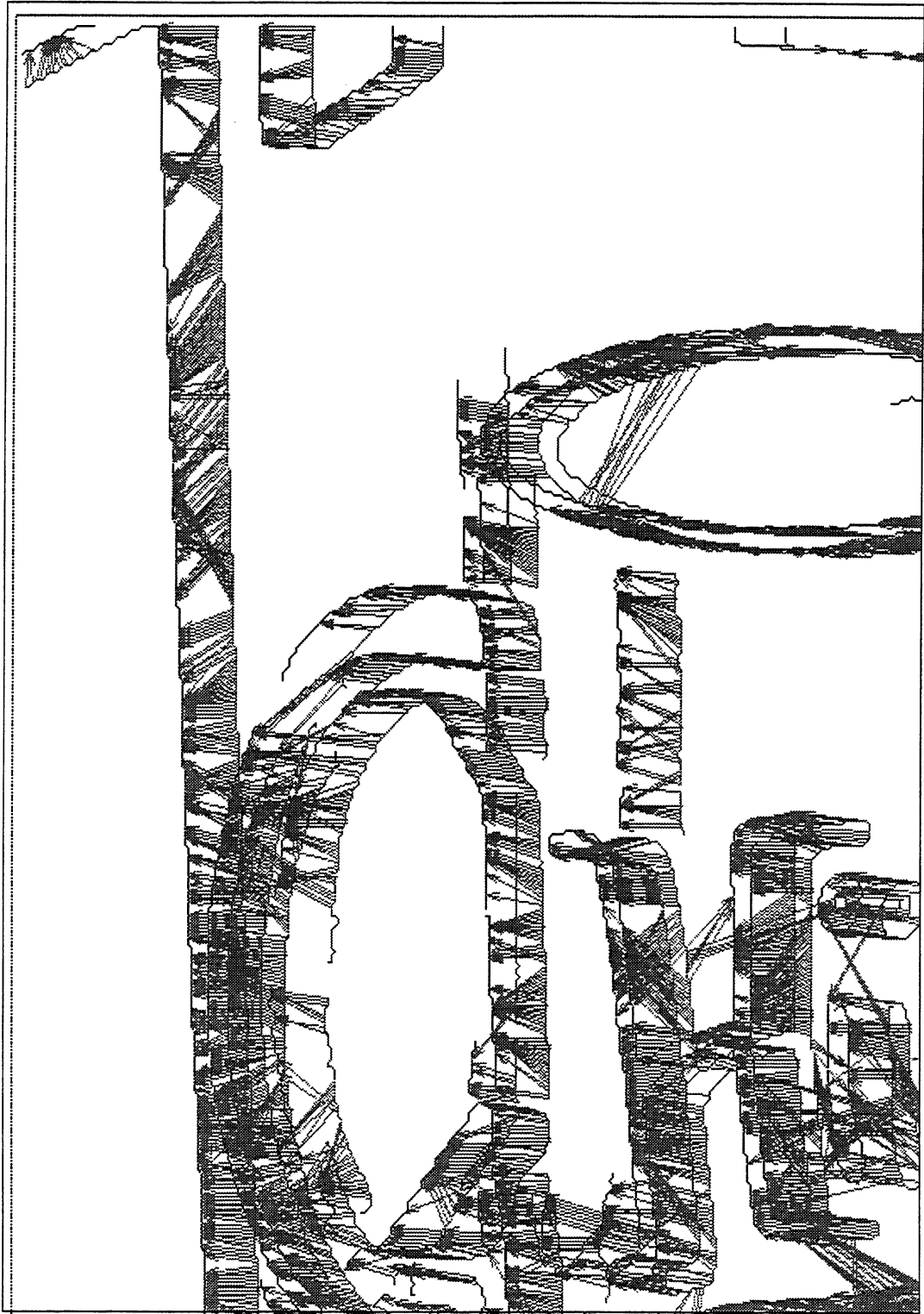


FIG. 1.13 - L'information de mise en correspondance (pyramide Passe-Bas, corrélation de gradient).

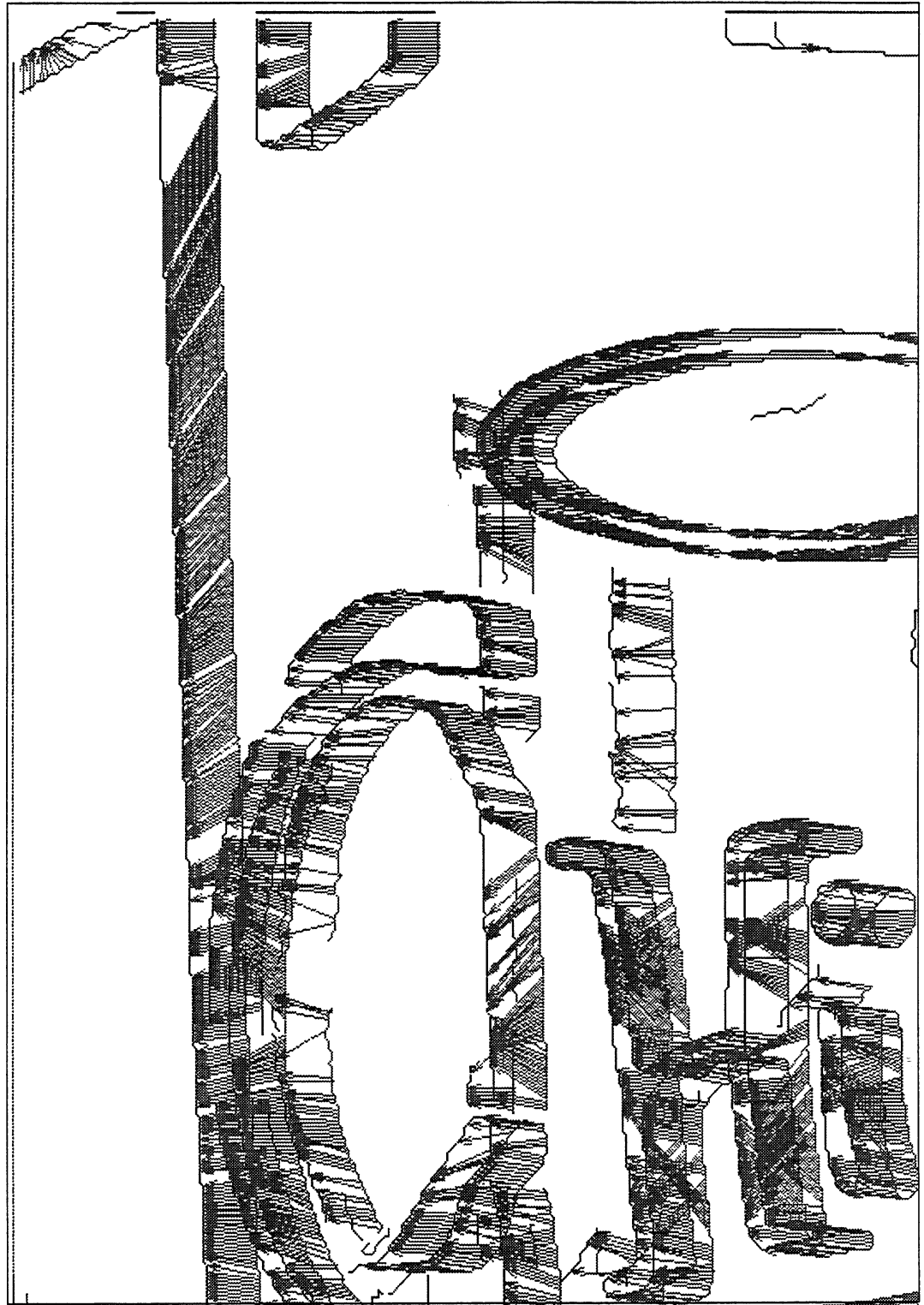


FIG. 1.14 - L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien).

La Figure 1.14 présente les résultats des mises en correspondance point à point de contours d'une portion d'image en effectuant la corrélation simple sur les images de Laplacien.

La Figure 1.15 présente les résultats des mises en correspondance aux différents niveaux; les points indiqués sont tous les points des contours de la première image qui ont un correspondant dans la deuxième image.

Dans la Figure 1.16, seul un point sur trois a été pris en compte.

Dans la Figure 1.14, nous remarquons que les flèches se croisent assez fortement entre certains contours. Cependant, la plupart des mises en correspondances sont correctes (95%).

– Une scène d'extérieur

Cette scène est fournie par le projet PROMETHEUS.

La Figure 1.17 présente deux images à apparier de taille 512×256 .

La Figure 1.18 présente les résultats des mises en correspondance point à point de contours d'une portion d'image en effectuant la corrélation simple sur les images de Laplacien.

Bien que ces deux images soient bruitées, nous observons que la plupart des contours sont bien mis en correspondance dans la Figure 1.18, à l'exception de quelques uns près du bord gauche.

Corrélation Normalisée

Pour la scène de la tasse, la taille de la fenêtre de recherche de corrélation normalisée est de 3×3 au niveau 4 de la pyramide.

Nous avons aussi testé plusieurs sortes de corrélation normalisée (voir le paragraphe 1.2.2) sur les différents types d'images: images de niveau de gris, images de gradient, images de Laplacien. Finalement, nous avons conclu que la corrélation normalisée donne des résultats plus intéressants que la corrélation simple.

Par exemple, la Figure 1.19 présente les résultats des mises en correspondance point à point de contours d'une portion d'image en effectuant la corrélation normalisée sur les images de Laplacien de la pyramide Gaussienne.

Corrélation dans la bande épipolaire

La corrélation présente le défaut d'être lente. Cela est dû en partie au fait que le parcours complet de la fenêtre de corrélation est très coûteux. Nous avons vu que la contrainte épipolaire permet de diminuer la zone de recherche de corrélation.

Dans notre recherche, nous nous intéressons aux extrêmes de courbure des contours après leur lissage par des B-splines. Ces points sont des primitives robustes de l'image. Nous effectuons les mises en correspondance de ces points par la corrélation [Moh 92], et nous appliquons la méthode de Faugeras [Fau 92b] pour calculer les matrices épipolaires des images. Puis nous calculons dans la deuxième image les lignes épipolaires correspondant aux points d'intérêt de la première image.

– Une scène de tasse

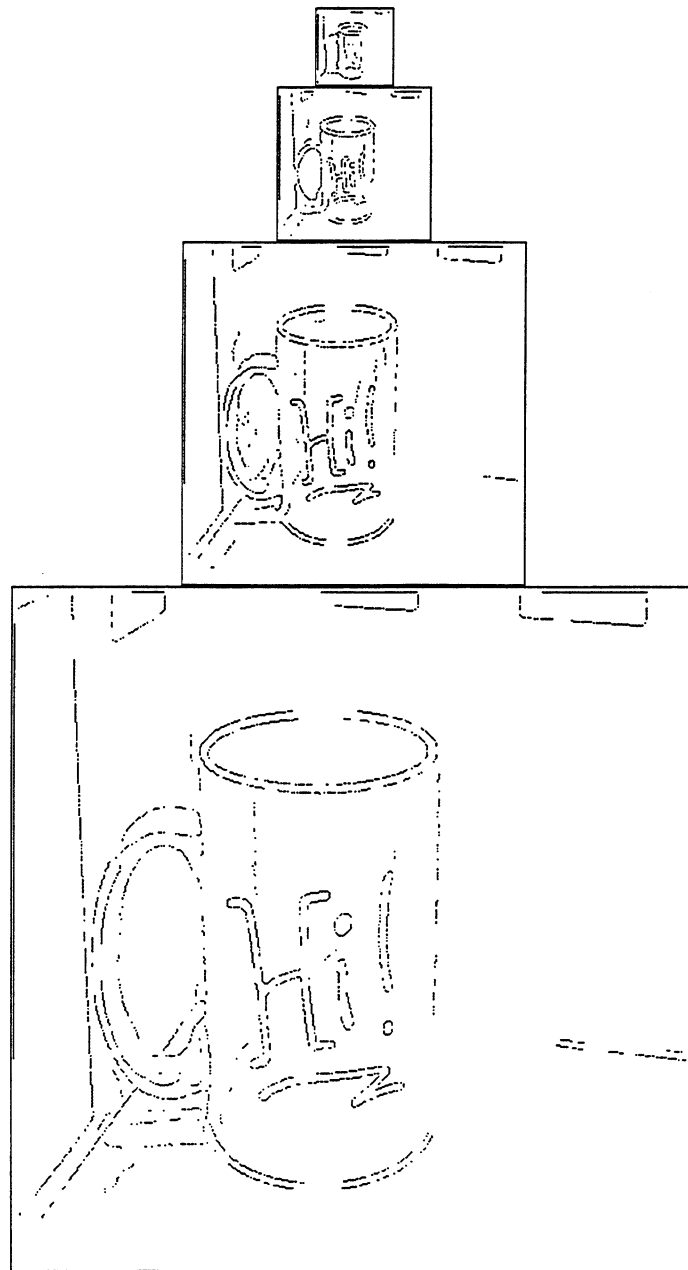


FIG. 1.15 - Les points indiqués sont tous les points des contours de la première image qui ont un correspondant dans la deuxième image.

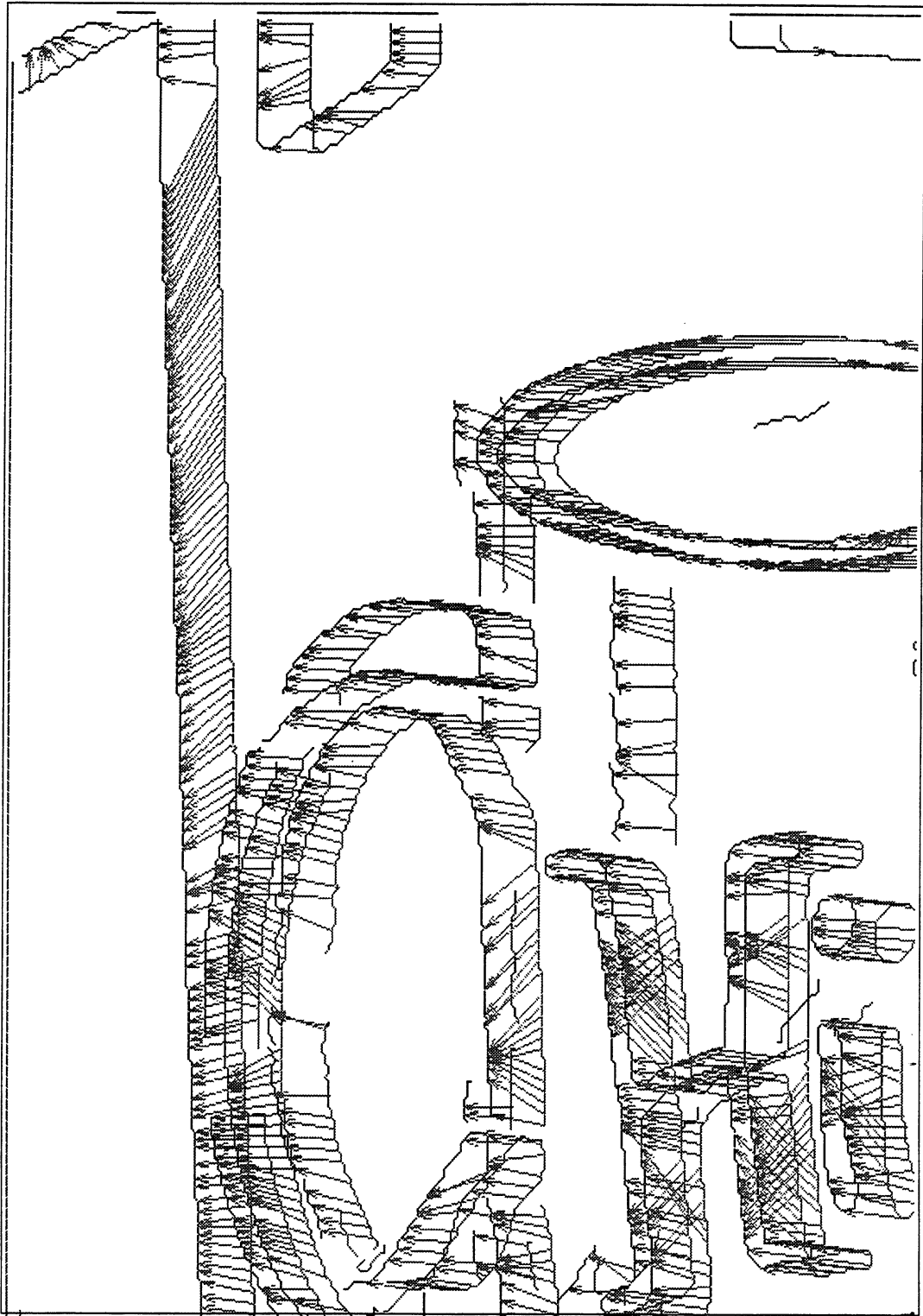


FIG. 1.16 - L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien, seul un point sur trois a été pris en compte).

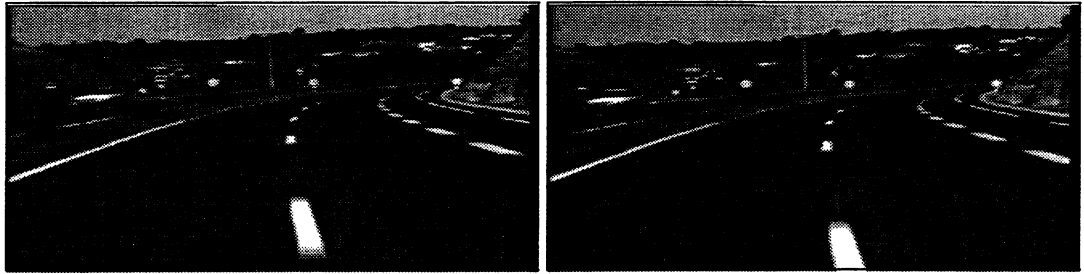


FIG. 1.17 - Deux images à apparier de taille 512×256

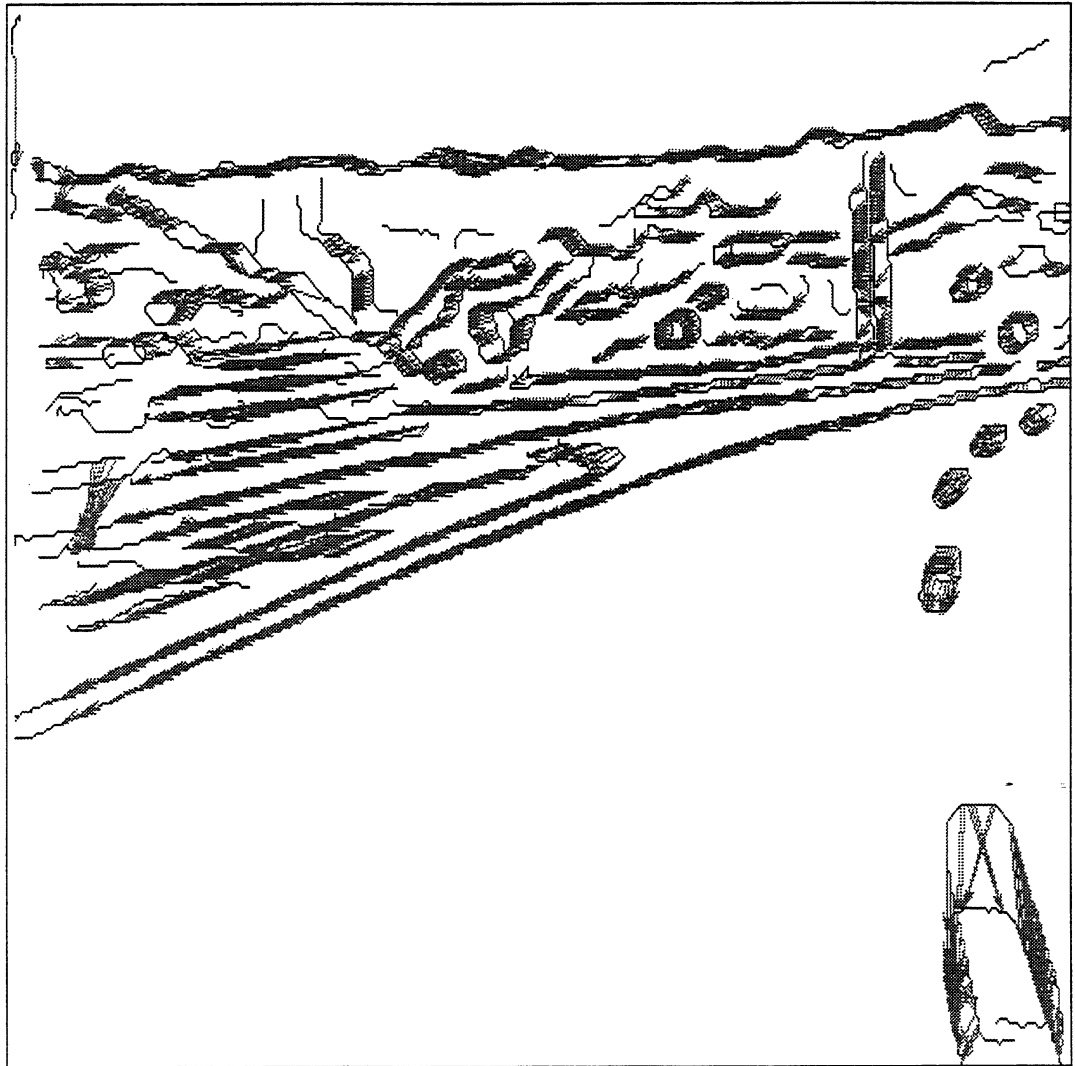


FIG. 1.18 - L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien).

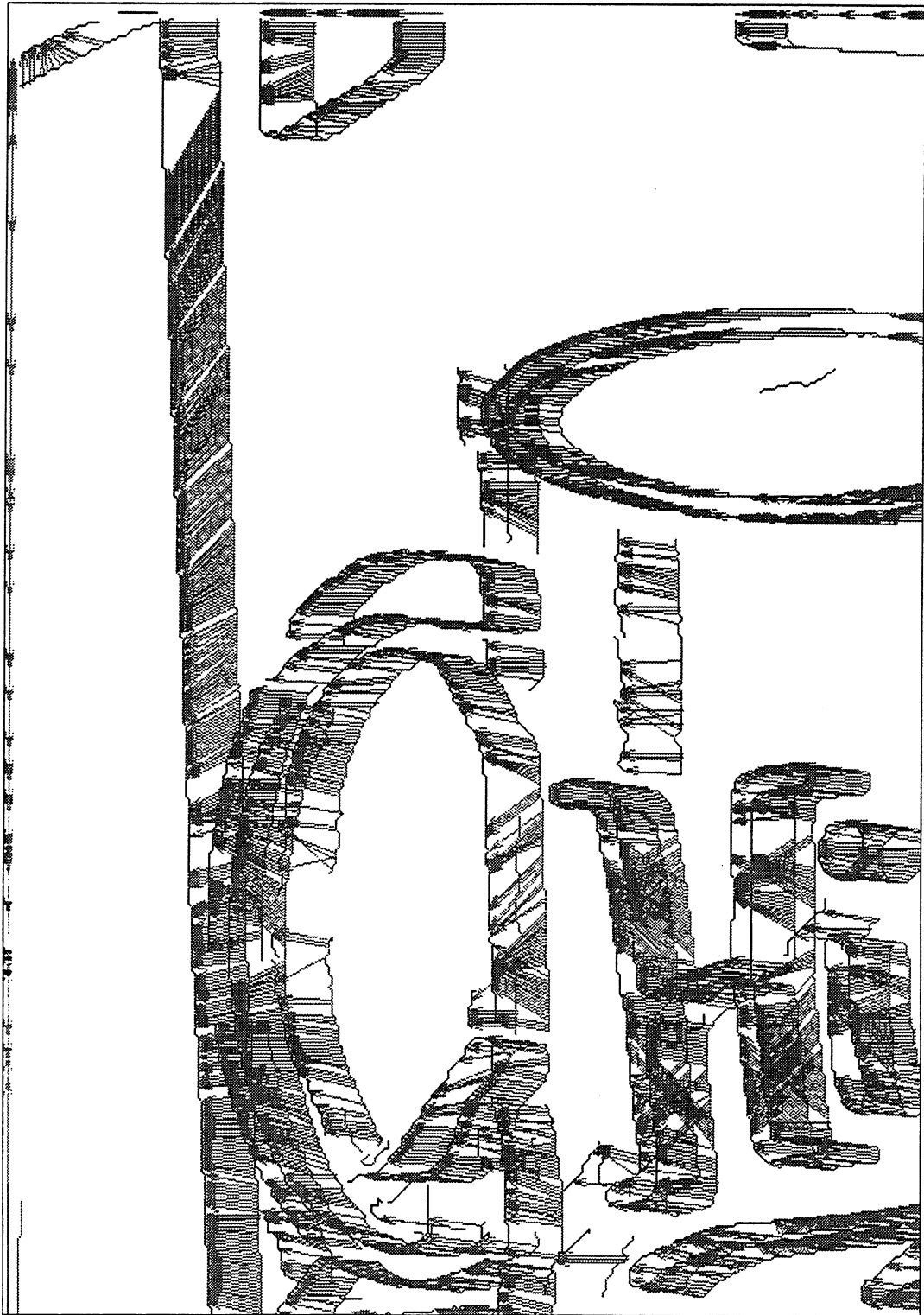


FIG. 1.19 - L'information de mise en correspondance (pyramide Gaussienne, corrélation normalisée de Laplacien).

La Figure 1.20 présente les lignes épipolaires de deux images à appairer de taille 512×512 .

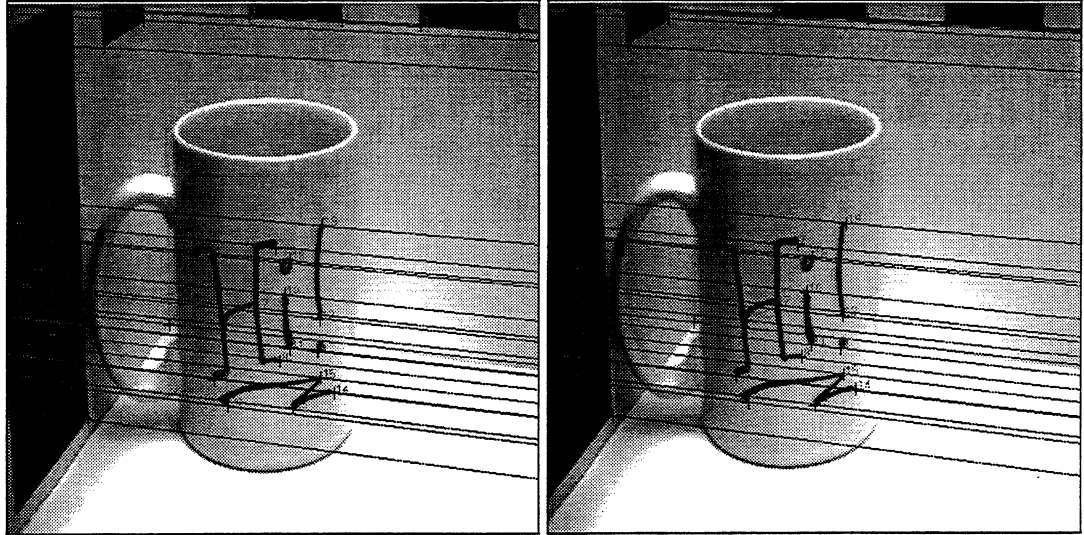


FIG. 1.20 - Deux images à appairer de taille 512×512 avec les lignes épipolaires

La Figure 1.21 présente les résultats des mises en correspondance point à point de contours d'une portion d'image. La taille de la bande épipolaire (fenêtre de recherche de la corrélation) au niveau 4 de la pyramide est 3×1 .

- Une scène de masque

Cette scène est fournie par le projet ORASIS.

La Figure 1.22 présente les lignes épipolaires de deux images à appairer de taille 512×512 .

La Figure 1.23 présente les résultats des mises en correspondance point à point de contours d'une portion d'image. La taille de la bande épipolaire (fenêtre de recherche de la corrélation) au niveau 4 de la pyramide est 2×1 .

Donc la fenêtre de recherche de corrélation épipolaire est plus petite que celle de la corrélation sans la contrainte épipolaire. Les résultats de mise en correspondance point à point sont très satisfaisants. Le temps de calcul est divisé par 3.

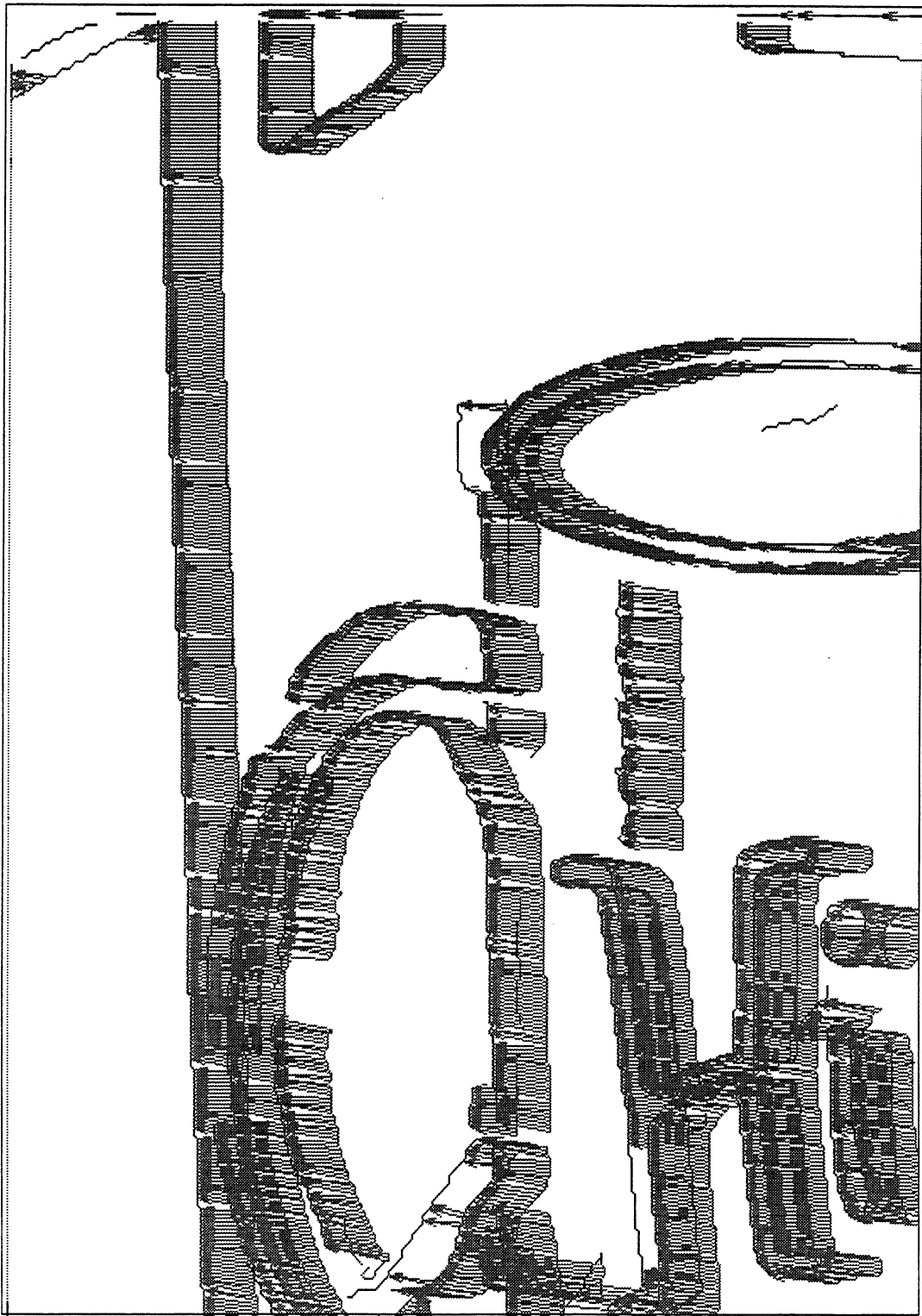


FIG. 1.21 - L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien).

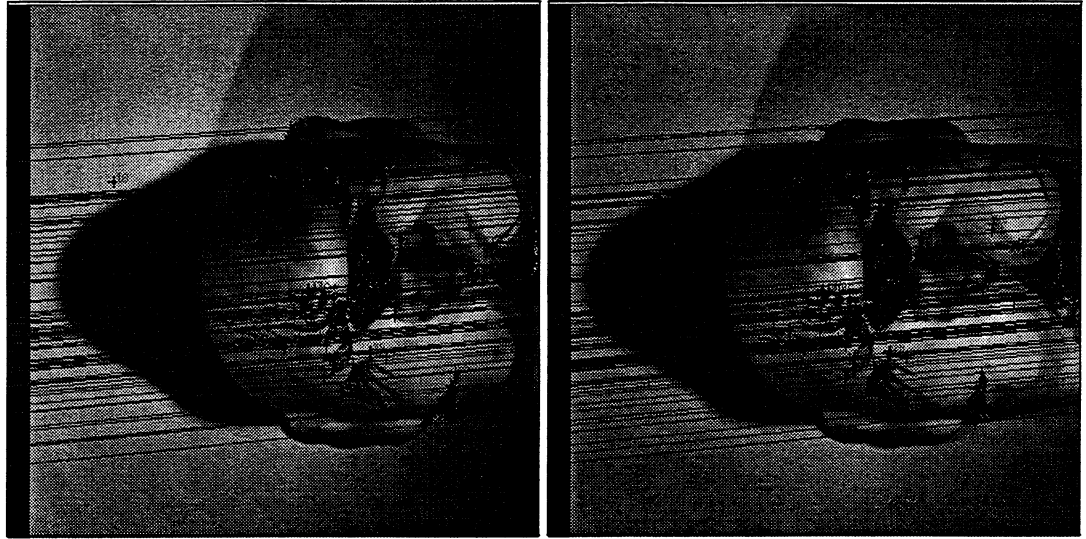


FIG. 1.22 - Deux images à apparier de taille 512×512 avec les lignes épipolaires

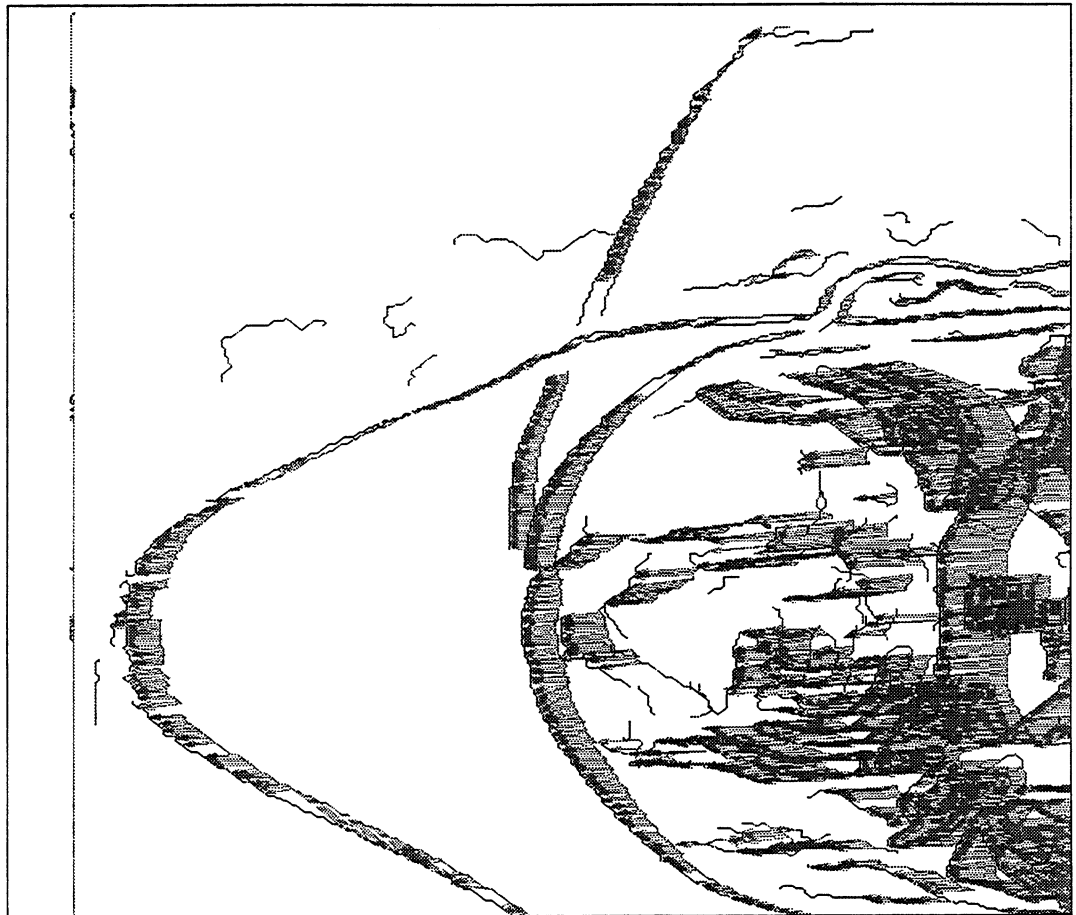


FIG. 1.23 - L'information de mise en correspondance (pyramide Gaussienne, corrélation de Laplacien).

1.4 Les raccordements et coupures de contours courbes

Dans ce paragraphe, nous nous proposons d'étudier les possibilités de correction des chaînes de contours.

De manière globale, notre objectif est de fournir les séquences de chaînes de contours les plus riches et les plus fiables possibles. Cette étude préalable doit permettre de faciliter l'étape de reconstruction des surfaces des objets de la scène perçue. Plus précisément, nous agissons dans les cinq directions suivantes :

1. mise en correspondance de contours.
2. détection et élimination des mises en correspondance peu fiables.
3. détection et élimination des chaînes de contours non productives⁴.
4. raccordement des chaînes de contours.
5. coupure des chaînes de contours.

1.4.1 Mise en correspondance de contours

La phase précédente nous permet de disposer de l'information locale de mise en correspondance point à point sur les contours. Celle-ci permet d'effectuer facilement la mise en correspondance entre contours. Les résultats sont satisfaisants dans le cas général. Certaines correspondances apparaissent toutefois surprenantes. Les flèches (mise en correspondance point à point) se croisent assez fortement sur certains contours. De petits décalages locaux entre les points appariés provoquent alors des superpositions et des ruptures de contours [Zha 92]. Ce phénomène est dû au fait qu'il n'est pas possible d'estimer exactement le flot optique mais seulement sa composante normale au contour [Hil 84, Nag 87]. Cependant, les décalages étant locaux, la mise en correspondance contour à contour reste assez robuste.

Coupure ou jonction de contours

C'est le cas (voir Figure 1.24) qui arrive le plus souvent en pratique. Le contour A de la première image se projette sur les contours B et C de la deuxième image. Nous les raccordons et nous les considérons comme un même contour sursegmenté par la technique de B-splines cubiques régularisées. Cette technique permet des interpolations, ce qui nous évite de revenir sur le signal image.

Réciproquement, si les contours B et C de la première image se projettent sur le contour A de la deuxième image, nous les raccordons de la même façon que précédemment.

Variation dans la segmentation de contours

Au cours de notre expérimentation, nous rencontrons un cas assez délicat : la branche horizontale de A est appariée avec la partie gauche de C , la partie verticale de A avec

⁴Ce sont des contours "bruités" ne correspondant à aucune réalité dans la scène perçue vis à vis de notre objectif de la reconstruction de surfaces.

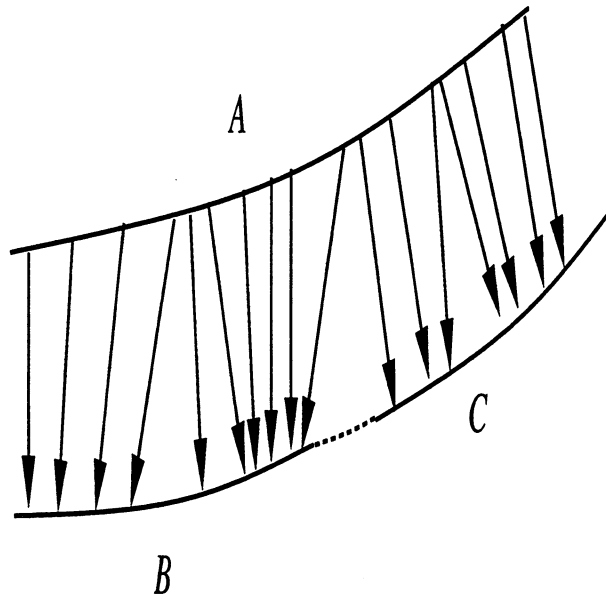


FIG. 1.24 - Cas de contours coupés

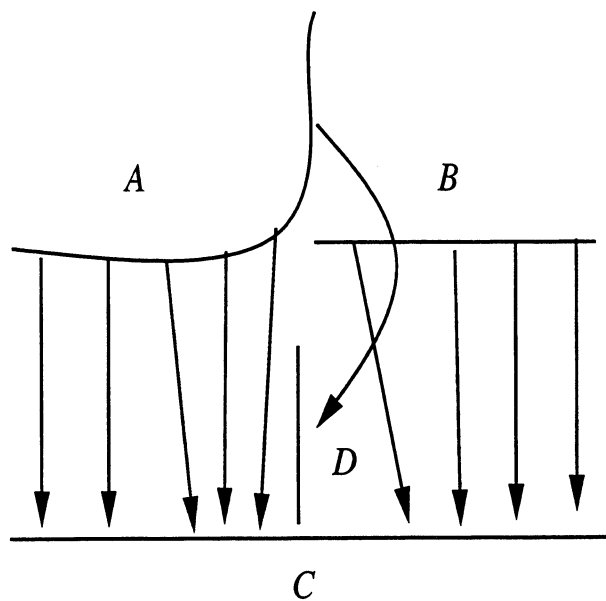


FIG. 1.25 - Cas de contours se segmentant de façon différente

D et B avec la partie droite de C (voir Figure 1.25). Nous coupons le contour A en deux parties, la branche verticale de A restant appariée avec D .

Erreurs de mise en correspondance

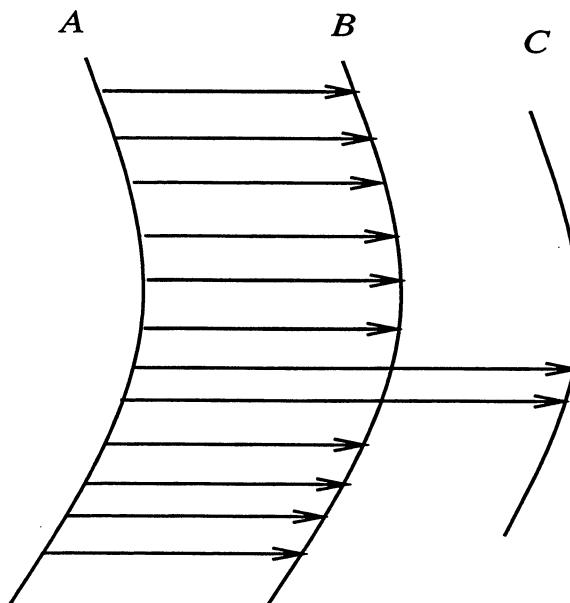


FIG. 1.26 - Cas d'erreurs de mise en correspondance de points sur les contours

Un cas typique d'erreur est celui pour lequel la plupart des points du contour A sont appariés avec les points du contour B , peu de points étant appariés avec les points de C (voir Figure 1.26). D'après la contrainte de continuité de contours, nous choisissons d'apparier le contour A avec B .

1.4.2 Détection et élimination des mises en correspondance chaîne à chaîne peu fiables

Motivations

Nous avons remarqué que les techniques actuelles de mises en correspondance point à point des chaînes de contours ne sont pas parfaites (voir Figure 1.26). En particulier, l'observation montre qu'un nombre réduit de points appariés entre deux chaînes de contours relativement longues ne constitue pas une information fiable permettant de conclure à la correspondance des deux chaînes.

Critères de fiabilité des appariements de contours

Au niveau bas du processus de vision qui est le nôtre, nous considérons une mise en correspondance entre deux chaînes de contours comme **non fiable**, lorsque le nombre de points suivis entre les deux chaînes, rapporté au nombre de points de la chaîne la moins longue, est inférieur à un certain seuil. Ce seuil a été ajusté en fonction des expérimentations.

1.4.3 Détection et élimination des chaînes de contours non productives

Motivations

Notre recherche se situe dans le cadre de la reconstruction des surfaces des objets d'une scène perçue. Dans cette optique, nous souhaitons mettre en évidence des séquences de chaînes de contours mises en correspondance par l'étape du suivi. Ces séquences ne sont intéressantes que dans la mesure où elles mettent en jeu un nombre suffisant d'images de la séquence.

Critères de non-productivité

Nous qualifions de **non productive** vis à vis de la reconstruction de surfaces, toute chaîne de la séquence qui à l'issue de l'étape d'élimination des appariements peu fiables n'est pas mise en correspondance, directement ou indirectement, avec un nombre suffisant de chaînes de contours de la séquence.

Le cas le plus simple de chaînes non productives est celui des chaînes isolées, qui ne font l'objet d'aucune mise en correspondance. L'observation montre que ces chaînes sont, dans la grande majorité des cas, des chaînes "bruitées" ne correspondant à aucune réalité dans la scène perçue.

1.4.4 Raccordement des chaînes de contours

Motivations

Nous observons très fréquemment que des chaînes de contours sont coupées en plusieurs sous-chaînes (voir Figure 1.24). Parfois même, une partie complète d'un contour de surface d'objet n'est pas représentée. Ces chaînes doivent être traitées de manière à fournir des images de chaînes de contours suffisamment riches pour la reconstruction des surfaces.

Critères de raccordement

Les raccordements doivent se faire sur la base de critères géométriques et photométriques. Les critères géométriques que nous avons retenus pour notre étude sont les suivants :

- proximité des extrémités à raccorder.
- continuité de tangence aux extrémités.
- continuité de courbure aux extrémités.

Nous n'avons retenu qu'un seul critère photométrique local :

- continuité d'orientation du gradient aux extrémités.

De nombreux autres critères peuvent être envisagés. Les paramètres à prendre en considération pour le choix de ces critères sont la vitesse de traitement de la séquence

et le degré de performance souhaité pour la correction des anomalies, tant sur le plan qualitatif que quantitatif.

Nous avons vu qu'à ce niveau bas de la vision, les informations photométriques relatives aux chaînes de contours sont relativement pauvres. Néanmoins, l'étape de mise en correspondance nous permet d'établir des liens photométriques entre les chaînes de contours. Nous verrons plus loin comment les exploiter.

1.4.5 Coupure des chaînes de contours

Motivations

Il arrive également parfois que deux courbes de natures différentes soient à l'origine d'une unique chaîne de contour (voir Figure 1.25). Ceci peut être dû à une mauvaise extraction de contours, ou à une mauvaise segmentation. Ce phénomène est indésirable et il nous faut le corriger.

Critères de coupure

Ce domaine de notre étude n'a pas fait l'objet de recherches approfondies. Nous ne nous intéresserons ici qu'à des critères de type contextuel. Nous n'aborderons pas le problème de la coupure de chaînes sur la base de critères intrinsèques : extrema de courbure, discontinuité de courbure, discontinuité de tangence, etc. . .

1.4.6 Une approche globale

Nous nous intéressons à l'élimination des erreurs produites par les étapes initiales du processus de vision. Nous développons ici une approche originale basée sur une technique de manipulation de graphes.

Après l'extraction des chaînes de contours et leurs mises en correspondance point à point par la technique de corrélation, nous tentons d'apporter une solution pour la correction des anomalies. L'étape de mise en correspondance point à point engendre elle-même ses propres erreurs. Nous verrons comment il est possible de prendre en compte ces erreurs.

Etant donnée une séquence d'images appariées de niveau de gris, après l'évaluation de la procédure de corrélation, nous produisons une séquence d'images de contours avec les coordonnées du déplacement des points de contours entre deux images appariées. A ce stade du processus de vision, nous disposons de quatre types d'informations :

1. des points de contours constituant des atomes.
2. ces points sont organisés en chaînes de contours constituant des éléments.
3. ces chaînes sont regroupées dans des images de chaînes de contours constituant des ensembles.
4. elles sont mises en relation par des relations de mise en correspondance point à point.

Cette observation nous conduit tout naturellement à une approche de type systématique, pour laquelle chaque constituant du système est en interrelation avec les autres constituants.

Afin de clarifier l'étude, nous distinguons trois familles de caractéristiques d'un contour :

- les caractéristiques internes.
- les caractéristiques statiques.
- les caractéristiques dynamiques.

Chaque chaîne de contour est considérée comme une entité présentant :

1. une longueur.
2. une évolution de la tangence tout au long de la chaîne.
3. une évolution de la courbure tout au long de la chaîne.
4. une proximité de ses extrémités.

Chaque chaîne est appréhendée comme un élément d'une image. A ce titre, elle entretient des relations tant géométriques que photométriques avec les autres chaînes de contours de l'image.

Nous introduisons ici la notion fondamentale d'affinité entre deux chaînes de contours.

Définition : Nous dirons que deux extrémités de chaînes présentent une **affinité** vis à vis du raccordement, lorsque certaines conditions de tangence, de courbure, de proximité et d'orientation du gradient sont observables au voisinage de ces extrémités.

Nous verrons plus loin comment coordonner la prise en compte des informations fournies par les quatre critères retenus.

A l'issue de l'étape de mise en correspondance point à point des chaînes de contours, les séquences d'images auxquelles nous nous intéressons sont structurées par la relation de correspondance.

Sous le nouvel angle adopté, chaque chaîne sera appréhendée comme un élément d'un ensemble de chaînes prises dans la séquence, toutes ces chaînes étant reliées entre elles, directement ou indirectement, par des relations de correspondance.

1.4.7 Modélisation par les graphes

La description du système, telle que nous l'avons présentée, nous a mené tout naturellement à une modélisation basée sur les graphes.

Nous précisons dans un premier temps les nouvelles instances du problème, puis nous montrons comment l'objectif à atteindre pour la séquence se réécrit dans ce nouveau contexte.

Définitions des instances, graphe engendré par une chaîne

Les graphes utilisés doivent rendre compte de la complexité du système, et sont donc de nature un peu particulière.

Nous les appellerons **graphes spatio-temporels**. La notion d'espace fait référence à la disposition des chaînes dans l'image. La notion de temps fait référence à la succession des images dans la séquence au cours du temps.

Chaque graphe est structuré en composantes dites composantes-images. Chaque composante-image renvoie à une image de chaînes de la séquence. La succession des images dans la séquence induit un "ordre" des composantes-images dans le graphe.

Chacune des composantes est constituée de sommets représentant chacun une chaîne de points de contours de la même image. Chaque sommet est porteur d'un attribut lié au type de la chaîne qui lui est associée (fermée ou ouverte), et possède deux pôles correspondant aux deux extrémités de la chaîne. Nous verrons plus loin la raison d'être de ces deux éléments.

Nous distinguons deux types d'arcs :

Arcs de liaison : ce sont des arcs orientés arbitrairement d'un sommet d'une composante-image vers un sommet de la composante-image suivante. Ils représentent les correspondances de points entre les 2 chaînes associées aux sommets. Ces arcs sont pondérés par la valeur suivante :

$$\frac{N_a}{N_b}$$

où N_a = "nombre de points mis en correspondance entre les deux chaînes", N_b = "nombre de points de la chaîne la moins longue".

Notons au passage que ce calcul mène parfois à une valeur supérieure à 100, dans le cas où plusieurs points sont mis en correspondance avec un même point.

Arcs d'affinité : ce sont des arcs orientés. Ils mettent en jeu des couples de sommets d'une même composante-image. Chaque arc d'affinité occupe de manière exclusive un pôle sur chaque sommet lié. Chacun représente la relation d'affinité entre les extrémités des chaînes associées aux sommets. Les chaînes de contours dont nous disposons sont orientées par la direction du gradient. Dans ce contexte, l'arc d'affinité (s_0, s_1) représente la relation d'affinité entre la fin de la chaîne associée à s_0 et l'origine de la chaîne associée à s_1 .

Le poids de ces arcs relève d'un calcul mettant en jeu les divers critères d'affinité mis en évidence précédemment. La problématique soulevée par ce calcul est abordée plus loin.

Nous appellerons **graphe engendré** par une chaîne de points de contours, un graphe construit selon les spécifications ci-dessus. Dans ce graphe, l'ensemble des chaînes de points de contours associées aux sommets sera l'ensemble des chaînes de la séquence reliées directement ou indirectement par la relation de correspondance à la chaîne d'origine.

La figure 1.27 présente un graphe engendré par une chaîne de contour d'une séquence de 7 images.

Caractérisation des instances

Deux éléments nous intéressent particulièrement dans les graphes engendrés : les embranchements, et la longueur des chaînes pendantes où des chaînes reliant 2 embranchements consécutifs.

Embranchements :

Ils peuvent traduire 4 réalités différentes sur la séquence.

1. une disjonction de surfaces d'un objet de la scène.
2. la présence d'une chaîne de contour incomplète.
3. une mauvaise segmentation de contours.
4. une erreur générée par l'étape de mise en correspondance point à point des chaînes de contours.

Pour les disjonctions de surface, l'embranchement correspond à une réalité dans la scène perçue. Cette configuration apparaît peu fréquemment. Toutefois, on peut toujours imaginer des scènes spécifiques contenant une multitude d'objets présentant cette caractéristique. Ce cas n'est donc pas à exclure.

Dans les 3 autres cas, nous sommes en présence d'erreurs. L'objet de notre étude est précisément d'éliminer ces anomalies. La difficulté consiste à différencier ces 4 cas. Pour cela, nous allons nous appuyer sur un deuxième élément.

Longueur critique :

Nous introduisons ici une nouvelle notion. Nous nous intéressons exclusivement aux chaînes de contours pendantes, et aux chaînes joignant deux embranchements consécutifs, ces chaînes n'utilisant que des arcs de liaisons parcourus tous dans le même sens.

Nous appellerons **longueur critique** d'un graphe la longueur en dessous de laquelle une chaîne du type spécifié ci-dessus ne "présente plus de sens" vis à vis de la scène perçue.

Notre hypothèse est qu'en dessous de cette longueur, les chaînes de contours mises en jeu par une chaîne dans le graphe sont à l'origine de surfaces reconstruites ne correspondant à aucune réalité dans la scène.

La valeur de cette longueur critique dépend de la nature de la scène et de la densité de la séquence d'images. Ce sera donc un paramètre important pour notre étude. La Figure 1.28 présente des exemples de chaînes de longueur 1 reliant 2 embranchements consécutifs. Ces chaînes n'ont pas de sens si la séquence est suffisamment dense.

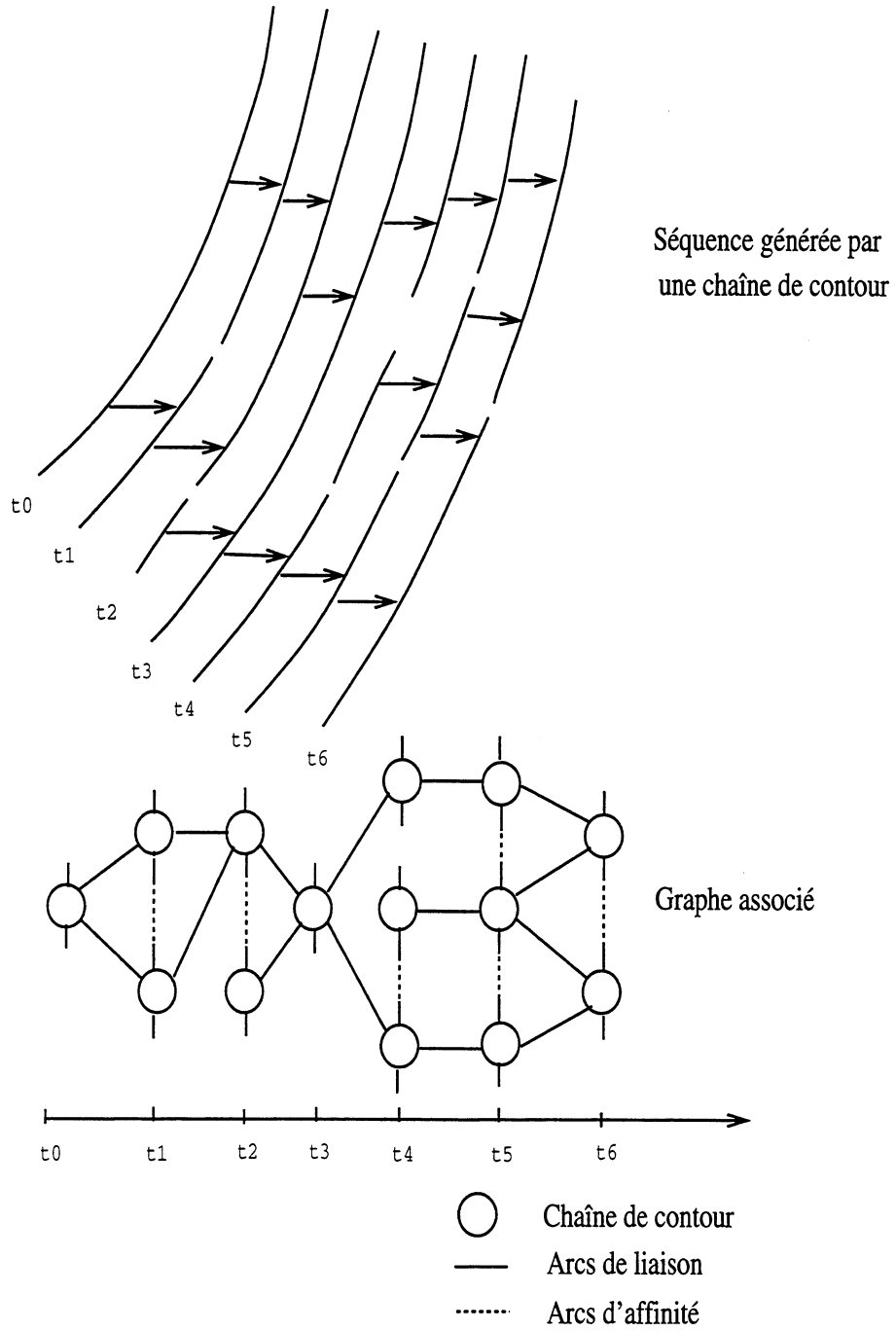


FIG. 1.27 - Graphe engendré par une chaîne de contour

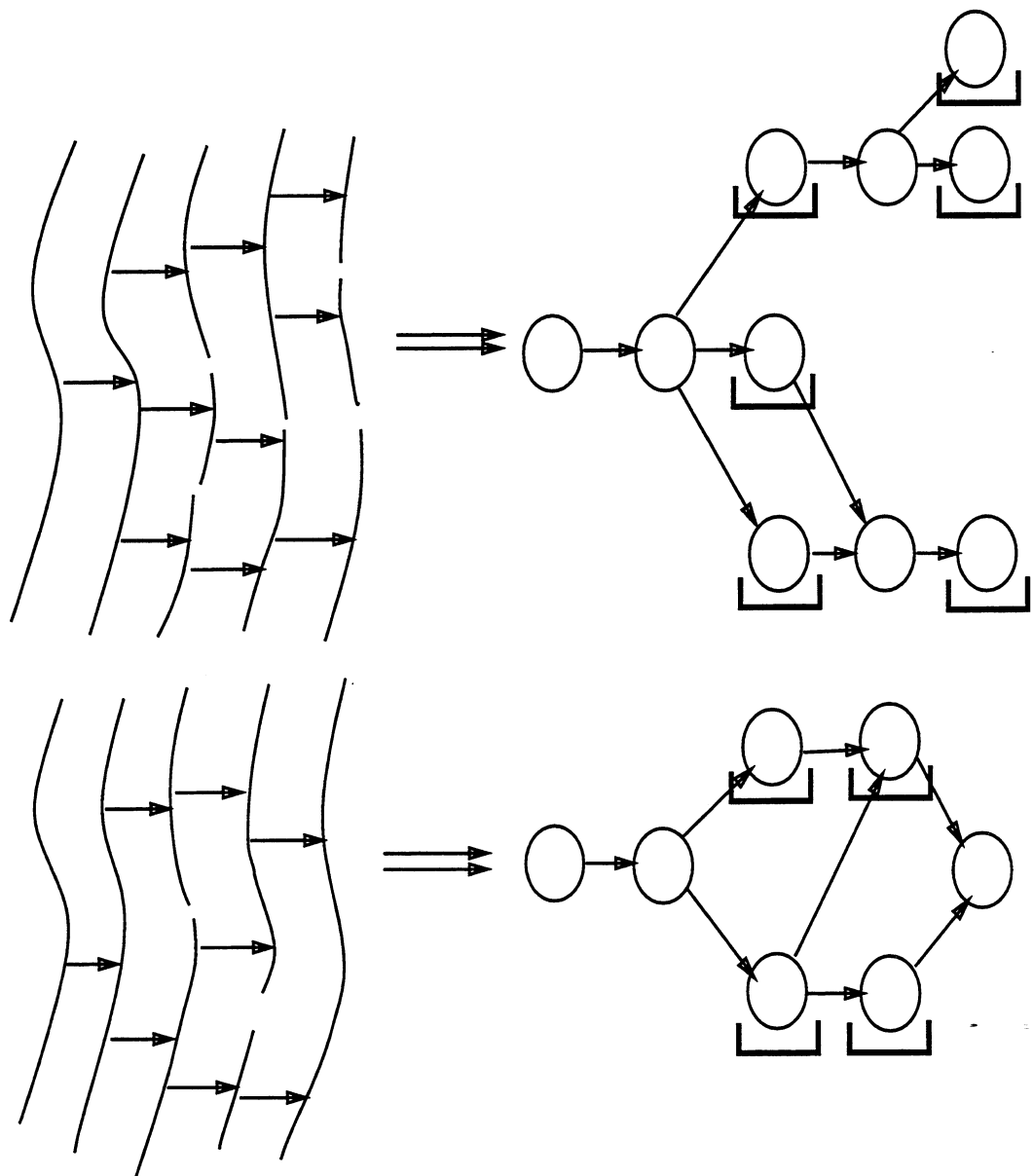


FIG. 1.28 - Exemples de chaînes de contenus sémantiques faibles

Nous notons au passage que pour une scène donnée, l'ajustement du déplacement de la caméra et de la fréquence des prises de vue permet d'assurer la stabilité de la longueur critique.

Les objectifs

Rappelons le quadruple objectif que nous nous étions fixé dans le paragraphe précédent pour la séquence :

- (1) détection et élimination des mises en correspondance chaîne à chaîne peu fiables.
- (2) détection et élimination des chaînes de contours non productives.
- (3) raccordement des chaînes de contours.
- (4) coupure des chaînes de contours.

L'objectif (1) se traduit trivialement sur les instances graphes par l'élimination des arcs de liaison dont le poids est inférieur à un certain seuil.

L'objectif (2) est également facilement transférable à notre nouveau contexte. Les chaînes de points de contours non productives génèrent des graphes de petite taille. Ces graphes sont facilement identifiables.

Pour les objectifs (3) et (4), nous avons envisagé dans le paragraphe précédent l'utilisation des critères suivants :

- (a) caractéristiques internes.
- (b) caractéristiques statiques.
- (c) caractéristiques dynamiques.

Ces rappels et ces remarques nous conduisent à formuler, dans le nouveau contexte, les objectifs suivants :

1. élimination des arcs de liaison dont le poids est inférieur au poids seuil (critère (c)).
2. non prise en considération des graphes engendrés dont la taille est inférieure à une valeur seuil (critère (c)).
3. traitement spécifique des sommets et des chaînes qui leur sont associées (critère (a)).
4. élimination des embranchements non pertinents (critère (c)).
5. élimination des chaînes de longueur inférieure ou égale à la longueur critique (critère (c)).

Ainsi, sur un plan qualitatif, les graphes, une fois traités, présenteront une diminution du nombre d'embranchements et du nombre de chaînes de longueur inférieure ou égale à la longueur critique.

1.4.8 Principe

Tous les objectifs formulés ci-dessus s'expriment en terme de transformations de graphes. Dans un premier temps, nous allons présenter les opérations de base que nous utiliserons dans le cadre de ces transformations.

Ces opérations étant définies, nous nous intéresserons à leurs conditions d'application. Pour résoudre ce problème, nous proposerons une méthode originale basée sur l'utilisation de règles de transformations de graphes.

Opérations de base

Cinq opérations de base sont nécessaires :

1. la destruction de graphes.

Cette opération s'applique aux graphes engendrés par des chaînes non productives. Pour la séquence, cette opération se traduit par l'élimination des chaînes de contours concernées et des relations de correspondance éventuellement mises en jeu.

2. la suppression d'arcs de liaison.

Pour la séquence, cette opération se traduit par l'élimination des relations d'appariement correspondant aux arcs supprimés (voir Figure 1.29).

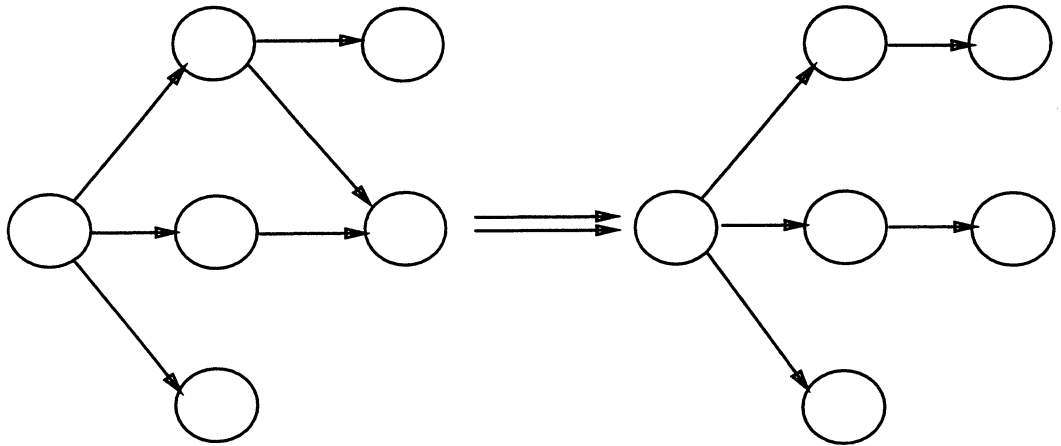


FIG. 1.29 - Suppression d'un arc de liaison dans un graphe spatio-temporel

3. la "fermeture" des sommets.

Cette opération correspond à la fermeture des chaînes de contours associées aux sommets.

4. la fusion de sommets.

Il s'agit, dans un graphe, de remplacer 2 sommets distincts par un seul et même nouveau sommet (voir Figure 1.30). Bien entendu, cette opération nécessite la gestion des arcs adjacents aux sommets fusionnés. Cette opération sur les graphes correspond dans la séquence à un raccordement de chaînes de contours. La fusion

de 2 sommets met en jeu un pôle unique sur chaque sommet. C'est la raison pour laquelle un sommet peut être concerné par 2 fusions : une sur chaque pôle.

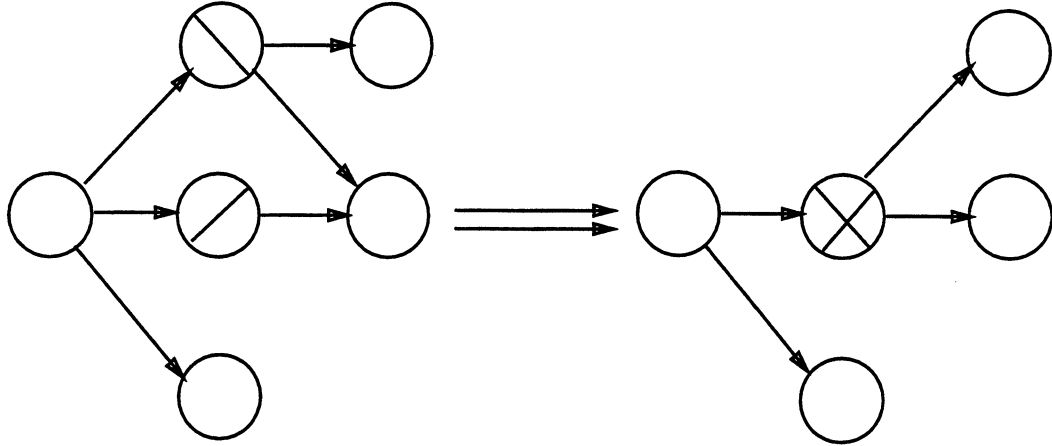


FIG. 1.30 - Jonction de sommets dans un graphe spatio-temporel

5. la disjonction de sommets.

Cette opération consiste à remplacer un sommet de graphe par 2 nouveaux sommets (voir Figure 1.31). Cette opération est beaucoup plus délicate que la précédente, en particulier pour la gestion des arcs adjacents au sommet disjoint. Elle correspond dans la séquence à une coupure de chaînes de contours.

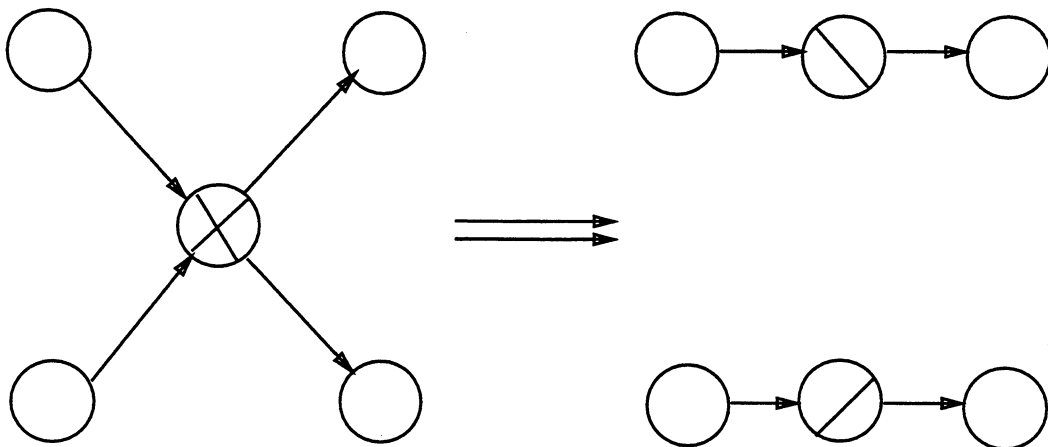


FIG. 1.31 - Disjonction de sommets dans un graphe spatio-temporel

Règles de transformations de graphes

Nous devons résoudre un triple problème.

- Quelles informations souhaitons-nous acquérir sur les graphes à traiter (prise d'informations)?

- Comment allons-nous traiter ces informations de manière efficace (traitement de l'information)?
- Comment allons nous décider de l'exécution de telle ou telle opération (prise de décision)?

Notre idée ici a été d'utiliser des règles de transformations de graphes. De quoi s'agit-il?

Une **règle** de transformations de graphes est spécifiée par la description d'une famille de graphes que nous appellerons motif, par une contrainte globale sur ce motif, par une liste d'opérations exécutables sur le motif et par une valeur de priorité. Un des intérêts de cette méthode est qu'elle est bien en accord avec l'approche de type systémique que nous avons adoptée. En effet, l'utilisation de motifs permet de rendre compte de la notion de contexte général et de contexte local.

En outre, cette technique de manipulation de graphes nous permet d'observer une certaine cohérence vis à vis de la modélisation choisie. Enfin, l'utilisation de ces règles nous permet d'aborder le problème à l'aide d'un outil formel.

Nous répondons aux 3 problèmes soulevés plus haut de la manière suivante :

1. la prise d'informations consiste en la recherche de sous-graphes dans un graphe.
2. la prise d'information et son traitement se font simultanément, puisqu'on ne recherche dans un graphe que des informations directement significatives pour notre problème.
3. la prise de décision pour une règle donnée est aisée puisqu'elle est décrite par la règle.

1.4.9 Description des règles de transformations de graphes

Les bases de notre approche globale étant maintenant clairement définies, nous allons étudier les problèmes particuliers apparus lors de la mise en œuvre et nous décrirons en détail les solutions retenues.

Différents éléments d'une règle

Comme nous l'avons écrit plus haut, une règle est composée de 4 éléments. Nous allons maintenant les détailler.

Motif :

Il spécifie une famille de graphes. Il est décrit par un graphe support de même nature que ceux utilisés pour la modélisation du problème. Des contraintes portant sur les arcs sont ajoutées : une valeur seuil et un type du seuil. Ce sont ces contraintes qui définissent la famille.

Rechercher un sous-graphe dans un graphe, c'est établir un isomorphisme entre les sous-graphe recherché et une partie du graphe. Une condition nécessaire pour qu'un représentant de motif soit reconnu dans un graphe est que pour chaque arc du graphe

associé à un arc du représentant, la contrainte définie par la règle soit respectée. Si le seuil est de type bas (resp. haut), le poids de l'arc doit être supérieur (resp. inférieur) à la valeur du seuil.

Notons que pour la suite, lorsque le contexte ne sera pas ambigu, nous assimilerons la notion de motif et celle de représentant de motif.

Contrainte globale :

Comme son nom l'indique, elle permet d'exprimer une contrainte globale sur le motif. Elle est décrite également par un type de seuil, et par une valeur seuil. Cette valeur représente la moyenne des poids des arcs. Son intérêt est d'autoriser des poids faibles sur certains arcs, tout en maintenant une forte cohésion du motif.

Liste des transformations :

Chaque transformation est décrite par la liste des sommets concernés, et par le type de l'opération à réaliser.

Nous rappelons les 3 opérations possibles :

1. fermeture: S(hut).
2. raccord: J(oin).
3. coupure: C(ut).

Priorité :

Elle est représentée par un entier. La plus haute priorité est la priorité 0, puis la priorité diminue lorsque cet entier croît. Nous verrons plus loin l'utilité de cet attribut.

Grammaire $LL(1)$

Nous avons cherché un moyen simple de décrire ces règles. Notre souhait était de pouvoir les produire facilement, et de pouvoir également vérifier, à l'aide d'un automate, que ces règles étaient syntaxiquement et sémantiquement correctes.

Nous avons utilisé la grammaire $LL(1)$ que voici :

Vocabulaire terminal :

```
Sv = {'graphe', 'globalContraint', 'transformations', 'priority',
      'dim', 'component', 'nodes', 'arcs', 'EQUAL', 'CLOSED',
      '=', '+', '*', 'H', 'L', 'J', 'C', 'S', ',', ';', ':',
      '[', ']', '(', ')', '0', '1', '2',
      '3', '4', '5', '6', '7', '8', '9'}
```

```

<Rule>          == 'graphe' ':' Graphe;
                  'globalContraint' ':' GlobalContraint;
                  'transformations' ':' LTransformations;
                  'priority' ':' Priority;

<Graphe>        == 'dim' ':' Integer /*Number of graphe components*/
                  Component LComponent

<LComponents>   == Component LComponents |
                  epsilon
<Component>     == 'component' ':'
                  'nodes' '=' Node LNodes
                  'arcs' '=' LArcs
<LNodes>        == Node LNodes |
                  epsilon
<Node>          == '[' CompAccess ',' NodeAccess ','
                  NodeType ',' Extension ']'
<CompAccess>    == Integer; /*First component is 0*/
<NodeAccess>    == Integer; /*First node of a component is 0*/
<NodeType>      == EQUAL | CLOSED
<Extension>     == '=' | '+' | '*' /* '=' = not extended */
                  /* '+' = half_extended */
                  /* '*' = extended */

<LArcs>         == Arc LArcs |
                  epsilon
<Arc>           == '(' CompAccess ',' NodeAccess ','
                  CompAccess ',' NodeAccess ','
                  Weight ',' ThreshType ')'
<Weight>        == Integer; /*Between 0 and 100*/
<Thresh_Type>   == 'H' | 'L' /*H = High, L = Low*/

<GlobalContraint> == Thresh Thresh_Type
<Thresh>        == Integer /*Between 0 and 100*/

<LTransformations> == Transformation LTransformations |
                  epsilon
<Transformation> == '[' CompAccess ',' NodeAccess ','
                  NodeAccess ']' Operation
<Operation>     == 'J' | 'C' | 'S' /*J = Join, C = Cut*/
                  /*S = Shut*/

<Priority>       == Integer /*Between 0 and 100*/

```

```

<Integer>      == Number Integer
<Number>      == '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'|
                epsilon

```

Sémantique

<Graphe> :

La dimension (“dim”) d’un graphe est le nombre de composantes-images (nombre d’images concernées) du graphe.

<NodeType> :

Un nœud peut être de 2 types :

’EQUAL’ : le type quelconque.

’CLOSED’ : la chaîne associée au nœud est fermée.

<Extension> :

Chaque nœud d’un motif de règle est soumis à des contraintes sur le nombre d’arcs entrants et sortants. Ces contraintes sont décrites par l’attribut <Extension>. 3 valeurs sont autorisées :

’=’ : le nœud doit se présenter exactement dans la configuration décrite (nombre d’arcs entrants et sortants).

’*’ : le nœud doit supporter au moins les arcs spécifiés.

’+’ : comme précédemment, mais le nœud ne tolère que des arcs entrants ou que des arcs sortants.

<ThreshType> :

Pour chaque arc d’un motif, nous définissons un seuil. L’arc d’un graphe pourra convenir pour ce motif si le poids de cet arc est compatible avec la valeur et le type du seuil. Ce type peut être :

’L’ : le seuil est de type bas.

’H’ : le seuil est de type haut.

<Operation> :

Les trois opérations possibles dans un graphe sont :

’S’(hut) : fermeture de la chaîne associée au nœud (Shut). Dans ce cas, les 2 nœuds spécifiés pour la transformation sont identiques.

’J’(oin) : raccord des chaînes associées aux 2 nœuds (Join).

’C’(ut) : coupure de la chaîne associée au nœud spécifié (Cut).

Exemple

Voici la description d'une règle: O représente un sommet correspondant à une chaîne de nature quelconque. Les sommets d'une même composante (même image) sont représentés en colonne. Un arc reliant 2 sommets d'une même colonne est *un arc d'affinité*. Un arc reliant 2 sommets de 2 colonnes voisines est *un arc de liaison*.

```

/*****/
/* Rule 4.0 */
/*      Image-components */
/*      0 1 2 */
/* N 0      -0--0--0-    >>  -0--0--0- */
/* o          \ | / */
/* d           \ \| */
/* e 1          -0- */
/* s */
/* */
/*****/

```

graphe :

```

dim :
    3
component :
    nodes = [0, 0, EQUAL, *]
    arcs = (0, 0, 1, 0, 30, L)
           (0, 0, 1, 1, 30, L)
component :
    nodes = [1, 0, EQUAL, *]
           [1, 1, EQUAL, *]
    arcs = (1, 0, 1, 1, 40, L)
           (1, 1, 2, 0, 30, L)
           (1, 1, 2, 0, 30, L)
component :
    nodes = [2, 0, EQUAL, *]
    arcs =
;
globalContraint :
    40 L
;
transformations :
    [1, 0, 1] J
;
priority :
    1
;

```

Cette règle correspond au schéma suivant :

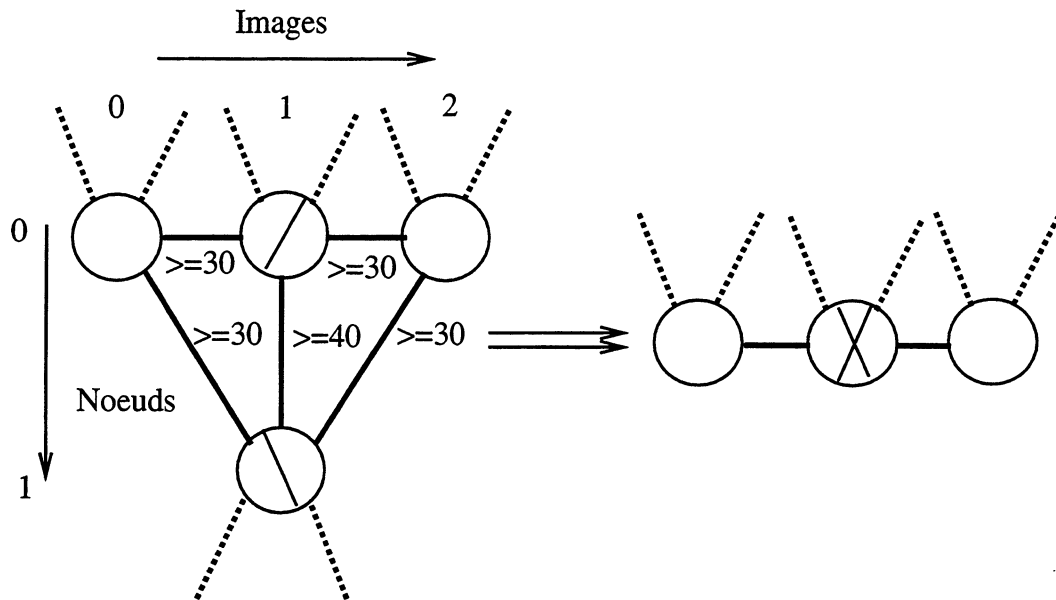


FIG. 1.32 - Représentation schématique de la règle

1.4.10 Choix des règles

Heuristique de construction des règles

Les règles choisies doivent permettre d'atteindre les objectifs spécifiés précédemment. A cet effet, elles doivent viser à assurer la fermeture des chaînes, à réduire le nombre d'embranchements et le nombre de chaînes de graphe de longueur inférieure ou égale à la longueur critique.

Nature des règles :

Le problème de la fermeture est un problème spécifique. Nous définissons une règle de fermeture reposant uniquement sur des critères intrinsèques : règle 1.0 (voir Annexe B) ainsi que deux autres règles destinées à assurer la propagation de la fermeture : règles 1.1 et 1.2.

Nous proposons une règle de jonction se basant uniquement sur le contexte-image (règle 2.0). Toutes les autres règles de jonction sont bâties sur le même modèle. A partir d'un embranchement, nous étudions toutes les configurations pouvant suivre cet embranchement. Nous ne retenons que les configurations mettant en jeu des chaînes de graphe de longueur limitée. Pour notre étude, nous avons fixé la longueur critique à 3. Pour une longueur donnée, nous n'avons que trois configurations possibles. Nous sommes donc certains que le nombre de règles nécessaires reste dans des limites raisonnables.

Définissons une règle de disjonction : cette règle commande la disjonction d'un sommet, quand ce sommet présente un embranchement à "gauche" et à "droite" et que de chaque côté, les configurations ne comportent que des chaînes de longueur supérieure à la longueur critique.

Valeurs de seuil :

Nous n'avons pas de méthode *a priori* nous permettant de déterminer ces seuils. Un des objectifs de notre étude a été justement d'estimer expérimentalement leurs valeurs.

Attribution des priorités :

Nous avons choisi de donner la priorité la plus forte aux règles mettant en jeu le nombre de composantes-images le plus faible. Cela repose sur le fait que les structures les plus petites dans un graphe correspondent aux structures les moins significatives dans la scène. Ce sont elles que nous éliminons en priorité.

La réflexion sur l'attribution des priorités n'est pourtant pas définitive. Il est par exemple envisageable de considérer que certaines règles sont, pour des raisons intrinsèques, d'une priorité élevée. Cela pourrait être le cas des règles dont le motif contient un cycle. Nous présentons en effet intuitivement que cette structure particulière leur donne une certaine force. Ce point précis mérite d'être approfondi.

Règles retenues

La liste des règles que nous avons utilisées est présentée en Annexe B.

1.4.11 Traitement d'une séquence par application des règles

L'algorithme de base est le suivant :

```

Pour chaque chaîne de la séquence non encore visitée
  Construire le graphe qu'elle engendre
  Rechercher tous les motifs de règles inclus dans le graphe engendré
  Déterminer les transformations spécifiées par les règles
  Exécuter les transformations
FinPour

```

En réalité, le problème est un peu plus complexe pour 2 raisons :

1. des conflits ou des concurrences peuvent se manifester entre certaines règles.
2. l'application d'une transformation peut faire apparaître de nouveaux motifs dans le graphe engendré, ou en faire disparaître.

Apparition et disparition de motifs

L'application de règles dans un graphe a pour conséquence une modification de sa structure. Bien souvent, nous constatons à l'issue d'une première série de transformations l'apparition de nouveaux motifs de règles, ou le maintien d'anciens motifs.

Il est donc nécessaire de procéder en plusieurs passes.

Malheureusement, pour des graphes complexes, nous ne pouvons prévoir le nombre de passes à exécuter. D'ailleurs, rien ne permet d'affirmer que l'algorithme va converger, et la preuve de cette convergence reste à étudier.

Pour la réalisation de notre travail, nous avons retenu la solution suivante : la recherche de motifs de règles s'arrête dès qu'une passe n'a donné lieu à aucune transformation dans le graphe, ou dès que le nombre de passes a atteint un seuil haut fixé par l'utilisateur.

D'autres critères sont envisageables. Nous pourrions imaginer par exemple une fonction définie sur les graphes que nous chercherions à optimiser. L'arrêt de l'algorithme serait commandé par la stagnation de la valeur de cette fonction, ou par l'atteinte d'un certain seuil.

Concurrence entre règles de transformations

Il est fréquent que dans un même graphe, nous puissions observer simultanément la présence de plusieurs motifs. Les motifs respectant les contraintes imposées par leurs règles respectives vont donner lieu à un certain nombre de transformations. A ce stade là, il arrive que des conflits apparaissent. En effet, 2 sommets d'un graphe peuvent être l'objet de 2 transformations contradictoires, fusion et non-fusion par exemple. Nous observons également le cas où 2 règles souhaitent fusionner un même sommet (sur le même pôle) avec 2 autres sommets distincts.

Ces situations sont relativement fréquentes. Les notions de priorité de règle et de robustesse de graphe nous permettent de traiter ce problème. Nous avons vu plus haut ce que nous entendons par priorité de règle.

Définition : La **robustesse** est une information globale sur un graphe. Elle rend compte de sa "force de cohésion". L'estimation de la robustesse consiste à calculer la valeur moyenne des arcs du graphe.

L'algorithme est simple. En cas de conflit entre 2 règles R_1 et R_2 , nous procédons de la façon suivante :

```

Si priorité( $R_1$ ) > priorité( $R_2$ )
    Retenir transformation  $R_1$ 
Sinon
    Si priorité( $R_2$ ) > priorité( $R_1$ )
        Retenir transformation  $R_2$ 
    Sinon
        Si robustesse(motif( $R_1$ ))  $\geq$  robustesse(motif( $R_2$ ))
            Retenir transformation  $R_1$ 
        Sinon
            Retenir transformation  $R_2$ 
    FinSi
FinSi
FinSi

```

Algorithme de traitement d'une séquence

Compte tenu des observations précédentes, l'algorithme est le suivant :

```

Pour chaque chaîne de la séquence non encore visitée
  Construire le graphe qu'elle engendre;
  nombre_passes ← 0; transforme ← vrai
  Tantque ((transforme) et (nombre_passes < maximum_passes))
    Rechercher tous les motifs de règles inclus dans le graphe
    Pour chaque motif repéré
      Pour chaque transformation de la règle
        Retenir la transformation si elle n'est en concurrence avec
        aucune autre, sinon retenir la transformation élue
        par l'algorithme ci-dessus.
      FinPour
    FinPour
    Si Des transformations sont demandées
      Exécuter les transformations
    Sinon
      Transforme ← faux
    FinSi
    Nombre_passes ← Nombre_passes + 1
  FinTantque
FinPour

```

1.4.12 Inclusion de graphes

L'application des règles dans un graphe est basée sur la mise en évidence de sous-graphes. Ce problème est celui de l'isomorphisme de graphes.

Pour le résoudre, nous avons choisi de l'aborder sous l'angle de la satisfaction de contraintes. Soient un premier graphe G_1 et un deuxième graphe G_2 dont on cherche à trouver toutes les occurrences dans G_1 (pour l'inclusion des graphes).

D'après Mackworth [Mac 77], le problème peut être décrit de la façon suivante :

$A = \{b, c, \dots\}$ est l'ensemble des nœuds de G_1 . $|A| = a$.
 $N = \{i, j, \dots\}$ est l'ensemble des nœuds de G_2 . $|N| = n$.
 $E = \{(i, j) / (i, j) \text{ appartient à } N \times N\}$ est l'ensemble des arcs de G_2 . $|E| = n$.
 $A_i = \{b \mid b \text{ est dans } A \text{ et } (i, b) \text{ est admissible}\}$.
 R_1 est une relation unaire et (i, b) est admissible si $R_1(i, b)$.
 R_2 est une relation binaire et $(i, b) - (j, c)$ est admissible si $R_2(i, b, j, c)$.

Nous souhaitons trouver tous les n-uplets de A^n vérifiant les relations R_1 et R_2 . La relation R_1 va nous permettre d'exprimer la consistance des nœuds, la relation R_2 la consistance des arcs.

Consistance des nœuds (R_1)

La première étape va consister à déterminer, pour chaque nœud i de G_2 l'ensemble des nœuds b de G_1 qui ne sont pas admissibles en i (que nous ne saurions superposer avec i).

Nous introduisons la notion de motif primaire.

Définition : Un **motif primaire** est composé d'un sommet pris dans un graphe, et de ses arcs adjacents. Nous avons défini pour nos besoins 6 motifs primaires.

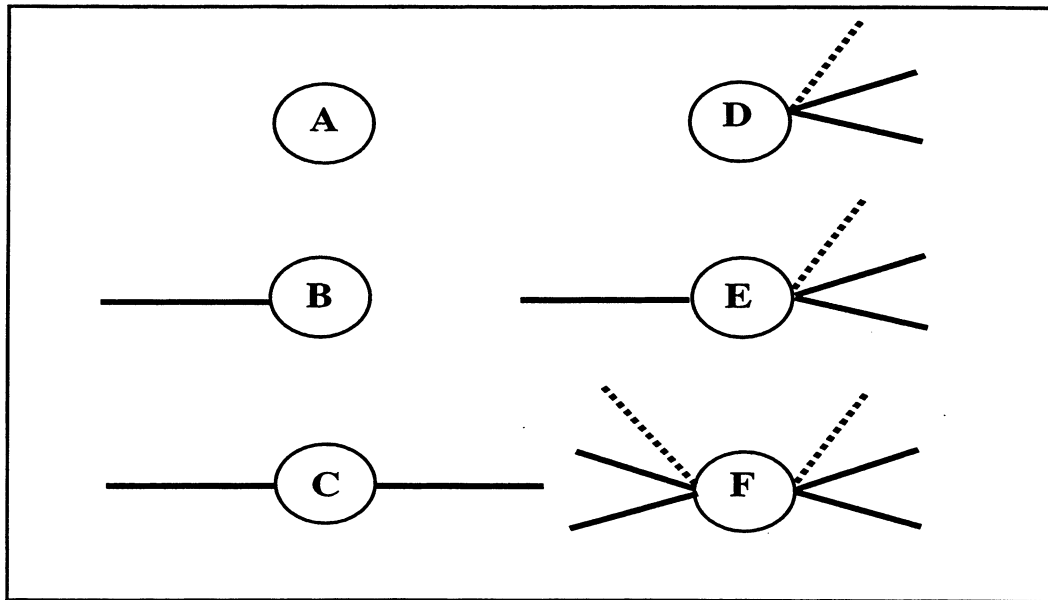


FIG. 1.33 - Liste des motifs primaires

Pour vérifier la consistance des nœuds, nous appliquons le principe suivant :

Pour chaque nœud b de G_1 , nous identifions le type de b par rapport à ces 6 motifs primaires. Ce type est comparé à celui des nœuds de G_2 . Comme indiqué dans la grammaire de description des règles, pour chaque sommet du motif sont spécifiés les arcs adjacents incontournables (attribut Extension). Un attribut permet de préciser si ces sommets peuvent accepter des arcs supplémentaires, d'un côté et de l'autre. Nous devons tenir compte de cet attribut pour la comparaison des types des nœuds. A cet effet, nous avons défini 2 relations d'ordre sur les motifs primaires : une relation d'ordre partiel ' \succ ', et une relation d'ordre total ' \gg '.

Soient X et Y deux motifs primaires.

1. Nous avons $Y \gg X$ si et seulement si de chaque côté de X , le nombre d'arcs adjacents à X est inférieur ou égal au nombre d'arcs adjacents à Y .
2. Nous avons $Y \succ X$ si et seulement si X et Y ont au moins un côté identique, et si de l'autre côté, le nombre d'arcs adjacents à X est inférieur au nombre d'arcs adjacents à Y .

Nous avons donc les ordres suivants :

$$\begin{array}{ll} A \gg A & B \gg B \\ B \gg A & C \gg B \\ C \gg A & D \gg B \\ D \gg A & E \gg B \\ E \gg A & F \gg B \\ F \gg A & \end{array}$$

$$\begin{array}{ll} C \gg C & D \gg D \\ D \gg C & E \gg D \\ E \gg C & F \gg D \\ F \gg C & \end{array}$$

$$\begin{array}{ll} E \gg E & F \gg F \\ F \gg E & \end{array}$$

$$\begin{array}{l} B \succ A \\ D \succ A \\ D \succ B \\ E \succ C \\ F \succ E \end{array}$$

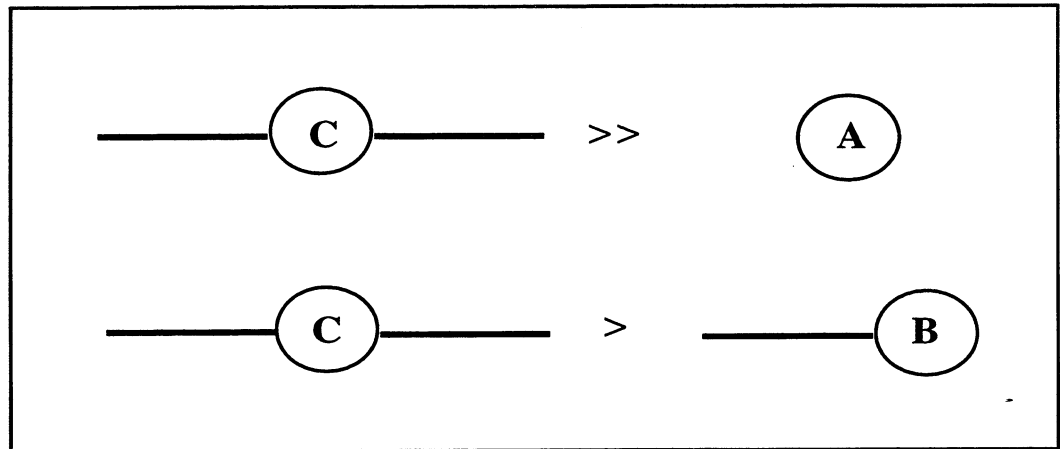


FIG. 1.34 - Ordre des motifs primaires

Ces deux relations d'ordre (voir Figure 1.34) nous permettent de régler tous les cas de comparaison de nœuds en fonction de leurs types.

Consistance des arcs (R_2)

Les candidats en chaque nœud de G_2 étant désormais précisés, nous nous intéressons maintenant aux associations $(i, b) - (j, c)$ admissibles, pour chaque arc (i, j) de G_2 .

Notre problème est de savoir si c est possible en j lorsque b est en i . Pour cela, deux conditions doivent être respectées :

Consistance des composantes : Si i et j sont dans la même composante-image (resp. dans 2 composantes-images successives) de G_2 , b et c doivent également être dans la même composante-image (resp. dans 2 composantes-images successives) de G_1 .

Consistance des arcs : b et c doivent être reliés dans G_1 par un arc. Le poids de cet arc doit être compatible avec la valeur et le type du seuil de l'arc (i, j) .

Afin de vérifier la consistance des arcs de G_2 dans G_1 , nous utilisons l'algorithme $AC - 4$ [Moh 86]. Cet algorithme proposé par Roger Mohr et Thomas C. Henderson est optimal en temps. En revanche, il est coûteux en mémoire, et ne peut s'appliquer qu'à des graphes de taille limitée.

Production des motifs

A l'issue des 2 étapes de consistance des nœuds et de consistance des arcs, nous connaissons, pour chaque nœud i de G_2 , la liste des nœuds de G_1 possibles en i , et la liste des incompatibilités.

Grâce à ces informations, à la description du motif et à l'orientation arbitraire des graphes, la production des motifs se fait sans problèmes particuliers.

1.4.13 Affinités entre chaînes de contours

L'affinité entre 2 extrémités de chaînes de contours est un indice variant de 0 à 100 destiné à juger de leur aptitude au raccordement.

Comme nous l'avons précisé, 4 critères sont pris en compte pour le calcul de l'affinité : la proximité, la continuité de tangence, la continuité de courbure, la continuité d'orientation du gradient.

Chacun de ces critères apporte, par une pondération, sa contribution pour le calcul de l'affinité finale.

La proximité se mesure en pixels à partir des coordonnées des extrémités. Nous avons choisi une fonction linéaire pour le calcul de cette affinité partielle notée A_p :

$$A_p = \frac{2 \times 100}{d + 1}$$

où d est la distance entre les deux extrémités a, b (voir Figure 1.35). L'affinité est donc de 100 pour une distance de 1 pixel, de 50 pour 3 pixels, de 20 pour 9 pixels.

Pour l'estimation de la tangente et de la courbure, les chaînes de contours sont approximées par des fonctions B-splines cubiques régularisées.

L'application qui nous intéresse dans cette étude est la reconstruction de surfaces, et nous avons besoin de pouvoir interpoler entre les pixels et de dériver jusqu'au deuxième ordre. L'approximation fonctionnelle des chaînes de points de contour a suscité notre intérêt. Puisqu'elle se propose de représenter le contour par une ou plusieurs fonctions "lisses" (au moins C^1). Elle permet d'obtenir une représentation continue du

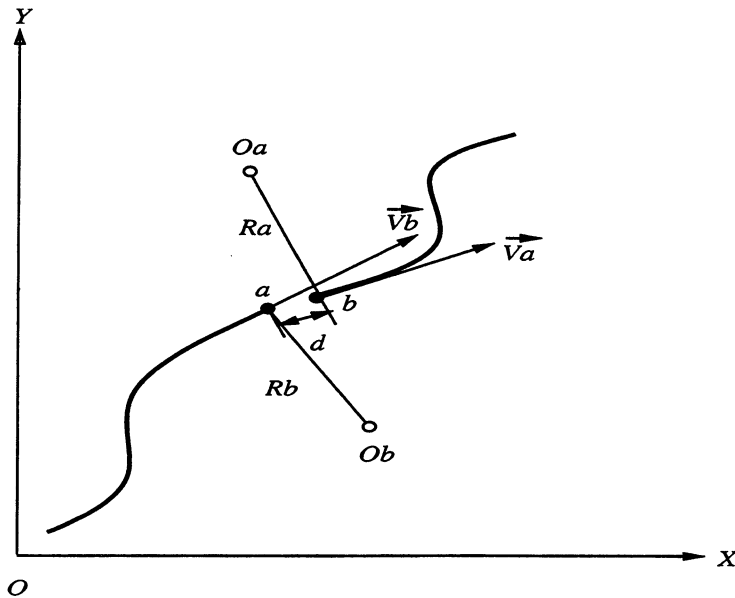


FIG. 1.35 - Critères de raccordement de chaînes de contours

signal [Arb 91a]. Le choix de la fonctionnelle dans notre recherche s'est porté sur les B-splines cubiques régularisées. L'avantage des B-splines est théoriquement bien étudié [Bar 87, Su 89]. Elles ont de nombreuses propriétés intéressantes. Dans notre application, il s'agit de reconstituer correctement une fonction dans le cas où il n'y pas de données dans certaines zones. L'interpolation le long des chaînes de contours est effectuée en utilisant cet outil. En pratique, nous complétons les trous entre deux contours (voir Figure 1.24).

Après l'approximation de B-splines de contours, chaque chaîne peut alors être décrite par un paramètre t : $(x(t), y(t))$. Nous pouvons ainsi caractériser toutes les propriétés intrinsèques et géométriques de la courbe d'une chaîne de contour par la méthode différentielle classique (voir Annexe A.1).

Ainsi, la forme extérieure d'une chaîne nous fournira parfois des indices utiles pour décider de la pertinence de telle ou telle transformation.

Le calcul des affinités se fait à partir des différences de courbures et de tangences.

Soient R_a et R_b les deux rayons de courbure aux deux extrémités a , b , et θ_a , θ_b les deux angles que font les tangentes \vec{V}_a , \vec{V}_b à la courbe avec l'axe X . Les affinités de courbure A_c et de tangence A_t sont calculées à partir des 2 paramètres suivants :

$$\begin{cases} A_c \geq \left| \frac{1}{R_a} - \frac{1}{R_b} \right| \\ A_t \geq |\theta_a - \theta_b| \end{cases} \quad (1.25)$$

Ces deux valeurs A_c , A_t ont été ajustées en fonction des expérimentations.

Le quatrième critère est celui de la continuité d'orientation du gradient.

En chaque point d'une chaîne de contour, le gradient est un vecteur perpendiculaire à la chaîne, orienté de la région sombre vers la région claire. Il arrive parfois que le long d'une même chaîne de contour, le gradient change d'orientation (voir Figure 1.36).

Dans un tel cas, nous faisons le choix de couper cette chaîne au niveau du changement d'orientation. Par conséquent, pour chaque chaîne, l'orientation du gradient devient continue.

Pour le problème du raccordement, l'affinité de gradient A_g est calculée à partir de l'angle formé par les vecteurs gradients aux 2 extrémités concernées.

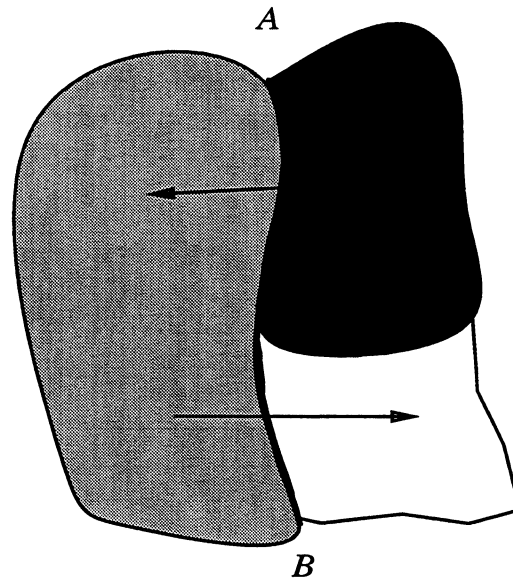


FIG. 1.36 - Un cas de direction du gradient variant sur la même chaîne

1.4.14 Résultats expérimentaux

Dans ce paragraphe, nous exposons les résultats de correction des anomalies des chaînes de contours d'une séquence d'images de la tasse.

La Figure 1.37 présente une séquence de six images initiales de contours que nous voulons traiter.

La Figure 1.38 présente la séquence des six images finales de contours après application de la procédure de raccordement précédente.

Le temps de traitement de la séquence de dix images 512×512 est inférieur à 10 secondes sur une station de travail (Sun sparc2). La plupart de corrections des anomalies sont correctes (80%).

La Figure 1.39 montre la deuxième image initiale de contours de la séquence.

La Figure 1.40 montre la deuxième image finale de contours après la procédure de raccordement.

La Figure 1.41 montre la sixième image initiale de contours.

La Figure 1.42 montre la sixième image finale de contours après la procédure de raccordement.

Si nous comparons les contours d'images entre la Figure 1.37 et la Figure 1.38, nous nous apercevons que certains petits contours sont éliminés, et que quelques contours sont raccordés. Le tableau 1.1 décrit les modifications des chaînes de contours de la séquence.

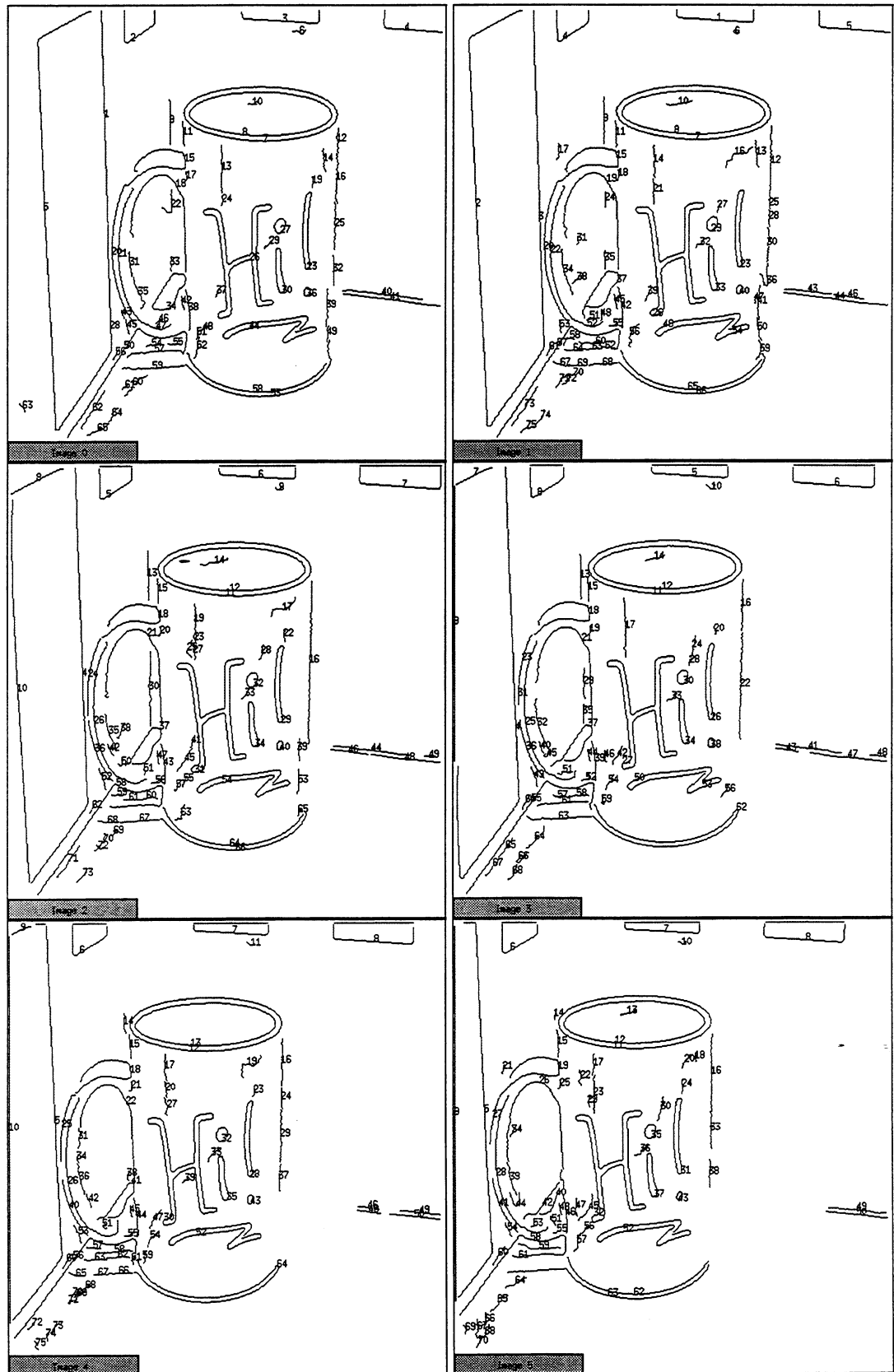


FIG. 1.37 - Une séquence de 6 images initiales de contours

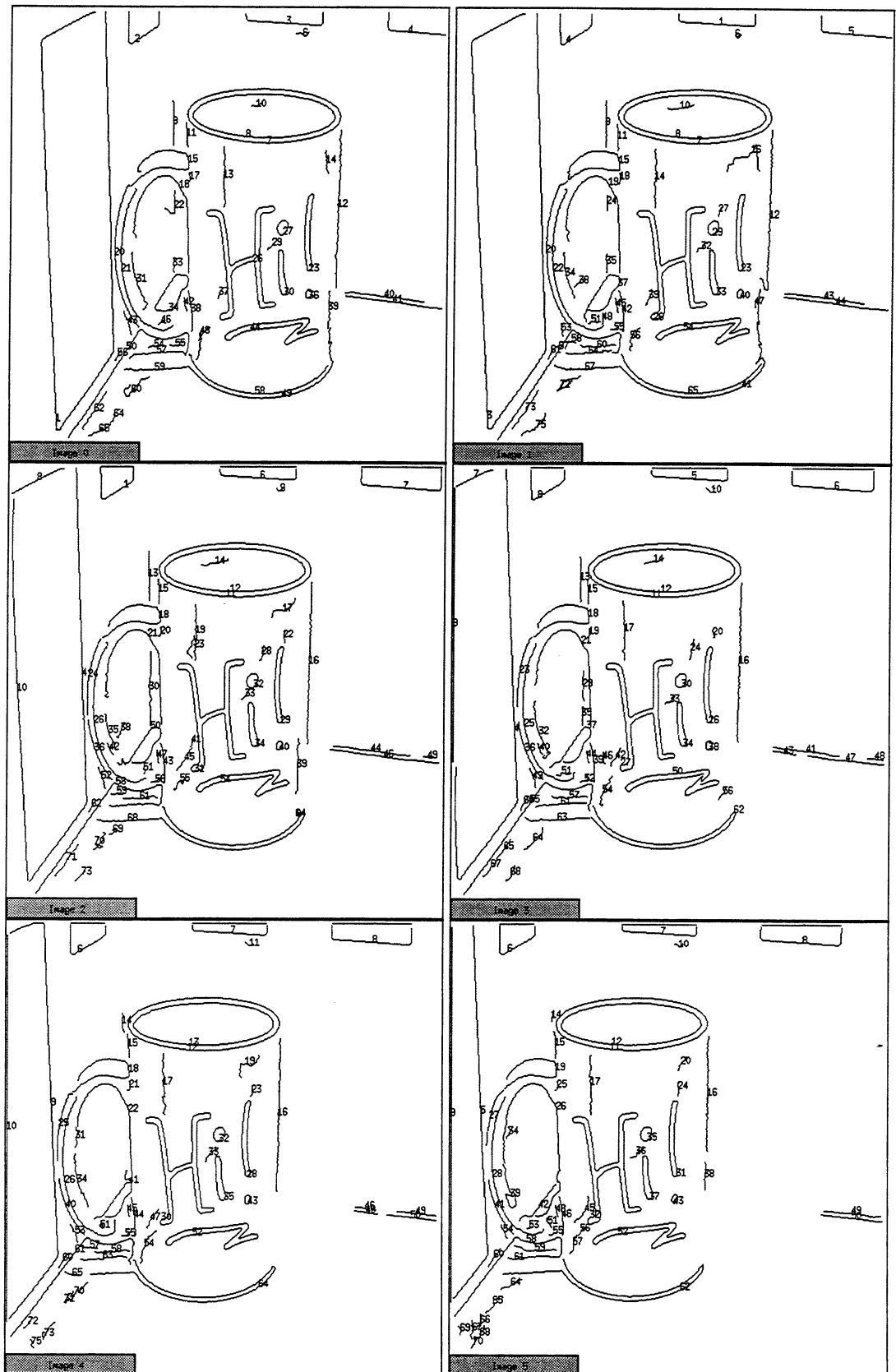


FIG. 1.38 - Une séquence de 6 images finales de contours après le traitement

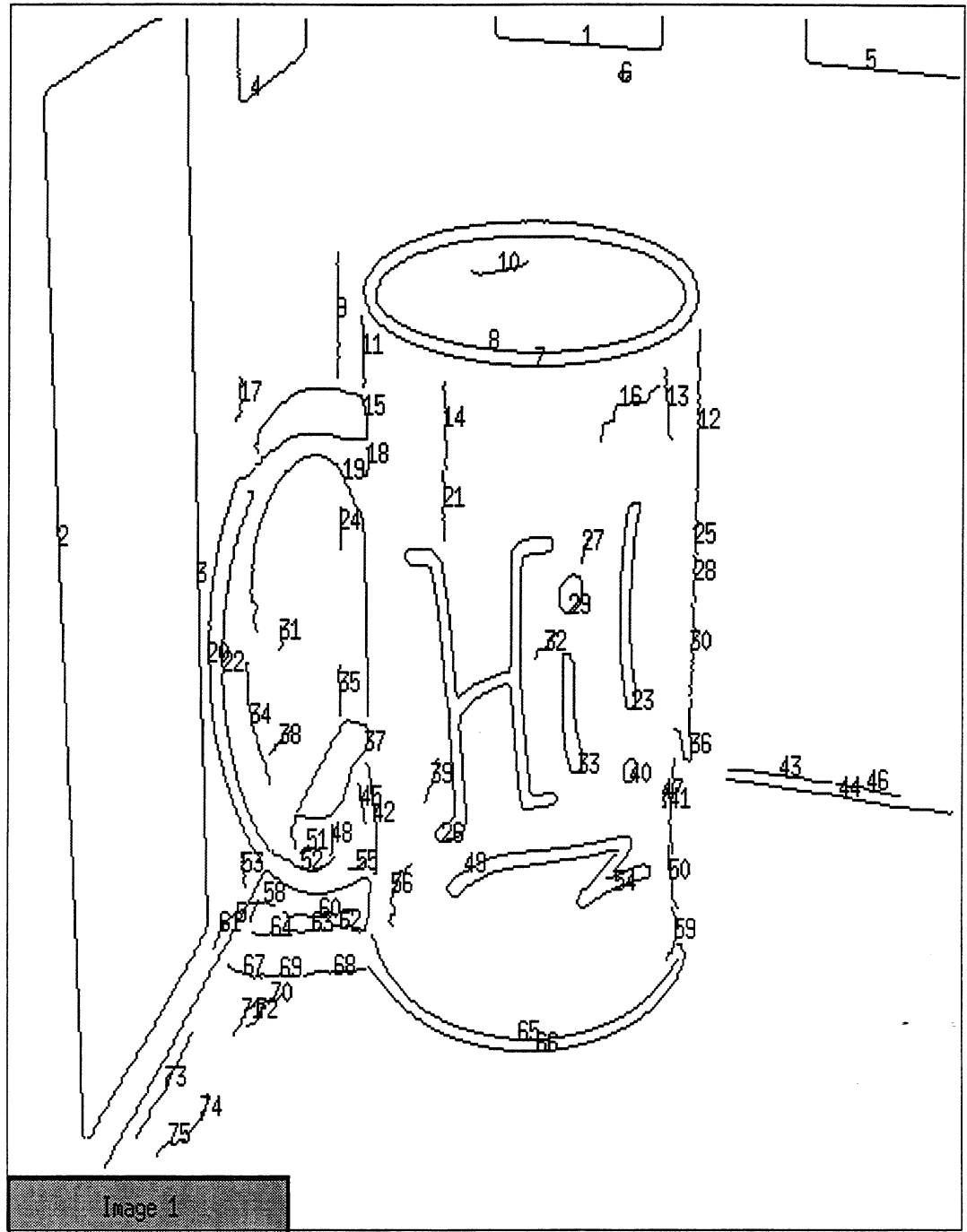


FIG. 1.39 - Deuxième image initiale de contours

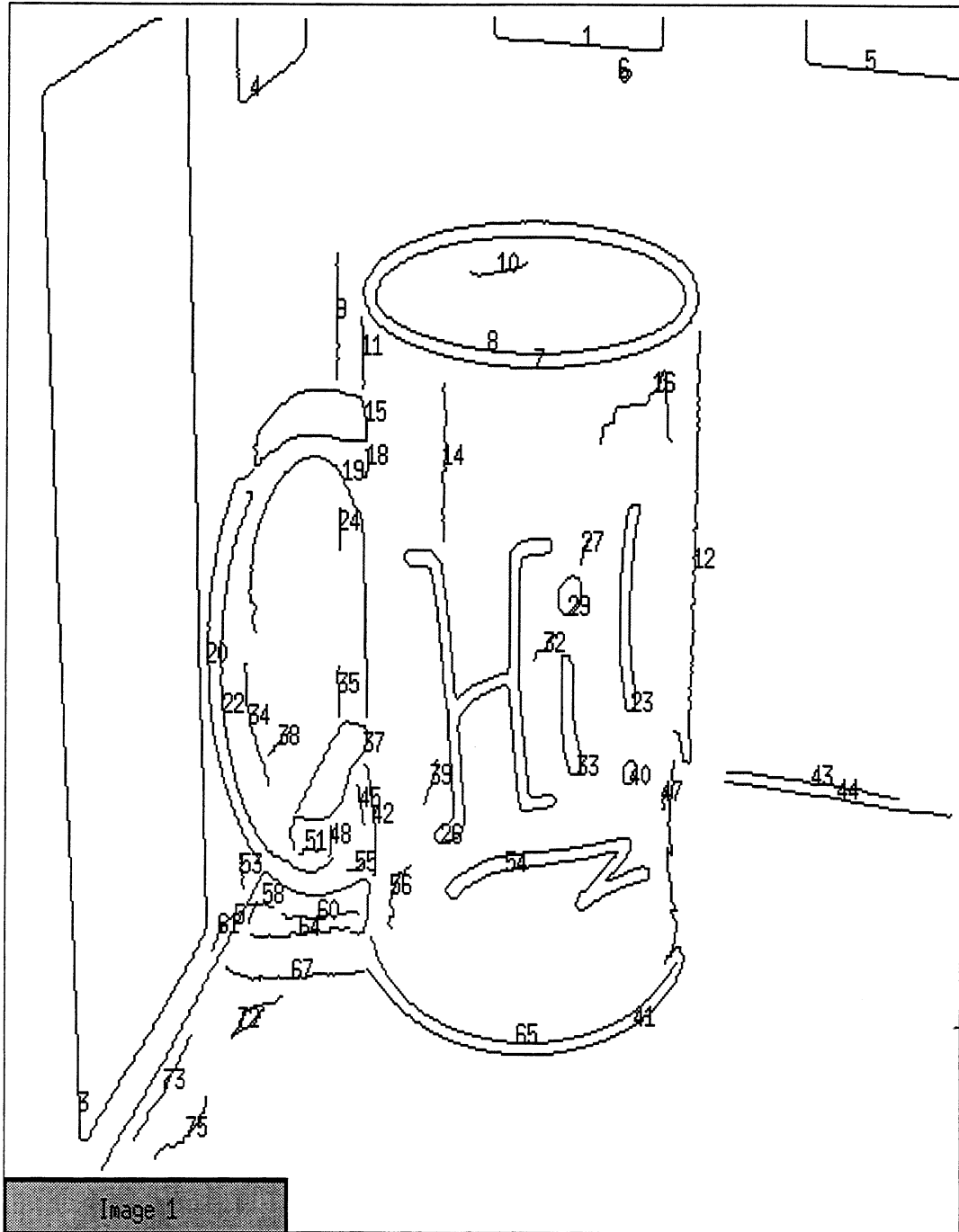


FIG. 1.40 - Deuxième image finale de contours après le traitement

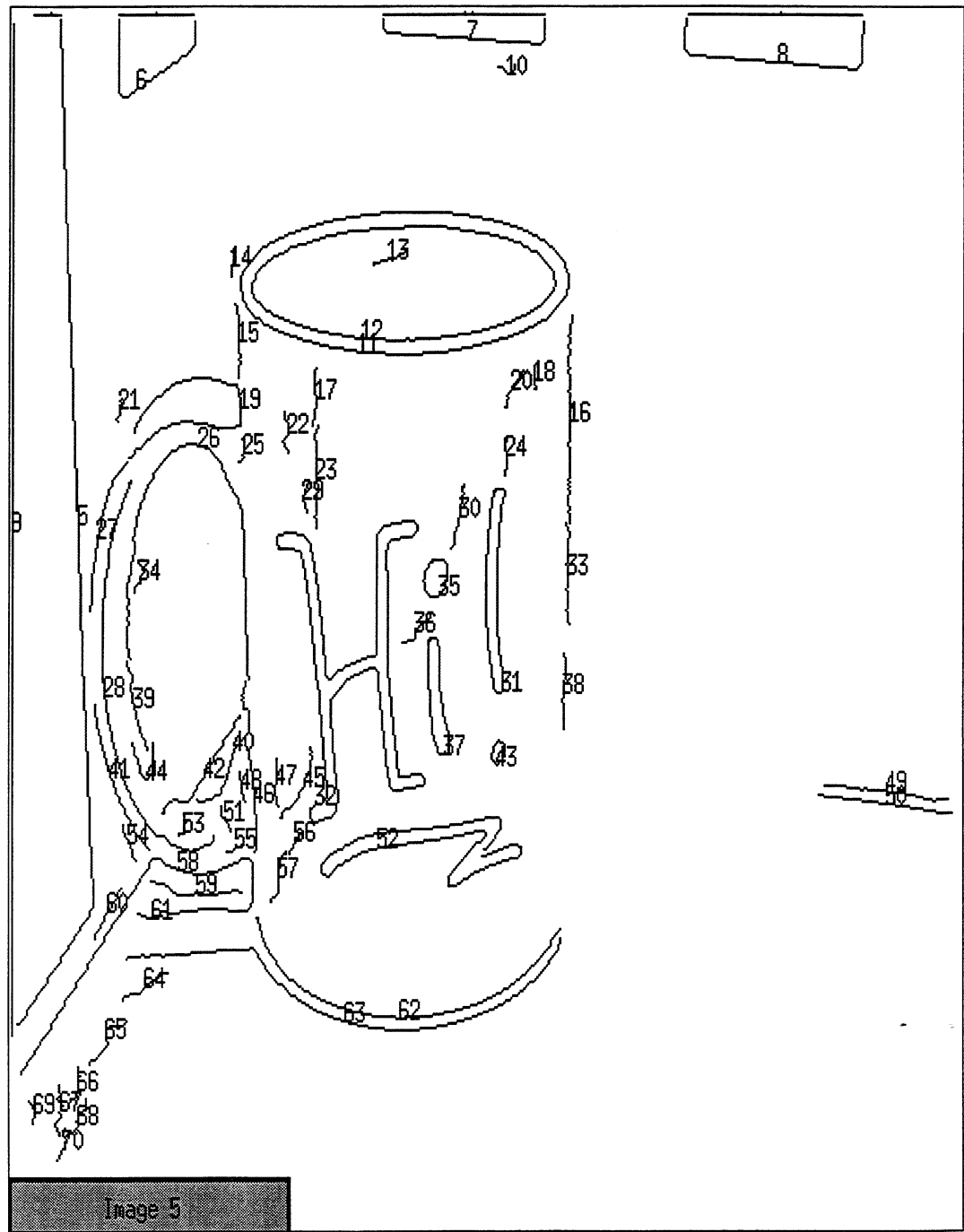


FIG. 1.41 - Sixième image initiale de contours

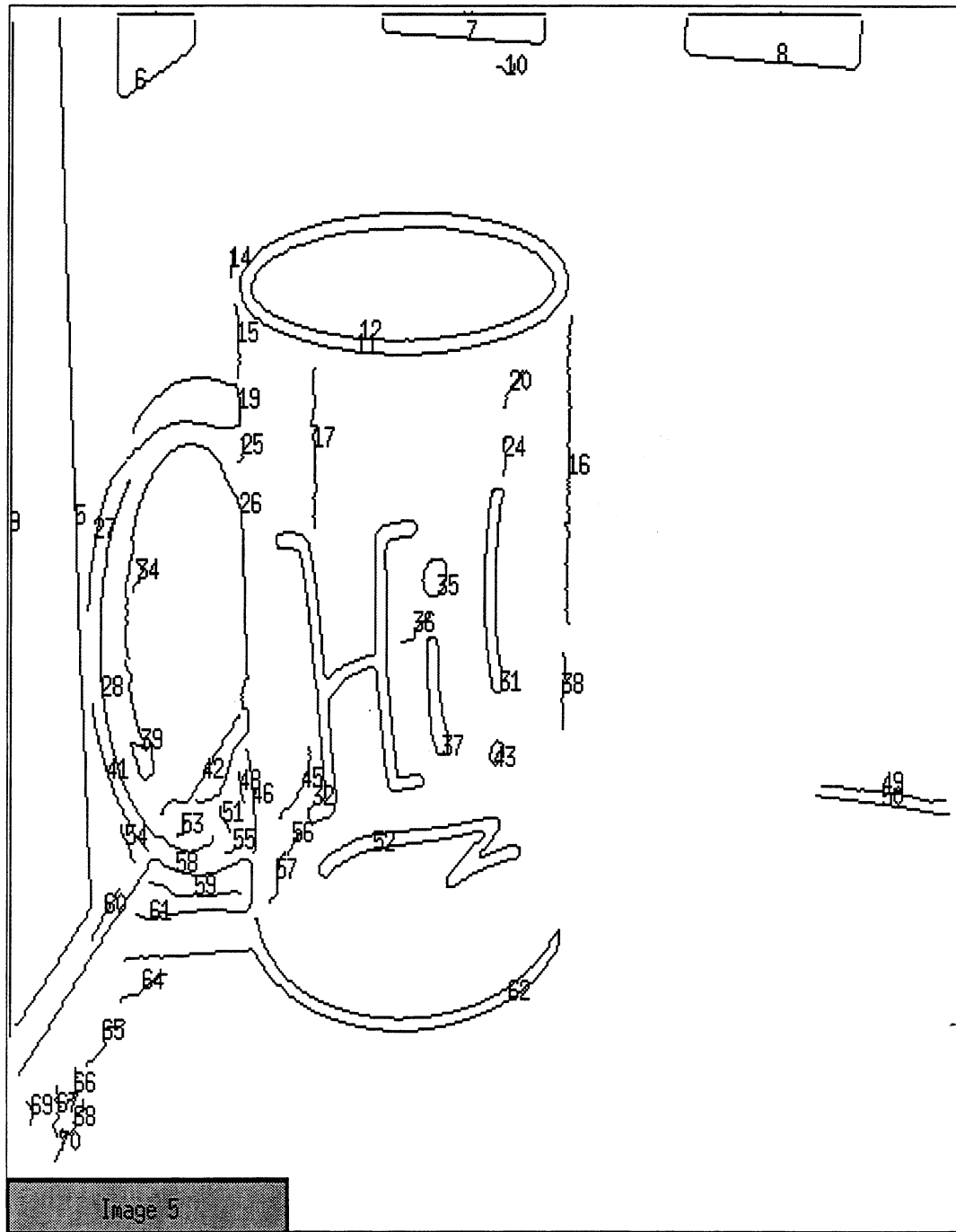


FIG. 1.42 - Sixième image finale de contours après le traitement

<i>Numéro d'image</i>	<i>Numéro des chaînes éliminées</i>	<i>Numéro des chaînes regroupées</i>
1	19, 63, ...	12, 16, 25, 32 vers 12
		21, 47 vers 21
		39, 49 vers 39, ...
2	17, ...	12, 25, 28, 30, 36 vers 12
		14, 21 vers 14
		43, 46 vers 43, ...
3	65, ...	39, 53 vers 39
		46, 48 vers 46
		64, 66 vers 64, ...
4	66, ...	16, 22 vers 16
		23, 31 vers 23
		54, 59 vers 54, ...
5	39, ...	16, 24, 29, 37 vers 16
		34, 36, 42 vers 34
		54, 59 vers 54, ...
6	13, 18, 21, ...	16, 33 vers 16
		17, 23 vers 17
		62, 63 vers 62, ...

TAB. 1.1 - Table des modifications des chaînes de contours

Nous avons aussi remarqué que la chaîne 38 de la sixième image n'est pas raccordée avec les chaînes 33 et 16 car la distance entre les deux extrémités des chaînes 38 et 33 est trop grande par rapport à notre seuil. Mais avec le critère de continuité de tangence, on aurait le raccordement.

D'autre part, bien que peu visible sur la Figure 1.38, certaines chaînes de contours ont été fermées. Par exemple, la chaîne de contours 12 de la quatrième image, la chaîne de contours 52 de la cinquième image, ...

1.5 Perspectives et discussion

Notre objectif est de mener une recherche sur les possibilités de correction des anomalies dans une séquence d'images de chaînes de contours. Cette étude permet de faciliter l'étape de reconstruction de surfaces. Notre approche consiste à rechercher un état du système globalement cohérent à partir de l'ensemble des informations de mises en correspondance point à point d'une séquence.

Les résultats sur les séquences tests nous paraissent très encourageants.

Il faut bien avoir à l'esprit que, comme pour toutes les étapes du processus de vision, l'étape de correction des anomalies ne saurait être exempte d'erreurs ! Nous ne sommes donc pas à la recherche d'une solution parfaite, mais plutôt d'un outil nous assurant un degré minimum de fiabilité. Toutefois, en l'état actuel des choses, nous ne saurions apporter une conclusion définitive. Nous disposons d'un outil d'étude performant. La phase d'expérimentation doit maintenant commencer.

Les grands axes à suivre sont désormais les suivants :

- caractérisation précise des séquences à traiter, des motifs apparaissant le plus souvent dans les graphes spatio-temporels. Cette étape doit permettre d'affiner le choix des règles et leur collaboration.
- réflexion plus approfondie sur le problème de la coupure des chaînes de contours.
- réflexion sur la part respective de chaque critère d'affinité dans le calcul de l'affinité globale entre 2 extrémités de chaînes de contours.
- étude de l'incidence des priorités des règles sur le traitement des séquences.
- preuve de la convergence de l'algorithme d'applications des règles.
- analyse qualitative et quantitative des résultats obtenus pour plusieurs types de séquences d'images.

En complément de ces six points, nous proposons également un sujet de réflexion venant terminer notre étude.

Nous avons expliqué pourquoi certaines informations du suivi de points sont peu fiables, et comment nous les éliminons sur la base du poids des arcs de liaison. Il arrive malheureusement que le poids de ces arcs incohérents soit fort, et dans ce cas, notre méthode est sans effet.

L'observation de ces anomalies semble indiquer qu'elles se propagent rarement le long de la séquence. Nous pensons que ce problème pourrait être convenablement traité par l'utilisation de nouvelles règles de transformations de graphes, dont l'objet ne serait plus d'opérer sur les sommets, mais plutôt d'opérer sur les arcs.

1.6 Conclusion

Dans ce chapitre, nous avons présenté un algorithme pyramidal de suivi de contours courbes basé sur les points de contours en utilisant un outil puissant de corrélation. Le résultat des mises en correspondance point à point est tout à fait satisfaisant.

Par notre expérimentation, nous trouvons :

d'une part, qu'il est préférable d'effectuer la corrélation simple ou normalisée sur les images de Laplacien que sur les images de gradient ou les images de niveau de gris. d'autre part que la meilleure pyramide est la Gaussienne.

Le meilleur compromis prenant en compte de traitement du temps et qualité du résultat, est d'effectuer la corrélation simple sur des images laplaciennes de la pyramide gaussienne.

La technique de corrélation est vraiment très simple. Notamment, l'approche pyramidale permet une réduction importante des temps de calcul par rapport à une approche multi-résolution classique où la résolution est définie par la taille du filtre d'extraction des contours.

En revanche, notre algorithme de suivi de contours est basé sur un appariement global. Son temps de calcul est assez coûteux. Nous avons vu que, si nous appliquons la contrainte épipolaire, nous faisons simplement la corrélation sur la droite de ligne épipolaire. D'une part, cela permet de diminuer la zone de recherche de corrélation, d'autre part, le résultat de la mise en correspondance point à point est plus précis que pour la corrélation bidimensionnelle.

Notre approche originale de correction des anomalies de contours d'une séquence donne des résultats très encourageants. Plus particulièrement, cette approche montre que la technique de manipulation de graphes spatio-temporels est bien adaptée pour ce problème. Mais nous n'avons pas encore profondément abordé le problème de coupure de contours à cause de la difficulté de définition des critères.

Chapitre 2

Reconstruction des surfaces B-splines

La reconstruction de surfaces 3D d'un objet non polyédrique est un problème difficile de la vision par ordinateur. Dans ce chapitre, une nouvelle approche est présentée pour la reconstruction des surfaces tridimensionnelles à partir de l'observation du mouvement des contours occultants dans une séquence d'images calibrées. La représentation de la surface de cet objet est donnée par les surfaces B-splines uniformes et bicubiques. Nous ramenons le problème de la reconstruction des surfaces à la résolution d'un système d'équations non linéaires qui détermine leurs points de contrôle. Deux méthodes numériques de résolution du problème sont utilisées : Levenberg-Marquardt et Quasi-Newton. L'approche proposée s'applique dans le cas d'un mouvement quelconque d'une caméra calibrée, avec des surfaces C^2 , par morceaux au moins. Pour éviter le calibrage classique nécessitant une mire, nous utilisons des points stables de la scène pour auto-calibrre la caméra. Des résultats expérimentaux sur des données synthétiques et réelles sont enfin présentés.

2.1 Introduction

Les problèmes de représentation et de reconstruction des surfaces tridimensionnelles d'un objet en vision par ordinateur ont beaucoup été étudiés ces dernières années. La plupart du temps ce sont des méthodes utilisant en entrée des informations tridimensionnelles fournies par des capteurs télémétriques [Gri 82, Ter 88, San 90, Sze 91, Tau 92, Pen 92].

Une bonne représentation d'une surface 3D implique souvent une reconstruction facile. Du point de vue de la géométrie différentielle classique, il existe trois sortes de représentations pour une surface : explicite, implicite et paramétrée [Car 76]. Plusieurs méthodes de reconstruction de surfaces utilisant respectivement les trois représentations précédentes sont introduites dans [Bol 91]. Dans cette étude, nous utilisons la forme des surfaces B-splines uniformes et bicubiques pour représenter la surface d'un objet 3D à reconstruire. Les surfaces B-splines sont des représentations classiques pour les objets 3D dans l'espace [Bar 87, Su 89]. Elles sont bien étudiées en CAO¹ et dans le domaine des images de synthèse, car elles ont des propriétés très intéressantes : une surface B-spline est en effet une combinaison linéaire des produits tensoriels des fonctions de base des courbes B-splines. Un morceau d'une surface B-spline bicubique est déterminé complètement par 16 points de contrôle.

Notre objectif est de reconstruire la surface 3D d'un objet à partir de l'observation du mouvement des contours occultants dans une séquence d'images. Nous supposons que la surface de cet objet n'est ni plane, ni concave, mais C^2 , par morceaux au moins. Ce problème de la reconstruction des surfaces avait été abordé par Giblin et Weiss [Gib 87]. Ils se sont ramenés au problème de reconstruction d'une enveloppe de ses plans tangents en résolvant un ensemble d'équations. Les propriétés des contours occultants sont théoriquement bien étudiées par Koenderink [Koe 84, Koe 90]. Il a établi les relations mathématiques entre la courbure radiale de la surface, la courbure de la projection et la courbure gaussienne de la surface au point donné. Par ce genre d'informations sur les contours occultants, nous pouvons inférer des résultats qualitatifs et quantitatifs sur la surface tridimensionnelle observée : les convexités ou les concavités du contour correspondent à des morceaux de surface. Les travaux de Brady [Bra 85], Lim [Lim 88], Kutulakos [Kut 91], et Seales [Sea 92] nous ont montré également que les contours occultants sont de riches sources d'informations sur la surface. L'approche de Vaillant [Vai 92] est particulièrement intéressante. Il a originalement développé une méthode de détection des contours d'occultation en observant trois images d'un objet, puis il a généralisé certains résultats de Koenderink dans le cadre de la projection perspective. L'étude de Cipolla [Cip 89] se situe aussi dans ce cadre. Il a développé une méthode d'estimation de paramètres de surface (courbures gaussiennes et moyennes) par la géométrie différentielle en prenant un modèle de caméra sphérique. C'est en fait la solution du problème initial posé par Giblin et Weiss.

Ici nous nous proposons de reconstruire directement un élément de surface par l'estimation de ses paramètres au lieu de passer par la reconstruction des positions dans l'espace. La surface de cet objet est paramétrée par des surfaces B-splines uniformes et bicubiques. Notre but final sera de construire une surface complète en fusionnant un ensemble des surfaces B-splines les unes avec les autres. Notre recherche reprend différemment les travaux d'Arbogast [Arb 91a, Arb 91b], développés au LIFIA, en montrant que la reconnaissance de la surface spatio-temporelle est une étape intermédiaire nécessaire pour reconstruire la surface d'un objet 3D. L'étude de Bolles [Bol 87] montre que nous pouvons déterminer l'espace libre 3D par la reconnaissance de la surface spatio-temporelle.

Ce chapitre est organisé de la manière suivante :

Dans la première partie, nous introduisons quelques définitions et notions essen-

¹ Conception Assistée par Ordinateur.

tielles de notre approche.

Dans la deuxième partie, nous posons mathématiquement le modèle de la reconstruction des surfaces B-splines en utilisant une caméra en mouvement calibrée. Tout d'abord, nous présentons une méthode de l'autocalibration de la caméra en mouvement en utilisant les points de référence de la scène. Ensuite, nous ramenons le problème de la reconstruction des surfaces à un problème de résolution d'un système d'équations non linéaires déterminant leurs points de contrôle.

Dans la troisième partie, nous montrons les résultats de la reconstruction des surfaces B-splines avec des données synthétiques et réelles. Nous discutons des stratégies d'amélioration des résultats et de notre recherche future.

2.2 Définitions et notations préliminaires

Dans ce paragraphe, nous allons tout d'abord présenter quelques définitions et notions utilisées dans notre recherche.

2.2.1 Courbes et surfaces B-splines uniformes de degré d

Une étude détaillée des présentations générales de *B-splines* est faite par Bartels [Bar 87] et Su [Su 89]. Dans ce paragraphe, nous nous intéressons aux courbes et surfaces *B-splines* uniformes de degré d .

Courbes B-splines uniformes de degré d

Nous supposons qu'une courbe *B-spline* est donnée par deux composantes x et y du plan correspondant à un paramètre t de $[a, b]$. Dans la suite nous ne nous occuperons que d'une des composantes, par exemple x .

Soit $a = u_0 < u_1 \dots < u_n = b$ une subdivision régulière de l'intervalle $[a, b]$, de pas $h = u_{i+1} - u_i = (b - a)/n$. Notons S l'espace des fonctions définies sur $[a, b]$ qui sont formées de polynômes de degré d sur chaque intervalle $[u_i, u_{i+1}]$, $i = 0, \dots, n - 1$, et qui sont de classe C^{d-1} . Nous voyons facilement que S est de dimension $n + d$. Dans notre étude, nous envisageons seulement la valeur $d = 3$ [Zha 93a, Zha 93b, Zha 93c].

Soit B_j la *B-spline* de degré d dont le support est $[u_j, u_{j+d+1}]$, $j = -d, \dots, n - 1$. Nous introduisons ici les nœuds supplémentaires u_{-d}, \dots, u_{-1} , et u_{n+1}, \dots, u_{n+d} . Toute fonction Q appartenant à S peut être mise sous la forme :

$$Q(t) = \sum_{j=-d}^{n-1} \alpha_j B_j(t) \quad (2.26)$$

Les coefficients α_j sont appelés *points de contrôle*. Les *fonctions de base de B-splines* B_j sont définies [Lau 87] comme suit :

Si p et q ($p + q = 1$) représentent la localisation de t dans $[u_i, u_{i+1}]$, $p = (t - u_i)/h$, $q = (u_{i+1} - t)/h$, Nous avons :

pour $d=1$:

$$\begin{cases} B_{i-1}(t) = q \\ B_i(t) = p \end{cases} \quad (2.27)$$

pour $d=2$:

$$\begin{cases} B_{i-2}(t) = \frac{q^2}{2} \\ B_{i-1}(t) = \frac{1+2pq}{2} \\ B_i(t) = \frac{p^2}{2} \end{cases} \quad (2.28)$$

pour $d=3$:

$$\begin{cases} B_{i-3}(t) = \frac{q^3}{6} \\ B_{i-2}(t) = \frac{3q(pq+1)}{6} \\ B_{i-1}(t) = \frac{3p(pq+1)}{6} \\ B_i(t) = \frac{p^3}{6} \end{cases} \quad (2.29)$$

Les courbes *B-splines* possèdent deux propriétés intéressantes :

– Les courbes *B-splines* sont à support compact.

Pour un $t \in [u_i, u_{i+1}]$, les seules valeurs de j pour lesquelles il faut calculer $B_j(t)$ sont $j = i - d, \dots, i$ (les autres étant nécessairement nulles). Donc

$$Q(t) = \sum_{j=-d}^{n-1} \alpha_j B_j(t) = \sum_{j=i-d}^i \alpha_j B_j(t) \quad (2.30)$$

– Les points de contrôle des courbes *B-splines* commutent avec toute transformation affine.

Supposons que $\Phi : \mathbb{R} \longrightarrow \mathbb{R}$ soit une transformation affine, alors nous avons l'égalité suivante :

$$\Phi(Q(t)) = \Phi\left(\sum_{j=-d}^{n-1} \alpha_j B_j(t)\right) = \sum_{j=-d}^{n-1} \Phi(\alpha_j B_j(t)) = \sum_{j=-d}^{n-1} \Phi(\alpha_j) B_j(t)$$

Surfaces *B-splines* uniformes de degré d

Les surfaces *B-splines* sont des généralisations des courbes *B-splines* en dimension deux.

Soient $a_s = u_0 < u_1 \dots < u_m = b_s$ une subdivision régulière de l'intervalle $I = [a_s, b_s]$ sur l'axe s , de pas $h_s = u_{i+1} - u_i = (b_s - a_s)/m$, et $a_t = v_0 < v_1 \dots < v_n = b_t$ une autre subdivision régulière de l'intervalle $J = [a_t, b_t]$ sur l'axe t , de pas $h_t = v_{j+1} - v_j = (b_t - a_t)/n$.

Soient B_i , la *B-spline* de degré d dont le support est $[u_i, u_{i+d+1}]$, $i = -d, \dots, m-1$, et B_j , la *B-spline* de degré d dont le support est $[v_j, v_{j+d+1}]$, $j = -d, \dots, n-1$. Nous introduisons ici les nœuds supplémentaires u_{-d}, \dots, u_{-1} , et u_{m+1}, \dots, u_{m+d} pour la *B-spline* B_i , et v_{-d}, \dots, v_{-1} , et v_{n+1}, \dots, v_{n+d} pour la *B-spline* B_j .

La forme générale des surfaces *B-splines* uniformes de degré d s'exprime ainsi :

$$\begin{aligned}
\vec{Q}(s, t) &= \sum_{i=-d}^{m-1} \sum_{j=-d}^{n-1} \vec{V}_{ij} B_i(s) B_j(t) \\
&= \sum_{i=-d}^{m-1} \sum_{j=-d}^{n-1} (\alpha_{ij} B_i(s) B_j(t), \beta_{ij} B_i(s) B_j(t), \gamma_{ij} B_i(s) B_j(t)) \quad (2.31)
\end{aligned}$$

où $s, t \in I \times J$ et $I, J \subset \mathbb{R}$

Les vecteurs $\vec{V}_{ij} = (\alpha_{ij}, \beta_{ij}, \gamma_{ij})$ sont appelés *points de contrôle* des surfaces *B-splines*. Les B_i, B_j sont des *fonctions de base* des courbes *B-splines* définies comme précédemment. Le produit $B_i(s) B_j(t)$ est le *produit tensoriel* de deux *B-splines*. Les surfaces *B-splines* ont les mêmes propriétés que celles des courbes *B-splines*:

– Elles sont à support compact.

Pour un couple $(s, t) \in [u_k, u_{k+1}] \times [v_l, v_{l+1}]$, les seules valeurs de i, j pour lesquelles il faut calculer les produits $B_i(s) B_j(t)$ (les autres étant nécessairement nulles) sont $i = k - d, \dots, k, j = l - d, \dots, l$. Donc

$$\vec{Q}(s, t) = \sum_{i=k-d}^k \sum_{j=l-d}^l (\alpha_{ij} B_i(s) B_j(t), \beta_{ij} B_i(s) B_j(t), \gamma_{ij} B_i(s) B_j(t)) \quad (2.32)$$

– Les points de contrôle de surfaces *B-splines* commutent avec toute transformation affine.

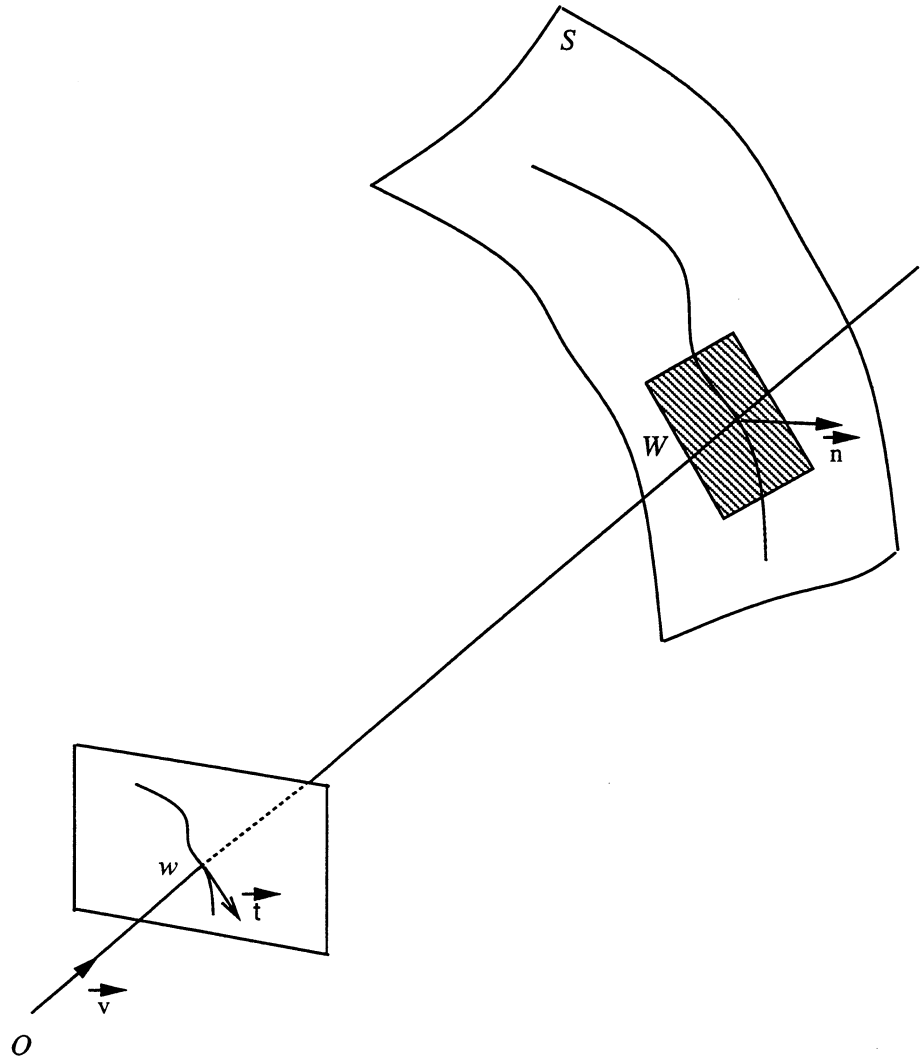
Supposons que $\Phi = (\Phi_1, \Phi_2, \Phi_3) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ soit une transformation affine, alors nous avons l'égalité suivante :

$$\begin{aligned}
&\Phi(\vec{Q}(s, t)) \\
&= \Phi(\sum_{ij} \vec{V}_{ij} B_i(s) B_j(t)) \\
&= \Phi(\sum_{ij} (\alpha_{ij} B_i(s) B_j(t), \beta_{ij} B_i(s) B_j(t), \gamma_{ij} B_i(s) B_j(t))) \\
&= \sum_{ij} \Phi(\alpha_{ij} B_i(s) B_j(t), \beta_{ij} B_i(s) B_j(t), \gamma_{ij} B_i(s) B_j(t)) \\
&= (\sum_{ij} \Phi_1(\alpha_{ij} B_i(s) B_j(t)), \sum_{ij} \Phi_2(\beta_{ij} B_i(s) B_j(t)), \sum_{ij} \Phi_3(\gamma_{ij} B_i(s) B_j(t))) \\
&= (\sum_{ij} \Phi_1(\alpha_{ij}) B_i(s) B_j(t), \sum_{ij} \Phi_2(\beta_{ij}) B_i(s) B_j(t), \sum_{ij} \Phi_3(\gamma_{ij}) B_i(s) B_j(t)) \\
&= \sum_{ij} \Phi(\vec{V}_{ij}) B_i(s) B_j(t)
\end{aligned}$$

2.2.2 Contour occultant

Les objets que nous étudierons seront les surfaces physiques des objets, les lignes tracées sur ces surfaces, ainsi que les contours sur un plan d'image, résultant de l'observation de ces surfaces et lignes par projection perspective.

Pour une surface S et un point d'observation O donnés, un *contour occultant* est l'ensemble des points w des projections de points W de la surface, tels que le rayon optique OW appartienne au plan tangent à la surface. La Figure 2.1 montre un contour d'occultation, où \vec{n} est le vecteur normal unitaire de la surface au point W , \vec{v} est le vecteur unitaire du rayon optique OW , et \vec{t} est le vecteur tangent au point w du contour d'occultation.

FIG. 2.1 - Une surface S et le *contour occultant* relativement au point O

2.2.3 Paramétrisation épipolaire d'une surface spatio-temporelle

Surface spatio-temporelle

Lors du mouvement relatif de la caméra par rapport à un objet (une surface), les contours perçus par la caméra se déplacent dans l'image. Au cours du temps, ils décrivent une surface appelée surface *spatio-temporelle*; de fait le terme de *spatio-temporel* pourrait avantageusement être remplacé par celui de *spatio-spatial* puisque le déplacement des contours est dû au déplacement relatif de la caméra par rapport à l'objet et non au simple fait de l'écoulement du temps.

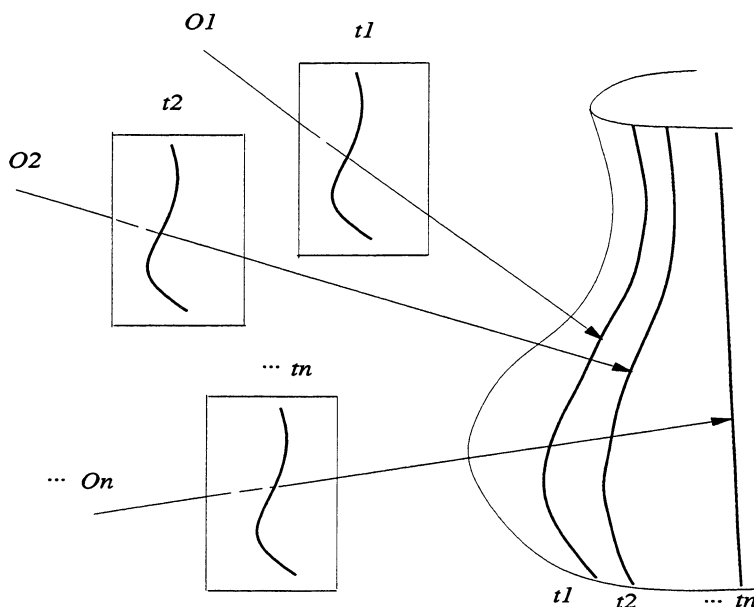


FIG. 2.2 - Contours perçus par la caméra en mouvement

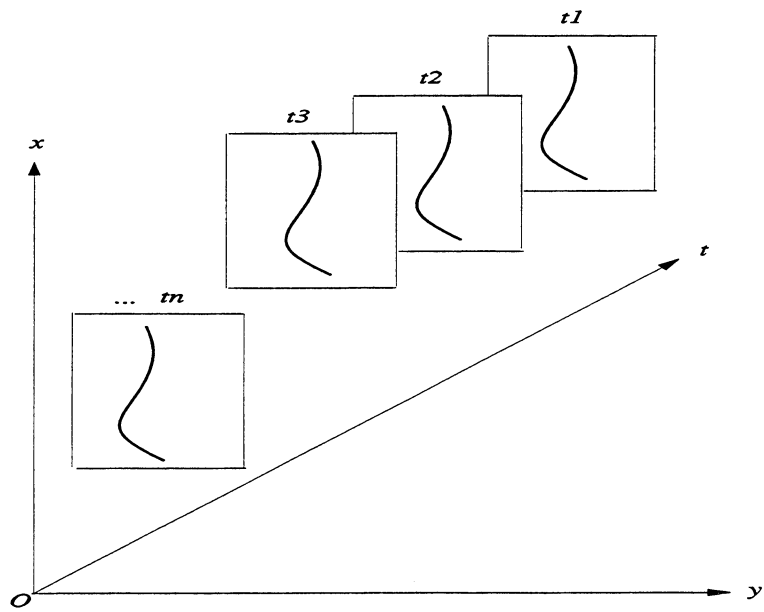
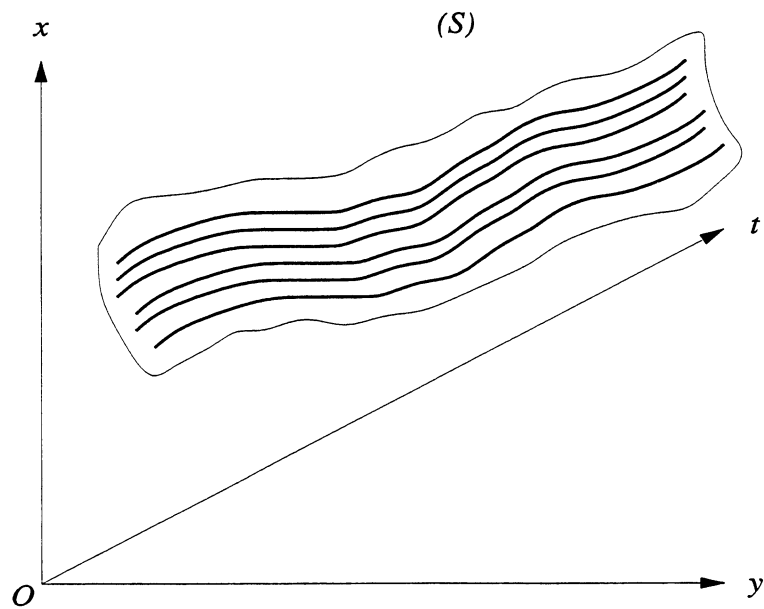
La Figure 2.2 illustre la formation d'un contour occultant apparent sur le plan d'image, pour un certain nombre de positions de la caméra.

La Figure 2.3 montre comment l'ensemble de ces contours est structuré temporellement, et la Figure 2.4 la création de la surface *spatio-temporelle*, en passant "au continu".

Paramétrisation épipolaire

La paramétrisation de la surface de l'objet et celle de la surface *spatio-temporelle* sont directement reliées. Une fois la surface *spatio-temporelle* entièrement connue, sa description et sa manipulation nécessite de définir une paramétrisation de la surface.

Chaque contour qui compose la surface *spatio-temporelle* possède sa propre paramétrisation; il faut donc établir une correspondance entre les contours de façon à définir une paramétrisation de la surface en s et t . Dans notre étude, le paramètre t est imposé par la démarche expérimentale: en effet, le paramètre t est lié au mouvement de la caméra, et est donc imposé; le paramètre s est imposé par l'abscisse curviligne des contours occultants. La surface *spatio-temporelle* $T(s, t)$ est en fait un maillage entre des courbes $T(s, t_0)$ (courbes à t constant, correspondant aux contours observés) et

FIG. 2.3 - Formation d'une surface *spatio-temporelle* dans un repère absoluFIG. 2.4 - Surface *spatio-temporelle*: passage au continu

$T(s_0, t)$ (courbes à s constant) où la correspondance à s constant est pour l'instant non définie (voir Figure 2.5).

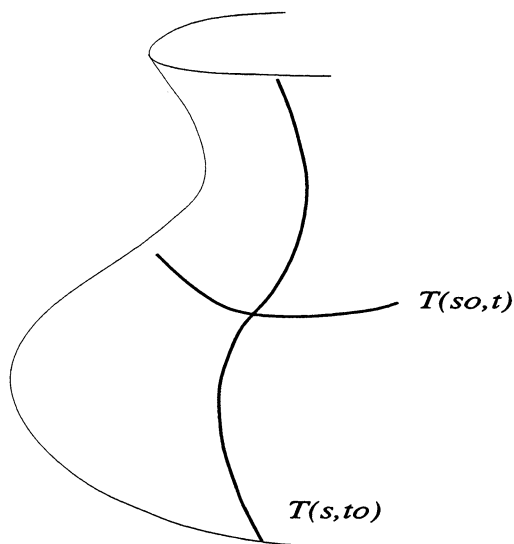


FIG. 2.5 - Paramétrisation de la surface *spatio-temporelle*

Cette correspondance doit vérifier les notions de voisinage et définir une véritable paramétrisation de la surface, mais n'a pas de contrainte particulière supplémentaire à respecter. Il est naturel d'utiliser la géométrie épipolaire pour déterminer les correspondances et ce choix donne des résultats plus stables [Cip 89, Arb 91a].

Cette nappe *spatio-temporelle* est définie intrinsèquement indépendamment d'une paramétrisation quelconque [Cip 89, Arb 91a].

La Figure 2.6 présente graphiquement la paramétrisation épipolaire :

Supposons que nous ayons deux images successives dans une séquence, le contour C_1 de celle prise au temps t_1 est mis en correspondance avec le contour C_2 de celle prise au temps t_2 . Soit a_1 un point quelconque sur C_1 , a_2 est le point d'intersection entre le plan épipolaire et le contour occultant C_2 (voir Figure 2.6). Si a_1 à l'instant t_1 possède pour abscisse s sur C_1 , alors nous définirons l'abscisse de a_2 à l'instant t_2 comme étant s également. Cette géométrie épipolaire permet de construire le correspondant sur C_2 d'un point de C_1 . Par cette construction, nous pouvons facilement mettre en correspondance les points du contour C_1 avec ceux du contour C_2 sauf lorsque la courbe devient tangente à la droite épipolaire, ce dernier cas est dégénéré et ne permet pas de reconstruire la surface [Arb 91a, Vai 90a, Cip 89]. Nous construisons donc la surface *spatio-temporelle* (S) (voir Figure 2.4) dans le repère $Oxyt$ en exprimant la direction de la ligne de vue dans $Oxyt$ pour chaque point des contours occultants suivis pendant cette séquence d'images.

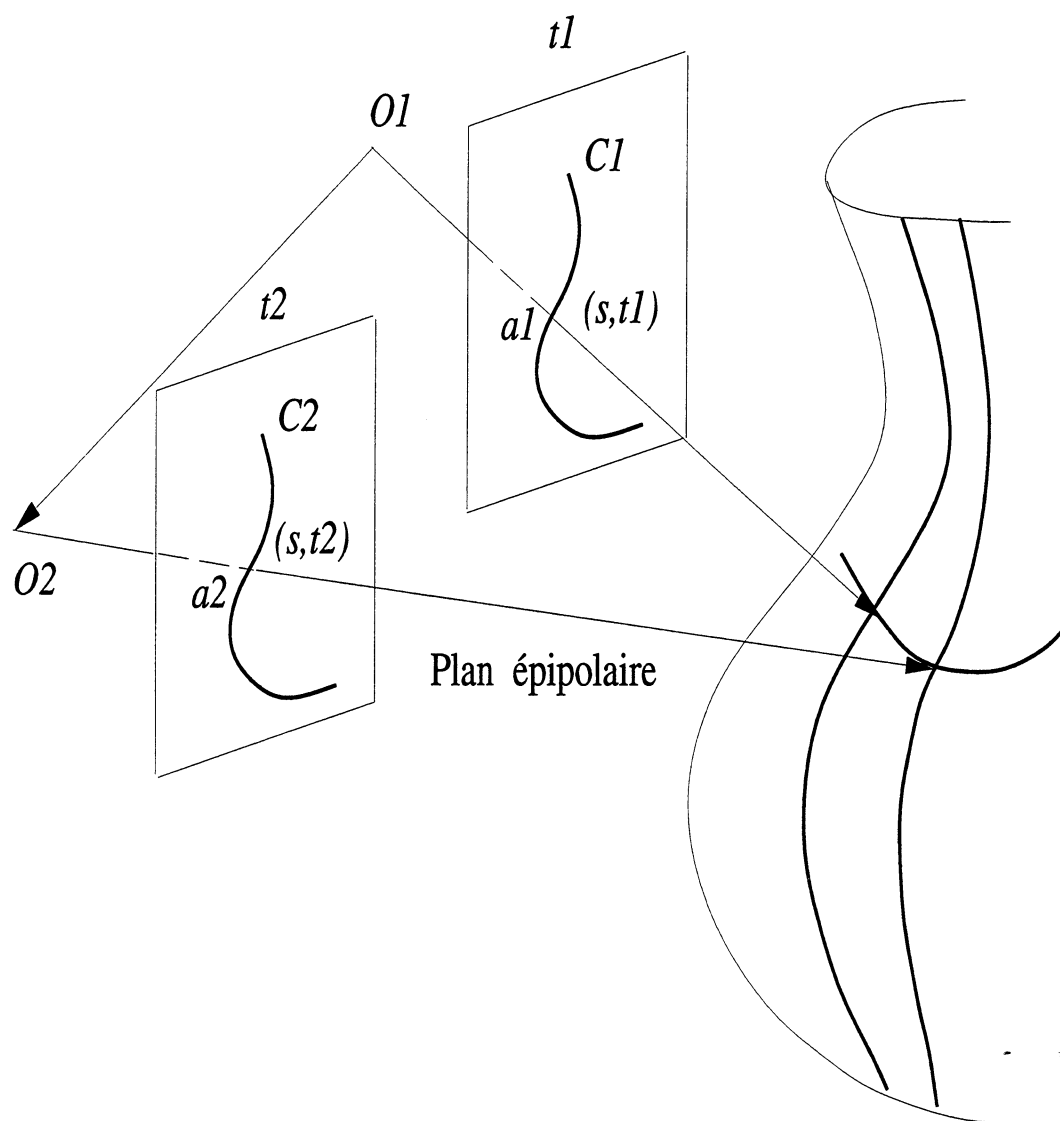


FIG. 2.6 - Paramétrisation épipolaire

2.3 Calibrage d'une caméra en mouvement

Le calibrage d'une caméra est un problème important et délicat en vision par ordinateur [Fau 87, Tsa 87]. Il consiste à déterminer cinq paramètres intrinsèques ou paramètres intérieurs, et six paramètres extrinsèques ou paramètres extérieurs. Ces onze paramètres de la caméra sont déterminés parfaitement par la matrice de projection perspective.

Calibrer une caméra nécessite de disposer de points ou de courbes [Fau 87, Fau 89, Fau 92b, May 92] dans l'espace. Les coordonnées de ces points ou de ces courbes sont supposées connues dans un certain repère. La méthode classique utilise l'image d'une mire constituée d'une ou plusieurs grilles. Nous estimons donc facilement la position des points d'intérêts (par exemple les coins de la grille) ou des droites dans un repère lié à cette mire afin de calibrer la caméra.

Cette méthode révèle quelques difficultés : Les paramètres du calibrage sont très sensibles aux modifications de l'environnement tels qu'un changement de la température ou un choc mécanique, etc . . . Dans le contexte de notre reconstruction des surfaces 3D d'un objet, la caméra est obligée de se déplacer autour de l'objet à reconstruire. Donc il faut un calibrage préalable à chaque point de vue, ce qui est très difficile à réaliser en pratique.

Pour éviter le calibrage classique nécessitant une mire, nous proposons ici une méthode de calcul des matrices de projection perspective à partir d'images 2D (au moins 2) obtenues avec une ou plusieurs caméras en mouvement. Aucune connaissance sur la ou les caméras ni sur le ou les mouvements n'est requise, seul cinq points de la scène sont supposés connus et sont utilisés comme repère relatif.

2.3.1 Le modèle de caméra

Le modèle le plus généralement utilisé est le modèle *sténopé* (voir Figure 2.7). La caméra est définie par un centre de projection O et une droite Oz perpendiculaire au plan image R passant par O , l'axe principal. La distance principale est f ; c'est la distance du point O au plan image R [Fau 87, Len 87].

2.3.2 Les équations

Nous considérons k images ($k \geq 2$) d'une scène rigide représentée par l points. Certains points apparaissent et disparaissent d'une image à une autre, cependant pour simplifier, les points sont supposés apparaître dans toutes les images, ce qui nous donne $k \times l$ points images.

Les points tridimensionnels de la scène sont introduits par W_i , $i = 1, \dots, l$. Pour chaque image j , le point W_i représenté par ses coordonnées homogènes $(x_i, y_i, z_i, t_i)^T$ se projette en w_{ij} noté par ses coordonnées homogènes $(u_{ij}, v_{ij}, \xi_{ij})^T$ ou bien par ses coordonnées image $(U_{ij}, V_{ij})^T$. Soit M_j la matrice de projection 3×4 de la $j^{\text{ème}}$ image.

Pour les coordonnées homogènes nous avons la relation :

$$\rho_{ij} w_{ij} = M_j W_i, \quad i = 1, \dots, l, \quad j = 1, \dots, k \quad (2.33)$$

où ρ_{ij} est un facteur d'échelle inconnu.

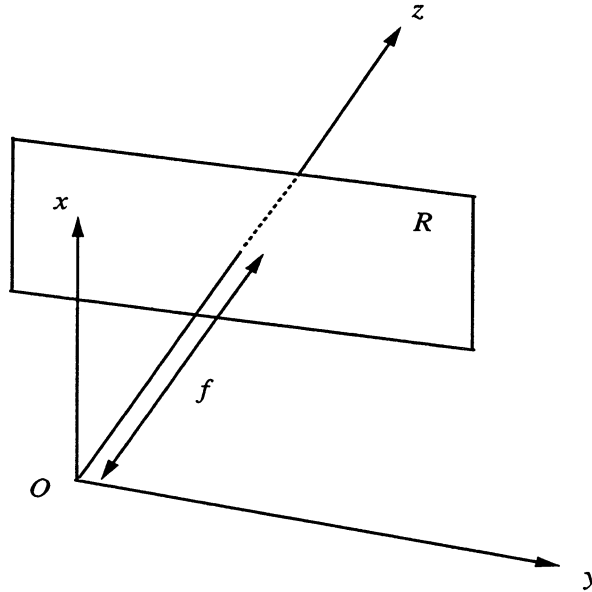


FIG. 2.7 - Le modèle géométrique d'une caméra

L'équation (2.33) peut être écrite sous une autre forme plus utilisée dans la littérature :

$$\begin{cases} U_{ij} = \frac{m_{11}^{(j)}x_i + m_{12}^{(j)}y_i + m_{13}^{(j)}z_i + m_{14}^{(j)}t_i}{m_{31}^{(j)}x_i + m_{32}^{(j)}y_i + m_{33}^{(j)}z_i + m_{34}^{(j)}t_i} \\ V_{ij} = \frac{m_{21}^{(j)}x_i + m_{22}^{(j)}y_i + m_{23}^{(j)}z_i + m_{24}^{(j)}t_i}{m_{31}^{(j)}x_i + m_{32}^{(j)}y_i + m_{33}^{(j)}z_i + m_{34}^{(j)}t_i} \end{cases} \quad (2.34)$$

Ces équations expriment tout simplement la colinéarité entre les points 3D et leur points 2D correspondants.

Comme nous avons l points et k images, cela nous donne un total de $2 \times l \times k$ équations. Pour ce qui est des inconnues, nous en avons $11 \times k$ pour les matrices de projection et $3 \times l$ pour les points 3D. Dès que le nombre des équations est plus grand que celui des inconnues, le système (2.34) devient résolvable. La solution de celui-ci ne peut être définie qu'à une transformation projective près. En effet, si M_j et W_i sont solutions du système, il en est de même pour $M_j N^{-1}$ et NW_i , où N est une transformation projective de l'espace, soit une matrice 4×4 inversible.

En conséquence, une base quelconque peut être choisie dans l'espace 3D. Pour l'espace projectif \mathbb{P}^3 , cinq points indépendants (non quatre à quatre coplanaires) forment une base [Sem 52]. Cette base peut être associée à une base canonique :

$$(1, 0, 0, 0)^T, (0, 1, 0, 0)^T, (0, 0, 1, 0)^T, (0, 0, 0, 1)^T \text{ et } (1, 1, 1, 1)^T$$

Ainsi les matrices de projection M_j , $j = 1, \dots, k$ sont estimées par des méthodes d'optimisation non linéaires comme l'algorithme de Levenberg-Marquardt. Les travaux de [Moh 92, Bou 93] détaillent cette méthode.

2.4 Modélisation mathématique

Nous allons dans ce paragraphe établir les trois équations élémentaires pour notre approche de la reconstruction de surfaces. Par hypothèse, nous supposons que la surface est représentée par des surfaces B-splines uniformes et bicubiques. Dans notre méthode, nous construisons la surface complète en fusionnant un ensemble des surfaces B-splines les unes avec les autres. Chaque surface B-spline peut être définie par sa représentation paramétrée:

$$\begin{aligned}\vec{X}(s, t) &= (X_1(s, t), X_2(s, t), X_3(s, t)) \\ &= \sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} (\alpha_{ij} B_i(s) B_j(t), \beta_{ij} B_i(s) B_j(t), \gamma_{ij} B_i(s) B_j(t))\end{aligned}\quad (2.35)$$

Il convient de remarquer que la reconstruction d'une surface B-spline est équivalente à trouver les valeurs de ces $3 \times (m+3) \times (n+3)$ coefficients inconnus α_{ij} , β_{ij} , γ_{ij} , $i = -3, \dots, m-1$, $j = -3, \dots, n-1$ dans la formule (2.35).

Nous remarquons que les points de contrôle $(\alpha_{ij})_{i=1, \dots, m+3, j=1, \dots, n+3}$ de la première composante $X_1(s, t)$ peuvent s'écrire sous une forme matricielle ainsi que les points de contrôle $(\beta_{ij})_{i=1, \dots, m+3, j=1, \dots, n+3}$ de la deuxième composante $X_2(s, t)$, et les points de contrôle $(\gamma_{ij})_{i=1, \dots, m+3, j=1, \dots, n+3}$ de la troisième composante $X_3(s, t)$. Leurs matrices sont les suivantes :

$$(\alpha_{ij})_{i=1, \dots, m+3, j=1, \dots, n+3} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1(n+3)} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2(n+3)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{(m+3)1} & \alpha_{(m+3)2} & \dots & \alpha_{(m+3)(n+3)} \end{pmatrix}\quad (2.36)$$

$$(\beta_{ij})_{i=1, \dots, m+3, j=1, \dots, n+3} = \begin{pmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1(n+3)} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2(n+3)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{(m+3)1} & \beta_{(m+3)2} & \dots & \beta_{(m+3)(n+3)} \end{pmatrix}\quad (2.37)$$

$$(\gamma_{ij})_{i=1, \dots, m+3, j=1, \dots, n+3} = \begin{pmatrix} \gamma_{11} & \gamma_{12} & \dots & \gamma_{1(n+3)} \\ \gamma_{21} & \gamma_{22} & \dots & \gamma_{2(n+3)} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{(m+3)1} & \gamma_{(m+3)2} & \dots & \gamma_{(m+3)(n+3)} \end{pmatrix}\quad (2.38)$$

Nous pouvons regrouper ces trois matrices (2.36), (2.37), (2.38) sous la forme vectorielle ci-dessous :

$$\underbrace{(x_1, x_2, \dots, x_{(m+3)(n+3)})}_{\alpha_{ij}}, \underbrace{(x_{(m+3)(n+3)+1}, \dots, x_{2(m+3)(n+3)})}_{\beta_{ij}}, \underbrace{(x_{2(m+3)(n+3)+1}, \dots, x_{3(m+3)(n+3)})}_{\gamma_{ij}}$$

Plus précisément,

$$(x_l)_{l=1, \dots, 3(m+3)(n+3)} = \left\{ \begin{array}{ll} x_{(i-1)(n+3)+j} & \text{Si } x_l \text{ est inclus dans } \alpha_{ij} \\ x_{(m+3)(n+3)+(i-1)(n+3)+j} & \text{Si } x_l \text{ est inclus dans } \beta_{ij} \\ x_{2(m+3)(n+3)+(i-1)(n+3)+j} & \text{Si } x_l \text{ est inclus dans } \gamma_{ij} \end{array} \right\}\quad (2.39)$$

Comme les surfaces B-splines sont à support compact, nous avons alors le lemme qui suit :

Lemme :

Pour un couple $(s, t) \in [u_k, u_{k+1}] \times [v_l, v_{l+1}]$, chacune des matrices précédentes possède 16 points de contrôle qui permettent de calculer la valeur de chaque composante. Ces 16 points de contrôle sont pour la première composante :

$$\begin{pmatrix} \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \alpha_{kl} & \alpha_{k(l+1)} & \alpha_{k(l+2)} & \alpha_{k(l+3)} & \dots \\ \dots & \alpha_{(k+1)l} & \alpha_{(k+1)(l+1)} & \alpha_{(k+1)(l+2)} & \alpha_{(k+1)(l+3)} & \dots \\ \dots & \alpha_{(k+2)l} & \alpha_{(k+2)(l+1)} & \alpha_{(k+2)(l+2)} & \alpha_{(k+2)(l+3)} & \dots \\ \dots & \alpha_{(k+3)l} & \alpha_{(k+3)(l+1)} & \alpha_{(k+3)(l+2)} & \alpha_{(k+3)(l+3)} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad (2.40)$$

De même, pour la deuxième composante :

$$\begin{pmatrix} \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \beta_{kl} & \beta_{k(l+1)} & \beta_{k(l+2)} & \beta_{k(l+3)} & \dots \\ \dots & \beta_{(k+1)l} & \beta_{(k+1)(l+1)} & \beta_{(k+1)(l+2)} & \beta_{(k+1)(l+3)} & \dots \\ \dots & \beta_{(k+2)l} & \beta_{(k+2)(l+1)} & \beta_{(k+2)(l+2)} & \beta_{(k+2)(l+3)} & \dots \\ \dots & \beta_{(k+3)l} & \beta_{(k+3)(l+1)} & \beta_{(k+3)(l+2)} & \beta_{(k+3)(l+3)} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad (2.41)$$

et pour la troisième composante :

$$\begin{pmatrix} \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \gamma_{kl} & \gamma_{k(l+1)} & \gamma_{k(l+2)} & \gamma_{k(l+3)} & \dots \\ \dots & \gamma_{(k+1)l} & \gamma_{(k+1)(l+1)} & \gamma_{(k+1)(l+2)} & \gamma_{(k+1)(l+3)} & \dots \\ \dots & \gamma_{(k+2)l} & \gamma_{(k+2)(l+1)} & \gamma_{(k+2)(l+2)} & \gamma_{(k+2)(l+3)} & \dots \\ \dots & \gamma_{(k+3)l} & \gamma_{(k+3)(l+1)} & \gamma_{(k+3)(l+2)} & \gamma_{(k+3)(l+3)} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad (2.42)$$

Démonstration :

Pour un couple $(s, t) \in [u_k, u_{k+1}] \times [v_l, v_{l+1}]$, les seules valeurs de i, j pour lesquelles il faut calculer les produits $B_i(s)B_j(t)$ sont $i = k, \dots, k+3, j = l, \dots, l+3$ (les autres étant nécessairement nulles), ce qui conduit à calculer 16 points de contrôle.

Finalement, nous avons la proposition suivante :

Proposition :

Si $(s, t) \in [u_k, u_{k+1}] \times [v_l, v_{l+1}]$, alors $\vec{X}(s, t) = (X_1(s, t), X_2(s, t), X_3(s, t))$ où

$$X_1(s, t) = \sum_{i=k}^{k+3} \sum_{j=l}^{l+3} x_{(i-1)(n+3)+j} B_i(s) B_j(t)$$

$$X_2(s, t) = \sum_{i=k}^{k+3} \sum_{j=l}^{l+3} x_{(m+3)(n+3)+(i-1)(n+3)+j} B_i(s) B_j(t)$$

$$X_3(s, t) = \sum_{i=k}^{k+3} \sum_{j=l}^{l+3} x_{2(m+3)(n+3)+(i-1)(n+3)+j} B_i(s) B_j(t) \quad (2.43)$$

où les polynômes B_i et B_j sont définis dans (2.29).

2.4.1 Equations élémentaires de reconstruction de surfaces

Notre but est de reconstruire la surface tridimensionnelle d'un objet observé par une caméra en mouvement. Nous avons donc deux sortes d'informations disponibles : les coordonnées des contours occultants de l'objet perçus par la caméra dans l'image, et la connaissance du mouvement de la caméra qui est représentée par une matrice de projection perspective. Dans la suite, nous allons voir comment modéliser le problème de la reconstruction des surfaces à partir de ces informations.

La matrice de projection perspective

Le modèle de caméra utilisé est le modèle sténopé. La caméra est définie par un centre de projection O (voir Figure 2.7) et l'image est obtenue par une projection perspective parfaite de centre O . Elle se forme sur le plan rétinien. La relation géométrique entre les coordonnées homogènes d'un point $W = [X, Y, Z, 1]^T$ de la surface S et son image $w = (x, y)^T$ peut s'écrire très simplement :

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.44)$$

où $M = (m_{pr})_{p=1,\dots,3,r=1,\dots,4}$ est une matrice 3×4 connue dans la littérature sous le nom de matrice de projection perspective, x et y sont les coordonnées du point image exprimées en pixels.

M est définie à un facteur d'échelle près ainsi que le montre l'équation suivante qui exprime directement x et y en fonction des éléments de la matrice :

$$\begin{cases} x = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \\ y = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \end{cases} \quad (2.45)$$

Ces deux équations suffisent à représenter toute l'information qui est nécessaire pour calculer des quantités tridimensionnelles.

Nous pouvons facilement calculer la ligne de vue correspondant à un point donné de l'image à partir de la matrice de projection perspective connue.

En effet, ces deux équations du système (2.45) présentent deux plans dans l'espace. L'intersection de ces deux plans est une droite passant par deux points w et W . Donc la direction de cette droite est la ligne de vue \vec{v} correspondant au point w . Elle est calculée par le produit vectoriel de deux vecteurs normaux de ces deux plans.

Plus précisément, nous avons les relations suivantes en posant $\vec{v} = (v_1, v_2, v_3)$:

$$\begin{cases} v_1 = m_{12}m_{23} - ym_{12}m_{33} - xm_{32}m_{23} - m_{22}m_{13} + xm_{22}m_{33} + ym_{32}m_{13} \\ v_2 = m_{21}m_{13} - xm_{21}m_{33} - ym_{31}m_{13} - m_{11}m_{23} + ym_{11}m_{33} + xm_{31}m_{23} \\ v_3 = m_{11}m_{22} - ym_{11}m_{32} - xm_{31}m_{22} - m_{21}m_{12} + xm_{21}m_{32} + ym_{31}m_{12} \end{cases} \quad (2.46)$$

Propriétés d'un contour occultant

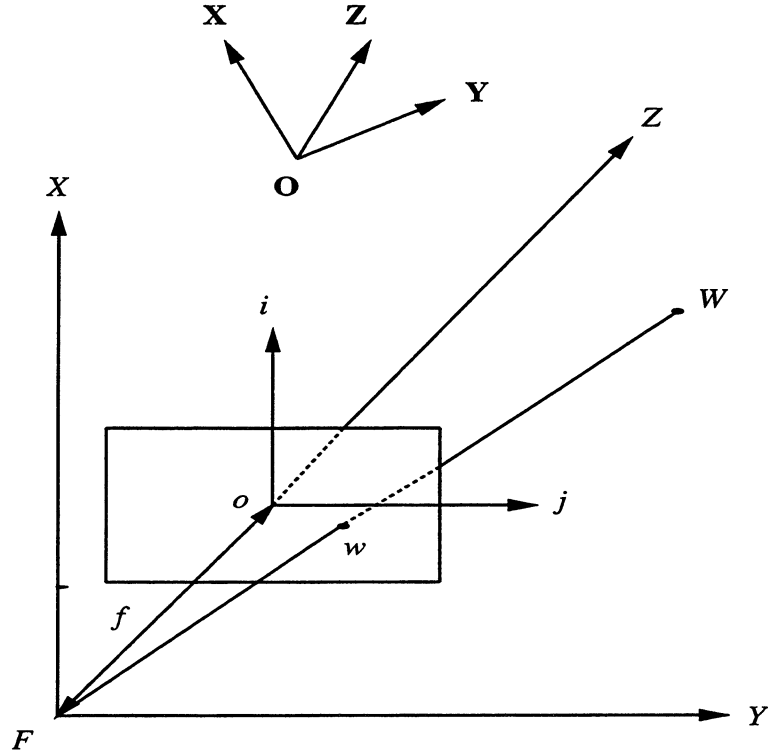


FIG. 2.8 - Le modèle sténopé d'une caméra

Soient $[X_1(s, t), X_2(s, t), X_3(s, t), 1]$ les coordonnées homogènes d'un point W de la surface S , (x, y) les coordonnées de w de la projection de W dans le plan image (voir Figure 2.8).

Selon le système d'équations (2.45) du paragraphe 2.4.1, nous avons les relations suivantes :

$$\begin{cases} x = \frac{m_{11}X_1(s,t) + m_{12}X_2(s,t) + m_{13}X_3(s,t) + m_{14}}{m_{31}X_1(s,t) + m_{32}X_2(s,t) + m_{33}X_3(s,t) + m_{34}} \\ y = \frac{m_{21}X_1(s,t) + m_{22}X_2(s,t) + m_{23}X_3(s,t) + m_{24}}{m_{31}X_1(s,t) + m_{32}X_2(s,t) + m_{33}X_3(s,t) + m_{34}} \end{cases} \quad (2.47)$$

Trivialement, (2.47) peut s'écrire sous la forme suivante en effectuant des remplacements par les formules (2.29), (2.35) et en multipliant les deux membres de l'équation par le dénominateur :

$$P_1(x, s, t, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}) = 0 \quad P_2(y, s, t, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}) = 0 \quad (2.48)$$

Il est très facile de remarquer que les deux équations (2.48) sont linéaires par rapport aux inconnues $\alpha_{ij}, \beta_{ij}, \gamma_{ij}$.

Comme w est un point du contour d'occultation, la ligne de vue OW est tangentielle à la surface (voir Figure 2.9). Supposons que $\vec{v} = (v_1, v_2, v_3)$ soit la direction de la ligne de vue et \vec{n} soit le vecteur normal à la surface au point W . Le plan tangent au point W est engendré par les vecteurs suivants : $\frac{\partial \vec{X}(s,t)}{\partial s}$ et $\frac{\partial \vec{X}(s,t)}{\partial t}$.

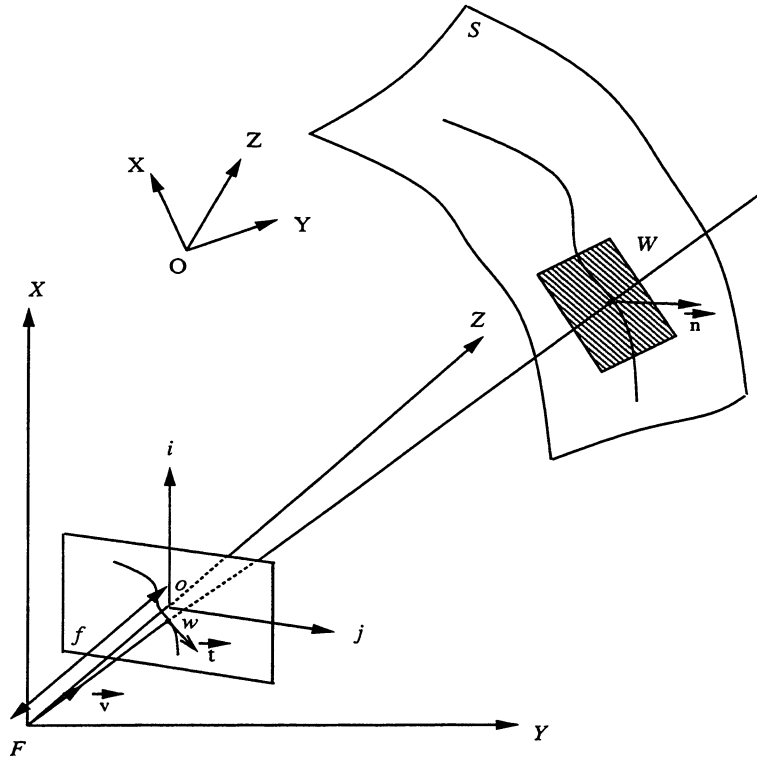


FIG. 2.9 - Propriété d'un contour occultant

Donc le vecteur normal \vec{n} au point W est donné par :

$$\vec{n} = \frac{\partial \vec{X}(s, t)}{\partial s} \times \frac{\partial \vec{X}(s, t)}{\partial t} \quad (2.49)$$

D'après la définition d'un contour occultant, nous avons cette égalité :

$$\vec{n} \cdot \vec{v} = 0 \quad (2.50)$$

Donc l'équation (2.50) peut s'écrire sous la forme suivante :

$$\vec{n} \cdot \vec{v} = \left(\frac{\partial \vec{X}(s, t)}{\partial s} \times \frac{\partial \vec{X}(s, t)}{\partial t} \right) \cdot \vec{v} = \begin{vmatrix} \frac{\partial X_1}{\partial s} & \frac{\partial X_2}{\partial s} & \frac{\partial X_3}{\partial s} \\ \frac{\partial X_1}{\partial t} & \frac{\partial X_2}{\partial t} & \frac{\partial X_3}{\partial t} \\ v_1 & v_2 & v_3 \end{vmatrix} \quad (2.51)$$

Trivialement, nous pouvons écrire l'équation (2.51) en effectuant des remplacements par les formules (2.29), (2.35), (2.46) et (2.49) :

$$P_3(x, y, s, t, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}) = 0 \quad (2.52)$$

Malheureusement, nous nous apercevons que l'équation (2.52) n'est pas linéaire par rapport aux inconnues α_{ij} , β_{ij} , γ_{ij} .

Comme le plan tangent au point W de la surface S est aussi défini par sa ligne de vue \vec{v} et le vecteur tangent \vec{t} au point w du contour occultant (voir Figure 2.9), donc nous avons l'équation ci-dessous :

$$(\vec{v} \times \vec{t}) \times \vec{n} = 0 \quad (2.53)$$

Cependant, cette équation (2.53) donne les mêmes informations que l'équation (2.51). En effet, il décrit aussi la propriété d'orthogonalité entre le vecteur normal et le plan tangent au point W de la surface S . Donc elle est supprimée.

Quelle relation existe-t-il entre s et x, y, t dans P_1, P_2, P_3 ?

Intuitivement, s et t sont deux paramètres des surfaces B-splines. (x, y) sont les coordonnées d'un point de contour occultant d'une image. D'après la construction de la paramétrisation épipolaire d'une surface spatio-temporelle du paragraphe précédent, nous avons la conséquence suivante: s est l'abscisse curviligne d'un point (x, y) du contour occultant à l'instant donné t dans une séquence d'images. L'origine de l'abscisse curviligne est choisie au premier point.

2.4.2 Système d'équations non linéaires pour une séquence d'images

Finalement, étant donné un point $w = (x, y)$ du contour occultant d'une image quelconque, nous pouvons calculer facilement les deux paramètres de la surface spatio-temporelle s et t qui lui correspondent. Nous avons le système d'équations non linéaires suivant, d'après les paragraphes précédents :

$$\begin{cases} P_1(x, s, t, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}) = 0 \\ P_2(y, s, t, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}) = 0 \\ P_3(x, y, s, t, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}) = 0 \end{cases} \quad (2.54)$$

Nous généralisons le système (2.54) pour une séquence d'images en posant k comme étant le numéro d'image de la séquence, et l étant le numéro d'un point de contour occultant de la $k^{\text{ème}}$ image de la séquence. Donc nous avons le système d'équations non linéaires suivant pour une séquence d'images :

$$\begin{cases} q_1(x_l^k, s_l^k, t_l^k, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}^k) = 0 \\ q_2(y_l^k, s_l^k, t_l^k, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}^k) = 0 \\ q_3(x_l^k, y_l^k, s_l^k, t_l^k, \alpha_{ij}, \beta_{ij}, \gamma_{ij}, m_{pr}^k) = 0 \end{cases} \quad (2.55)$$

Nous nous apercevons facilement que ce système d'équations (2.55) n'est pas linéaire par rapport aux inconnues $\alpha_{ij}, \beta_{ij}, \gamma_{ij}$.

2.5 Résolution du système

Pour k_o images d'une séquence et l_o points extraits par image, $l_o \gg m$ et $l_o \gg n$, alors nous avons un système redondant de $3 \times k_o \times l_o$ équations non linéaires avec $(m+3) \times (n+3)$ points de contrôle à estimer. Le nombre d'inconnues est $3 \times (m+3) \times (n+3)$ d'après le système d'équations (2.55).

Le problème se pose naturellement de la manière suivante: Est-ce qu'il existe une solution pour le système (2.55)? Si oui, est ce qu'elle est unique?

Notre approche de la reconstruction des surfaces tridimensionnelles d'un objet est effectuée dans le contexte de l'étude du mouvement des contours occultants d'une séquence d'images. Si la solution du système est correcte, alors elle est *a priori* unique. En effet, notre caméra observe toujours la même surface d'un objet lors de notre expérimentation. Aussi, nous ne nous sommes pas intéressés à prouver si le système admettait une solution et une seule.

De nombreuses méthodes numériques sont appliquées pour résoudre un système d'équations non linéaires dans le domaine de la vision par ordinateur : la méthode classique de Gauss-Newton, la méthode d'élimination [Kri 90], la méthode de continuité [Wam 88], la méthode d'homotopie [Fau 92b]. Au cours de cette étude, nous avons expérimenté les deux méthodes suivantes : Quasi-Newton et Levenberg-Marquardt.

2.5.1 Energie et moindres carrés

Le problème de notre approche aux moindres carrés se pose de la manière suivante :

Soient k_o le nombre d'images d'une séquence et l_o le nombre de points choisis du contour occultant par image. L'approche consiste à minimiser l'énergie qui suit :

$$\Theta = \sum_{k=1}^{k_o} \sum_{l=1}^{l_o} [\lambda (q_1^2 + q_2^2) + \mu q_3^2] \quad (2.56)$$

où λ et μ sont deux coefficients réels de pondération.

2.5.2 Energie et moindres carrés avec régularisation

L'approche précédente ne permet pas d'ajuster facilement la forme de la surface à reconstruire dans le cas où il y a du bruit présent dans les images. Donc nous imposons un terme supplémentaire qui s'appelle *terme de régularisation* [Gri 82, Pog 85, Ter 86, Bla 87, Lau 87]. Ce terme fut introduit par Laurent [Lau 87] dans le cas des courbes B-splines à une dimension. Nous généralisons ici ce terme pour les surfaces B-splines. Il nous permet de minimiser les variations de l'orientation de la surface locale pour avoir une surface lissée. Donc il est équivalent de minimiser les quantités de dérivées du deuxième ordre de la surface.

Soient $\vec{X}(s, t) = (X_1(s, t), X_2(s, t), X_3(s, t))$ la représentation paramétrée de la surface, E, F, G trois coefficients de la première forme fondamentale, et L, M, N trois coefficients de la deuxième forme fondamentale (voir Annexe A.2). Les courbures gaussiennes K_g et moyennes K_m en un point de la surface sont données par les formules suivantes :

$$\begin{aligned} K_g &= \frac{LN - M^2}{EG - F^2} \\ K_m &= \frac{LG + EN - 2FM}{2 \times (EG - F^2)} \end{aligned} \quad (2.57)$$

Dans ce paragraphe, plusieurs possibilités en ce qui concerne le terme de régularisation sont présentées.

Régularisateur de courbure gaussienne

Le premier terme de régularisation est posé par la courbure gaussienne. C'est-à-dire nous minimisons la formule ci-dessous :

$$\Delta_1 = \int_I \int_J K_g^2 dsdt = \int_I \int_J \left(\frac{LN - M^2}{EG - F^2} \right)^2 dsdt \quad (2.58)$$

Régularisateur de courbure moyenne

Le deuxième terme de régularisation est posé par la courbure moyenne. C'est-à-dire nous minimisons la formule suivante :

$$\Delta_2 = \int_I \int_J K_m^2 dsdt = \int_I \int_J \left[\frac{LG + EN - 2FM}{2 \times (EG - F^2)} \right]^2 dsdt \quad (2.59)$$

Régularisateur de variation quadratique

Supposons que H soit une fonction de deux variables s et t avec $s, t \in I \times J$ et $I, J \subseteq \mathbb{R}$. La *variation quadratique* [Gri 82] de H est définie par la formule qui suit :

$$\delta = \int_I \int_J \left[\left(\frac{\partial H}{\partial s} \right)^2 + \left(\frac{\partial H}{\partial t} \right)^2 \right] dsdt \quad (2.60)$$

Dans notre cas, nous définissons deux fonctions $H_i^1(s, t)$, $H_i^2(s, t)$ telles que

$$\begin{aligned} H_i^1(s, t) &= \frac{\partial X_i}{\partial s} \\ H_i^2(s, t) &= \frac{\partial X_i}{\partial t} \end{aligned} \quad (2.61)$$

où $i = 1, 2, 3$.

Leurs variations quadratiques δ_i^1 , δ_i^2 de $H_i^1(s, t)$, $H_i^2(s, t)$ sont définies par les formules ci-dessous :

$$\begin{aligned} \delta_i^1 &= \int_I \int_J \left[\left(\frac{\partial^2 X_i}{\partial^2 s} \right)^2 + \left(\frac{\partial^2 X_i}{\partial s \partial t} \right)^2 \right] dsdt \\ \delta_i^2 &= \int_I \int_J \left[\left(\frac{\partial^2 X_i}{\partial t \partial s} \right)^2 + \left(\frac{\partial^2 X_i}{\partial^2 t} \right)^2 \right] dsdt \end{aligned} \quad (2.62)$$

où $i = 1, 2, 3$.

Or $X_i(s, t)$ est C^2 , donc nous avons cette égalité :

$$\frac{\partial^2 X_i}{\partial t \partial s} = \frac{\partial^2 X_i}{\partial s \partial t} \quad (2.63)$$

où $i = 1, 2, 3$.

Donc nous proposons le terme de régularisation qui suit :

$$\Delta_3 = \sum_{i=1}^3 (\delta_i^1 + \delta_i^2) = \sum_{i=1}^3 \int_I \int_J \left[\left(\frac{\partial^2 X_i}{\partial^2 s} \right)^2 + 2 \times \left(\frac{\partial^2 X_i}{\partial s \partial t} \right)^2 + \left(\frac{\partial^2 X_i}{\partial^2 t} \right)^2 \right] dsdt \quad (2.64)$$

Ce terme revient à minimiser la variation de la courbure. Cela permet de lisser la surface des zones qui sont particulièrement bruitées, et souvent caractérisées par une fonction de courbure très bruitée. Ceci se justifie par le fait que la courbure est une fonction des dérivées du deuxième ordre par rapport à la fonction d'origine.

Régularisateur simplifié

Puisque nous appliquons les surfaces B-splines bicubiques, donc le degré maximal de s et t dans le terme X_i , où $i = 1, 2, 3$, est 3. Nous remarquons que $\frac{\partial^4 X_i}{\partial^2 s \partial^2 t}$, où $i = 1, 2, 3$ est une fonction linéaire de s et t . Pour simplifier le terme de régularisation, nous pouvons proposer d'utiliser le régularisateur ci-dessous :

$$\Delta = \sum_{i=1}^3 \int_I \int_J \left(\frac{\partial^4 X_i}{\partial^2 s \partial^2 t} \right)^2 ds dt \quad (2.65)$$

Finalement, dans notre recherche, nous avons utilisé le terme de régularisation (2.65). Donc nous minimisons l'énergie suivante [Zha 93b, Zha 93c, Zha 94c] :

$$\Theta = \sum_{k=1}^{k_o} \sum_{l=1}^{l_o} [\lambda (q_1^2 + q_2^2) + \mu q_3^2] + \tau \int_I \int_J \left[\left(\frac{\partial^4 X_1}{\partial^2 s \partial^2 t} \right)^2 + \left(\frac{\partial^4 X_2}{\partial^2 s \partial^2 t} \right)^2 + \left(\frac{\partial^4 X_3}{\partial^2 s \partial^2 t} \right)^2 \right] ds dt$$

où λ et μ sont deux coefficients réels de pondération. Le facteur τ est la *tension* des surfaces B-splines, la partie de l'intégrale double est le *terme de régularisation* des surfaces B-splines. Ce facteur de tension permet de passer continûment de la solution exacte aux moindres carrés à une solution dominée par ce terme du régularisateur.

2.6 Implémentation

Dans ce paragraphe, nous allons décrire le schéma général du processus de reconstruction des surfaces B-splines.

Il est évident que les surfaces B-splines sont entièrement déterminées par un ensemble de leurs points de contrôle dans l'espace. De plus, d'après les propriétés des surfaces B-splines, les positions des points de contrôle sont assez proches de ces surfaces. Cela nous permet de déterminer les points de départ de notre système non linéaire en calculant les profondeurs de certains points sur la surface. La méthode adaptée est une simple *triangulation* (voir Figure 2.10).

Dans les paragraphes précédents, nous avons décrit le modèle mathématique de la reconstruction des surfaces. Il s'agit d'un système d'équations non linéaires que nous devons résoudre. Le schéma général du processus de reconstruction des surfaces B-splines uniformes et bicubiques est le suivant :

- Extraction des contours et des coins.
- Mise en correspondance des contours occultants et des coins.
- Calcul d'une représentation sous forme de B-splines uniformes, cubiques et régularisées des chaînes de contours qui ont été mises en correspondance.

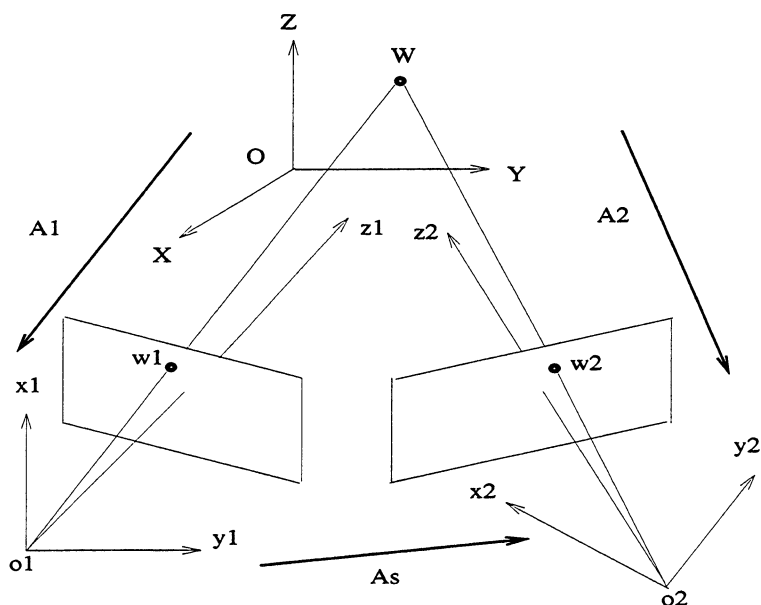


FIG. 2.10 - Reconstruction par *triangulation* à l'aide de deux caméras. Ces deux caméras sont calibrées, les matrices de transformation du repère absolu dans le repère de la caméra A_1 et A_2 sont donc connues. Nous pouvons déterminer la matrice de transformation entre le repère caméra gauche et le repère caméra droite A_s par $A_2 \cdot A_1^{-1}$.

- Calcul des matrices essentielles et des lignes épipolaires.
- Calcul des matrices projectives.
- Calcul des points d'intersection entre les lignes épipolaires et les contours occultants.
- Démarrage de notre programme de reconstruction des surfaces: estimation par triangulation stéréo des points de contrôle, minimisation de l'énergie Θ .

Lors de notre expérimentation, nous avons appliqué deux méthodes d'optimisation au sens des moindres carrés: **Quasi-Newton** et **Levenberg-Marquardt**.

Quasi-Newton : Nous programmons entièrement en **Maple**. Par expérience, nous savons qu'il est difficile de choisir les valeurs initiales, les coefficients λ et μ de pondération et le facteur de tension τ pour que le système converge correctement.

Levenberg-Marquardt : Nous programmons en **C**. L'expérience montre que cette méthode converge bien et que le choix des valeurs initiales est très aisé par la méthode de triangulation.

Le lecteur intéressé pourra trouver les détails dans [Gil 89]. Dans notre étude, nous utilisons la procédure de la librairie qui est décrite dans le livre [Pre 88].

2.7 Résultats expérimentaux

Nous avons testé notre méthode sur des données synthétiques et sur des données réelles.

2.7.1 Données synthétiques

Dans ce paragraphe, nous allons décrire les résultats obtenus par notre algorithme avec des données synthétiques. Des expériences ont été menées avec des sphères, des segments, des cylindres, des hyperboloïdes car la complexité des calculs menés exigeait de comparer chacune des quantités calculées avec sa valeur nominale. Cela nous a permis une juste appréciation de la sensibilité au bruit des différentes étapes.

Le principe étant similaire sur les différents exemples, nous en donnons ici le détail dans le cas de la sphère.

Nous supposons que le modèle de la caméra que nous avons utilisé est parfaitement sténopé. La distance focale f de notre caméra est $f = 1 \text{ cm}$. L'image générée est binaire, c'est-à-dire que les pixels du fond sont mis à zéro. Les autres sont mis à 255. Le bruit associé à la digitalisation est présent dans les mesures.

Sphères

La séquence utilisée est constituée de 10 images d'une demi-sphère de rayon 4 cm . Nous supposons que la sphère est statique, son centre se trouve sur l'axe Z . Elle est définie par :

$$x^2 + y^2 + (z - 8)^2 = 4^2$$

La caméra est en mouvement le long de l'axe Y , et est bien calibrée. La distance entre la sphère et la caméra est environ de 12 cm . Le déplacement de la caméra entre deux images successives est assez petit; la translation est environ de 1 cm et la rotation est à peu près de 6 degrés. Nous supposons aussi que les contours occultants sont détectés et suivis dans la séquence. Le schéma d'obtention des données synthétiques est le suivant :

- Nous projetons cette demi-sphère sur le plan image 512×512 , afin d'obtenir les contours occultants dans la séquence.
- Nous choisissons 30 points sur la demi-sphère que nous projetons respectivement sur chaque image. Nous obtenons donc le suivi de ces 30 points. Nous appliquons la méthode de Faugeras [Fau 92b] pour calculer les matrices essentielles des images de la séquence.
- Nous choisissons 60 points initiaux sur le contour occultant de la première image, et nous calculons dans la deuxième image les lignes épipolaires correspondant aux points choisis afin d'obtenir les points d'intersection entre ces lignes épipolaires et le contour occultant. Nous traitons ces points d'intersection comme les points initiaux et répétons la procédure précédente jusqu'à la dernière image. Finalement, nous avons les coordonnées x, y, s, t de ces points dans la séquence.

- Nous remplaçons les coordonnées f, x, y, s, t dans le système (2.55) pour calculer les points de contrôle des surfaces B-splines par les deux méthodes suivantes : Quasi-Newton, Levenberg-Marquardt.

La Figure 2.11 montre une séquence de huit images d'une demi-sphère avec les lignes épipolaires.

Avant de présenter les résultats obtenus, nous introduisons quelques paramètres de notre procédure de reconstruction des surfaces B-splines :

- NI : nombre d'images utilisées.
- NP : nombre de points choisis par image.
- NS : nombre d'intervalles choisis pour les B-splines sur l'axe s .
- NT : nombre d'intervalles choisis pour les B-splines sur l'axe t .
- NV : nombre d'inconnues à estimer.
- NC : nombre de points de contrôle à estimer.
- NL : nombre d'itérations par la méthode Levenberg-Marquardt.

La Figure 2.12 montre une partie d'une sphère observée lors du mouvement de la caméra.

La Figure 2.13 présente des surfaces B-splines reconstruites de la sphère avec les points de contrôle. Les paramètres utilisés pour cette reconstruction sont : NI=8, NP=75, NS=6, NT=2, NV=135, NC=45, NL=3.

La Figure 2.14 présente la superposition des surfaces reconstruites et de la partie de cette sphère observée dans l'espace.

Evidemment, les zones où nous arrivons à reconstruire les surfaces B-splines passent par les lignes épipolaires.

La Figure 2.15 présente les graphes de la courbure gaussienne et moyenne des surfaces reconstruites de la sphère.

La Figure 2.16 montre le résultat de la reconstruction de la sphère en introduisant un bruit blanc gaussien à la fois sur les données x, y avec un écart-type de trois pixels, et sur la rotation avec un écart-type d'un dixième de degré. Les paramètres utilisés pour cette reconstruction sont : NI=8, NP=75, NS=6, NT=2, NV=135, NC=45, NL=3.

Nous constatons que la forme des surfaces B-splines est relativement peu sensible à la présence du bruit.

La Figure 2.17 présente les graphes de la courbure gaussienne et moyenne des surfaces B-splines reconstruites de la sphère avec le bruit blanc gaussien précédent.

Pour estimer la qualité de reconstruction des surfaces, après la superposition de la sphère et des surfaces B-splines reconstruites, nous les coupons avec un plan.

La Figure 2.18 montre les deux courbes d'intersection obtenues après le découpage.

Le graphe de la Figure 2.18 montre bien que les surfaces B-splines reconstruites collent bien avec la sphère. Le graphe droit montre aussi que les surfaces B-splines reconstruites se déforment localement en introduisant un bruit blanc gaussien à la fois sur les données x, y avec un écart-type de trois pixels, et sur la rotation avec un écart-type d'un dixième de degré. La distance maximale entre la courbe sur la sphère et la courbe sur les surfaces B-splines reconstruites est de 0.08 cm.

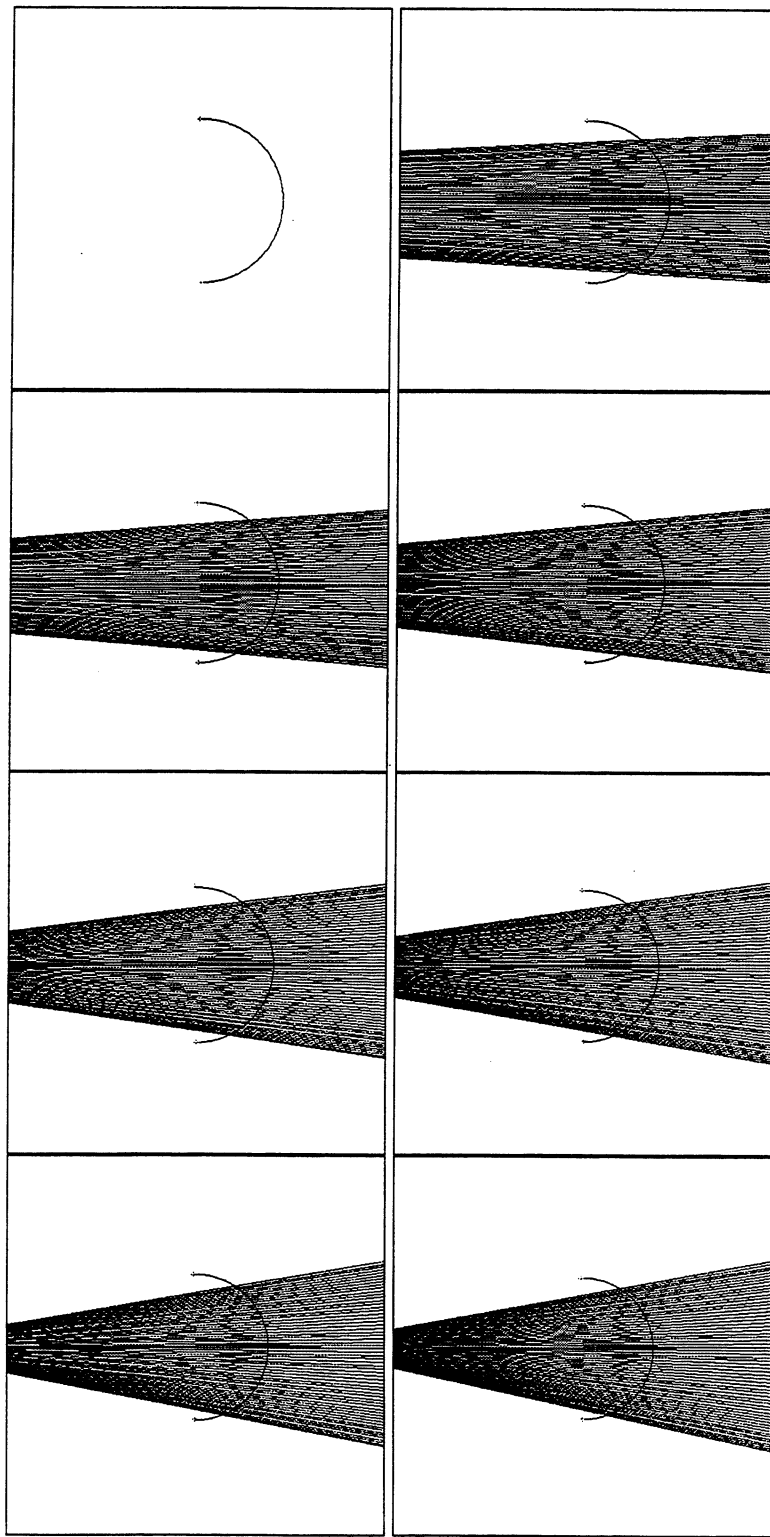


FIG. 2.11 - Une séquence de huit images d'une demi-sphère avec les lignes épipolaires

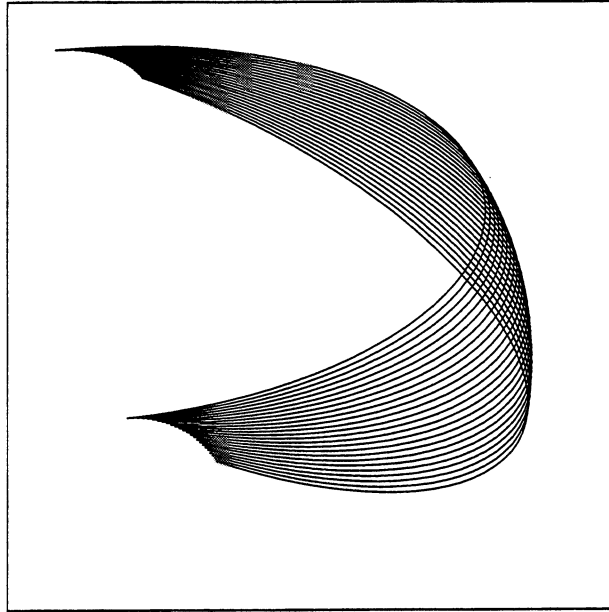


FIG. 2.12 - Partie d'une sphère observée lors du mouvement de la caméra

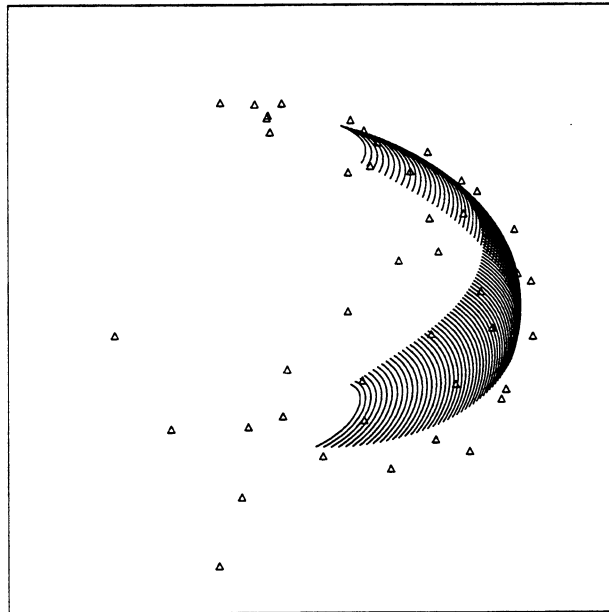


FIG. 2.13 - Surfaces B-splines reconstruites correspondant à la partie d'une sphère observée avec les points de contrôle

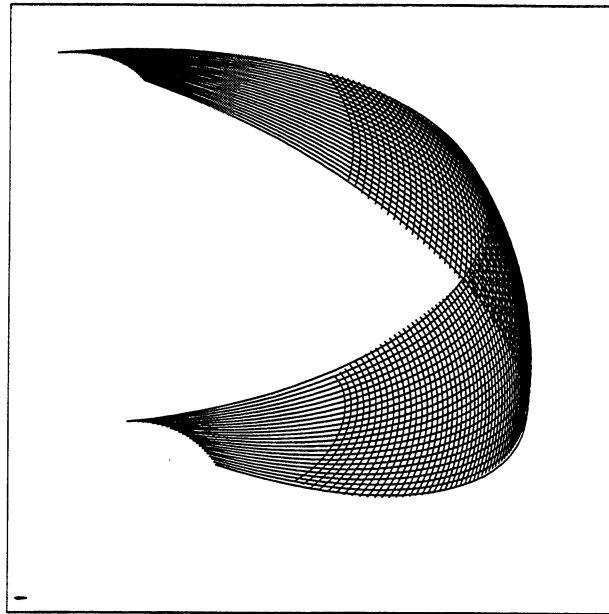


FIG. 2.14 - Superposition des surfaces B-splines reconstruites et de la partie d'une sphère observée

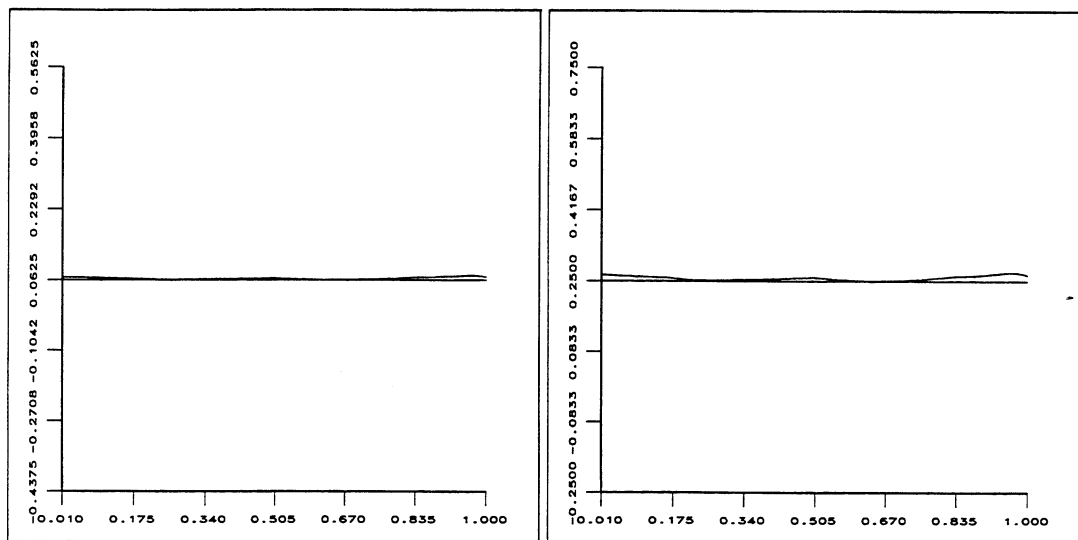


FIG. 2.15 - Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) des surfaces B-splines reconstruites

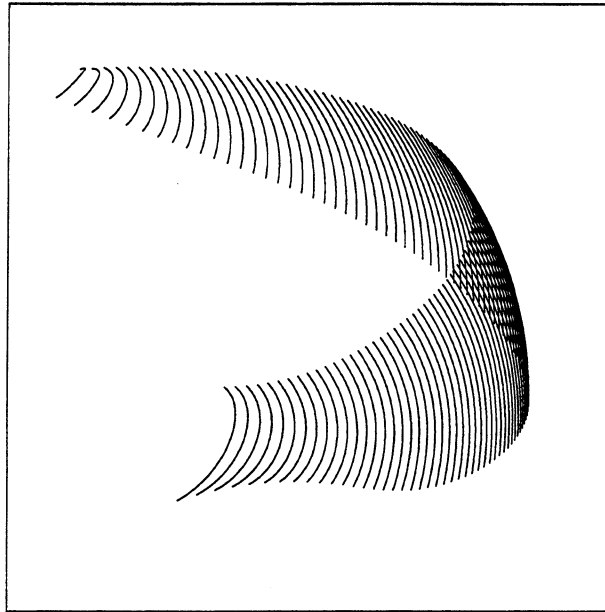


FIG. 2.16 - Surfaces B-splines reconstruites correspondant à la partie d'une sphère observée avec un bruit blanc gaussien

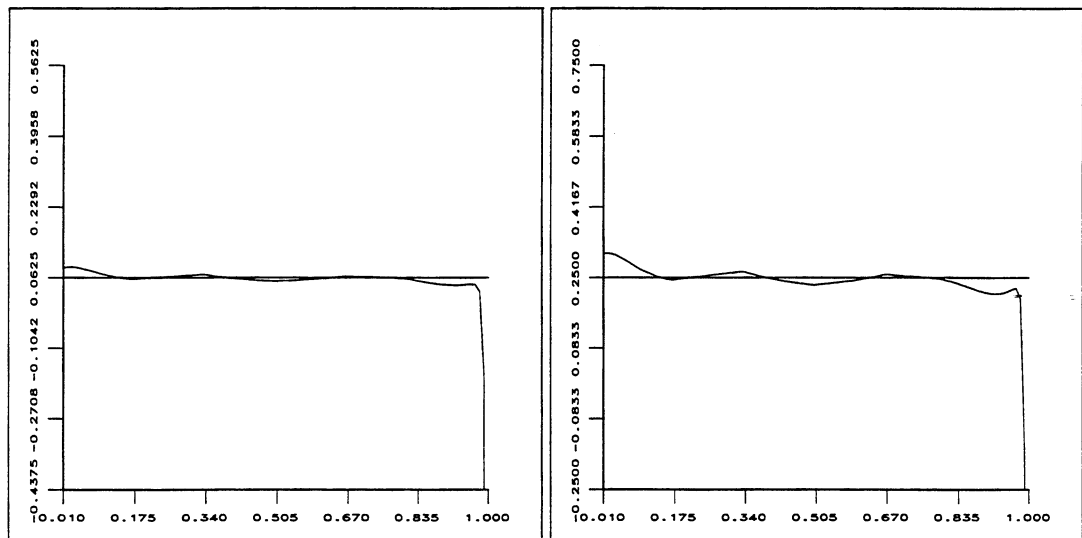


FIG. 2.17 - Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) des surfaces B-splines reconstruites avec un bruit blanc gaussien

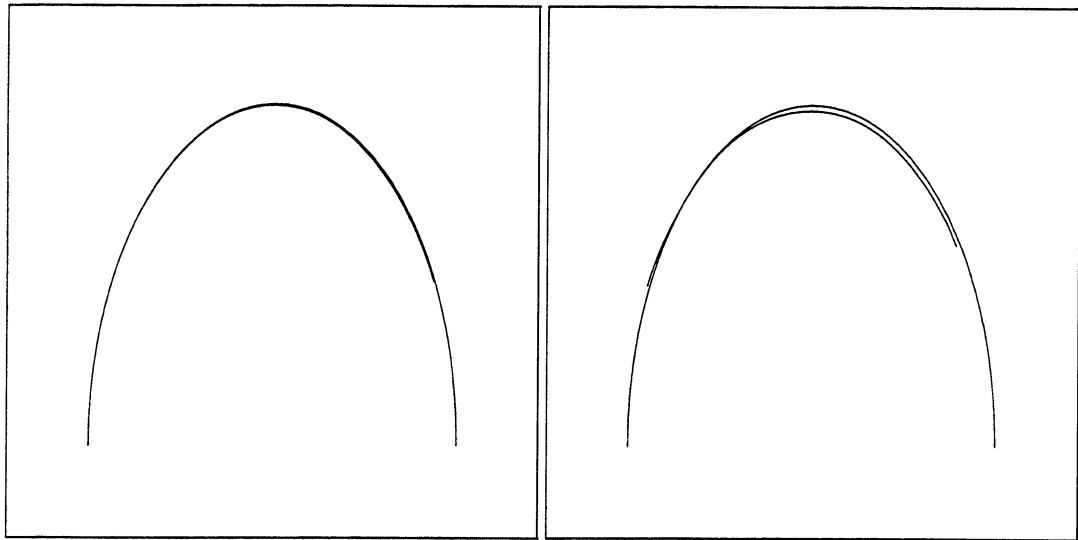


FIG. 2.18 - Courbes d'intersection après le découpage; résultats sur les données parfaites (à gauche); résultats sur les données bruitées (à droite)

Segments

Le segment que nous traitons se trouve sur l'axe Z , il est défini par :

$$z = 8$$

La Figure 2.19 présente la superposition des surfaces B-splines reconstruites d'un segment et du segment réel dans l'espace. Les paramètres utilisés pour cette reconstruction sont : $NI=10$, $NP=80$, $NS=6$, $NT=1$, $NV=108$, $NC=36$, $NL=4$.

Le tableau 2.1 fournit la précision de certaines coordonnées du segment reconstruit.

La Figure 2.20 présente la superposition des surfaces B-splines reconstruites d'un segment et du segment réel dans l'espace en introduisant un bruit blanc gaussien à la fois sur les données x , y avec un écart-type de deux pixels, et sur la rotation avec un écart-type d'un dixième de degré. Les paramètres utilisés pour cette reconstruction sont : $NI=10$, $NP=80$, $NS=8$, $NT=1$, $NV=132$, $NC=44$, $NL=4$.

Le tableau 2.2 montre la précision de certaines coordonnées d'une partie entre le segment reconstruit et le segment réel en introduisant le bruit blanc gaussien.

La Figure 2.21 présente trois sortes de points de contrôle des mêmes surfaces B-splines reconstruites suivant le nombre d'intervalles sur les axes s et t .

Les points de contrôle représentés par des "carrés" dans la Figure 2.21 sont obtenus avec les valeurs suivantes : $NI=10$, $NP=80$, $NS=1$, $NT=1$, $NV=48$, $NC=16$, $NL=5$.

Les points de contrôle représentés par des "triangles" dans la Figure 2.21 sont obtenus avec les valeurs suivantes : $NI=10$, $NP=80$, $NS=10$, $NT=1$, $NV=156$, $NC=52$, $NL=2$.

Les points de contrôle représentés par des "croix" dans la Figure 2.21 sont obtenus avec les valeurs suivantes : $NI=10$, $NP=80$, $NS=15$, $NT=2$, $NV=270$, $NC=90$, $NL=3$.

Nous observons que les points de contrôle s'approchent de plus en plus des surfaces B-splines reconstruites lorsque nous augmentons le nombre d'intervalles des surfaces B-splines sur les axes s et t .

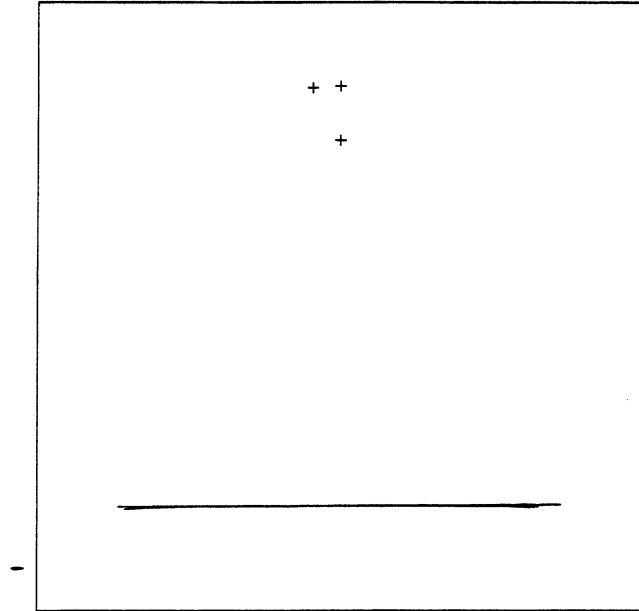


FIG. 2.19 - Superposition des surfaces B-splines reconstruites d'un segment et du segment réel dans l'espace

<i>Coordonnées 3D</i>			<i>Coordonnées reconstruites</i>			<i>Erreurs</i>	
X	Y	Z	X_r	Y_r	Z_r	$Y_r - Y$	$Z_r - Z$
-5.81818	0.00000	8.00000	-5.92024	-0.00154	8.00462	-0.00154	0.00462
-5.89899	0.00000	8.00000	-5.95573	-0.00128	8.00400	-0.00128	0.00400
-5.97980	0.00000	8.00000	-5.99201	-0.00097	8.00330	-0.00097	0.00330
-6.06061	0.00000	8.00000	-6.02909	-0.00062	8.00251	-0.00062	0.00251
-6.14141	0.00000	8.00000	-6.06694	-0.00022	8.00164	-0.00022	0.00164
-6.22222	0.00000	8.00000	-6.10554	0.00023	8.00069	0.00023	0.00069
-6.30303	0.00000	8.00000	-6.14489	0.00073	7.99965	0.00073	-0.00035
-6.38384	0.00000	8.00000	-6.18497	0.00128	7.99852	0.00128	-0.00148
-6.46465	0.00000	8.00000	-6.22576	0.00189	7.99732	0.00189	-0.00268
-6.54545	0.00000	8.00000	-6.26724	0.00256	7.99603	0.00256	-0.00397

TAB. 2.1 - Certaines coordonnées d'une partie d'un segment reconstruit 3D et des erreurs

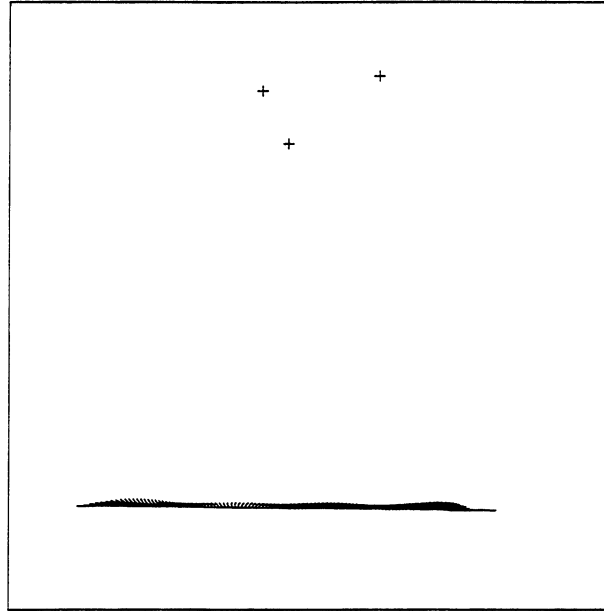


FIG. 2.20 - Superposition des surfaces B-splines reconstruites d'un segment et du segment réel dans l'espace avec un bruit blanc gaussien

<i>Coordonnées 3D</i>			<i>Coordonnées reconstruites</i>			<i>Erreurs</i>	
X	Y	Z	X_r	Y_r	Z_r	$Y_r - Y$	$Z_r - Z$
-5.17172	0.00000	8.00000	-5.25417	-0.00233	7.95953	-0.00233	-0.04047
-5.25253	0.00000	8.00000	-5.27172	-0.00182	7.95823	-0.00182	-0.04177
-5.33333	0.00000	8.00000	-5.29030	-0.00129	7.95686	-0.00129	-0.04314
-5.41414	0.00000	8.00000	-5.30991	-0.00072	7.95543	-0.00072	-0.04457
-5.49495	0.00000	8.00000	-5.33054	-0.00013	7.95394	-0.00013	-0.04606
-5.57576	0.00000	8.00000	-5.35216	0.00048	7.95239	0.00048	-0.04761
-5.65657	0.00000	8.00000	-5.37476	0.00112	7.95078	0.00112	-0.04922
-5.73737	0.00000	8.00000	-5.39833	0.00179	7.94912	0.00179	-0.05088
-5.81818	0.00000	8.00000	-5.42285	0.00247	7.94741	0.00247	-0.05259
-5.89899	0.00000	8.00000	-5.44831	0.00318	7.94564	0.00318	-0.05436

TAB. 2.2 - Certaines coordonnées d'une partie d'un segment reconstruit 3D et des erreurs avec un bruit blanc gaussien

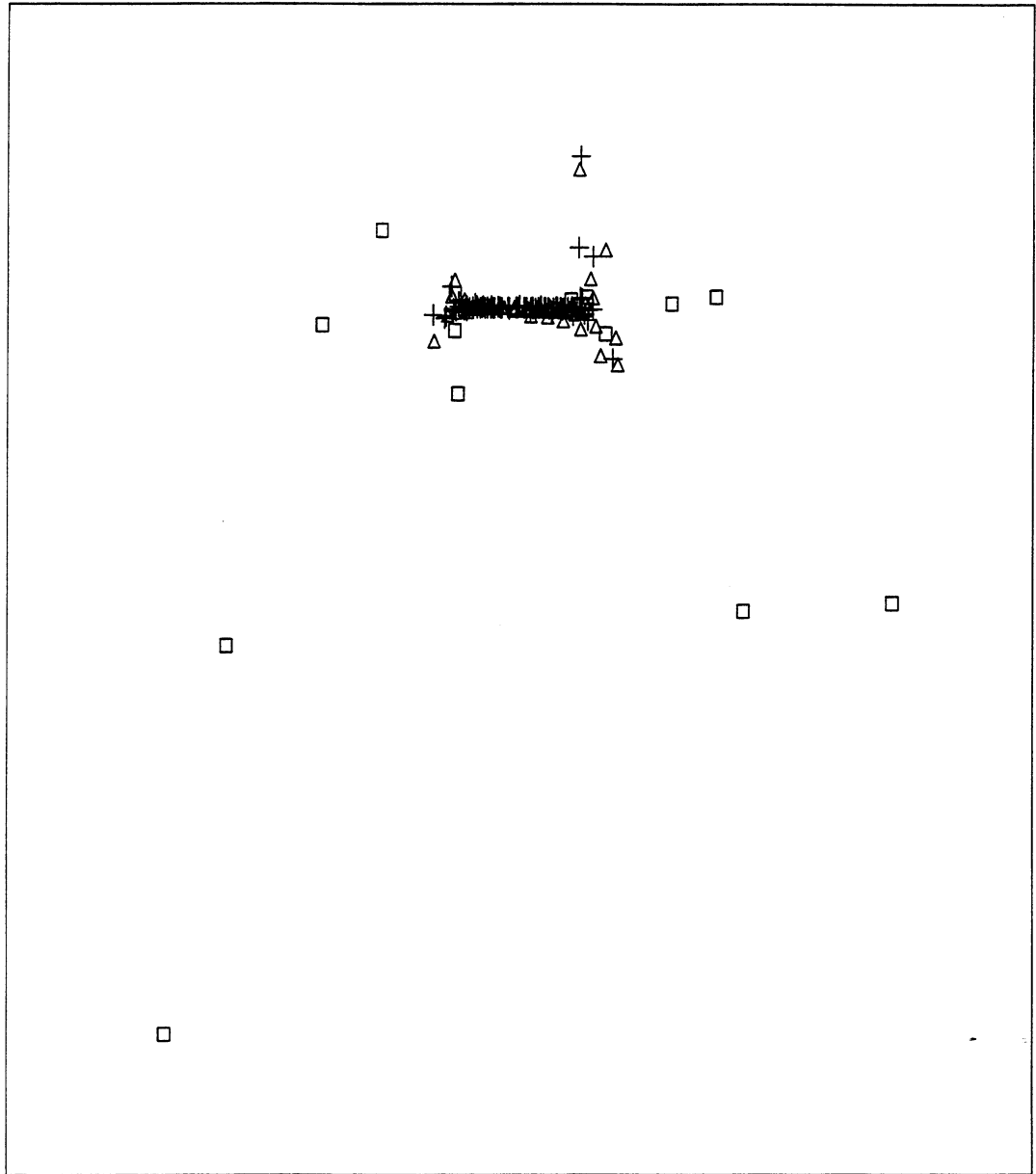


FIG. 2.21 - Points de contrôle des mêmes surfaces B-splines avec les différents nombres d'intervalles choisis sur les axes s et t

Cylindres

La Figure 2.22 montre une partie d'un cylindre observée lors du mouvement de la caméra.

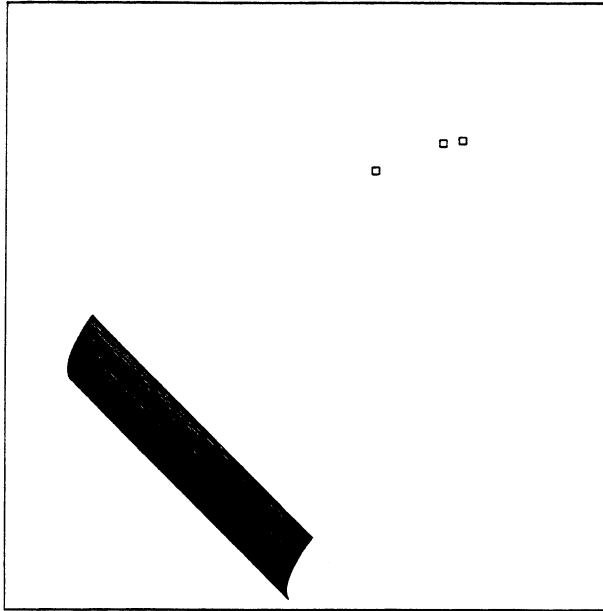


FIG. 2.22 - Partie d'un cylindre observée par la caméra

La Figure 2.23 présente la reconstruction des surfaces B-splines d'un cylindre dans l'espace. Les paramètres utilisés pour cette reconstruction sont : $NI=10$, $NP=60$, $NS=8$, $NT=2$, $NV=165$, $NC=55$, $NL=3$.

Hyperboloïdes

La Figure 2.24 montre une partie d'un hyperboloïde observée lors du mouvement de la caméra.

La Figure 2.25 présente la superposition des surfaces B-splines reconstruites d'un hyperboloïde et de la partie de cet hyperboloïde observée dans l'espace. Les paramètres utilisés pour cette reconstruction sont : $NI=10$, $NP=60$, $NS=8$, $NT=2$, $NV=165$, $NC=55$, $NL=3$.

La Figure 2.26 présente deux surfaces B-splines reconstruites d'un hyperboloïde avec deux différents mouvements de la caméra dans l'espace.

Incertitude des points de contrôle des surfaces B-splines

Pour estimer l'incertitude des points de contrôle des surfaces, nous obtenons la matrice de covariance à partir de la procédure de Levenberg-Marquardt. La Figure 2.27 présente l'incertitude des points de contrôle des surfaces B-splines reconstruites d'un segment. Chaque parallélépipède dans la Figure 2.27 correspond à l'incertitude d'un point de contrôle. Les paramètres utilisés pour cette reconstruction sont : $NI=10$, $NP=80$, $NS=15$, $NT=2$, $NV=270$, $NC=90$, $NL=3$.

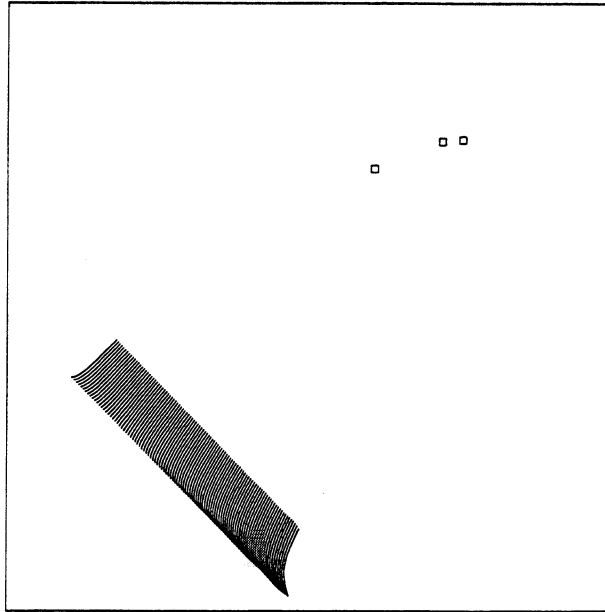


FIG. 2.23 - Surfaces B-splines reconstruites d'un cylindre

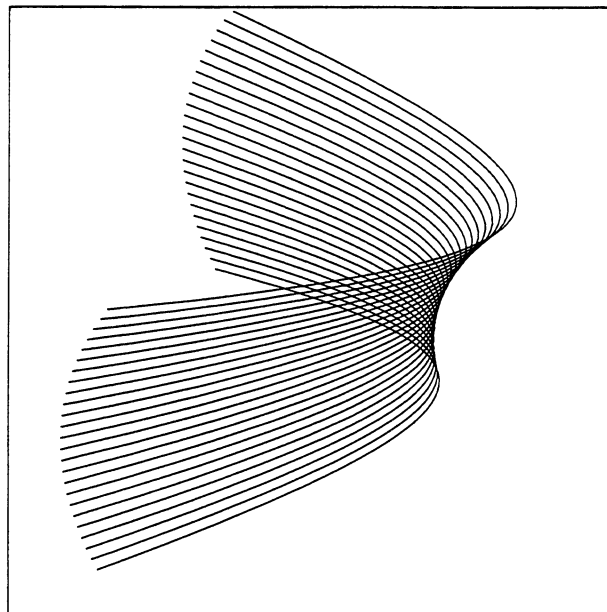


FIG. 2.24 - Partie d'un hyperboloïde observée lors du mouvement de la caméra

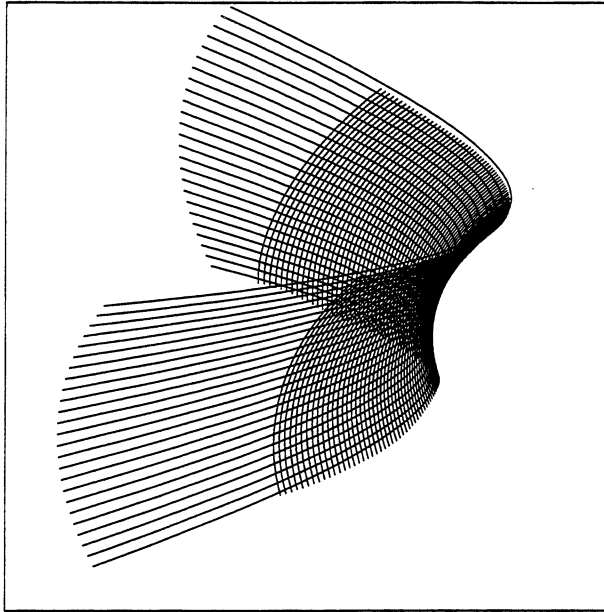


FIG. 2.25 - Superposition des surfaces B-splines reconstruites et de la partie d'un hyperboloïde observée

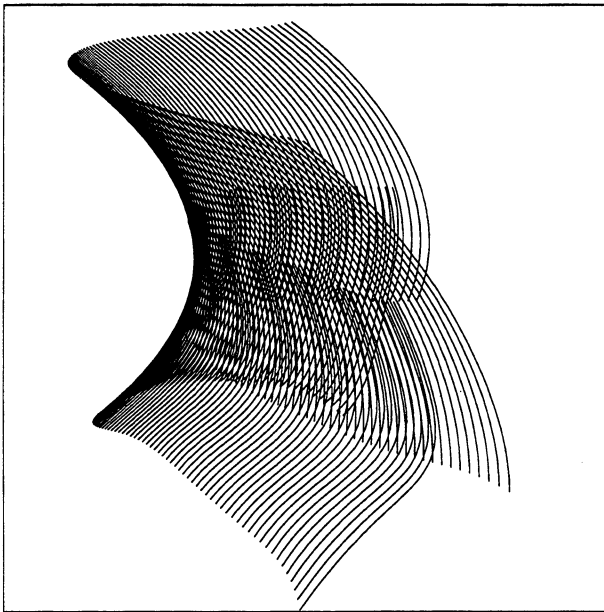


FIG. 2.26 - Deux surfaces B-splines reconstruites d'un hyperboloïde avec deux différents mouvements de la caméra

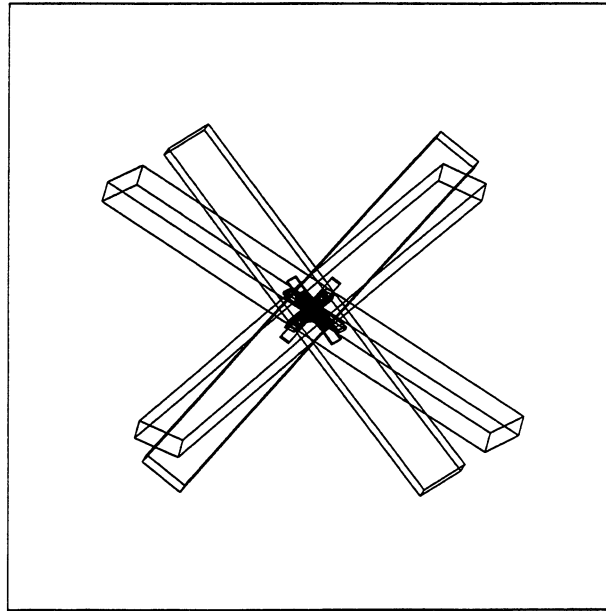


FIG. 2.27 - Incertitude des points de contrôle des surfaces B-splines reconstruites d'un segment

La Figure 2.28 présente l'incertitude des points de contrôle des surfaces B-splines reconstruites d'une sphère. Chaque parallélépipède dans la Figure 2.28 correspond à l'incertitude d'un point de contrôle. Les paramètres utilisés pour cette reconstruction sont : $NI=8$, $NP=75$, $NS=6$, $NT=2$, $NV=135$, $NC=45$, $NL=3$.

Après une vérification de la position des points de contrôle dans l'espace, nous nous apercevons que les quatre grands parallélépipèdes correspondent bien aux quatre points de contrôle qui se trouvent près de quatre sommets des surfaces reconstruites. Par conséquent, les autres points de contrôle sont relativement stables.

Objets composés par deux surfaces de révolution

Les objets entrant dans la catégorie des surfaces de révolution sont souvent utilisés en matière de la modélisation d'objets 3D. Ces surfaces ont des propriétés très intéressantes. Géométriquement, elles sont engendrées par la rotation d'une courbe-plane autour d'un axe [Car 76]. Sans perte de généralité, nous supposons que l'axe Z soit l'axe de rotation, la courbe plane C se trouve sur le plan XZ . Elle est définie par :

$$x = f(v), \quad z = g(v), \quad a < v < b, \quad f(v) > 0$$

Nous notons que u est l'angle de rotation par rapport à l'axe Z . Alors la surface de révolution S est définie par :

$$\begin{aligned} \vec{X} : U &\longrightarrow S \\ (u, v) &\longrightarrow (f(v)\cos(u), f(v)\sin(u), g(v)) \end{aligned} \quad (2.66)$$

où U est un ensemble ouvert défini par :

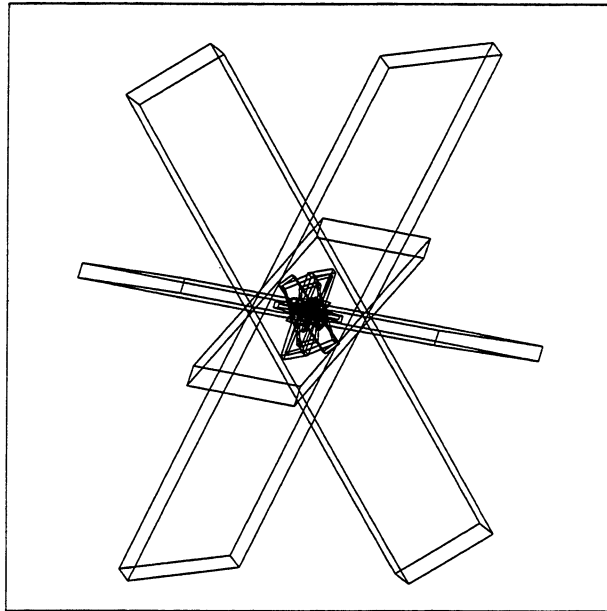


FIG. 2.28 - Incertitude des points de contrôle des surfaces B-splines reconstruites d'une sphère

$$U = \{(u, v) \in \mathbb{R}^2; 0 < u < 2\pi, a < v < b\}$$

La courbe C est appelée *génératrice* de S . Les cercles décrits par les points de C sont appelés *parallèles* de S . Les différentes positions de C sur S sont appelées *méridiens* de S .

L'objet que nous traitons ici est composé par une surface de révolution d'ellipsoïde et une surface de révolution d'hyperboloïde.

La Figure 2.29 présente un des contours occultants de cet objet dans la séquence.

La Figure 2.30 présente la reconstruction dans l'espace des surfaces B-splines. Les paramètres utilisés pour cette reconstruction sont : $NI=8$, $NP=75$, $NS=4$, $NT=2$, $NV=105$, $NC=35$, $NL=4$.

Pour estimer la qualité de reconstruction des surfaces, après la superposition de la surface composée et des surfaces B-splines reconstruites, nous les coupons avec un plan.

La Figure 2.31 montre les courbes d'intersection obtenues après le découpage.

La Figure 2.32 présente les graphes de la courbure gaussienne et moyenne des surfaces B-splines reconstruites de cet objet.

Nous remarquons facilement dans la Figure 2.32 qu'il y a une variation assez forte de courbure sur l'endroit de jonction entre l'ellipsoïde et l'hyperboloïde. De fait il y a une discontinuité dans les dérivées secondes. En revanche, les surfaces reconstruites sont continues et on voit l'effet de raccordement obtenu.

Conséquences d'un suivi de contours erroné

Dans ce paragraphe, nous traitons un cas intéressant; nous supposons que nous observons deux cylindres parallèles avec une caméra calibrée (voir Figure 2.33). Au

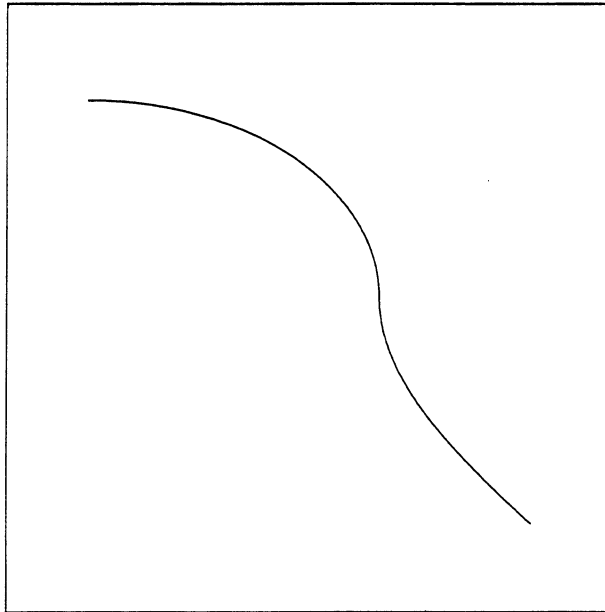


FIG. 2.29 - Contour d'occultation de l'objet composé par un ellipsoïde et un hyperboloïde

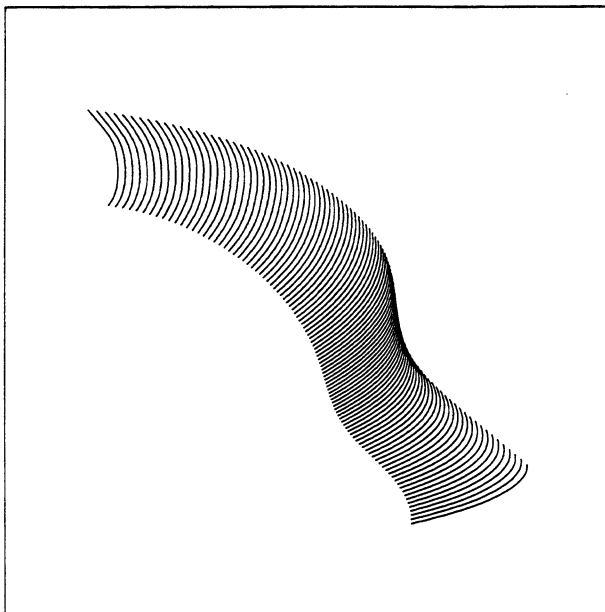


FIG. 2.30 - Surfaces B-splines reconstruites lors du mouvement de la caméra

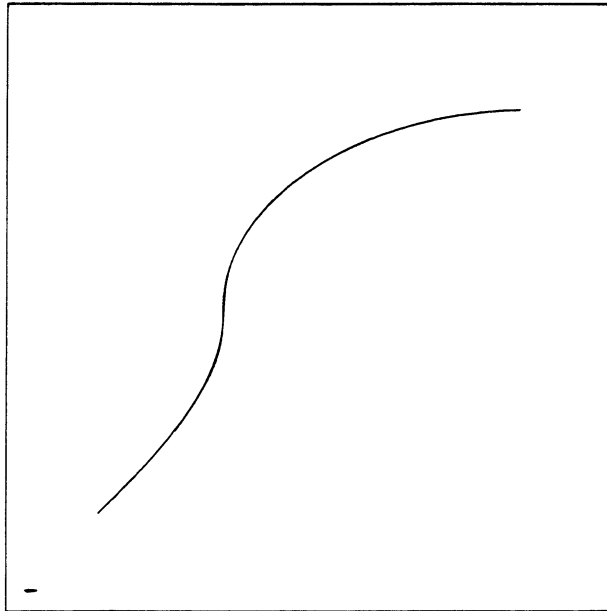


FIG. 2.31 - Courbes d'intersection obtenues après le découpage

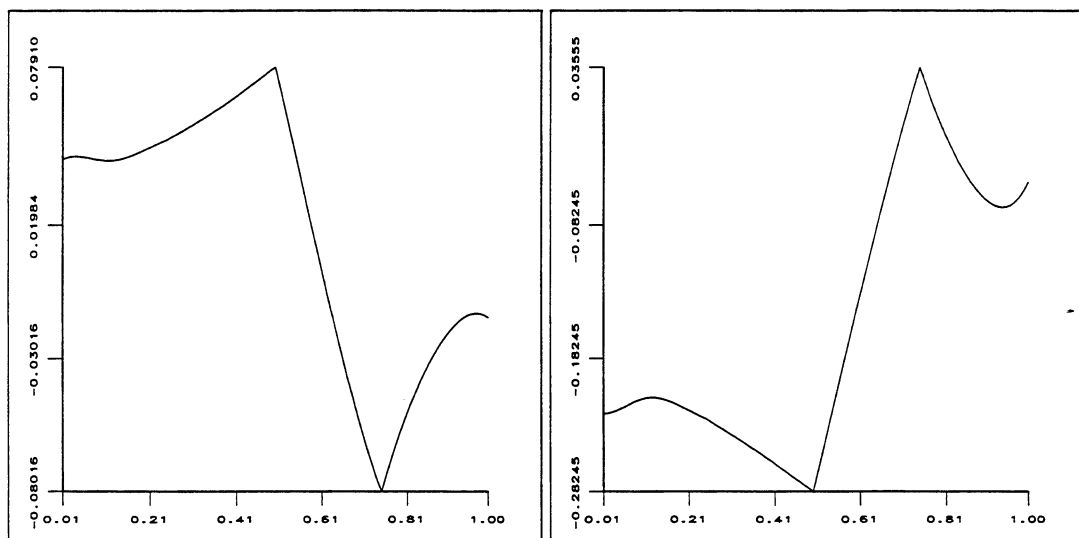


FIG. 2.32 - Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) des surfaces B-splines reconstruites de l'objet composé par un ellipsoïde et un hyperboloïde

début, la caméra observe les contours occultants du premier cylindre, au bout d'un certain temps, elle observe ceux du deuxième cylindre.

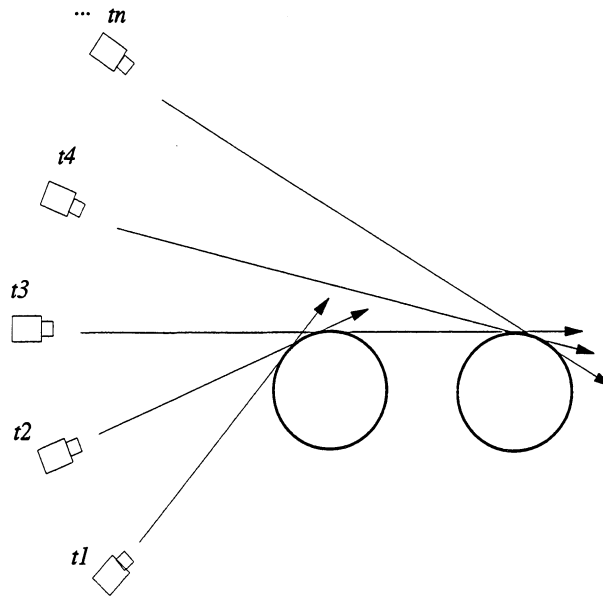


FIG. 2.33 - Mouvement de la caméra autour de deux cylindres parallèles

La Figure 2.34 présente la superposition des surfaces B-splines reconstruites et de deux portions de cylindres observées dans l'espace.

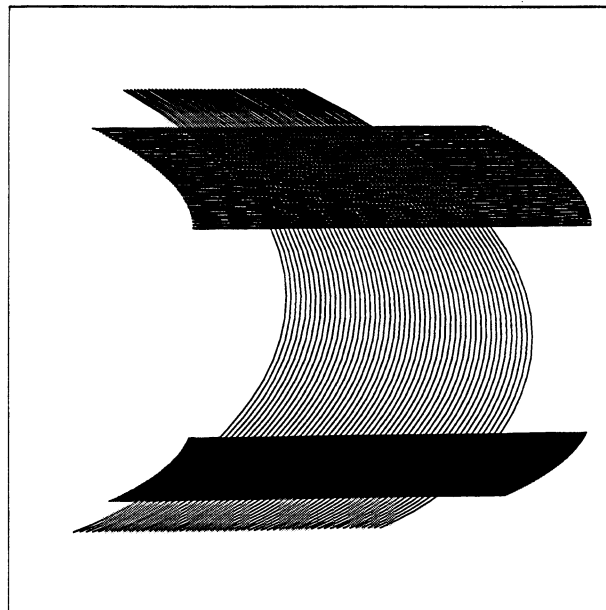


FIG. 2.34 - Superposition des surfaces B-splines reconstruites et des portions observées de deux cylindres lors du mouvement de la caméra

Pour estimer la qualité de reconstruction des surfaces, après la superposition de

deux portions de cylindres observées et des surfaces B-splines reconstruites, nous les coupons avec un plan. La Figure 2.35 montre les courbes d'intersection obtenues après le découpage.

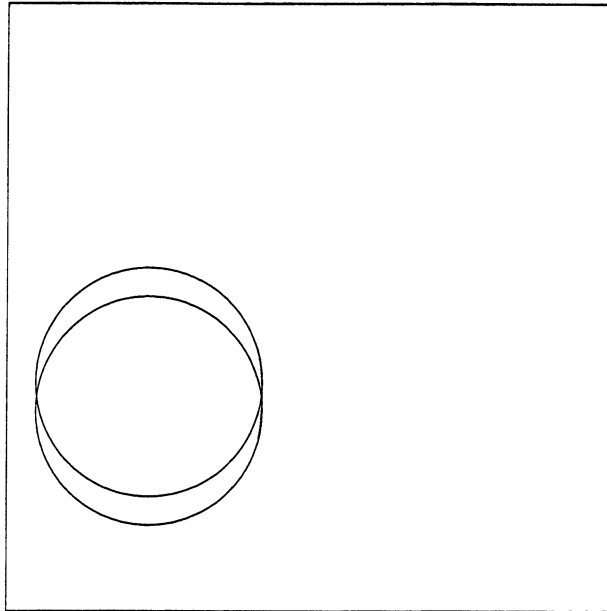


FIG. 2.35 - Courbes d'intersection obtenues après le découpage

Puisque nous introduisons une régularisation sur les surfaces tridimensionnelles, cela nous permet de compléter les zones où les données manquent. Les résultats de reconstruction des surfaces sont satisfaisants. Cependant, si ces deux cylindres sont assez éloignés, notre système d'équations non linéaires ne converge pas.

Que pourrions nous faire? En fait, nous envisageons actuellement d'appliquer une nouvelle paramétrisation de la surface. Le paramètre s est toujours déduit de l'abscisse curviligne du contour occultant. Mais pour t , il est déterminé par la distance tridimensionnelle tel que le montre la Figure 2.36.

La Figure 2.37 montre les courbes d'intersection obtenues après le découpage. Nous nous apercevons que la surface reconstruite entre deux cylindres reste plane et tangente aux cylindres lorsque ils s'éloignent. Cela nous montre que le nouveau type de paramétrisation de la surface est efficace dans ce cas.

2.7.2 Données réelles

Nous avons testé notre algorithme avec des données réelles. Toutes nos expériences sont effectuées avec une caméra 765 Pulnix, un objectif avec une focale de 18mm et une carte de FG150 de "Imaging-Technology".

Le modèle de notre caméra est supposé parfaitement sténopé. Donc nous n'avons pas corrigé la distorsion de la caméra. Or le problème du calibrage est difficile. En pratique, la précision de la matrice projective (translation et rotation) n'atteint pas celle que nous voulons. Donc nous avons utilisé les points stables de la scène afin de calculer la bonne matrice projective.

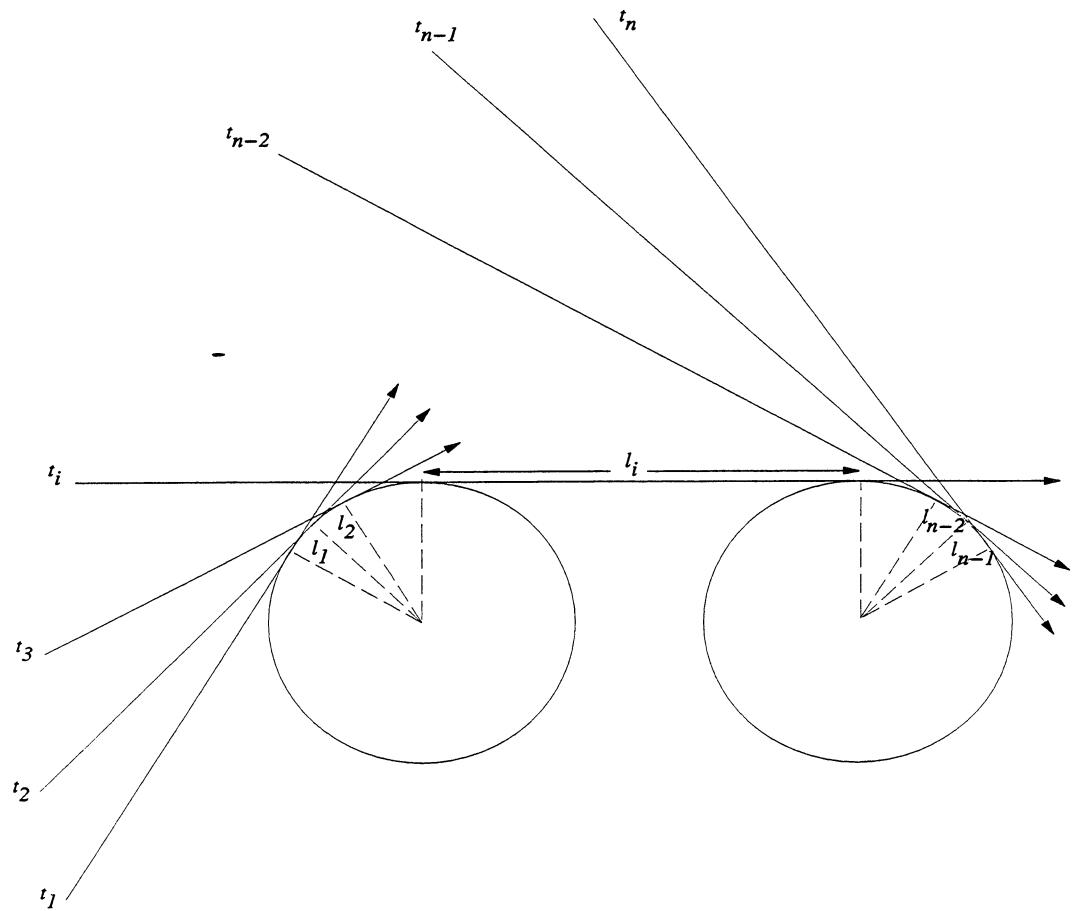


FIG. 2.36 - Une nouvelle paramétrisation de la surface. Le paramètre t est déterminé à partir de la distance tridimensionnelle entre deux contours consécutifs. l_i présente la distance de deux contours entre l'instant t_i et t_{i+1} .

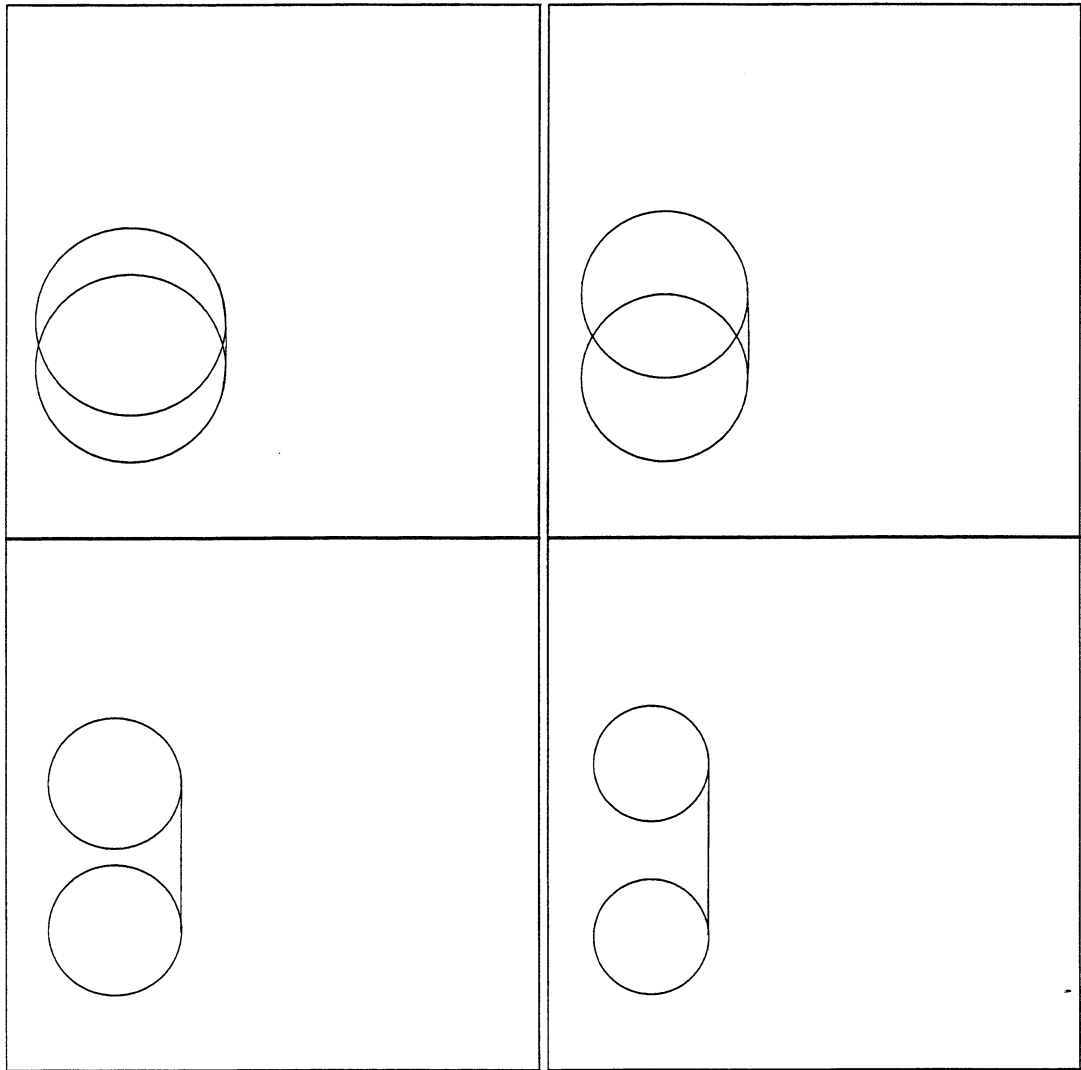


FIG. 2.37 - Courbes d'intersection obtenues après le découpage

Carafe avec un cube transparent

Nous avons fabriqué un cube transparent, et mis une carafe à l'intérieur (voir Figure 2.38). Le cube est composé par douze barres du fer et huit balles de ping-pong. Les balles de ping-pong sont colorées en noir. La longueur de chaque barre est environ de 50.0 cm. Les contours de l'image sont obtenus à l'aide de l'extracteur de contours de Deriche [Der 87]. Ils sont filtrés par le lissage des B-splines cubiques régularisées.

Les coins des feuilles de papier polyédriques de la scène sont détectés par une méthode développée dans notre laboratoire. Ces points sont des primitives robustes de l'image. Nous avons effectué les mises en correspondance de ces points par une technique standard de corrélation. Pour calculer le centre de chaque balle, nous considérons le point d'intersection des axes médians de chaque barre. Finalement, nous avons choisi un repère absolu attaché au cube transparent, et les bonnes matrices projectives de la séquence sont alors obtenues. Cette méthode a été proposée par Mohr.

La Figure 2.38 ne montre que la première et la dernière image de la séquence de dix images de notre carafe. Le mouvement de la caméra est quelconque. Le déplacement entre deux images successives est relativement petit. La rotation totale entre la première et la dernière image est environ de 40 degrés.

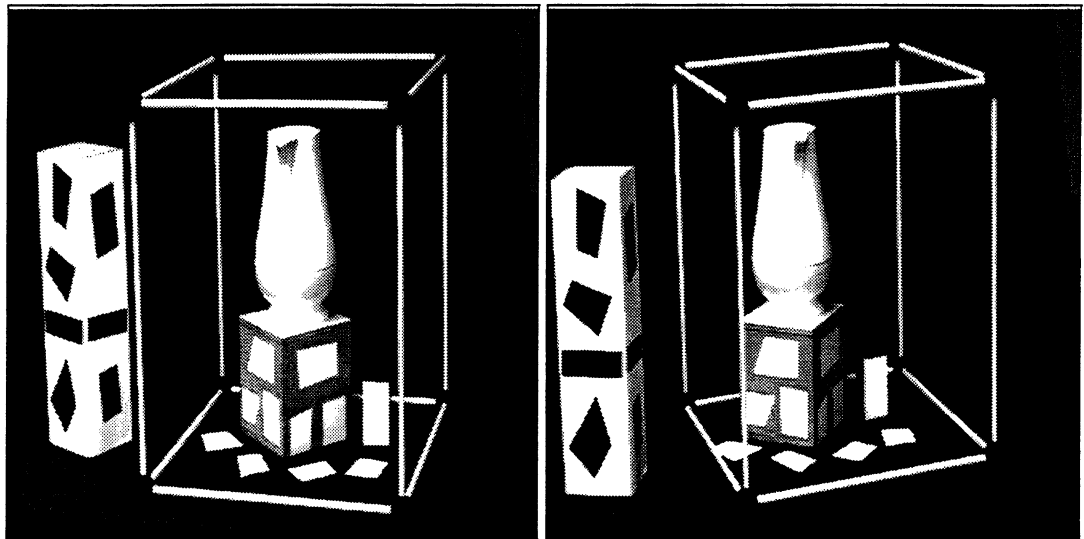


FIG. 2.38 - Première (à gauche) et dernière (à droite) image d'une séquence d'images de la carafe

La Figure 2.39 montre les surfaces B-splines reconstruites de la carafe en utilisant dix images. Les paramètres utilisés pour cette reconstruction sont : $NI=10$, $NP=70$, $NS=3$, $NT=1$, $NV=72$, $NC=24$, $NL=3$. En particulier, le temps total de ces trois itérations est 99 secondes sur une machine SunSparc10.

Nous remarquons dans la Figure 2.40 que la reconstruction des surfaces est qualitativement bonne.

La Figure 2.41 présente les graphes de la courbure gaussienne et moyenne des surfaces B-splines reconstruites de la carafe.

Nous notons dans la Figure 2.41 que les valeurs de la courbure gaussienne de certains points sur les surfaces B-splines reconstruites sont négatives. Ceci correspond aux

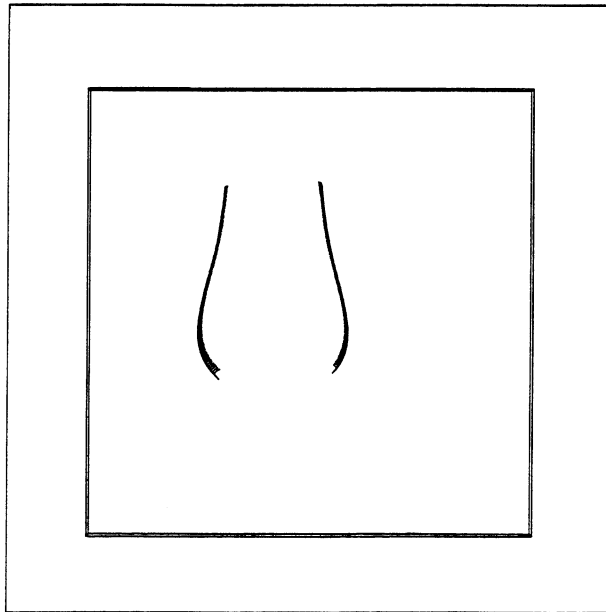


FIG. 2.39 - Surfaces B-splines reconstruites avec le cube transparent

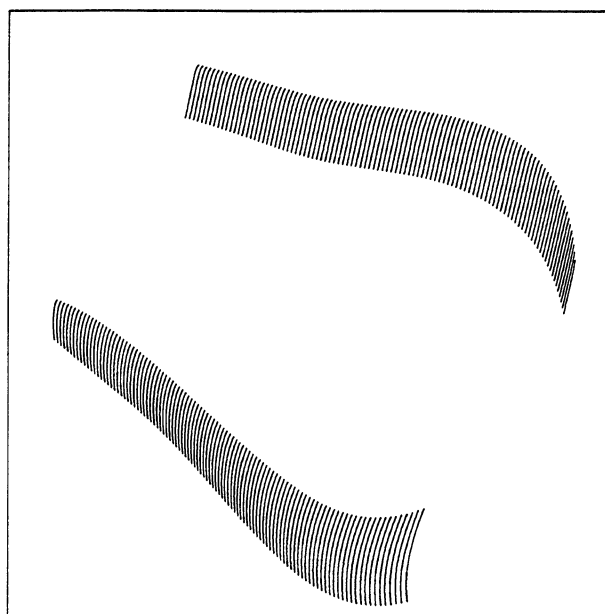


FIG. 2.40 - Une vue des surfaces B-splines reconstruites sans le cube transparent

parties elliptiques et hyperboliques de la carafe.

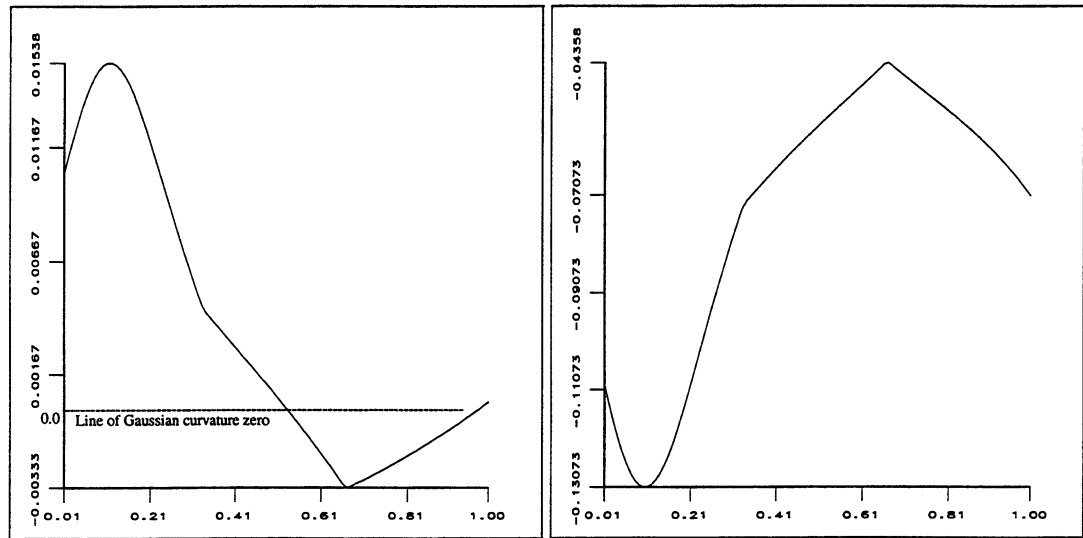


FIG. 2.41 - Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) de surfaces B-splines reconstruites de notre carafe

La Figure 2.42 montre deux graphes superposés, représentant chacun la valeur de la courbure gaussienne pour deux surfaces différentes. Les deux surfaces choisies correspondant à la reconstruction de deux côtés opposés de la carafe. Nous observons sur cette figure que les deux graphes sont de formes identiques. Nous pouvons donc en déduire que ces surfaces B-splines reconstruites sont qualitativement de la même forme. Comme nous l'avons vu (Figures 2.31 et 2.32), il est impossible d'inférer une valeur quantitative sur la précision de la reconstruction à partir de ces courbures.

Pour estimer quantitativement la qualité de notre reconstruction de surface, nous avons comparé le diamètre calculé et le diamètre réel de la carafe. Les diamètres calculés sont obtenus en utilisant les surfaces reconstruites de deux côtés opposés de la carafe. Les diamètres réels de la carafe à tous les niveaux ont été mesurés avec un pied à coulisse.

La table 2.3 suivante montre les résultats obtenus en prenant les diamètres calculés et les diamètres mesurés de la carafe. Les erreurs proviennent de plusieurs sources : la régularisation de la surface reconstruite de la carafe, les incertitudes relatives à la localisation des points de contours, et l'imprécision des calculs sur les matrices de projection perspective.

Tube avec un cube transparent

La Figure 2.43 ne montre que la première et la dernière image de la séquence de dix images de notre tube. Le déplacement entre deux images successives est assez faible.

La Figure 2.44 montre les surfaces B-splines reconstruites du tube en utilisant dix images.

Les paramètres utilisés pour cette reconstruction sont : $NI=10$, $NP=75$, $NS=4$, $NT=2$, $NV=105$, $NC=35$, $NL=3$.

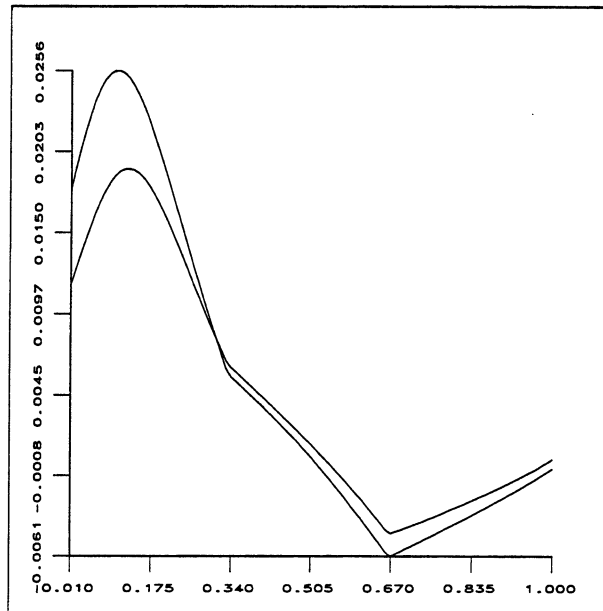


FIG. 2.42 - Superposition de deux graphes de la courbure gaussienne des surfaces B-splines reconstruites de notre carafe

<i>Hauteur</i>	<i>Diamètre réel</i>	<i>Diamètre calculé</i>	<i>Erreur</i>
60.0	144.0	139.2	4.8
70.0	148.5	143.1	5.4
80.0	150.2	147.5	2.7
90.0	148.5	149.2	-0.7
100.0	146.0	148.8	-2.8
110.0	142.0	146.6	-4.6
120.0	139.5	143.9	-4.4
130.0	135.0	140.3	-5.3

TAB. 2.3 - Erreurs sur les diamètres de notre carafe reconstruite (en *mm*)

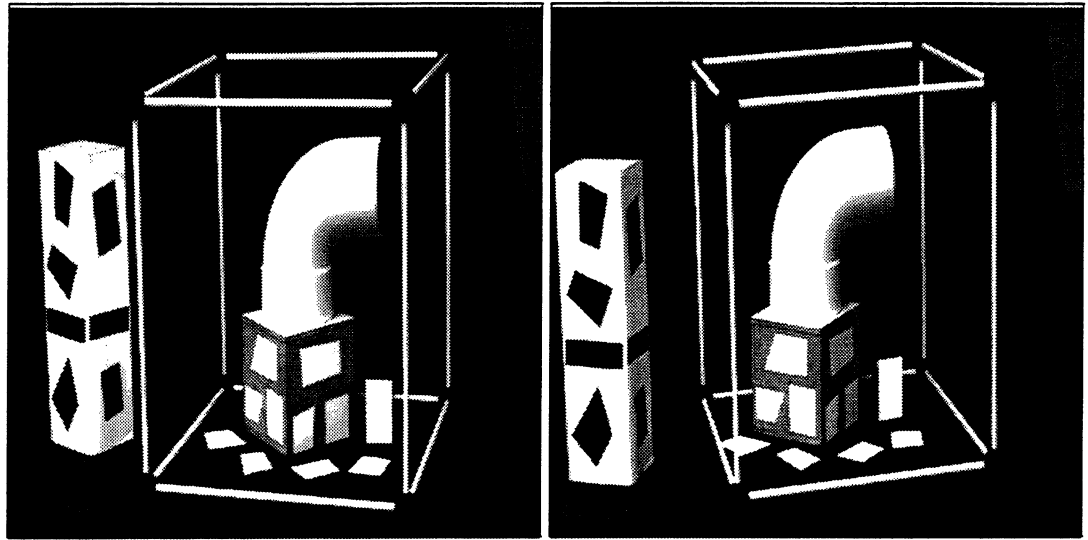


FIG. 2.43 - Première (à gauche) et dernière (à droite) image d'une séquence d'images d'un tube

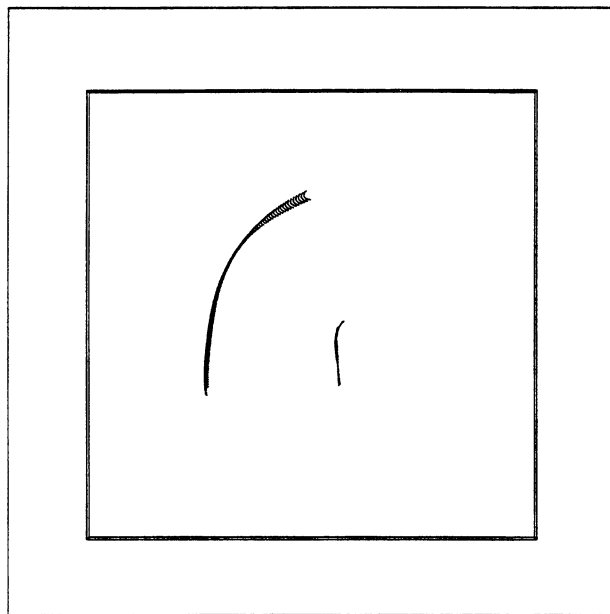


FIG. 2.44 - Surfaces B-splines reconstruites avec le cube transparent

Nous remarquons dans la Figure 2.45 que la reconstruction des surfaces est qualitativement bonne.

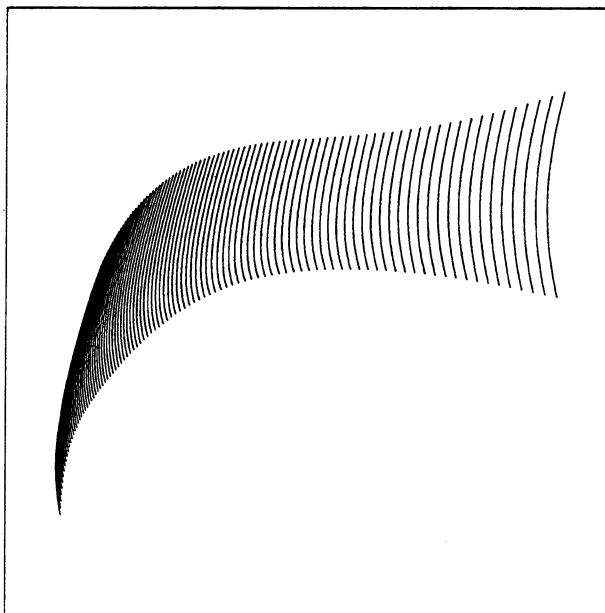


FIG. 2.45 - Une vue des surfaces B-splines reconstruites sans le cube transparent

Nous notons aussi qu'il y a une rupture sur ce tube, mais nous n'arrivons pas à caractériser ce phénomène par notre reconstruction des surfaces.

Dans la Figure 2.46, après un zoom sur les pixels de cette zone de rupture, nous nous apercevons que la localisation du détecteur de contours est insuffisante. Une solution à ce problème serait d'utiliser un détecteur de contours avec localisation "sub-pixel", mais nous n'en avons malheureusement pas à notre disposition.

La Figure 2.47 présente les graphes de la courbure gaussienne et moyenne des surfaces B-splines reconstruites du tube.

Carafe avec deux cubes

Dans ce paragraphe, nous voulons reconstruire la surface complète de la carafe lorsque notre caméra fait un tour complet autour de l'objet.

Pour cela, notre carafe est posée sur deux cubes dont les sommets sont traités comme des points de référence et ainsi la scène dispose d'un repère absolu. La longueur du petit cube est environ de 25.0 *cm*. La longueur du grand cube est environ de 35.0 *cm*. La scène contient des papiers noirs ou blancs pour calibrer la caméra en mouvement. Le contraste entre l'objet et l'arrière-plan est important pour avoir de bons contours dans les images.

La Figure 2.48 présente six images extraites de la séquence complète de la carafe. Le nombre d'images de notre séquence est de 150. La rotation entre deux images successives de la séquence est environ de 2,4 degrés.

La Figure 2.49 montre un ensemble de surfaces B-splines reconstruites de la carafe. Sur cette figure, nous voyons que la surface de la carafe est pratiquement reconstruite. Nous pouvons distinguer les différentes surfaces qui ont servi à cette reconstruction.

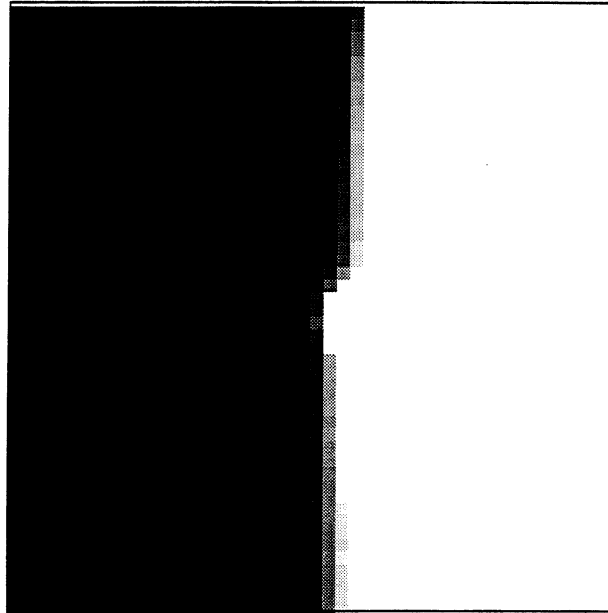


FIG. 2.46 - Zoom dans la zone de rupture du tube

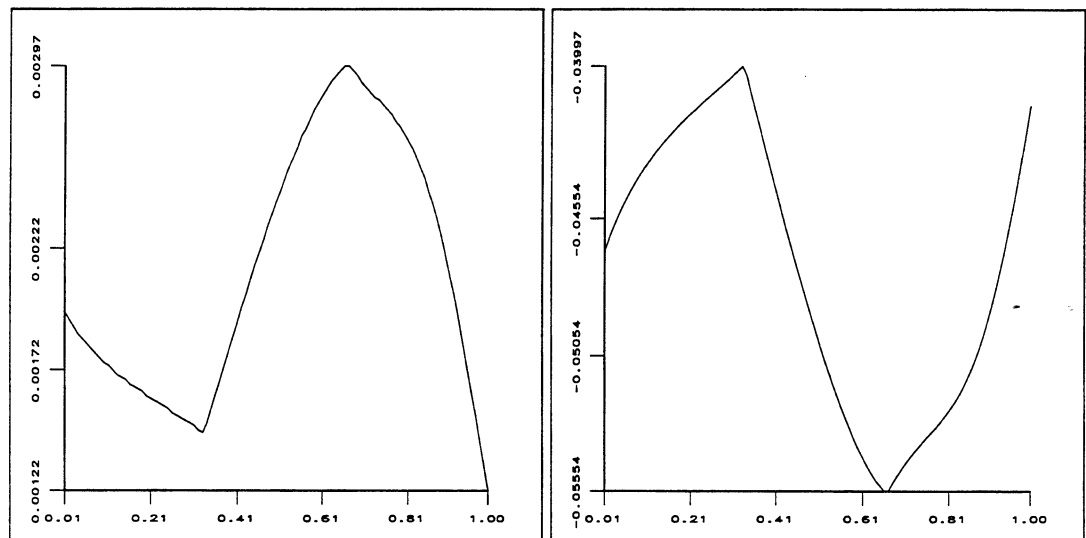


FIG. 2.47 - Graphes de la courbure gaussienne (à gauche) et moyenne (à droite) des surfaces B-splines reconstruites de notre tube

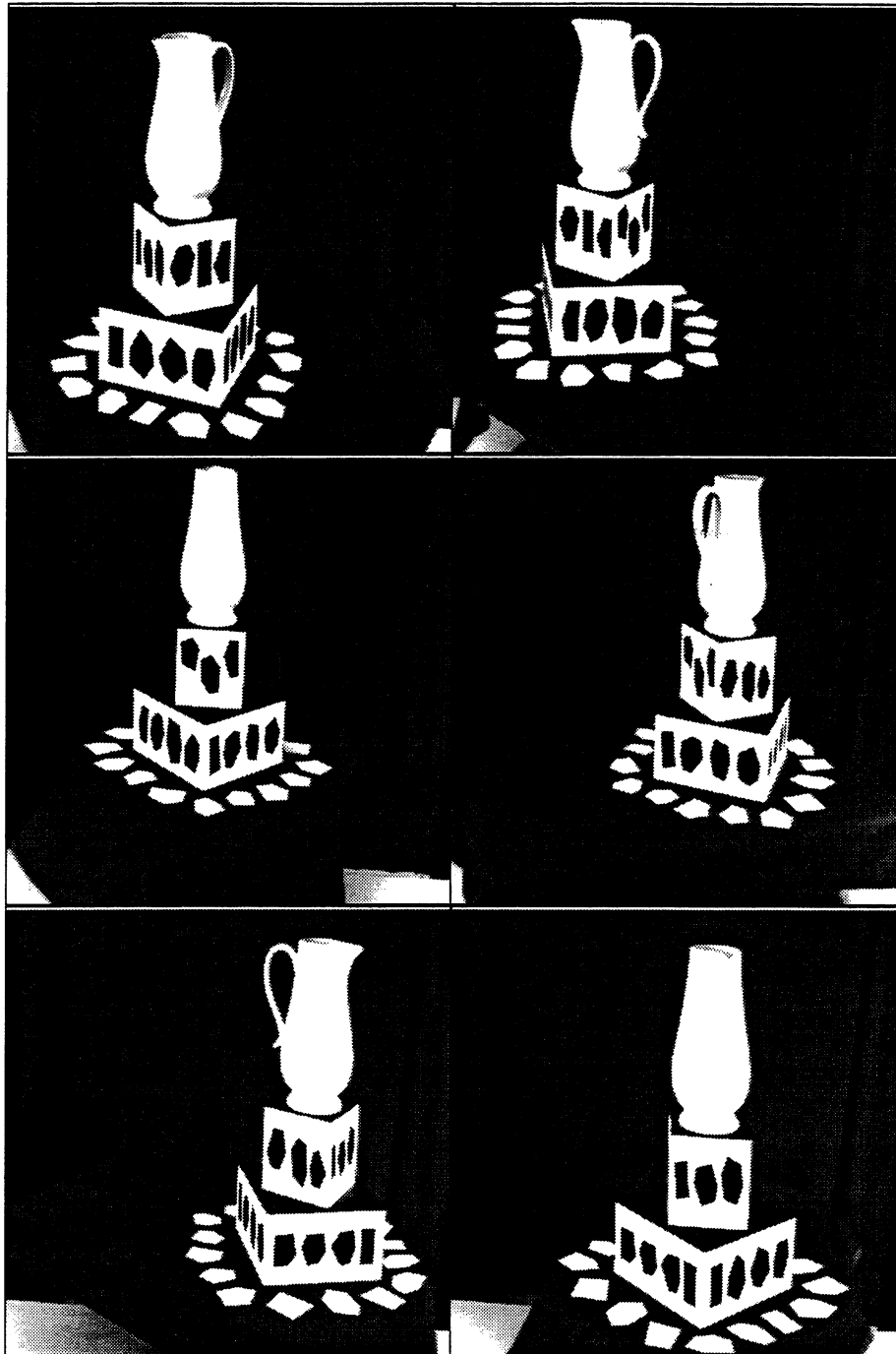


FIG. 2.48 - Différentes images extraites d'une séquence où l'objet à reconstruire est une carafe. Le mouvement est une rotation. Nous pouvons suivre les déplacements successifs de gauche à droite et de haut en bas

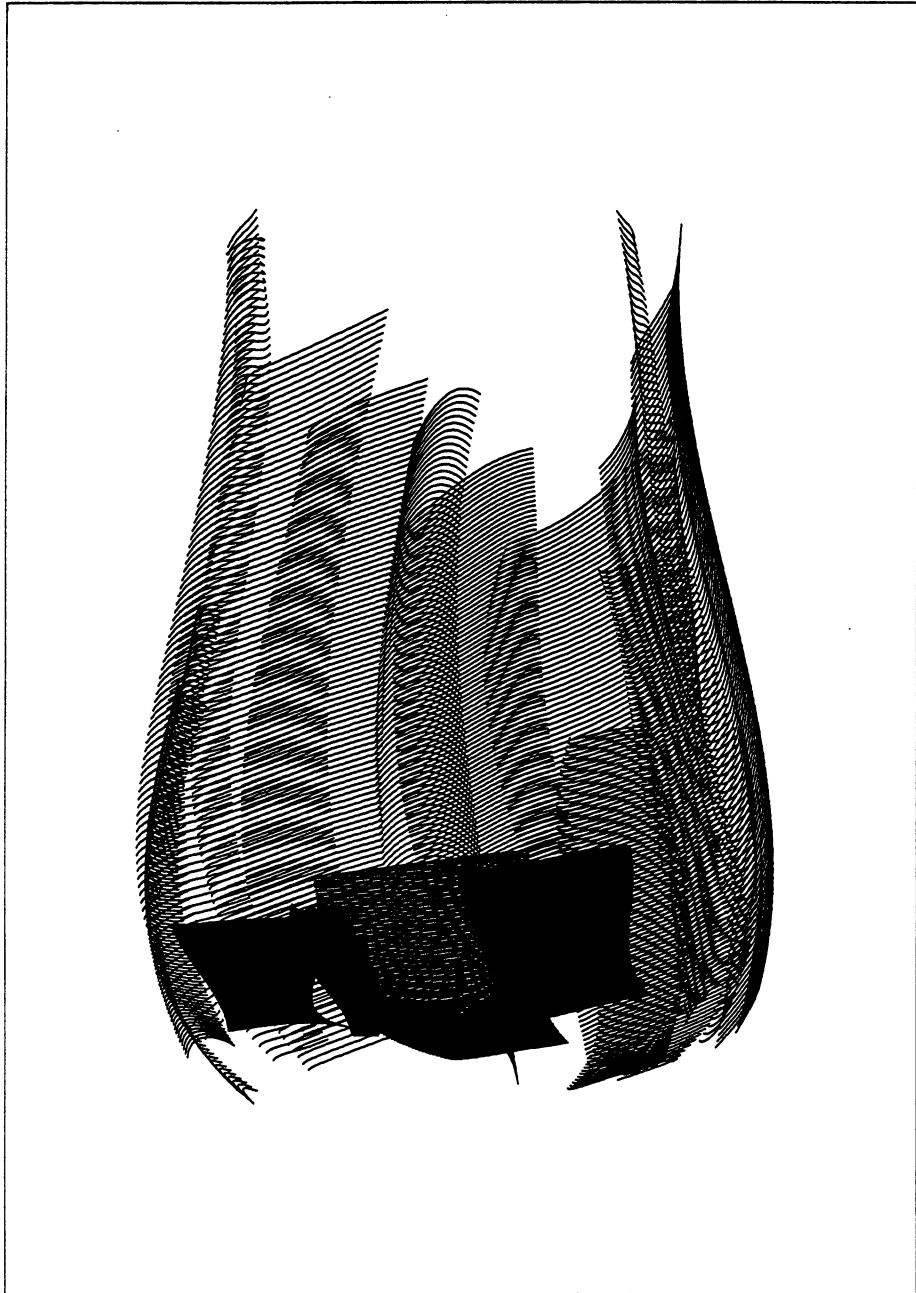


FIG. 2.49 - Ensemble de surfaces B-splines reconstituant la surface de notre carafe

La Figure 2.50 présente une vue de dessus de 21 surfaces B-splines qui reconstituent la surface de notre carafe avec deux cubes.

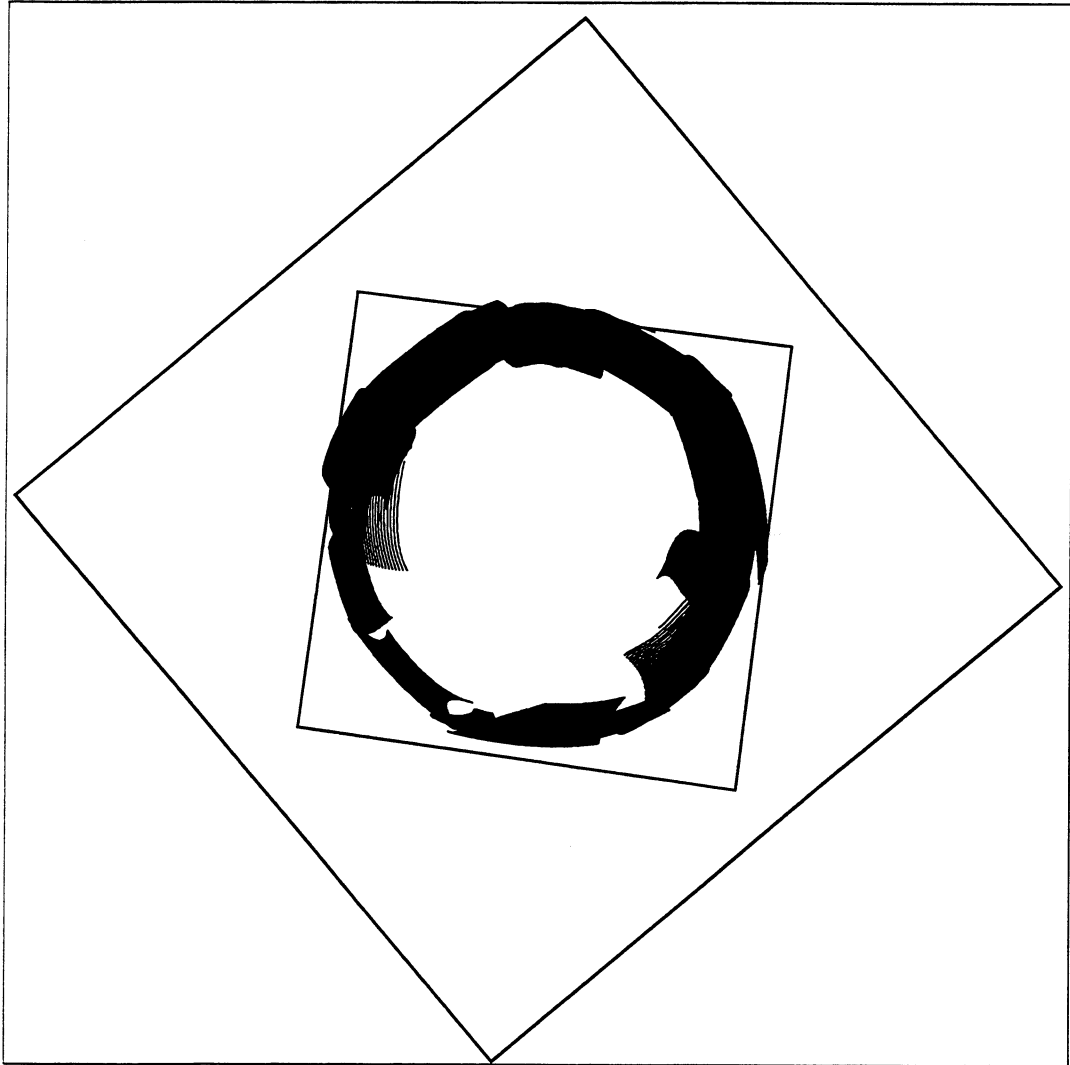


FIG. 2.50 - Une vue de dessus d'un ensemble de surfaces B-splines reconstituant la surface de notre carafe

Nous remarquons facilement que les résultats de la reconstruction des surfaces de la carafe sont qualitativement et quantitativement bons. Cependant, il y a certaines surfaces qui sont localement déformées à cause de l'imprécision du calcul des matrices de projection perspective de notre caméra, et la localisation des contours occultants d'images. Cela provient du choix des points de référence (choix des sommets de deux cubes), de la précision des coins stables détectés dans la scène, et de l'éclairage dans la séquence lors de notre expérimentation. Le bruit est en effet toujours présent dans les images.

Dans la partie d'expérimentation de reconstruction des surfaces, nous avons posé la carafe à reconstruire sur une table tournante. Nous avons tourné la table en fixant la caméra pour prendre une séquence complète d'images. Nous pouvons imaginer que

c'est la caméra qui pourrait tourner autour de l'objet si elle était montée sur le bras d'un robot mobile contrôlable par l'utilisateur. Nous n'avons malheureusement pas pu le faire dans le cadre de ce travail.

2.8 Perspectives et discussion

La reconstruction des surfaces tridimensionnelles d'un objet non polyédrique, à partir de l'observation des contours occultants dans une séquence d'images, est un problème difficile de la vision par ordinateur. Dans ce mémoire, nous avons développé une méthode originale de reconstruction des morceaux surfaciques tridimensionnels à l'aide des surfaces B-splines. Cette méthode permet de reconstituer la surface tridimensionnelle d'un objet morceau par morceau. Notre but est de construire la surface complète en fusionnant un ensemble des surfaces B-splines les unes avec les autres.

Notre méthode de reconstruction des surfaces est effectuée dans le contexte de l'étude du mouvement des contours occultants dans une séquence d'images. Nous extrayons sur un plan image les contours résultant de l'observation de la surface physique de l'objet, des lignes tracées sur ces surfaces par projection perspective (contours occultants). Nous les poursuivons dans la séquence d'images par la méthode pyramidale développée dans ce mémoire. L'intérêt de la méthode de reconstruction tient au fait que nous ne "touchons" pas à l'objet à reconstruire. Elle se distingue des méthodes actives qui utilisent un laser pour parcourir la surface. Ici nous ne travaillons que sur un plan image et nous disposons donc que d'informations bidimensionnelles.

Un point du contour occultant dans le plan image nous fournit exactement trois équations indépendantes. Deux parmi ces trois équations proviennent de la propriété naturelle de la projection perspective de la caméra. La troisième présente la propriété d'orthogonalité de contour occultant : un *contour occultant* est un ensemble de points d'image dont la ligne de vue est tangente à la surface de l'objet observé. A partir de ces trois équations, le problème de la reconstruction est ramené à un problème de résolution d'un système d'équations non linéaires qui détermine les points de contrôle des surfaces B-splines. Finalement, les petits morceaux de la surface de l'objet sont reconstruits. Notre méthode s'applique dans le cas où les surfaces sont non planes, non concaves, mais au moins C^2 par morceaux.

Cette recherche se situe dans la suite des travaux de Giblin [Gib 87], Cipolla [Bla 89, Bla 90, Cip 89, Cip 92], Vaillant [Vai 90a, Vai 90b], et Arbogast [Arb 91a] qui permettent de calculer aux points observés de l'objet non polyédrique les courbures gaussiennes, moyennes, et normales, ainsi que les courbures et les directions principales. La contribution originale de notre recherche est de caractériser les propriétés locales de la surface à reconstruire, et aussi reconstruire la surface globale, ce qui n'avait encore jamais été fait à notre connaissance. Nous avons reformulé le problème de reconstruction de surfaces sous forme d'optimisation en utilisant des surfaces B-splines comme modélisation de la surface d'un objet, et cette façon de procéder a permis d'aboutir à des résultats quantitatifs non encore égalés [Zha 94c, Zha 94d]. Pour notre étude, la caméra non calibrée (ou l'objet) est montée sur un support mobile (par exemple sous le bras d'un robot mobile qui peut être commandé parfaitement par l'utilisateur), afin d'acquérir une séquence complète autour de l'objet dont les surfaces sont reconstruites. En plus, notre méthode de reconstruction introduit directement un terme de régularisation [Zha 94a, Zha 94e] sur les surfaces tridimensionnelles qui permet d'ajuster facilement la forme des surfaces à reconstruire dans le cas où il y a du bruit dans les images.

Nous utilisons les surfaces B-splines qui ont les propriétés les plus intéressantes. Il faudrait essayer avec d'autres représentations des surfaces pour comparer les résultats.

Par exemple, nous pourrions expérimenter notre méthode avec les surfaces de Bézier, ou les surfaces de Coons [Shi 87], etc ...

2.9 Conclusion

Dans ce chapitre, nous avons développé une méthode de reconstruction des éléments surfaciques 3D à l'aide des surfaces B-splines à partir de l'observation de contours occultants. Nous introduisons une régularisation sur la surface 3D au lieu de réaliser le lissage sur les contours occultants 2D comme cela a été fait dans les approches antérieures. Notre approche montre que les résultats de reconstruction des surfaces 3D sont satisfaisants. Les simulations montrent bien que les surfaces sont peu affectées par l'introduction de bruit dans les images synthétiques ainsi que dans les données de déplacement de la caméra. Les résultats obtenus sur les données réelles prouvent bien que notre méthode de reconstruction est robuste en présence du bruit dans les images. Notamment, l'algorithme itératif de Levenberg-Marquardt est bien adapté à ce problème. Il est très facile de trouver les valeurs initiales assurant la convergence par la méthode classique de triangulation de stéréovision. La convergence est généralement obtenue avec un nombre d'itération compris entre deux et dix. Notre objectif final est de fusionner un ensemble de surfaces B-splines reconstruites les unes avec les autres afin d'obtenir une surface complète de l'objet, et de procéder à des expérimentations en grandeur réelle. Cette étape sera présentée dans le chapitre suivant.

Chapitre 3

Raccordement des surfaces B-splines

3.1 Introduction

Dans ce chapitre, nous allons présenter une nouvelle approche pour traiter le problème du raccordement de deux surfaces B-splines uniformes et bicubiques. Ce problème a déjà été posé dans les domaines de la CAO¹ et de la CFAO² [Per 92] qui utilisent, en effet, les modèles volumiques à l'aide d'arbres de construction, ou les modèles surfaciques pour la création de formes. Notamment, les modèles surfaciques sont largement utilisés dans les industries aéronautiques, automobiles, navales et autres. Ils permettent de concevoir des objets qui peuvent être représentés par leur "peau". La conception d'une surface complexe est en général réalisée "par morceaux", construits indépendamment les uns des autres ou non, suivant la méthode de création utilisée. Deux types de problèmes se posent alors :

- la création de surfaces gauches ainsi que le positionnement des différents morceaux dans l'espace.
- le raccordement des différentes surfaces afin d'obtenir une surface globale.

Ici, nous sommes dans un contexte de vision par ordinateur. Nous avons un ensemble de surfaces B-splines qui constituent la surface d'un objet à reconstruire. Ces surfaces

¹Conception Assistée par Ordinateur.

²Conception et Fabrication Assistées par Ordinateur.

possédant des zones de recouvrement, nous voulons les fusionner afin d'obtenir la surface complète.

Ce chapitre est organisé de la manière suivante :

- Tout d'abord, nous décrivons et justifions, dans chacun des paragraphes qui vont suivre, les méthodes que nous avons développées pour résoudre le problème du raccordement. Nous fournissons les algorithmes utilisés à chaque étape de la résolution.
- Ensuite, nous présentons les résultats obtenus et les améliorations à apporter pour que cette recherche s'applique dans des cas plus généraux.

3.2 Notre approche

Dans ce paragraphe, nous allons introduire une nouvelle méthode pour le raccordement de deux surfaces B-splines.

Avant de détailler notre algorithme de raccordement, nous allons d'abord étudier les résultats des surfaces reconstruites d'un hyperboloïde (voir Figure 3.1) lorsque notre caméra se déplace dans l'espace. La surface de la Figure 3.1a est le résultat de l'observation d'un hyperboloïde dans l'espace avec un premier mouvement de la caméra. Dans la Figure 3.1b, nous avons ajouté une deuxième surface à la première. Celle-ci a été obtenue par un deuxième mouvement de la caméra. Ces deux surfaces ont une zone de recouvrement.

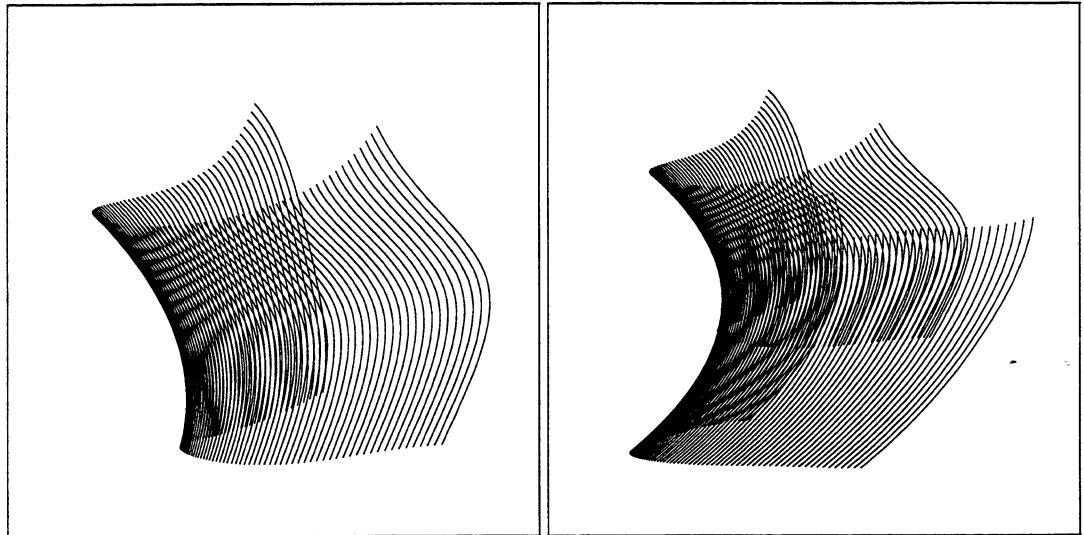


FIG. 3.1 - a) Une portion observée d'un hyperboloïde lors du mouvement de la caméra (à gauche). b) Deux surfaces B-splines reconstruites d'un hyperboloïde avec deux mouvements différents de la caméra (à droite).

Ici nous considérons le problème du raccordement pour deux surfaces seulement. Le concept peut être aisément généralisé par récurrence pour une série de surfaces.

Nous pouvons déjà nous poser une première question : est-ce que la paramétrisation est une représentation suffisante de la surface dans notre problème du raccordement ?

En effet, la représentation par les surfaces B-splines nous fournit d'importantes informations géométriques telles que le rayon de courbure ou la normale à un point. Cependant, les représentations des surfaces en informatique sont généralement des représentations discrètes. De plus, si nous voulons raccorder deux surfaces entre elles, la paramétrisation de surfaces B-splines n'a plus lieu d'être, incapable de décrire la nouvelle surface qui sera créée. Il faut ainsi déterminer une nouvelle représentation des surfaces qui s'adapte à notre problème.

Nous pouvons aussi formuler une deuxième remarque : le problème du raccordement se ramène à un traitement des zones de recouvrement de deux surfaces. Ce sont ces zones qui posent un problème puisque nous pouvons encore distinguer l'appartenance des points à l'une ou à l'autre surface. Ce problème ne peut pas se résoudre en une étape unique.

Après avoir détecté les zones de recouvrement, il faudra traiter ces zones irrégulières en les fusionnant. De plus, comme les bords sont les parties les plus bruitées, il faudra lisser l'ensemble. Puis il faudra fusionner ces zones aux surfaces dont elles étaient extraites au départ afin de créer une seule surface.

Nous choisissons ainsi de procéder selon quatre étapes dans la résolution que nous allons adopter :

- le maillage : nous définirons un nouveau modèle caractérisant une surface B-spline. Aussi, nous ajouterons d'autres informations qui permettront de résoudre les étapes suivantes.
- le découpage : nous déterminerons les zones communes à deux surfaces à partir des informations disponibles dans le nouveau modèle et nous les extrairons.
- la reconstitution des parties découpées : nous traiterons les zones extraites par interpolation.
- le raccordement : nous fusionnerons ces zones aux surfaces d'origine afin de constituer une nouvelle surface.

Dans la suite, nous détaillons chaque étape de la résolution du raccordement de deux surfaces B-splines.

3.3 Le maillage

L'utilisation d'un maillage pour la représentation discrète de surfaces permet facilement d'établir des relations d'adjacence entre les points des surfaces. Les informations géométriques et topologiques des points des surfaces sont ainsi plus aisément implantées et gérées.

Dans ce paragraphe, nous allons fabriquer un maillage régulier tout en conservant certaines informations fournies par les équations paramétrées des surfaces B-splines. Ceci constitue la première étape de la résolution.

3.3.1 Graphe

Les points des surfaces B-splines constituées de courbes B-splines sont définis dans \mathbb{R}^3 par l'équation paramétrée (2.31) du paragraphe 2.2.1.

Nous remarquons qu'à partir de cette équation, nous pouvons déterminer un maillage si nous discrétisons l'espace des paramètres.

En effet, un point p est défini pour $s = p_s$ et $t = p_t$ par $\vec{Q}(p_s, p_t)$. Il aura alors une relation d'adjacence avec les points représentés par $\vec{Q}(p_s - h_s, p_t)$, $\vec{Q}(p_s - h_s, p_t - h_t)$, $\vec{Q}(p_s - h_s, p_t + h_t)$, $\vec{Q}(p_s, p_t - h_t)$, $\vec{Q}(p_s, p_t + h_t)$, $\vec{Q}(p_s + h_s, p_t)$, $\vec{Q}(p_s + h_s, p_t + h_t)$, $\vec{Q}(p_s + h_s, p_t - h_t)$.

La surface peut ainsi être modélisée par un graphe. Les sommets de ce graphe seront les points fournis par les surfaces B-splines. Les arêtes modéliseront les relations de voisinage entre les points. Nous voyons qu'un sommet du graphe peut avoir au maximum huit sommets adjacents ($s \in [a_s, b_s]$ et $t \in [a_t, b_t]$) et deux au minimum ($s = a_s, s = b_s, t = a_t, t = b_t$).

Cette représentation est intéressante car elle fournit des informations topologiques immédiates. De plus, nous n'avons plus besoin de recalculer les coordonnées des points car ces informations sont contenues dans le graphe. Ce graphe permet de produire

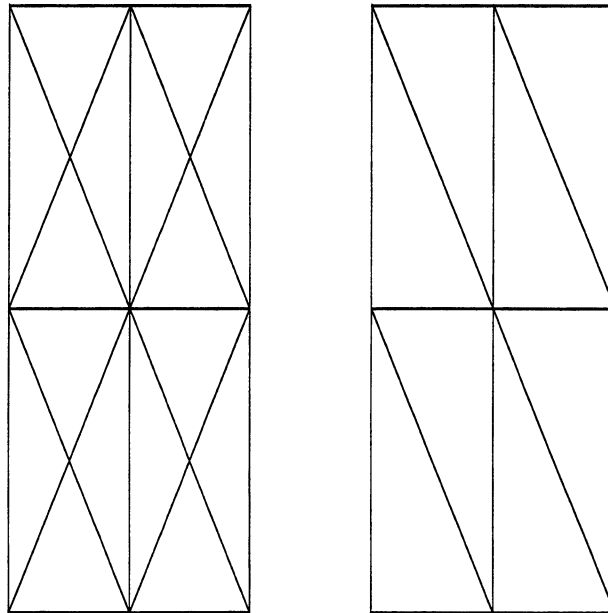


FIG. 3.2 - Maillage à huit sommets adjacents (à gauche) et à six adjacents (à droite) pour neuf points d'une surface projetée sur un plan. On remarquera l'intersection des arêtes dans le premier cas.

un maillage régulier. Nous préférons utiliser un maillage triangulaire à six sommets adjacents. Cette triangulation est plus économique (voir Figure 3.2) et conserve les informations principales d'adjacence. Pour cela, nous n'utiliserons plus que six sommets adjacents pour chaque sommet. Les sommets, voisins de $\vec{Q}(p_s, p_t)$ choisis sont: $\vec{Q}(p_s - h_s, p_t)$, $\vec{Q}(p_s - h_s, p_t - h_t)$, $\vec{Q}(p_s, p_t - h_t)$, $\vec{Q}(p_s, p_t + h_t)$, $\vec{Q}(p_s + h_s, p_t)$, $\vec{Q}(p_s + h_s, p_t + h_t)$.

Nous représentons donc la surface par un graphe $G(X, E)$ où :

- $|X| = n \times m,$

- $|E| = n \times m + (n - 1) \times (m - 1)$,
- $\Delta(X) = 6$ et $\delta(X) = 2$ (degré maximum et minimum d'un sommet).

3.3.2 Structure et algorithme

La topologie de graphe permet une bonne implémentation informatique. Nous définissons une nouvelle structure qui représentera un sommet du graphe et contiendra les informations nécessaires pour la résolution du problème du raccordement. Cette structure doit d'abord contenir des informations géométriques :

- les coordonnées du point dans l'espace tridimensionnel.
- la normale en ce point. Le vecteur normal calculé à partir de $\vec{Q}(s, t)$ servira dans les étapes suivantes.

Elle contiendra des informations topologiques :

- la liste des sommets adjacents,
- sa position dans le graphe (bord, intérieur).

L'algorithme de maillage d'une surface B-spline s'effectue à partir de son équation paramétrique. L'algorithme est constitué de deux boucles principales sur s et sur t . Chaque nouveau point de la surface est calculé, ajouté au graphe ainsi qu'à ses sommets adjacents. Suivant les valeurs de s et de t , nous déterminons si le point possède zéro, un ou trois sommets adjacents au départ. Nous décrivons informellement l'algorithme qui construit la structure contenant le maillage d'une surface :

```

pour  $i$  de  $a_s$  jusqu'à  $b_s$  par pas de  $h_s$ 
  pour  $j$  de  $a_t$  jusqu'à  $b_t$  par pas de  $h_t$ 
    calculer  $x$  pour  $\vec{Q}(i, j)$ ;  $\{x$  est le sommet courant $\}$ 
     $X = X \cup \{x\}$ ;  $\{\text{On travaille avec le graphe } G(X, E)\}$ 
    déterminer les sommets adjacents de  $x$  dans  $G$ ;
    ajouter  $x$  aux sommets adjacents de  $x$ ;
  fin pour
fin pour

```

La complexité temporelle de cet algorithme est linéaire par rapport au nombre de points du maillage.

3.4 L'intersection de deux surfaces

Maintenant, nous prenons deux surfaces maillées ayant une partie en commun telle que le montre la Figure 3.3.

Nous voulons extraire les zones d'intersection de ces deux surfaces pour un traitement ultérieur. Pour cela, nous devons d'abord repérer et délimiter les parties communes aux deux surfaces. Il faut donc définir un critère d'appartenance à une zone de recouvrement pour les points des surfaces.

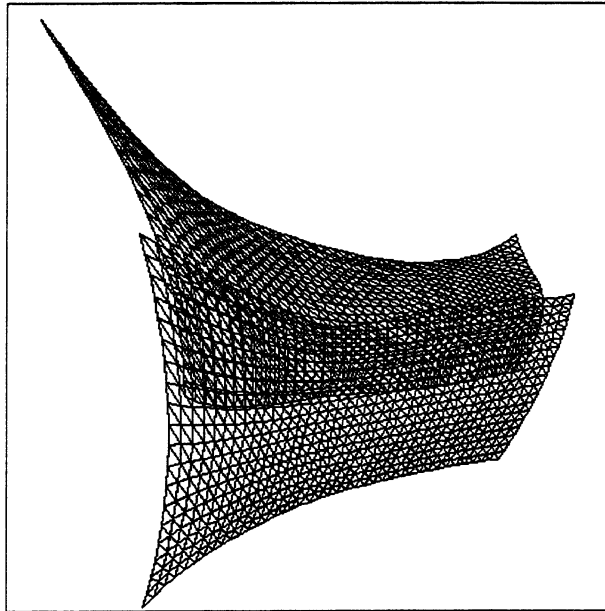


FIG. 3.3 - Deux surfaces d'une hyperboloïde qui se superposent.

3.4.1 Critères de détection

Nous voulons aboutir à une détection rapide des zones communes à deux surfaces. Plusieurs méthodes peuvent être mises en œuvre.

La projection des surfaces suivant leur direction moyenne est une méthode intéressante. La *direction moyenne* d'une surface est définie par l'orientation générale des vecteurs normaux de ses points (voir Figure 3.4). Cette projection doit se faire en plusieurs fois si le rayon de courbure de la surface varie de façon trop importante. Il suffit, ensuite, de comparer dans le plan de projection la distance entre les points des deux surfaces.

Cependant, cette méthode est trop lente et trop fastidieuse à mettre en œuvre pour un grand nombre de points. De plus, il est difficile de distinguer des zones de recouvrement nombreuses.

Pour pallier à ce problème de rapidité, nous nous appuyons sur une constatation évidente : quand une partie de surface recouvre une autre surface, un morceau de son contour appartient nécessairement à la zone de recouvrement. Ceci constitue la base de notre méthode. Nous allons donc parcourir les points du contour d'une des deux surfaces pour vérifier si une partie des surfaces se superpose.

Mais comment vérifier si les points appartiennent réellement aux zones communes ?

Nous travaillons avec un critère de voisinage entre points de surfaces différentes. Soient deux surfaces A et B . Prenons un point $a \in A$. Pour que ce point appartienne à une zone de recouvrement de A et de B , il faut qu'il existe au moins un point $b \in B$ tel que :

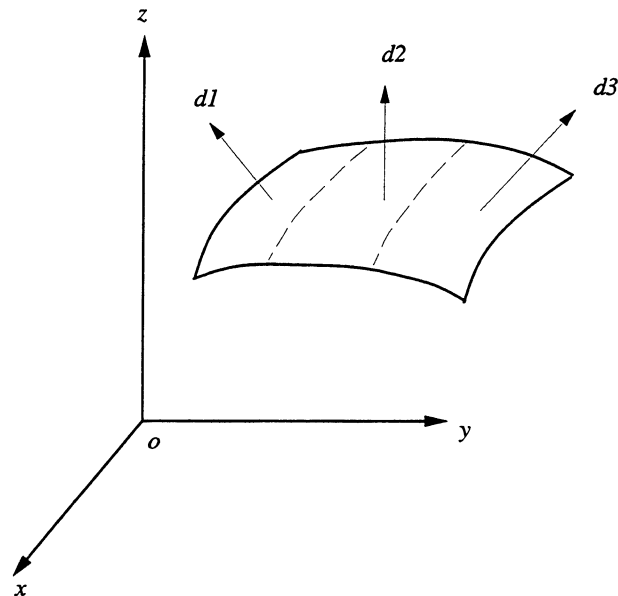


FIG. 3.4 - Méthode de projection d'une surface par morceaux. Ici, la surface sera projetée en trois fois. d_i représente la direction moyenne de i -ième sous-surface.

$$\begin{cases} |a - b| < \varepsilon \\ \langle \vec{N}_a, \vec{N}_b \rangle \geq \theta \end{cases} \quad (3.67)$$

où \vec{N}_a et \vec{N}_b sont les normales aux points a et b , et où ε et θ sont deux seuils.

La première condition implique qu'un point a de la surface A appartient à une zone commune s'il existe un point de B qui est suffisamment proche de a .

La deuxième condition permet d'éliminer les points proches, mais ayant des courbures trop différentes (par exemple, dans le cas où deux surfaces s'intersectent, mais sans se superposer). Les seuils ε et θ sont obtenus par expérimentation.

3.4.2 Algorithme des zones communes de deux surfaces

Plus précisément, l'algorithme des zones communes de deux surfaces B-splines est le suivant :

Soient A et B deux ensembles de points de surfaces maillées différentes.

Soit $C \subseteq A$ (resp. $D \subseteq B$) l'ensemble des points formant le contour de A (resp. B).

Soit Ω l'ensemble des zones communes détectées, et I_k la k -ième zone commune détectée.

(1) Initialisation : $k = 1$; $\Omega = \emptyset$; $C = \text{Contour}(A)$;

(2) Tant que $C \neq \emptyset$
 prendre $c \in C$;
 fin = FAUX;


```

D = B;
Tant que D ≠ ∅ ∨ -fin
  prendre b ∈ D;
  si Propriétés(c, b) {les deux points sont presque confondus}
    Ik = Diffuser(c, b); {les sommets adjacents vérifient-ils la propriété?}
    Ω = Ω ∪ Ik; {on ajoute une nouvelle zone à Ω}
    A = A \ (A ∩ Ik); {on retire de A les points appartenant à la zone}
    B = B \ (B ∩ Ik); {idem pour B}
    k = k + 1;
    fin = VRAI;
  sinon D = D \ {b};
C = C \ {c};

```

(3) $A = A \cup B$. {on assemble les deux surfaces}

Dans cette procédure, il y a deux sous procédures *Propriétés* et *Diffuser*:

La procédure *Propriétés*(a, b) vérifie que pour deux points a et b donnés, $|a - b| < \varepsilon$ et $\langle \vec{N}_a, \vec{N}_b \rangle \leq \theta$.

La procédure *Diffuser* permet d'opérer le même traitement sur les points adjacents à a et b . Cette procédure est récursive. Nous la détaillons ici :

Soit I le résultat de la recherche.

Soit P l'ensemble des points adjacents déjà traités.

Soit Aa l'ensemble des adjacents d'un point a .

Soient Ab l'ensemble des adjacents d'un point b et Ab' une partie ouverte de Ab .

(1) Initialisation : $I = \emptyset$; $Ab' = \emptyset$; $P = \emptyset$; $Aa = \text{Adjacent}(a)$; $Ab = \text{Adjacent}(b)$;

(2) $I = I \cup a$; $I = I \cup b$;

(3) Tant que $Aa \neq \emptyset$

prendre $x \in Aa$;

$Ab' = Ab \setminus P$;

Tant que $Ab \neq \emptyset$

prendre $y \in Ab$;

si *Propriétés*(x, y)

$I = I \cup \text{Diffuser}(x, y)$;

$P = P \cup y$;

sinon $Ab = Ab \setminus \{y\}$;

$Aa = Aa \setminus \{x\}$;

(4) Retourner I .

Quand une zone est extraite, les points bordant cette zone dans les surfaces d'origine sont estampillés d'une information de lien et du numéro de la zone. Nous serons alors en mesure de raccorder les points aux surfaces d'origine.

3.5 Traitement des zones extraites

La précédente étape a permis d'extraire les zones communes sous forme d'ensemble de points 3D. Les points de la surface rendue sont encore distribués de façon irrégulière. Nous pouvons encore différencier les sous-ensembles de points appartenant à chaque surface (voir Figure 3.5).

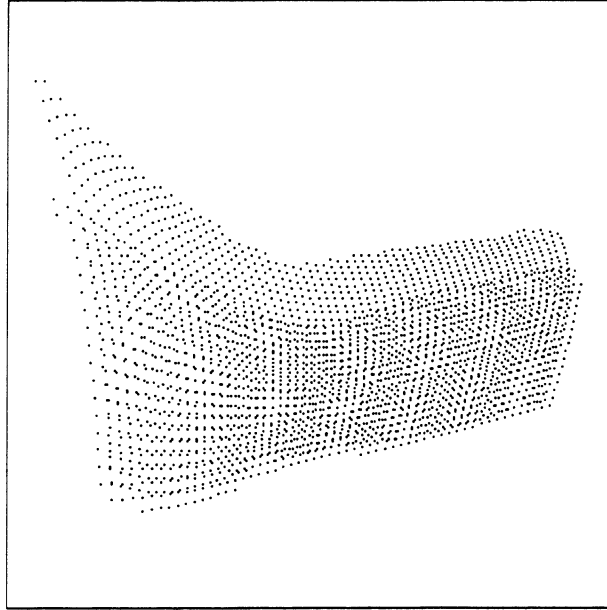


FIG. 3.5 - Zone commune extraite de deux surfaces. On distingue encore l'appartenance des points à l'une et à l'autre surface

Nous voulons reconstruire une nouvelle surface maillée avec cet ensemble de points. En effet, cette surface sera fusionnée ensuite aux surfaces initiales dont la zone était extraite au départ. Nous nous ramenons ainsi à un problème classique : l'interpolation de surface [Sze 90, Sze 91, Pen 92].

Nous nous appuyons sur la méthode d'estimation aux moindres carrés de points 3D en utilisant les surfaces B-splines. Nous en développerons les grandes lignes en Annexe C.

3.5.1 Changement de repère

Pour l'interpolation des points 3D, nous avons besoin de la direction moyenne de la surface. Supposons que l'ancien repère orthonormé $(Oxyz)$ est $(O, \vec{i}, \vec{j}, \vec{k})$, le nouveau repère orthonormé $(OXYZ)$ est $(O, \vec{I}, \vec{J}, \vec{K})$ où O est le point d'origine du repère. La direction moyenne $O\vec{M} = (x_M, y_M, z_M)^T$ d'un ensemble de points $(x_k, y_k, z_k)_{k=1, \dots, p}$ est définie par :

$$\begin{cases} x_M = \sum_{k=1}^p x_k / p \\ y_M = \sum_{k=1}^p y_k / p \\ z_M = \sum_{k=1}^p z_k / p \end{cases} \quad (3.68)$$

Après le changement de repère, la direction de l'axe OZ sera la direction moyenne car nous utiliserons les hauteurs pour l'approximation.

Donc, il ne reste plus qu'à interpoler les points dans ce nouveau repère.

3.5.2 Interpolation d'un ensemble des points 3D

Dans ce paragraphe, nous voulons établir le modèle de la surface interpolée d'un ensemble de points 3D par les surfaces B-splines. Supposons que la surface soit représentée par les surfaces B-splines uniformes et bicubiques suivantes :

$$F(x, y) = \sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} \alpha_{ij} B_i(x) B_j(y) \quad (3.69)$$

Il convient de remarquer que l'interpolation de surface par les surfaces B-splines est équivalente à trouver les valeurs de ces $(m+3) \times (n+3)$ coefficients inconnus α_{ij} , $i = -3, \dots, m-1$, $j = -3, \dots, n-1$ dans la formule (3.69).

Etant donné un ensemble de points $(x_k, y_k, z_k)_{k=1, \dots, p}$ dans le repère orthonormé $(O, \vec{I}, \vec{J}, \vec{K})$, où $x_k \in I \subseteq \mathbb{R}$, $y_k \in J \subseteq \mathbb{R}$, $z_k \in K \subseteq \mathbb{R}$, nous voulons déterminer la surface qui approche au mieux ces points. Ici, nous proposons une méthode aux moindres carrés. Notre approche consiste à minimiser une énergie analogue à celle développée en Annexe C :

$$\Theta = \sum_{k=1}^p [F(x_k, y_k) - z_k]^2 + \tau \int_I \int_J \left(\frac{\partial^6 F}{\partial^3 x \partial^3 y} \right)^2 dx dy \quad (3.70)$$

où le facteur τ est la *tension* des surfaces B-splines, la partie de l'intégrale double étant le *terme de régularisation* des surfaces B-splines.

3.5.3 Réduction de la surface

Nous avons obtenu par une approximation aux moindres carrés une surface composée de surfaces B-splines uniformes et bicubiques. En comparant cette surface à l'ensemble des points 3D de départ, nous remarquons qu'une partie des points doit être éliminée.

En effet, la paramétrisation ne permet pas d'approcher exactement la forme du maillage de départ. Nous allons donc conserver les points de la surface qui se trouvent à l'intérieur de l'ensemble et rejeter les autres.

L'algorithme de la réduction de la surface est le suivant :

Nous projetons la surface suivant l'axe OZ . Nous comparons ensuite la distance entre les points de la surface interpolée et ceux de la zone d'intersection de départ.

Soit F un ensemble de points à interpoler.

Soit F' une variable recopiant F .

Soit P l'ensemble des points de la surface interpolée à partir de F .

Soit P' l'ensemble de points à éliminer.

Soit P'' une variable recopiant P .

- (1) Initialisation : $P' = \emptyset$; $P'' = P$;
- (2) Tant que $P'' \neq \emptyset$
 - prendre $p \in P''$;
 - $F' = F$;
 - fin=FAUX;
 - Tant que $F' \neq \emptyset \vee \neg \text{fin}$
 - prendre $f \in F'$;
 - si $|p - f| < \varepsilon$
 - $P' = P' \cup \{p\}$;
 - fin=VRAI;
 - sinon $F' = F' \setminus \{f\}$;
 - $P'' = P'' \setminus \{p\}$;
- (3) $P = P \setminus P'$.

Enfin, les coordonnées restantes des surfaces résultant de l'interpolation doivent être replacées dans l'ancien repère.

Finalement, nous pouvons récapituler l'ensemble des étapes de cette partie :

- changement de repère et de coordonnées. Le nouvel axe OZ est parallèle à la direction moyenne de l'ensemble de points 3D traités.
- interpolation par les surfaces B-splines.
- maillage des surfaces obtenues pour pouvoir les fusionner ensuite aux surfaces d'origines.
- changement des coordonnées des points des nouvelles surfaces vers l'ancien repère.

Il ne reste plus alors que la dernière étape du raccordement de deux surfaces.

3.6 Le raccordement

Après avoir effectué toutes les étapes précédentes, nous pouvons raccorder les surfaces créées à partir de l'interpolation de l'ensemble de points 3D aux anciennes surfaces.

Lors de la phase de détection, nous avons marqué chaque point des surfaces d'origine bordant une zone commune d'une estampille (voir Figure 3.6). D'une part, celle-ci indique que ces points se différencient des autres puisqu'ils bordent une zone extraite. D'autre part, nous avons ajouté un numéro à ces points signalant à quelle zone extraite ils se rapportent.

Ces deux informations vont permettre de fusionner la zone interpolée correspondante et manquante aux surfaces d'origine.

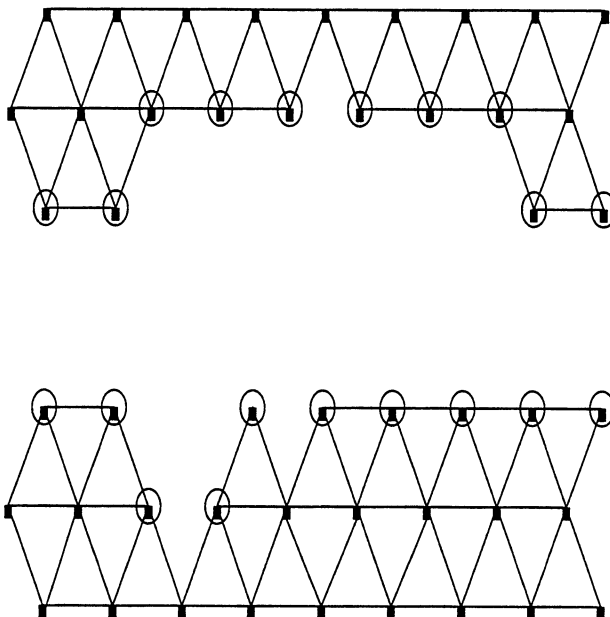


FIG. 3.6 - Deux surfaces privées de leur zone commune. Les points cerclés sont les points possédant l'estampille

3.6.1 Difficulté du raccordement

Nous trouvons de nombreuses contraintes (voir Figure 3.7) qui peuvent diminuer la qualité du raccordement :

- certains points à raccorder peuvent se superposer sur une surface (c'est-à-dire que ces points sont contenus dans cette surface) et il n'y a pas de moyen de le détecter.
- quelques points estampillés n'auront pas d'arêtes avec les points à raccorder. En effet, nous avons émis l'hypothèse au paragraphe 3.3 qu'un point avait au plus six sommets adjacents. Donc un point possédant déjà six liaisons ne pourra pas en avoir une nouvelle avec un point estampillé proche.
- un grand nombre de points ne seront pas reliés entre eux si la différence d'échantillonnage entre les surfaces est trop importante.

La première méthode à laquelle nous avons pensé est l'ajout de relations d'adjacence entre les points estampillés et les points à raccorder. L'idée est de prendre les points les plus proches des points estampillés et d'ajouter une arête. Mais cette méthode ne supprime pas l'ensemble des problèmes exposés.

Aussi, nous avons élaboré une seconde méthode qui permet d'éliminer la plupart des cas difficiles.

3.6.2 La fusion des points

Notre méthode est relativement simple. Nous marquons le contour de la zone à raccorder d'une information *bord*. Ainsi seuls ces points, à l'exclusion des autres, seront

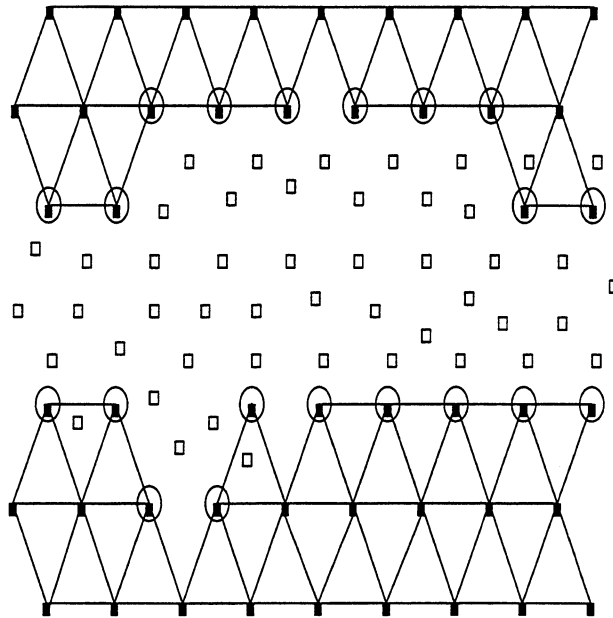


FIG. 3.7 - Ajout de la zone à raccorder. Les différents problèmes énumérés plus tôt y sont exposés.

choisis pour le raccordement. Nous allons fusionner chaque point estampillé avec le point le plus proche dans le maillage à raccorder possédant l'attribut *bord* (voir Figures 3.8 et 3.9). La seule condition à respecter est que le nombre de sommets adjacents aux points à fusionner soit inférieur ou égal à six.

Après la réunion de tous les points estampillés avec un point de la zone à raccorder, nous éliminons les points qui étaient des candidats possibles pour la fusion mais qui ont été évincés par des points plus proches qu'eux.

Les seuls points que nous ne pouvons pas éliminer sont les points qui pourraient se situer dans une surface (voir Figure 3.10).

3.6.3 Algorithme

Nous décrivons l'algorithme utilisé :

Soient P et Q deux surfaces que l'on cherche à raccorder.

Soit R l'ensemble des points à fusionner de P .

Soit B l'ensemble des points formant le contour de Q .

(1) Initialisation : $R = \text{Raccord}(P)$; $B = \text{Bord}(Q)$;

(2) Tant que $R \neq \emptyset$

 prendre $r \in R$;

$R = R \setminus \{r\}$;

$b = \text{ChercherPlusProcheBord}(r, B)$;

 si $b \neq \emptyset$

 Fusionner(r, b);

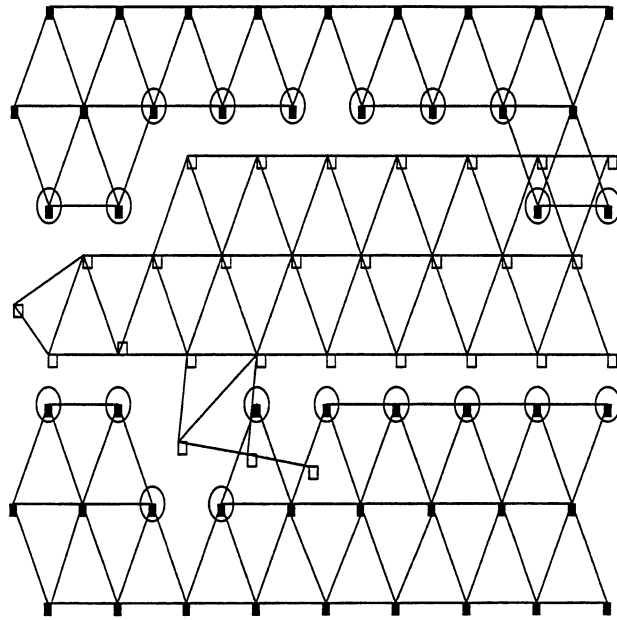


FIG. 3.8 - Fusion à effectuer entre deux zones.

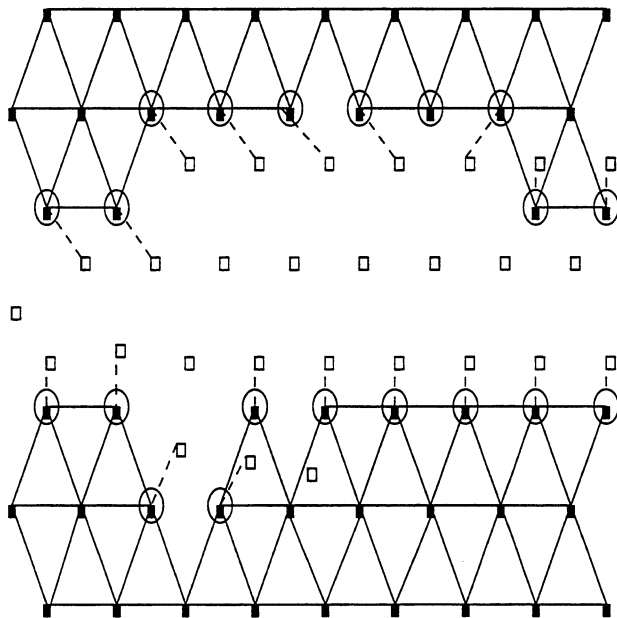


FIG. 3.9 - Exemple de raccordement : les traits en pointillés représentent les futures liaisons.

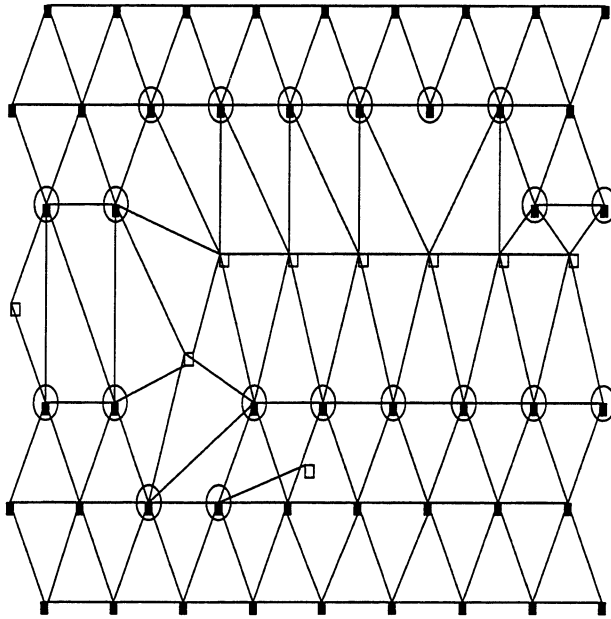


FIG. 3.10 - Résultat de la fusion.

(3) $EliminerCandidatsNonPris(Raccord(P), B)$;

(4) $P = P \cup Q$.

$ChercherPlusProcheBord(x, X)$ est une fonction qui permet de déterminer le point le plus proche de x dans l'ensemble X et possédant l'information *bord*. Nous ne la détaillons pas. Elle consiste en une boucle et un test de distance et d'attribut.

Nous ne détaillons pas non plus la procédure $Fusionner(a, b)$ (a hérite simplement des sommets adjacents de b). Quant à $EliminerCandidatsNonPris$, elle supprime les points qui aurait pu être fusionné au départ.

3.7 Résultats expérimentaux

Nous avons testé la méthode de raccordement sur des données synthétiques ainsi que sur des données réelles.

3.7.1 Données synthétiques

Nous présentons d'abord le résultat du maillage pour deux surfaces d'un hyperboloïde obtenus par des mouvements distincts de la caméra.

La Figure 3.11 présente deux surfaces maillées reconstruites à partir d'un hyperboloïde lorsque notre caméra se déplace dans l'espace.

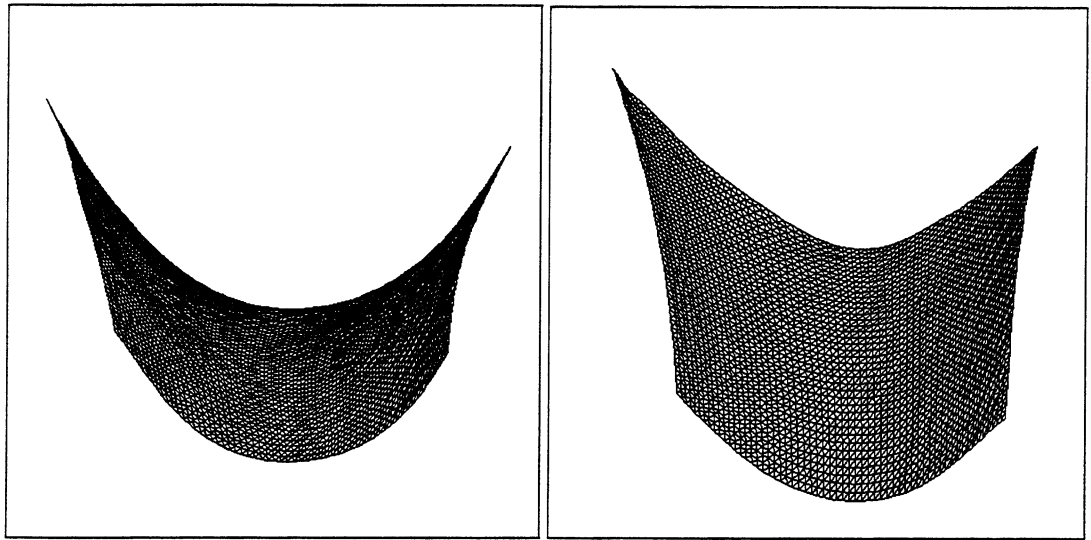


FIG. 3.11 - Première surface (à gauche) et seconde surface (à droite) reconstruites à partir d'un hyperboloïde.

La Figure 3.12 présente les deux surfaces B-splines maillées se superposant, et reconstruites à partir d'un hyperboloïde pour deux mouvements différents de la caméra.

La Figure 3.13 présente les mêmes surfaces dont nous avons extrait la zone commune.

La Figure 3.14 montre le résultat de la fusion de deux surfaces.

3.7.2 Données réelles

Nous avons aussi testé la méthode de raccordement sur deux surfaces reconstruites à partir d'une carafe.

La Figure 3.15 montre la première et la dernière image de la séquence de vingt images de notre carafe. Les deux cubes qui sont placés sous la carafe permettant de calibrer le mouvement de la caméra.

La Figure 3.16 présente le résultat du maillage de deux surfaces reconstruites à partir de notre carafe.

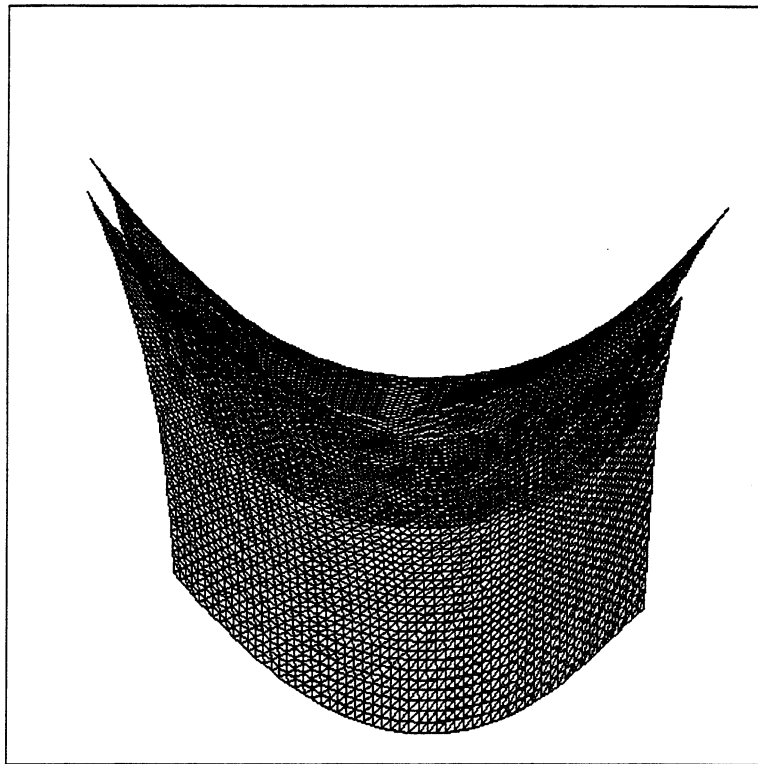


FIG. 3.12 - Superposition de deux surfaces.

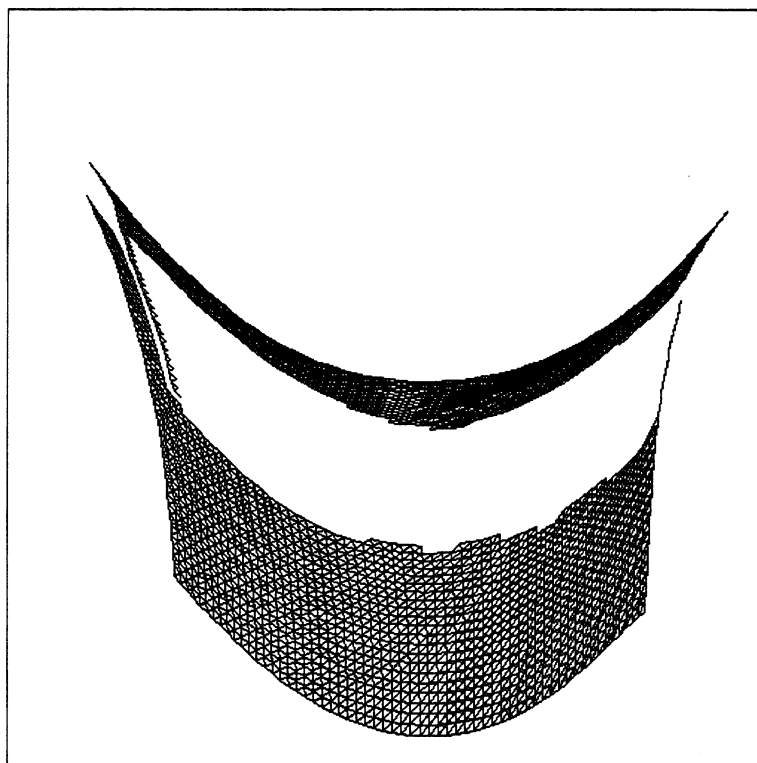


FIG. 3.13 - Extraction des zones communes de deux surfaces.

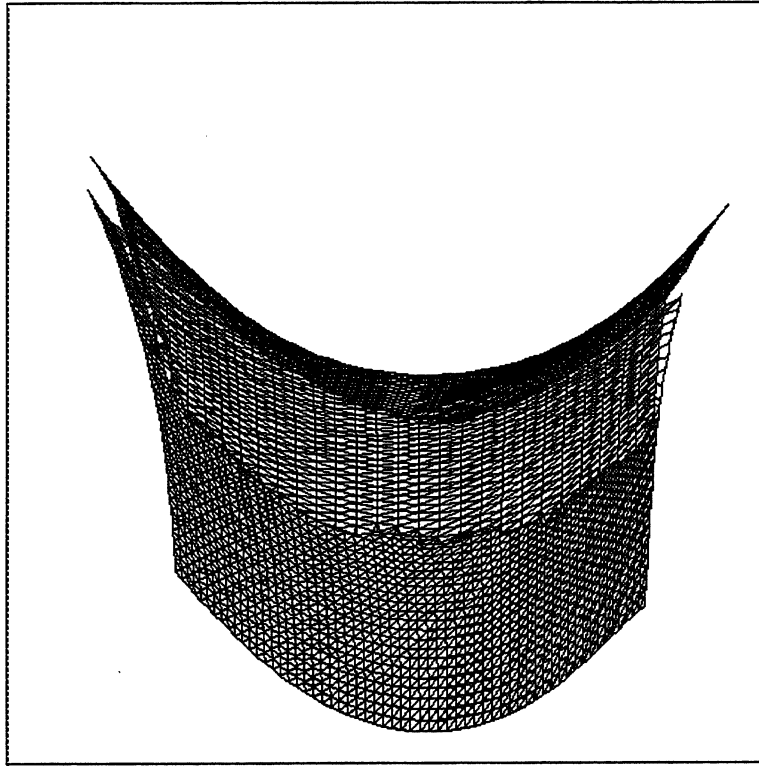


FIG. 3.14 - Fusion de deux surfaces d'un hyperboloïde.

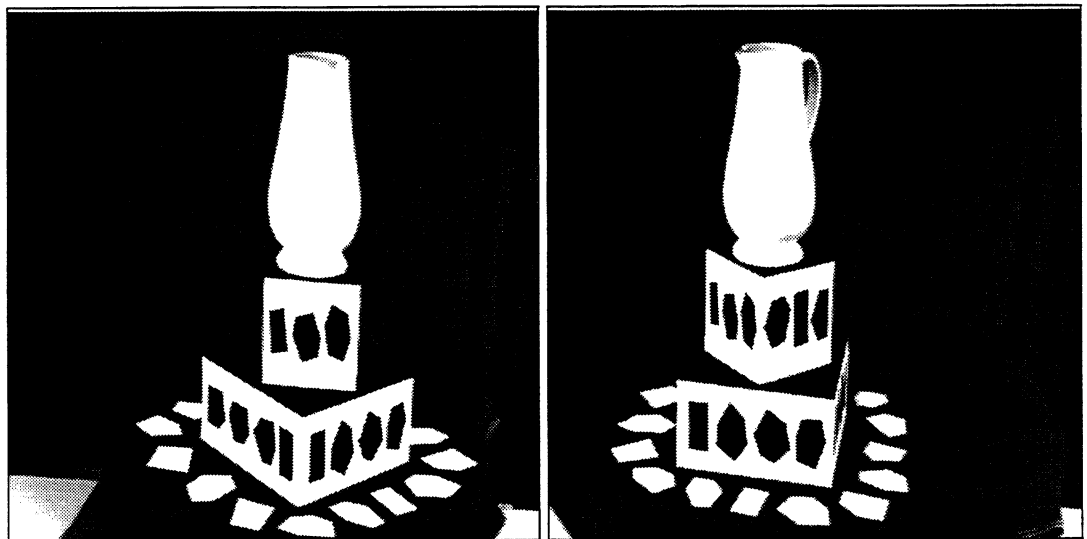


FIG. 3.15 - Première (à gauche) et dernière (à droite) image d'une séquence d'images de la carafe.

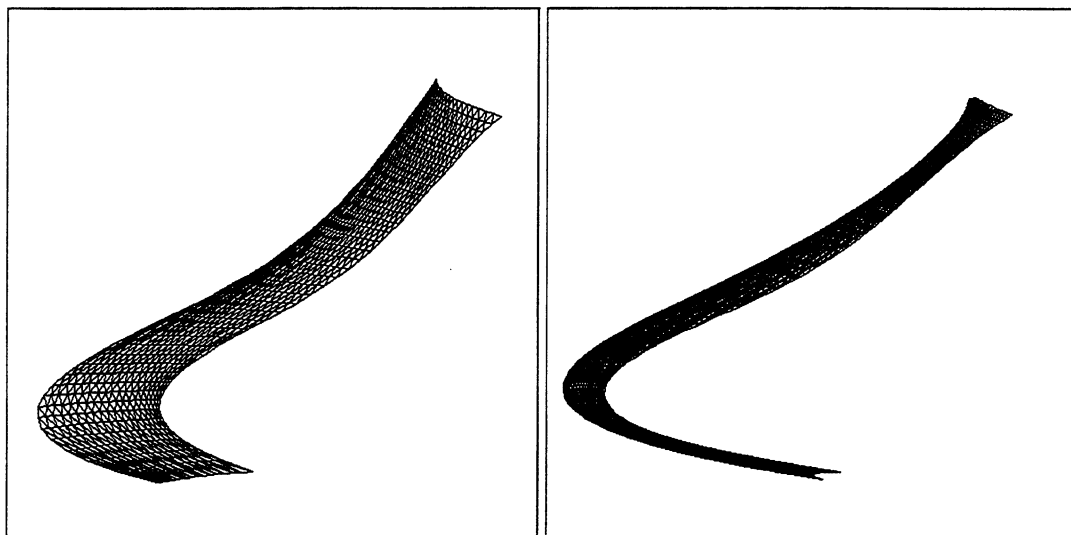


FIG. 3.16 - Première surface (à gauche) et seconde surface (à droite) des surfaces reconstruites à partir de notre carafe.

La Figure 3.17 présente les surfaces maillées des deux surfaces B-splines reconstruites et leur recouvrement.

La Figure 3.18 présente les mêmes surfaces dont nous avons extrait la zone commune. Nous remarquons que certains points n'ont pas pu être extraits. En effet, ces points ne vérifiaient pas les critères de sélection donnés par les deux inégalités (3.67). Les valeurs de ε et θ sont en effet difficiles à choisir.

La Figure 3.19 montre le résultat de la fusion. Ce résultat est correct. Nous nous apercevons facilement qu'au bord, la courbure variant fortement, la qualité des fusions se dégrade. Nous remarquons également le changement d'orientation du maillage triangulaire entre les surfaces d'origine et la zone découpée.

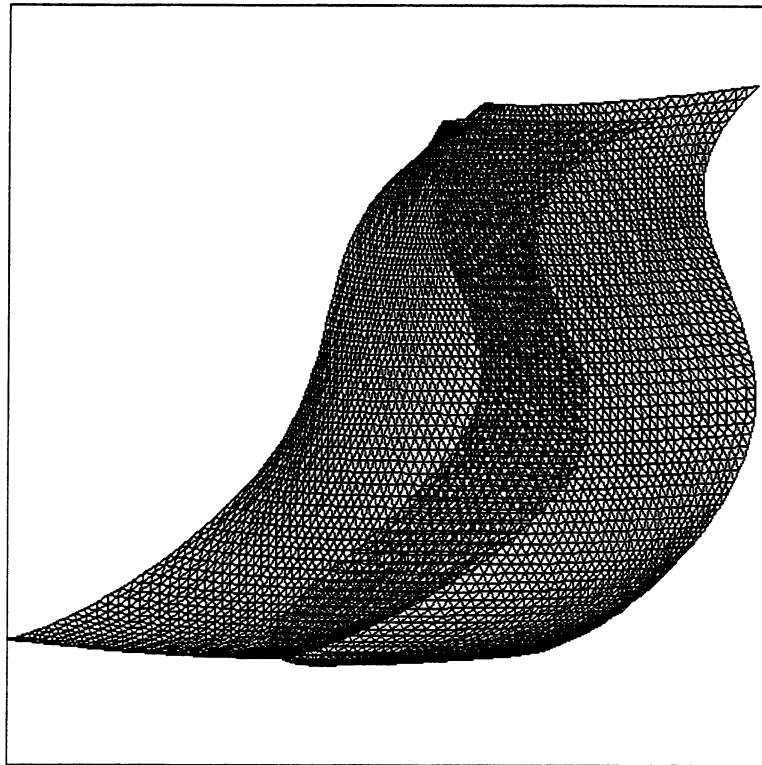


FIG. 3.17 - Superposition de deux surfaces.

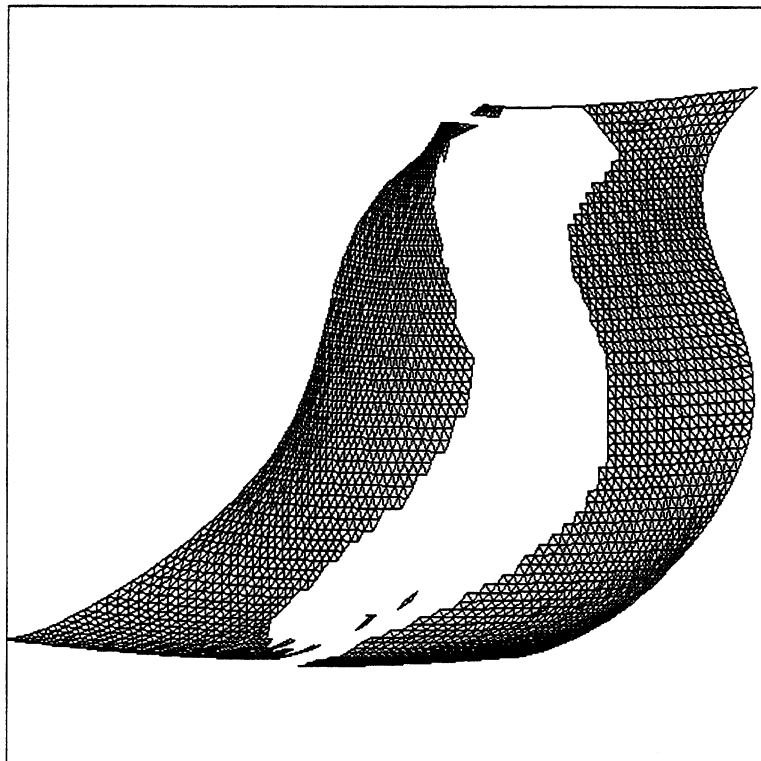


FIG. 3.18 - Extraction de deux surfaces.

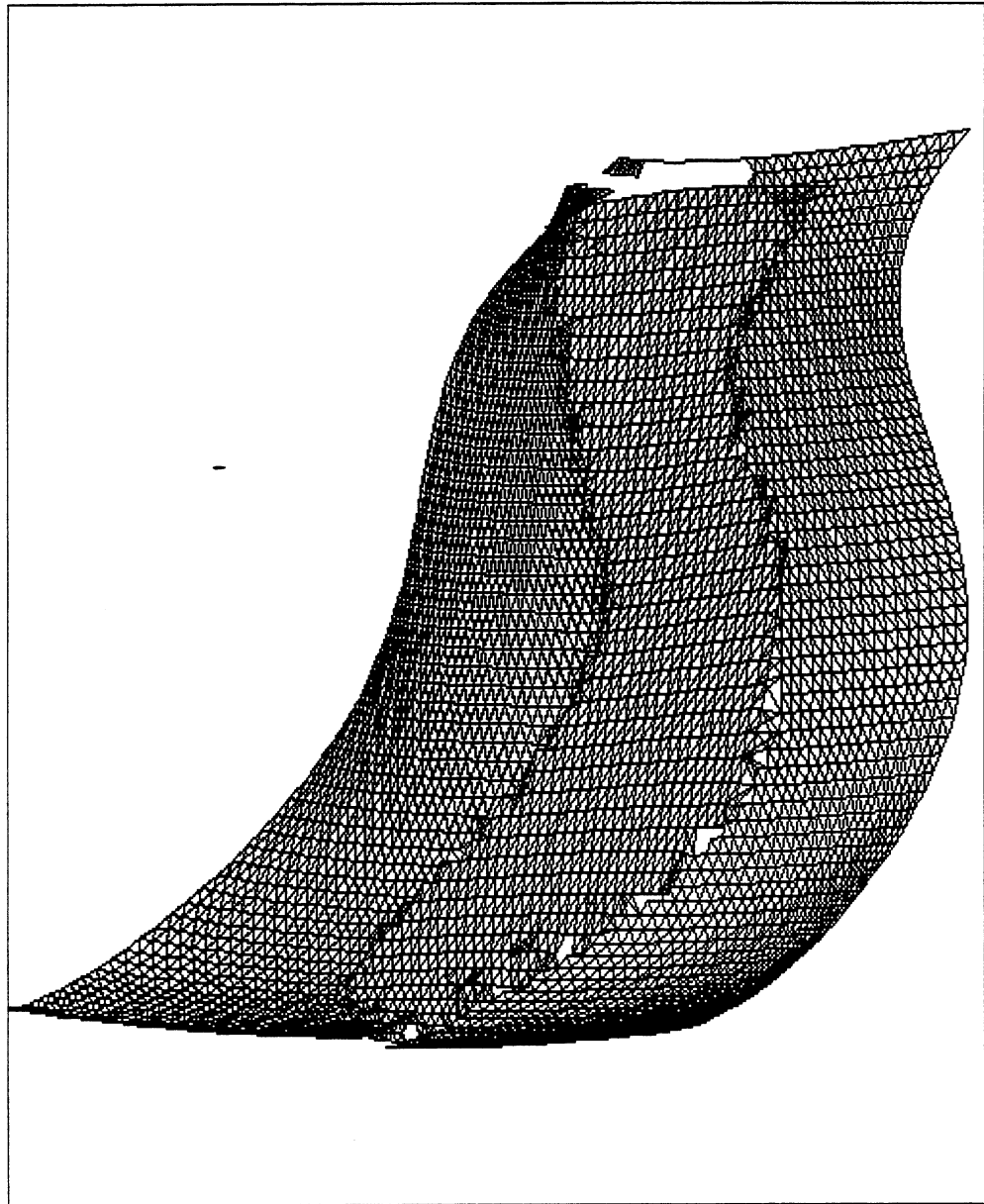


FIG. 3.19 - Fusion de deux surfaces.

3.8 Perspectives et discussion

Comme nous avons pu le voir, le problème de raccordement de surfaces B-splines n'est pas trivial.

La solution adoptée par maillages successifs des surfaces reconstruites est relativement satisfaisante.

L'ajout d'informations topologiques à la surface paramétrée permet une détection rapide des zones de recouvrement et une fusion correcte après traitement. L'interpolation peut être améliorée car le calcul de la direction moyenne dans le cas de surfaces à fort changement de courbure n'est pas assez précis.

Nous avons testé les algorithmes sur des données synthétiques et réelles. Si le raccordement de surfaces s'avère bon dans le premier cas, il en est autrement pour les données réelles. En effet, la présence de bruit dans les images ne permet pas une reconstruction suffisamment fine pour le raccordement. Une des conséquences est que la variation de courbure aux bords des surfaces est importante [Zha 94f].

Les résultats pourront être améliorés par une meilleure saisie des données au départ. Il faut choisir les paramètres de sélection qui s'adaptent le mieux au cas étudié.

Notre approche s'appuie sur le concept de maillage triangulaire et l'interpolation de surface. La structure de maillage triangulaire permet de gérer facilement les informations géométriques et topologiques des points de surfaces B-splines. Notre méthode de l'interpolation de surface moindres carrés est simplement d'utiliser les surfaces B-splines uniformes et bicubiques. Plus particulièrement, cette approche consiste à résoudre un système d'équations linéaires déterminant les points de contrôle de surfaces B-splines. En vision par ordinateur, la méthode de l'interpolation de surface par les fonctions de base des ondelettes est souvent utilisée [Pen 92]. Cette méthode est aussi très efficace au niveau du temps de calcul. Dans cette étude, nous n'avons pas comparé les résultats obtenus par ces deux méthodes.

Notre méthode a été testée pour deux surfaces B-splines. Il reste à tester cette méthode récursivement pour reconstruire un objet entièrement.

Enfin, notre algorithme s'applique dans le cas de surfaces continues. Si les surfaces possèdent des trous ou des parties concaves, certaines zones de recouvrement ne seront pas détectées. Ce point précis doit donc être résolu.

3.9 Conclusion

Dans ce chapitre, nous avons présenté une méthode de raccordement de surfaces B-splines uniformes et bicubiques. Cette méthode s'est révélée performante sur les données synthétiques comme sur les données réelles.

L'expérimentation nous a montré:

- que les algorithmes proposés s'adaptent bien au problème de raccordement.

- que chaque objet reconstruit à sa propre particularité. C'est pourquoi les critères de sélection doivent être déterminés avec précision.
- qu'un maillage adéquate permet d'augmenter la vitesse des algorithmes de recherche.

Les résultats obtenus aux différents points de la méthode nous paraissent très satisfaisants. Cette méthode pourra s'appliquer dans des cas plus généraux de raccordement de surfaces.

Conclusion

En vision par ordinateur, les problèmes de représentation et reconstruction des surfaces tridimensionnelles d'objets non polyédriques sont difficiles. Dans ce mémoire, nous avons développé une nouvelle méthode de reconstruction des surfaces globales tridimensionnelles, à partir de l'observation du mouvement des contours occultants dans une séquence d'images [Zha 94b].

Nous énumérons ci-dessous les principaux résultats obtenus :

- Nous avons présenté un algorithme pyramidal de poursuite de contours courbes basé sur des primitives de type "points de contour" en utilisant une technique de corrélation. Après la phase de mise en correspondance des chaînes de contours, nous nous sommes intéressés à l'élimination des erreurs produites par les étapes initiales du processus de vision. Nous avons proposé une approche originale basée sur une technique de manipulation de graphes spatio-temporels.
- Nous avons étendu l'algorithme précédent du suivi de contours courbes en appliquant la contrainte épipolaire. D'une part, cela permet de diminuer la zone de recherche de corrélation, d'autre part, le résultat de la mise en correspondance point à point est plus précis que pour la corrélation bidimensionnelle.
- Nous avons développé une méthode originale de reconstruction des morceaux surfaciques tridimensionnels à l'aide des surfaces B-splines. L'intérêt de cette méthode est de caractériser les propriétés locales de la surface à reconstruire, de permettre une reconstruction globale. Cette approche permet de compenser les imprécisions dans les images par l'accumulation des données. Elle est également inspirée de la méthode de reconstruction relative de la scène tridimensionnelle à partir d'une séquence d'images non calibrées. En plus, notre méthode de reconstruction introduit directement un terme de régularisation sur les surfaces qui permet d'ajuster facilement la forme des surfaces à reconstruire dans le cas où il y a du bruit dans les images.

- Après l'étape de reconstruction des surfaces B-splines morceau par morceau, nous avons présenté une méthode du raccordement des différentes surfaces B-splines reconstruites afin d'obtenir une surface complète. Cette méthode s'appuie sur le concept de maillage triangulaire et l'interpolation de surface. Cette structure de maillage triangulaire permet d'établir des relations d'adjacence entre les points des surfaces. Les informations géométriques et topologiques des points des surfaces sont ainsi plus facilement exprimées et gérées. Le coût des algorithmes de traitement des patchs s'en trouve diminué.

Les travaux décrits dans ce mémoire ouvrent des perspectives nouvelles en matière de modélisation d'objets à l'aide d'une caméra en mouvement. Ils ont bien montré que la reconstruction des surfaces complètes de l'objet observé par une caméra en mouvement est possible lorsque notre caméra fait un tour autour de l'objet. Notamment, les surfaces B-splines reconstruites ont deux propriétés essentielles très intéressantes : ils sont à support compact et leurs points de contrôle commutent avec toute transformation affine.

Mathématiquement, nous formulons le problème de reconstruction de surfaces sous la forme d'un problème d'optimisation en utilisant des surfaces B-splines comme modélisation de la surface d'un objet, puis nous ramenons ce problème à un problème de résolution d'un système redondant d'équations non linéaires déterminant leurs points de contrôle. Nos expérimentations montrent que notre système de quelques mille équations non linéaires avec quelques centaines inconnues converge correctement. Sans tenir compte des applications potentielles, la résolution de ce problème nous semble déjà très intéressante au niveau scientifique.

Les surfaces tridimensionnelles obtenues à partir d'un objet inconnu peuvent être utilisées par exemple en robotique pour la manipulation et la saisie, en robotique mobile pour la planification d'un robot dans un environnement où il y a des obstacles, ou afin de gérer automatiquement des descriptions d'objets.

Après la phase de raccordement des surfaces B-splines morceau par morceau, nous pouvons retrouver la surface complète d'un objet, qui peut alors être utilisée, soit pour la visualisation de la forme de l'objet observé, soit comme données pour un traitement de haut niveau en vision.

Ce travail a révélé de nombreuses perspectives de recherche qui n'ont cependant pas pu être couvertes pendant cette thèse. Les problèmes suivants méritent d'être approfondis :

- Dans la partie de correction des anomalies de contours d'une séquence d'images, nous n'avons pas suffisamment abordé le problème de coupure de contours à cause de la difficulté de définition des critères.
- Nous supposons que le modèle de la caméra est parfaitement sténopé. Nous n'avons donc pas corrigé sa distorsion [Bey 92a, Bey 92b, Peu 92a, Peu 92b]. De plus, il faudrait améliorer les précisions du calcul de la matrice de projection perspective [Bou 93], et de la localisation des contours occultants. En particulier, il faudrait augmenter la qualité du détecteur des coins de la scène [Der 90b, Der 93].

Notre approche de reconstruction des surfaces s'applique dans le cas où les surfaces sont non planes, non concaves, mais au moins C^2 par morceaux. Pour la reconstruction des surfaces de la carafe, nous n'avons pas pu obtenir la surface de son anse. Il faudrait utiliser d'autres types d'informations fournies par d'autres systèmes de vision [Koe 80, Woo 89] pour compléter la partie manquante de la carafe.

Le problème de la reconstruction des surfaces peut aussi être abordé en combinant avec d'autres approches qui utilisent les contours liés aux ombres [Hor 77, Pen 88, Sax 89, JO 93]. En particulier, nous pouvons analyser la fonction d'intensité dans le plan image [Ike 81], afin d'obtenir les propriétés caractéristiques de la surface observée.

Le cœur de notre approche a été l'utilisation de contours d'occultation, ces contours pourraient-ils avoir d'autres utilisations ? Reconnaissance d'objets non polyédriques [Pon 89, For 91a, For 91b, For 92, Zis 92] ou au moins discrimination de différents objets entre eux, ...

En guise de conclusion, nous pouvons retenir comme ligne directrice du travail présenté dans cette thèse, l'utilisation des contours occultants d'une séquence d'images calibrées pour reconstruire la surface tridimensionnelle d'un objet non polyédrique.

Annexe A

Géométrie différentielle des courbes et des surfaces

A.1 Les courbes

Une courbe paramétrée différentiable est une application $\gamma: I \rightarrow \mathbb{R}^2$ d'un ouvert $I \subseteq \mathbb{R}$ dans \mathbb{R}^2 .

La paramétrisation γ est dite **régulière** si $\gamma'(t) \neq 0$ pour tout $t \in I$.

Soit M un point quelconque de la courbe $\gamma(t)$ (voir Figure A.1), le vecteur unitaire tangent $\vec{v} = (v_1, v_2)$ à la courbe en M est donné par la formule suivante :

$$\begin{cases} v_1 = \frac{\dot{x}(t)}{\sqrt{\dot{x}^2(t) + \dot{y}^2(t)}} \\ v_2 = \frac{\dot{y}(t)}{\sqrt{\dot{x}^2(t) + \dot{y}^2(t)}} \end{cases} \quad (\text{A.71})$$

où $\dot{x}(t)$ et $\dot{y}(t)$ sont les dérivées respectives de $x(t)$ et $y(t)$ par rapport à t .

L'angle $\theta(t)$ que fait la tangente à la courbe avec l'axe X est donné par :

$$\begin{cases} \cos\theta(t) = \frac{\dot{x}(t)}{\sqrt{\dot{x}^2(t) + \dot{y}^2(t)}} \\ \sin\theta(t) = \frac{\dot{y}(t)}{\sqrt{\dot{x}^2(t) + \dot{y}^2(t)}} \end{cases} \quad (\text{A.72})$$

La courbure est définie comme le rapport de variation angulaire en fonction de l'abscisse curviligne s est donnée par :

$$\kappa(s) = \frac{d\theta}{ds} \quad (\text{A.73})$$

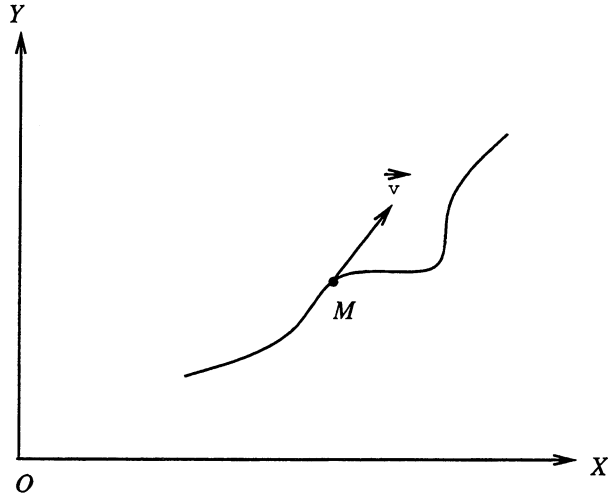


FIG. A.1 - Paramétrisation d'une courbe

avec

$$ds = \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} dt \quad (\text{A.74})$$

L'abscisse curviligne s de cette courbe à partir de t_0 est définie par la formule ci-dessous :

$$s(t) = \int_{t_0}^t \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} dt \quad (\text{A.75})$$

La courbure $c(t)$ de ce point M est donnée par la formule suivante :

$$c(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{[\dot{x}^2(t) + \dot{y}^2(t)]^{\frac{3}{2}}} \quad (\text{A.76})$$

Si $c(t) \neq 0$, alors $\frac{1}{c(t)}$ est appelé **rayon de courbure** en t à la courbe.

A.2 Première et deuxième formes fondamentales

Nous rappelons quelques notions sur la géométrie différentielle des surfaces [Car 76, Fau 90].

Considérons une surface (S) caractérisée par une équation paramétrique :

$$\begin{aligned} \vec{X} : U &\longrightarrow (S) \\ (s, t) &\longrightarrow (x_1(s, t), x_2(s, t), x_3(s, t)) \end{aligned} \quad (\text{A.77})$$

Les propriétés locales intrinsèques d'une telle surface sont caractérisées par deux formes quadratiques : *première et deuxième formes fondamentales*.

Soit $T_p(S)$ le plan tangent à la surface en un point p . Une base de ce plan est formée par $\frac{\partial \vec{X}(s, t)}{\partial s}$ et $\frac{\partial \vec{X}(s, t)}{\partial t}$. Donc un vecteur quelconque du plan tangent est une combinaison linéaire $\lambda \frac{\partial \vec{X}(s, t)}{\partial s} + \mu \frac{\partial \vec{X}(s, t)}{\partial t}$ où $\lambda, \mu \in \mathbb{R}$.

La première forme fondamentale Φ_1 est définie par :

$$\Phi_1 \left(\lambda \frac{\partial \vec{X}(s,t)}{\partial s} + \mu \frac{\partial \vec{X}(s,t)}{\partial t} \right) = \lambda^2 E + 2\lambda\mu F + \mu^2 G = \begin{pmatrix} \lambda & \mu \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \quad (\text{A.78})$$

où

$$\begin{aligned} E &= \left\| \frac{\partial \vec{X}(s,t)}{\partial s} \right\|^2 = \left(\frac{\partial x_1(s,t)}{\partial s} \right)^2 + \left(\frac{\partial x_2(s,t)}{\partial s} \right)^2 + \left(\frac{\partial x_3(s,t)}{\partial s} \right)^2 \\ F &= \frac{\partial \vec{X}(s,t)}{\partial s} \cdot \frac{\partial \vec{X}(s,t)}{\partial t} = \frac{\partial x_1(s,t)}{\partial s} \frac{\partial x_1(s,t)}{\partial t} + \frac{\partial x_2(s,t)}{\partial s} \frac{\partial x_2(s,t)}{\partial t} + \frac{\partial x_3(s,t)}{\partial s} \frac{\partial x_3(s,t)}{\partial t} \\ G &= \left\| \frac{\partial \vec{X}(s,t)}{\partial t} \right\|^2 = \left(\frac{\partial x_1(s,t)}{\partial t} \right)^2 + \left(\frac{\partial x_2(s,t)}{\partial t} \right)^2 + \left(\frac{\partial x_3(s,t)}{\partial t} \right)^2 \end{aligned} \quad (\text{A.79})$$

Le vecteur normale unitaire \vec{N}_p au point p est égal à :

$$\frac{\frac{\partial \vec{X}(s,t)}{\partial s} \times \frac{\partial \vec{X}(s,t)}{\partial t}}{\sqrt{EG - F^2}} \quad (\text{A.80})$$

Soit (S) une surface d'orientation \vec{N} . L'application $\vec{N}: S \rightarrow \mathbb{R}^3$ qui prend ses valeurs sur la sphère unité est appelée *carte de Gauss*.

La différentielle $d\vec{N}_p$ de l'application \vec{N} en un point $p: T_p(S) \rightarrow T_p(S)$ permet de définir la deuxième forme fondamentale Φ_2 :

$$\Phi_2(\vec{v}) = -d\vec{N}_p(\vec{v}) \cdot \vec{v} \quad (\text{A.81})$$

où $\vec{v} \in T_p(S)$.

La deuxième forme fondamentale est une notion reliée à la courbure normale.

Elle est définie par la formule suivante :

$$\Phi_2 \left(\lambda \frac{\partial \vec{X}(s,t)}{\partial s} + \mu \frac{\partial \vec{X}(s,t)}{\partial t} \right) = \lambda^2 L + 2\lambda\mu M + \mu^2 N \quad (\text{A.82})$$

où

$$L = \frac{\partial^2 \vec{X}(s,t)}{\partial s^2} \cdot \vec{N}_p = \frac{\left[\frac{\partial^2 \vec{X}(s,t)}{\partial s^2} \cdot \left(\frac{\partial \vec{X}(s,t)}{\partial s} \times \frac{\partial \vec{X}(s,t)}{\partial t} \right) \right]}{\sqrt{EG - F^2}} = \frac{\begin{vmatrix} \frac{\partial^2 x_1}{\partial s^2} & \frac{\partial^2 x_2}{\partial s^2} & \frac{\partial^2 x_3}{\partial s^2} \\ \frac{\partial x_1}{\partial s} & \frac{\partial x_2}{\partial s} & \frac{\partial x_3}{\partial s} \\ \frac{\partial x_1}{\partial t} & \frac{\partial x_2}{\partial t} & \frac{\partial x_3}{\partial t} \end{vmatrix}}{\sqrt{EG - F^2}} \quad (\text{A.83})$$

$$M = \frac{\partial^2 \vec{X}(s,t)}{\partial s \partial t} \cdot \vec{N}_p = \frac{\left[\frac{\partial^2 \vec{X}(s,t)}{\partial s \partial t} \cdot \left(\frac{\partial \vec{X}(s,t)}{\partial s} \times \frac{\partial \vec{X}(s,t)}{\partial t} \right) \right]}{\sqrt{EG - F^2}} = \frac{\begin{vmatrix} \frac{\partial^2 x_1}{\partial s \partial t} & \frac{\partial^2 x_2}{\partial s \partial t} & \frac{\partial^2 x_3}{\partial s \partial t} \\ \frac{\partial x_1}{\partial s} & \frac{\partial x_2}{\partial s} & \frac{\partial x_3}{\partial s} \\ \frac{\partial x_1}{\partial t} & \frac{\partial x_2}{\partial t} & \frac{\partial x_3}{\partial t} \end{vmatrix}}{\sqrt{EG - F^2}} \quad (\text{A.84})$$

$$N = \frac{\partial^2 \vec{X}(s,t)}{\partial^2 t} \cdot \vec{N}_p = \frac{\left[\frac{\partial^2 \vec{X}(s,t)}{\partial^2 t} \cdot \left(\frac{\partial \vec{X}(s,t)}{\partial s} \times \frac{\partial \vec{X}(s,t)}{\partial t} \right) \right]}{\sqrt{EG - F^2}} = \frac{\begin{vmatrix} \frac{\partial^2 x_1}{\partial^2 t} & \frac{\partial^2 x_2}{\partial^2 t} & \frac{\partial^2 x_3}{\partial^2 t} \\ \frac{\partial x_1}{\partial s} & \frac{\partial x_2}{\partial s} & \frac{\partial x_3}{\partial s} \\ \frac{\partial x_1}{\partial t} & \frac{\partial x_2}{\partial t} & \frac{\partial x_3}{\partial t} \end{vmatrix}}{\sqrt{EG - F^2}} \quad (\text{A.85})$$

Les directions principales de la surface en un point p sont les vecteurs propres de $d\vec{N}_p$, elles sont deux racines de l'équation qui suit :

$$(FL - EM)\lambda^2 + (GL - EN)\lambda\mu + (GM - FN)\mu^2 = 0 \quad (\text{A.86})$$

Donc nous avons :

$$\begin{aligned} \lambda &= EN - GL + \epsilon\sqrt{\Delta} \\ \mu &= 2(FL - EN) \end{aligned} \quad (\text{A.87})$$

où $\epsilon = \pm 1$ et $\Delta = (GL - EN)^2 - 4(FL - EM)(GM - FN)$.

Les courbures principales ρ_1, ρ_2 de la surface en un point p sont les valeurs propres de $d\vec{N}_p$, elles sont deux racines de l'équation ci-dessous :

$$(EG - F^2)\rho^2 + (LG + EN - 2FM)\rho + (LN - M^2) = 0 \quad (\text{A.88})$$

En particulier, leurs produits K_g et leurs demi-somme K_m sont les courbures gaussiennes et moyennes, qui sont données par les formules suivantes :

$$\begin{aligned} K_g &= \rho_1 \rho_2 = \frac{LN - M^2}{EG - F^2} \\ K_m &= \frac{\rho_1 + \rho_2}{2} = \frac{1}{2} \frac{LG + EN - 2FM}{EG - F^2} \end{aligned} \quad (\text{A.89})$$

Annexe B

Liste des règles de transformations de graphes

Cette annexe détaille les descriptions des règles de transformations de graphes utilisées dans la partie du raccordement et coupure des chaînes de contours d'images.

Afin de faciliter la lecture des règles, nous commençons par reprendre certaines conventions de représentation des règles.

B.1 Conventions de représentation

1. O : sommet correspondant à un objet de nature quelconque. Q : sommet correspondant à un objet fermé. Les sommets d'une même composante (même image) sont représentés en colonne.
2. Un arc reliant 2 sommets d'une même colonne est un arc d'affinité. Un arc reliant 2 sommets de 2 colonnes voisines est un arc de liaison. Chaque arc n'est déclaré qu'une fois. Pour les arcs de liaison, nous énonçons d'abord le nœud de la composante la plus à gauche (de plus petit indice). Exemple : (1, 0, 2, 0, ..., ...).
3. La dimension d'un graphe ('dim') est le nombre de composantes (nombre d'images concernées) du graphe.
4. L'indice de la première composante d'un graphe est toujours 0. Nous prenons pour composante 0 la composante de gauche. A droite de la composante 0, nous rencontrons successivement les composantes 1, 2, etc ...

5. L'indice du premier nœud d'une composante est toujours 0. Nous prenons pour nœud 0 le nœud du haut. En dessous du nœud 0, nous rencontrons successivement les nœuds 1, 2, etc ...

B.2 Règles de fermeture

```

/*****|
|
|           Rule 1.0
|
|      -0-   -->   -Q-
|
|
|
|*****/

```

```

graphe :
  dim :
    1
  component :
    nodes = [0, 0, EQUAL, *]
    arcs = (0, 0, 0, 0, 30, L)
;
globalContraint :
  30 L
;
transformations :
  [0, 0, 0] S
;
priority :
  0
;

```

```

/*****|
|
|           Rule 1.1
|
|      -Q--0   -->   -Q--Q
|
|
|
|*****/

```

```

graphe :
  dim :
    2
  component :
    nodes = [0, 0, CLOSED, *]
    arcs = (0, 0, 1, 0, 30, L)
  component :
    nodes = [1, 0, EQUAL, +]
    arcs = (1, 0, 1, 0, 30, L)
;
globalContraint :
  35 L
;
transformations :
  [1, 0, 0] S
;
priority :
  0
;

/*****|
|                                     |
|                               Rule 1.2 |
|                                     |
|                                     |
|          -Q--O--Q-  -->  -Q--Q--Q- |
|                                     |
|                                     |
|                                     |
|*****/

```

```

graphe :
  dim :
    3
  component :
    nodes = [0, 0, CLOSED, *]
    arcs = (0, 0, 1, 0, 30, L)
  component :
    nodes = [1, 0, EQUAL, *]
    arcs = (1, 0, 1, 0, 20, L)
           (1, 0, 2, 0, 30, L)
  component :
    nodes = [2, 0, CLOSED, *]
    arcs =
;
globalContraint :
  35 L

```

```

;
transformations :
    [1, 0, 0] S
;
priority :
    0
;

```

```

/*****|
|
|           Rule 2.0
|      -0-
|      |
|      |   -->   -0-
|      |
|      -0-
|
|*****|

```

```

graphe :
dim :
    1

component :
    nodes = [0, 0, EQUAL, *]
            [0, 1, EQUAL, *]
    arcs  = (0, 0, 0, 1, 60, L)
;
globalContraint :
    60 L
;
transformations :
    [0, 0, 1] J
;
priority :
    0
;

```

```

/*****|
|
|           Rule 3.0
|      -0
|      /|
|     -0 |   -->   -0--0
|      \|
|      -0
|
|*****|

```

```

|*****|
graphe :
  dim :
    2
  component :
    nodes = [0, 0, EQUAL, *]
    arcs = (0, 0, 1, 0, 15, L)
          (0, 0, 1, 1, 15, L)
  component :
    nodes = [1, 0, EQUAL, +]
          [1, 1, EQUAL, +]
    arcs = (1, 0, 1, 1, 20, L)
;
globalConstraint :
  35 L
;
transformations :
  [1, 0, 1] J
;
priority :
  1
;

```

```

/*****|
|
|           Rule 4.0           |
|      -0-                    |
|      /|\                    |
|     -0 | 0-  -->  -0--0--0-  |
|      \|\                    |
|     -0-                    |
|
|*****|

```

```

graphe :
  dim :
    3
  component :
    nodes = [0, 0, EQUAL, *]
    arcs = (0, 0, 1, 0, 15, L)
          (0, 0, 1, 1, 15, L)
  component :
    nodes = [1, 0, EQUAL, *]
          [1, 1, EQUAL, *]
    arcs = (1, 0, 1, 1, 20, L)
          (1, 0, 2, 0, 15, L)

```

```

(1, 1, 2, 0, 15, L)
component :
    nodes = [2, 0, EQUAL, *]
    arcs =
;
globalContraint :
    35 L
;
transformations :
    [1, 0, 1] J
;
priority :
    1
;

/*****|
|
|           Rule 4.1
|      -0--0-
|      /|
|      -0 |   -->   -0--0--0-
|      \|
|      -0
|
|*****|

graphe :
dim :
    3
component :
    nodes = [0, 0, EQUAL, *]
    arcs = (0, 0, 1, 0, 15, L)
           (0, 0, 1, 1, 15, L)
component :
    nodes = [1, 0, EQUAL, *]
           [1, 1, EQUAL, +]
    arcs = (1, 0, 1, 1, 20, L)
           (1, 0, 2, 0, 15, L)
component :
    nodes = [2, 0, EQUAL, *]
    arcs =
;
globalContraint :
    35 L
;
transformations :
    [1, 0, 1] J

```

```

;
priority :
    1
;

```

```

/*****|
|
|           Rule 4.2
|   -0--0-   -0-
|   /| |     /
| -0 | |   --> -0--0
|   \| |     \
|   -0--0-   -0-
|
|*****/

```

```

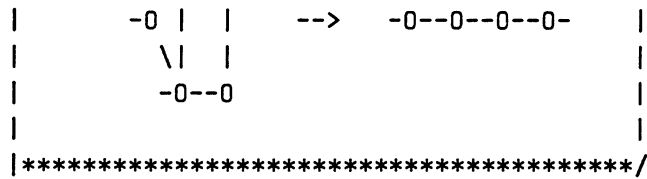
graphe :
dim :
    3
component :
    nodes = [0, 0, EQUAL, *]
    arcs = (0, 0, 1, 0, 15, L)
           (0, 0, 1, 1, 15, L)
component :
    nodes = [1, 0, EQUAL, *]
           [1, 1, EQUAL, *]
    arcs = (1, 0, 1, 1, 30, L)
           (1, 0, 2, 0, 15, L)
           (1, 1, 2, 1, 15, L)
component :
    nodes = [2, 0, EQUAL, *]
           [2, 1, EQUAL, *]
    arcs = (2, 0, 2, 1, 30, H)
;
globalContraint :
    35 L
;
transformations :
    [1, 0, 1] J
;
priority :
    2
;

```

```

/*****|
|
|           Rule 5.0
|
|*****/

```

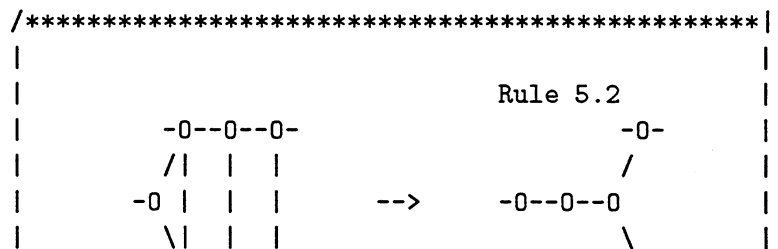



graphe :

```

dim :
    4
component :
    nodes = [0, 0, EQUAL, *]
    arcs = (0, 0, 1, 0, 15, L)
           (0, 0, 1, 1, 15, L)
component :
    nodes = [1, 0, EQUAL, *]
           [1, 1, EQUAL, *]
    arcs = (1, 0, 1, 1, 30, L)
           (1, 0, 2, 0, 15, L)
    -
           (1, 1, 2, 1, 15, L)
component :
    nodes = [2, 0, EQUAL, +]
           [2, 1, EQUAL, *]
    arcs = (2, 0, 2, 1, 30, L)
           (2, 0, 3, 0, 15, L)
component :
    nodes = [3, 0, EQUAL, *]
    arcs =
;
globalContraint :
    35 L
;
transformations :
    [1, 0, 1] J
    [2, 0, 1] J
;
priority :
    3
;

```



```

|          -0--0--0-          -0-          |
|          |          |          |
|*****|
graphe :
  dim :
    4
  component :
    nodes = [0, 0, EQUAL, *]
    arcs  = (0, 0, 1, 0, 15, L)
           (0, 0, 1, 1, 15, L)
  component :
    nodes = [1, 0, EQUAL, *]
           [1, 1, EQUAL, *]
    arcs  = (1, 0, 1, 1, 30, L)
           (1, 0, 2, 0, 15, L)
           (1, 1, 2, 1, 15, L)
  component :
    nodes = [2, 0, EQUAL, *]
           [2, 1, EQUAL, *]
    arcs  = (2, 0, 2, 1, 30, L)
           (2, 0, 3, 0, 15, L)
           (2, 1, 3, 1, 15, L)
  component :
    nodes = [3, 0, EQUAL, *]
           [3, 1, EQUAL, *]
    arcs  = (3, 0, 3, 1, 30, H)
;
globalContraint :
  35 L
;
transformations :
  [1, 0, 1] J
  [2, 0, 1] J
;
priority :
  2
;

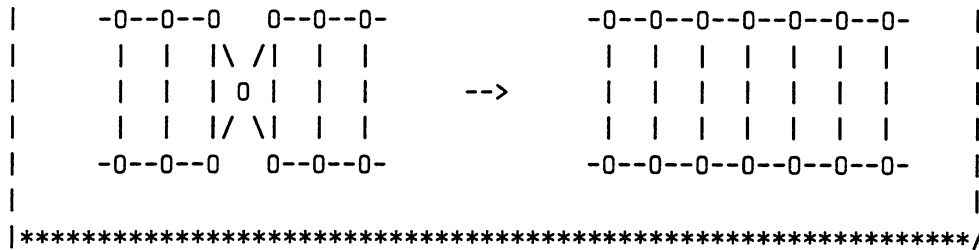
```

B.3 Règles de coupure

```

/*****|
|          |
|          Rule 3.1          |
|          |
|*****|

```



graphe :

```

dim :
    7
component :
    nodes = [0, 0, EQUAL, *]
           [0, 1, EQUAL, *]
    arcs = (0, 0, 0, 1, 60, H)
           (0, 0, 1, 0, 15, L)
           (0, 1, 1, 1, 15, L)
component :
    nodes = [1, 0, EQUAL, *]
           [1, 1, EQUAL, *]
    arcs = (1, 0, 1, 1, 60, H)
           (1, 0, 2, 0, 15, L)
           (1, 1, 2, 1, 15, L)
component :
    nodes = [2, 0, EQUAL, *]
           [2, 1, EQUAL, *]
    arcs = (2, 0, 2, 1, 60, H)
           (2, 0, 3, 0, 15, L)
           (2, 1, 3, 0, 15, L)
component :
    nodes = [3, 0, EQUAL, *]
    arcs = (3, 0, 4, 0, 15, L)
           (3, 0, 4, 1, 15, L)
component :
    nodes = [4, 0, EQUAL, *]
           [4, 1, EQUAL, *]
    arcs = (4, 0, 4, 1, 60, H)
           (4, 0, 5, 0, 15, L)
           (4, 1, 5, 1, 15, L)
component :
    nodes = [5, 0, EQUAL, *]
           [5, 1, EQUAL, *]
    arcs = (5, 0, 5, 1, 60, H)
           (5, 0, 6, 0, 15, L)
           (5, 1, 6, 1, 15, L)
component :
    nodes = [6, 0, EQUAL, *]

```

```
                                [6, 1, EQUAL, *]
                                arcs = (6, 0, 6, 1, 60, H)
;
globalContraint :
    35 L
;
transformations :
    [3, 0, 0] C
;
priority :
    4
;
```

Annexe C

Interpolation de points 3D par surfaces B-splines

Dans cette annexe, nous allons présenter une méthode de l'interpolation de surface par les surfaces B-splines uniformes et bicubiques étant donné un ensemble de points tridimensionnels dans l'espace.

Supposons que la surface soit représentée par les surfaces B-splines uniformes et bicubiques suivantes:

$$F(x, y) = \sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} \alpha_{ij} B_i(x) B_j(y) \quad (\text{C.90})$$

Il convient de remarquer que l'interpolation de surface par les surfaces B-splines est équivalente à trouver les valeurs de ces $(m + 3) \times (n + 3)$ points de contrôle α_{ij} , $i = -3, \dots, m - 1$, $j = -3, \dots, n - 1$ dans (C.90).

C.1 Energie et moindres carrés

Etant donné un ensemble de p points tridimensionnels $(x_k, y_k, z_k)_{k=1, \dots, p}$, nous voulons déterminer la surface¹ qui approche au mieux ces points. Notre approche aux moindres carrés consiste à minimiser l'énergie qui suit :

¹Ici nous utilisons les hauteurs de ces points pour l'approximation de la surface.

$$\Theta = \sum_{k=1}^p [F(x_k, y_k) - z_k]^2 = \sum_{k=1}^p \left[\sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} \alpha_{ij} B_i(x_k) B_j(y_k) - z_k \right]^2 \quad (\text{C.91})$$

Les conditions variationnelles d'optimalité qui expriment en fait la nullité de la dérivée partielle de Θ par rapport à $\alpha_{i_o j_o}$ s'écrivent :

$$\sum_{k=1}^p \left[\sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} \alpha_{ij} B_i(x_k) B_j(y_k) - z_k \right] B_{i_o}(x_k) B_{j_o}(y_k) = 0 \quad (\text{C.92})$$

où $i_o = -3, \dots, m-1, j_o = -3, \dots, n-1$.

et elles conduisent au système des $(m+3) \times (n+3)$ équations ci-dessous :

$$\sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} \alpha_{ij} \left[\sum_{k=1}^p B_i(x_k) B_j(y_k) B_{i_o}(x_k) B_{j_o}(y_k) \right] = \sum_{k=1}^p z_k B_{i_o}(x_k) B_{j_o}(y_k) \quad (\text{C.93})$$

où $i_o = -3, \dots, m-1, j_o = -3, \dots, n-1$.

D'après la propriété de surfaces B-splines, nous avons une remarque suivante :

Remarque :

Pour un couple $(x_k, y_k) \in [u_r, u_{r+1}] \times [v_l, v_{l+1}]$, les seules valeurs de i, j pour lesquelles il faut calculer les produits $B_i(x_k) B_j(y_k)$ dans (C.93) sont $i = r-3, \dots, r, j = l-3, \dots, l$ (les autres étant nécessairement nulles).

Le système (C.93) peut s'écrire sous le système suivant en posant $x_l = \alpha_{(i-4)(j-4)}$, où $l = (i-1)(n+3) + j, i = 1, \dots, (m+3), j = 1, \dots, (n+3)$:

$$\sum_{i=1}^{m+3} \sum_{j=1}^{n+3} x_{(i-1)(n+3)+j} \left[\sum_{k=1}^p B_i(x_k) B_j(y_k) B_{i_o}(x_k) B_{j_o}(y_k) \right] = \sum_{k=1}^p z_k B_{i_o}(x_k) B_{j_o}(y_k) \quad (\text{C.94})$$

où $i_o = 1, \dots, m+3, j_o = 1, \dots, n+3$.

Donc le système précédent (C.94) peut ainsi s'écrire sous la forme matricielle qui suit :

$$M \cdot X = B \quad (\text{C.95})$$

où

$$\begin{aligned} M &= (m_{rq})_{r=1, \dots, (m+3)(n+3), q=1, \dots, (m+3)(n+3)}, \\ X^T &= (x_1, x_2, \dots, x_l, \dots, x_{(m+3)(n+3)}), \\ B^T &= (b_1, b_2, \dots, b_l, \dots, b_{(m+3)(n+3)}) \end{aligned} \quad (\text{C.96})$$

avec

$$\begin{aligned}
m_{rq} &= \sum_{k=1}^p B_i(x_k) B_j(y_k) B_{i_o}(x_k) B_{j_o}(y_k), \\
b_l &= \sum_{k=1}^p z_k B_{i_o}(x_k) B_{j_o}(y_k), \\
i &= (q-1) \operatorname{div}(n+3) + 1, \\
j &= (q-1) \operatorname{mod}(n+3) + 1, \\
i_o &= (r-1) \operatorname{div}(n+3) + 1, \\
j_o &= (r-1) \operatorname{mod}(n+3) + 1, \\
l &= (i_o-1)(n+3) + j_o.
\end{aligned} \tag{C.97}$$

Nous remarquons facilement que $m_{rq} = m_{qr}$, où $r = 1, \dots, (m+3)(n+3)$, $q = 1, \dots, (m+3)(n+3)$. Donc M est une matrice symétrique de type $(m+3)(n+3) \times (m+3)(n+3)$.

C.2 Energie et moindres carrés avec régularisation

L'approche précédente ne permet pas d'ajuster facilement la forme des surfaces à reconstruire dans le cas où il y a du bruit présent dans les images. Donc nous imposons un terme supplémentaire qui s'appelle *terme de régularisation* [Gri 82, Pog 85, Ter 86, Bla 87, Lau 87]. Ce terme nous permet de minimiser les variations de l'orientation de la surface locale pour avoir une surface lissée. Il est équivalent de minimiser les quantités de dérivées du deuxième ordre de la surface.

Puisque la représentation paramétrée de la surface est définie par $(x, y, F(x, y))$.

Nous posons :

$$\begin{aligned}
F_x &= \frac{\partial F}{\partial x}, & F_y &= \frac{\partial F}{\partial y}, \\
F_{xx} &= \frac{\partial^2 F}{\partial^2 x}, & F_{xy} &= \frac{\partial^2 F}{\partial x \partial y}, & F_{yx} &= \frac{\partial^2 F}{\partial y \partial x}, & F_{yy} &= \frac{\partial^2 F}{\partial^2 y}
\end{aligned} \tag{C.98}$$

Donc les courbures gaussiennes K_g et moyennes K_m en un point de la surface sont données par les formules suivantes (voir Annexe A.2) :

$$\begin{aligned}
K_g &= \frac{F_{xx} F_{yy} - F_{xy}^2}{(1 + F_x^2 + F_y^2)^2}, \\
K_m &= \frac{(1 + F_x^2) F_{yy} - 2 F_x F_y F_{xy} + (1 + F_y^2) F_{xx}}{(1 + F_x^2 + F_y^2)^{\frac{3}{2}}}
\end{aligned} \tag{C.99}$$

Dans ce paragraphe, plusieurs possibilités en ce qui concerne le terme de régularisation sont présentées.

C.2.1 Régularisateur de courbure gaussienne

Soient deux intervalles $I = [0, 1]$, et $J = [0, 1]$. Le premier terme de régularisation est posé par la courbure gaussienne. C'est-à-dire nous minimisons la formule ci-dessous :

$$\Delta_1 = \int_I \int_J K_g^2 dx dy = \int_I \int_J \left(\frac{F_{xx}F_{yy} - F_{xy}^2}{(1 + F_x^2 + F_y^2)^2} \right)^2 dx dy \quad (C.100)$$

Supposons que F_x et F_y soient assez petits, donc la formule (C.100) peut s'écrire sous la forme qui suit :

$$\Delta_1 = \int_I \int_J K_g^2 dx dy = \int_I \int_J (F_{xx}F_{yy} - F_{xy}^2)^2 dx dy \quad (C.101)$$

C.2.2 Régularisateur de courbure moyenne

Le deuxième terme de régularisation est posé par la courbure moyenne. C'est-à-dire nous minimisons la formule suivante :

$$\Delta_2 = \int_I \int_J K_m^2 dx dy = \int_I \int_J \left[\frac{(1 + F_x^2)F_{yy} - 2F_xF_yF_{xy} + (1 + F_y^2)F_{xx}}{(1 + F_x^2 + F_y^2)^{\frac{3}{2}}} \right]^2 dx dy \quad (C.102)$$

Supposons que F_x et F_y soient assez petits, donc la formule (C.102) peut s'écrire sous la forme ci-dessous :

$$\Delta_2 = \int_I \int_J K_m^2 dx dy = \int_I \int_J (F_{xx} + F_{yy})^2 dx dy \quad (C.103)$$

C.2.3 Régularisateur de variation quadratique

Nous définissons deux fonctions $H_1(x, y)$, $H_2(x, y)$ telles que

$$\begin{aligned} H_1(x, y) &= F_x, \\ H_2(x, y) &= F_y \end{aligned} \quad (C.104)$$

Leurs variations quadratiques δ_1 , δ_2 de $H_1(x, y)$, $H_2(x, y)$ sont définies par les formules suivantes :

$$\begin{aligned} \delta_1 &= \int_I \int_J [(F_{xx})^2 + (F_{xy})^2] dx dy, \\ \delta_2 &= \int_I \int_J [(F_{yx})^2 + (F_{yy})^2] dx dy \end{aligned} \quad (C.105)$$

Or $F(x, y)$ est C^2 , donc nous avons cette égalité : $F_{xy} = F_{yx}$.

Donc nous proposons le terme de régularisation qui suit :

$$\Delta_3 = (\delta_1 + \delta_2) = \int_I \int_J [(F_{xx})^2 + 2 \times (F_{xy})^2 + (F_{yy})^2] dx dy \quad (C.106)$$

Ce terme revient à minimiser la variation de la courbure. Cela permet de lisser la surface des zones qui sont particulièrement bruitées, et souvent caractérisées par une fonction de courbure très bruitée. Ceci se justifie par le fait que la courbure est une fonction des dérivées du deuxième ordre par rapport à la fonction d'origine.

C.2.4 Régularisateur simplifié

Puisque nous appliquons les surfaces B-splines bicubiques, donc le degré maximal de x , y dans le terme $F(x, y)$ est 3. Pour simplifier le terme de régularisation, nous pouvons donc proposer d'utiliser les régularisateurs suivants:

$$\Delta = \int_I \int_J [(F_{xxx})^2 + (2 \times F_{xyx})^2 + (F_{yyx})^2] dx dy \quad (C.107)$$

ou

$$\Delta = \int_I \int_J [(F_{xxy})^2 + (2 \times F_{xyy})^2 + (F_{yyy})^2] dx dy \quad (C.108)$$

ou

$$\Delta = \int_I \int_J \left(\frac{\partial^6 F}{\partial^3 x \partial^3 y} \right)^2 dx dy \quad (C.109)$$

Finalement, dans notre étude, nous avons utilisé le terme de régularisation (C.109). Donc nous minimisons l'énergie ci-dessous :

$$\Theta = \sum_{k=1}^p [F(x_k, y_k) - z_k]^2 + \tau \int_I \int_J \left(\frac{\partial^6 F}{\partial^3 x \partial^3 y} \right)^2 dx dy \quad (C.110)$$

où le facteur τ est la *tension* des surfaces B-splines, la partie de l'intégrale double est le *terme de régularisation* des surfaces B-splines. Ce facteur de tension permet de passer continûment de la solution exacte aux moindres carrés à une solution dominée par ce terme de régularisateur.

Les conditions variationelles d'optimalité qui expriment en fait la nullité de la dérivée partielle Θ par rapport à $\alpha_{i_o j_o}$ s'écrivent :

$$\begin{aligned} & \sum_{k=1}^p \left[\sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} \alpha_{ij} B_i(x_k) B_j(y_k) - z_k \right] B_{i_o}(x_k) B_{j_o}(y_k) \\ & + \tau \int_I \int_J \left[\sum_{i=-3}^{m-1} \sum_{j=-3}^{n-1} \alpha_{ij} \frac{\partial^3 B_i(x)}{\partial^3 x} \frac{\partial^3 B_j(y)}{\partial^3 y} \right] \frac{\partial^3 B_{i_o}(x)}{\partial^3 x} \frac{\partial^3 B_{j_o}(y)}{\partial^3 y} dx dy = 0 \end{aligned} \quad (C.111)$$

où $i_o = -3, \dots, m-1$, $j_o = -3, \dots, n-1$.

Nous obtenons ainsi un système linéaire $M' \cdot X = B$, où X , B sont définis comme précédemment, et

$$\begin{aligned} M' &= (m'_{rq})_{r=1, \dots, (m+3)(n+3), q=1, \dots, (m+3)(n+3)}, \\ m'_{rq} &= m_{rq} + \tau \int_I \int_J \frac{\partial^3 B_i(x)}{\partial^3 x} \frac{\partial^3 B_j(y)}{\partial^3 y} \frac{\partial^3 B_{i_o}(x)}{\partial^3 x} \frac{\partial^3 B_{j_o}(y)}{\partial^3 y} dx dy \end{aligned} \quad (C.112)$$

Comme le terme de régularisation peut s'écrire sous la forme qui suit :

$$\int_I \int_J \frac{\partial^3 B_i(x)}{\partial^3 x} \frac{\partial^3 B_j(y)}{\partial^3 y} \frac{\partial^3 B_{i_0}(x)}{\partial^3 x} \frac{\partial^3 B_{j_0}(y)}{\partial^3 y} dx dy = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \int_{u_i}^{u_{i+1}} \int_{v_j}^{v_{j+1}} \frac{\partial^3 B_i(x)}{\partial^3 x} \frac{\partial^3 B_j(y)}{\partial^3 y} \frac{\partial^3 B_{i_0}(x)}{\partial^3 x} \frac{\partial^3 B_{j_0}(y)}{\partial^3 y} dx dy \quad (\text{C.113})$$

Donc nous avons la relation ci-dessous :

$$m'_{r_q} = m_{r_q} + \tau \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \int_{u_i}^{u_{i+1}} \int_{v_j}^{v_{j+1}} \frac{\partial^3 B_i(x)}{\partial^3 x} \frac{\partial^3 B_j(y)}{\partial^3 y} \frac{\partial^3 B_{i_0}(x)}{\partial^3 x} \frac{\partial^3 B_{j_0}(y)}{\partial^3 y} dx dy \quad (\text{C.114})$$

Nous remarquons que M' est une $(m+3)(n+3) \times (m+3)(n+3)$ matrice symétrique. Le système $M' \cdot X = B$ peut être résolu par les méthodes classiques.

- Décomposition de Choleski.
- Décomposition en valeur propre (Singular Value Decomposition).
- LU décomposition (le L signifiant lower triangular et le U upper triangular).

Bibliographie

- [Ana 89] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.
- [Arb 91a] E. Arbogast. *Modélisation automatique d'objets non polyédriques par observation monoculaire*. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 1991.
- [Arb 91b] E. Arbogast et R. Mohr. 3D structures inference from images sequences. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(5):749, 1991.
- [Aya 89] N. Ayache. *Vision Stéréoscopique et Perception Multisensorielle: Application à la Robotique Mobile*. Science Informatique. InterEditions, 1989.
- [Bal 82] D.H. Ballard et C.M. Brown. *Computer Vision*. Prentice Hall, 1982.
- [Bar 87] R. H. Bartels, J. C. Beatty, et B. A. Barsky. *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufman Pu. Inc., 1987.
- [Bey 92a] H.A. Beyer. Accurate calibration of CCD cameras. *Proceedings of the Conference on Computer Vision and Pattern Recognition, Urbana-Champaign, Illinois, USA*, pages 96–101, 1992.
- [Bey 92b] H.A. Beyer. *Geometric and Radiometric Analysis of a CCD-Camera Based Photogrammetric Close-Range System*. Thèse de Doctorat, ETH-Zurich, 1992.
- [Bla 87] A. Blake et A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, Massachusetts, 1987.

- [Bla 89] A. Blake et R. Cipolla. Robust Estimation of surface curvature from deformation of apparent contours. Rapport no. OUEL 1797/89, University of Oxford, Department of Engineering Science, Park Road, Oxford OX1 3PPJ, England, 1989.
- [Bla 90] A. Blake et R. Cipolla. Robust estimation of surface curvature from deformation of apparent contours. O. Faugeras, éditeur, *Proceedings of the 1st European Conference on Computer Vision, Antibes, France*, pages 465–474. Springer-Verlag, April 1990.
- [Bol 87] R.C. Bolles, H.H. Baker, et D.H. Marimont. Epipolar-plane image analysis: an approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55, 1987.
- [Bol 91] R.M. Bolle et B.C. Vemuri. On Three-Dimensional Surface Reconstruction Methodes. *IEEE Transactions on PAMI*, 13(1):1–13, January 1991.
- [Bou 93] B. Boufama, R. Mohr, et F. Veillon. Euclidian constraints for uncalibrated reconstruction. *Proceedings of the 4th International Conference on Computer Vision, Berlin, Germany*, pages 466–470, May 1993.
- [Bra 85] M. Brady, J. Ponce, A. Yuille, et H. Asada. Describing Surfaces. Hideo Hanfusa et Hirochika Inoue, éditeurs, *The Second International Symposium of Robotic Research*, pages 5–16. The MIT press, 1985.
- [Bur 81] P.J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, 16:20–51, 1981.
- [Can 86] J. Canny. A computational approach to edge detection. *IEEE Transactions on PAMI*, 8(6):679–698, 1986.
- [Car 76] M.P. Do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976.
- [Cip 89] R. Cipolla. Active Visual Inference of Qualitative Geometry. Rapport no. OUEL 1795/89, University of Oxford, Department of engineering science, Park Road, Oxford OX1 3PJ, U.K., July 1989.
- [Cip 92] R. Cipolla et A. Blake. Surface Shape from the Deformation of Apparent Contours. *International Journal of Computer Vision*, 9(2):83–112, 1992.
- [Cro 88] J.L. Crowley, C. Discours, et P. Stelmaszyk. Measuring image flow by tracking edges lines. *Proceedings of the 2nd International Conference on Computer Vision, Tampa, Florida, USA*, December 1988.
- [Cro 90] J.L. Crowley, A. Chehikian, F. Veillon, A. Favé, et S. Bardel. Definition of Image Primitives. Vision as Process, Esprit Basic Research Actions no. BR 3038 VAP, IMAG, Grenoble, France, 1990.
- [Der 87] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.

- [Der 90a] R. Deriche et O. Faugeras. Tracking line segments. *Proceedings of the 1st European Conference on Computer Vision, Antibes, France*, pages 259–267. Springer-Verlag, April 1990.
- [Der 90b] R. Deriche et G. Giraudon. Accurate corner detection: an analytical study. *Proceedings of the 3rd International Conference on Computer Vision, Osaka, Japan*, 1990.
- [Der 93] R. Deriche et T. Blaszk. Recovering and characterizing image features using an efficient model based approach. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New York, USA*, pages 530–535, June 1993.
- [Fau 87] O.D. Faugeras et G. Toscani. Camera calibration for 3D computer vision. *Proceedings of International Workshop on Machine Vision and Machine Intelligence, Tokyo, Japan*, 1987.
- [Fau 89] O.D. Faugeras et S. Maybank. Motion from point matches: multiplicity of solutions. *IEEE workshop on Computer Vision*, 1989.
- [Fau 90] O. Faugeras. On the motion of 3D curves and its relationship to optical flow. Rapport de Recherche no. 1183, INRIA, Sophia-Antipolis, France, March 1990.
- [Fau 92a] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? G. Sandini, éditeur, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 563–578. Springer-Verlag, May 1992.
- [Fau 92b] O.D. Faugeras, Q.T. Luong, et S.J. Maybank. Camera Self-Calibration: Theory and Experiments. G. Sandini, éditeur, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 321–334. Springer-Verlag, May 1992.
- [For 91a] D. Forsyth, J. L. Mundy, A. Zisserman, et C. Rothwell. Invariant descriptors for 3D object recognition and pose. *Proceeding of the DARPA-ESPRIT workshop on Applications of Invariants in Computer Vision, Reykjavik, Iceland*, pages 171–208, March 1991.
- [For 91b] D. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, et C. Rothwell. Invariant descriptors for 3D object recognition and pose. *IEEE Transactions on PAMI*, 13(10):971–991, October 1991.
- [For 92] D.A. Forsyth, J.M. Mundy, A. Zisserman, et C.A. Rothwell. Recognising rotationally symmetric surfaces from their outlines. G. Sandini, éditeur, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 639–647. Springer-Verlag, May 1992.
- [Fua 90] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 1990.

- [Gib 87] P. Giblin et R. Weiss. Reconstruction of surfaces from profiles. *Proceedings of the 1st International Conference on Computer Vision, London, England*, pages 136–144, London, England, 1987.
- [Gil 89] P. Gill, W. Murray, et M. H. Wright. *Practical Optimization*. Harcourt Brace Jovanovich, Publishers, 1989.
- [Gir 87] G. Giraudon. An efficient edge following algorithm. *Proceedings of the 5th Scandinavian Conference on Image Analysis, Stockholm, Sweden*, pages 547–554, 1987.
- [Gri 82] W.E.L. Grimson. A Computational Theory of Visual Surface Interpolation. Rapport, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts, USA, 1982.
- [Han 84] M.J. Hannah. Description of SRI's Baseline Stereo System. Technical Note no. 342, SRI, Stanford, 1984.
- [Hil 84] E. Hildreth. *The Measurement of Visual Motion*. M.I.T. Press, 1984.
- [Hor 77] B.K.P. Horn. Understanding image intensities. *Artificial Intelligence*, 8:201–231, 1977.
- [Hua 89] T. S. Huang et O. D. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on PAMI*, 11(12):1310–1312, December 1989.
- [Ike 81] K. Ikeuchi et B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(1–3):141–184, August 1981.
- [JO 93] P. Dupuis J. Oliensis. A global algorithm for shape from shading. *Proceedings of the 4th International Conference on Computer Vision, Berlin, Germany*, pages 692–701, May 1993.
- [Koe 80] J.J. Koenderink et A.J. Van Doorn. Photometric invariants related to solid shape. *Optica Acta*, 27:981–996, 1980.
- [Koe 84] J.J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.
- [Koe 90] J.J. Koenderink. *Solid Shape*. The MIT Press, Cambridge massachusetts, 1990.
- [Kol 90] D. Koller, N. Heinze, et H.H. Nagel. Algorithmic characterization of vehicle trajectories from image sequences by motion. *Computer Vision and Pattern Recognition*, 1990.
- [Kri 90] D.J. Kriegman et J. Ponce. On Recognizing and Positioning Curved 3-D Objects from Image Contours. *IEEE Transactions on PAMI*, 12(12):1127–1137, December 1990.

- [Kut 91] K.N. Kutulakos et C.R. Dyer. Recovering shape by purposive viewpoint adjustment. Technical Report no. 1035, University of Wisconsin, Madison, USA, August 1991.
- [Lau 87] P. J. Laurent. Courbes ouvertes ou fermées par B-splines régularisées. Rapport no. RR 652-M-, IMAG, Grenoble, France, March 1987.
- [Law 82] D.T. Lawton. Motion analysis via local translational processing. *IEEE Workshop on computer vision: Representation and Control*, pages 59–72, 1982.
- [Len 87] R.K. Lenz et R.Y. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology. *Proceedings of IEEE International Conference on Robotics and Automation, Raleigh, USA*, pages 68–75, 1987.
- [LH 81] H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. *Nature*, volume 293, pages 133–135. XX, September 1981.
- [Lim 88] H. S. Lim et T. O. Binford. Curved surface reconstruction using stereo correspondence. *Image Understanding Workshop*, pages 809–819, 1988.
- [Mac 77] A.K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, (8):99–118, 1977.
- [Mar 80] D. Marr et E. Hildreth. Theory of Edge Detection. Ai-memo 518, MIT, 1980.
- [May 79] P.S. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, 1979.
- [May 90] S.J. Maybank. The projective geometry of ambiguous surfaces. Rapport no. 1623, Long Range Laboratory, GEC, Wembley, Middlessex, England, July 1990.
- [May 92] S.J. Maybank et O.D. Faugeras. A theory of self calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.
- [Mey 89] A. Meygret. *Vision Automatique appliquée à la détection d'obstacles dans un environnement routier*. Thèse de doctorat, Université de Nice – Sophia-Antipolis, 1989.
- [Mey 90] A. Meygret, M. Thonnat, et M. Berthod. A pyramidal stereovision algorithm based on contour chain points. *Proceedings of the 1st European Conference on Computer Vision, Antibes, France*, pages 83–88, April 1990.
- [Moh 86] R. Mohr et T.C. Henderson. Arc and path consistency revisited. *Artificial Intelligence*, 28:225–233, 1986.
- [Moh 92] R. Mohr, L. Quan, F. Veillon, et B. Boufama. Relative 3D reconstruction using multiples uncalibrated images. Rapport no. RT 84-I-IMAG LIFIA 12, LIFIA-IRIMAG, 1992.

- [Shi 87] Y. Shirai. *Three-Dimensional Computer Vision*. Springer-Verlag, 1987.
- [Su 89] B. Q. Su et D. Y. Liu. *Computational Geometry - Curve and surface Modeling*. Harcourt Brace Jovanovich, Publishers, 1989.
- [Sze 90] R. Szeliski. Fast Surface Interpolation Using Hierarchical Basis Functions. *IEEE Transactions on PAMI*, 12(6):513–528, June 1990.
- [Sze 91] R. Szeliski et D. Tonnesen. Surface Modeling with Oriented Particle Systems. Rapport no. CRL 91/14, Digital Equipment Corporation, Cambridge research Lab, December 1991.
- [Tau 92] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, et D. Kriegman. Parameterized Families of Polynomiales for Bounded Algebraic Curve And Surface Fitting. Rapport no. RC-18065, IBM, June 1992.
- [Ter 86] D. Terzopoulos. Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Transactions on PAMI*, 8(4):413–424, July 1986.
- [Ter 88] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on PAMI*, 10(4):417–438, July 1988.
- [Tsa 84] R.Y. Tsai et T.S. Huang. Uniqueness and estimation of 3D motion parameters of rigid bodies with curved surfaces. *IEEE Transactions on PAMI*, volume 6, pages 13–27. IEEE, 1984.
- [Tsa 87] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.
- [Vai 89] R. Vaillant et O.D. Faugeras. Using occluding contours for recovering shape properties of objects. *IEEE Workshop on Interpretation of 3D scenes*, pages 26–32, November 1989.
- [Vai 90a] R. Vaillant. *Géométrie différentielle et vision par ordinateur : détection et reconstruction des contours d'occultation de surfaces*. Thèse de Doctorat, Université de Paris-Sud, Orsay, France, December 1990.
- [Vai 90b] R. Vaillant. Using occluding contours for 3D object modeling. O. Faugeras, éditeur, *Proceedings of the 1st European Conference on Computer Vision, Antibes, France*, pages 454–464. Springer-Verlag, April 1990.
- [Vai 92] R. Vaillant et O.D. Faugeras. Using Extremal Boundaries for 3-D Object Modeling. *IEEE Transactions on PAMI*, 14(2):157–173, February 1992.
- [Wam 88] C.W. Wampler, A.P. Morgan, et A.J. Sommese. Numerical continuation methods for solving polynomial systems arising in kinematics. Rapport no. Technical Report GMR-6372, General Motors Research Labs, August 1988.
- [Woo 89] R.J. Woodham. Determining surface curvature with photometric stereo. *International Conference of Robotics and Automation*, 1989.

- [Zha 92] C.S. Zhao, S. Plossu, et R. Mohr. Poursuite et Correction de Contours Courbes dans une Séquence d'Images. Rapport, LIFIA-IRIMAG, June 1992.
- [Zha 93a] C. S. Zhao et R. Mohr. Une reconstruction dynamique de surfaces 3D de B-splines par le mouvement d'une caméra. *Quatrièmes journées ORASIS GDR-PRC Communication Homme-Machine*, pages 86-89, Mulhouse, FRANCE, October 1993.
- [Zha 93b] C.S. Zhao et R. Mohr. B-spline Patches for Surface Reconstruction. P.J. Laurent, A. Le Méhauté, et L.L. Schumaker, éditeurs, *Curves and Surfaces, Second International Conference organized by Association Française d'Approximation*, page 177, Chamonix Mont-Blanc, FRANCE, June 1993.
- [Zha 93c] C.S. Zhao, R. Mohr, et L. Quan. Global surface reconstruction through regularized B-Spline patches. *Geometric Methods in Computer Vision II, SPIE's 1993 International Symposium on Optical Instrumentation and Applied Science*, pages 134-145, July 1993.
- [Zha 94a] C.S. Zhao et R. Mohr. B-spline patches for surface reconstruction in computer vision. P.J. Laurent, A. Le Méhauté, et L.L. Schumaker, éditeurs, *Curves and Surfaces II*, Academic Press, Boston, USA, 1994.
- [Zha 94b] C.S. Zhao et R. Mohr. Global three dimensional surface reconstruction from occluding contours, soumis. *Computer Vision, Graphics and Image Processing*, 1994.
- [Zha 94c] C.S. Zhao et R. Mohr. Relative 3D regularized B-spline surface reconstruction through image sequences. J. Eklundh, éditeur, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*. Springer-Verlag, May 1994.
- [Zha 94d] C.S. Zhao, R. Mohr, et B. Boufama. Reconstruction de surfaces B-splines régularisées et tridimensionnelles à partir de contours d'occultation. *Technique et Science Informatique*, 1994.
- [Zha 94e] C.S. Zhao, R. Mohr, et B. Boufama. Reconstruction de surfaces tridimensionnelles à partir d'une caméra en mouvement. *Actes du 9ème Congrès AF-CET de Reconnaissance des Formes et Intelligence Artificielle, Paris, France*, volume 1, pages 407-418, January 1994.
- [Zha 94f] C.S. Zhao, R. Mohr, et G. Koscielny. Fusion of three dimensional B-spline surface patches, soumis. *Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel*. IEEE Computer Society Press, October 1994.
- [Zis 92] A. Zisserman, D.A. Forsyth, J.L. Mundy, et C.A. Rothwell. Recognizing general curved objects efficiently. J. Mundy et A. Zisserman, éditeurs, *Geometric Invariance in Computer Vision*, pages 228-265. The MIT press, 1992.

Résumé

Le cadre général de ce travail est l'étude de la représentation et la reconstruction de surfaces tridimensionnelles d'objets non polyédriques, lorsqu'ils sont observés avec des systèmes de vision passifs. Cette thèse est plus spécialement consacrée à la reconstruction de surfaces tridimensionnelles en utilisant des surfaces B-splines, à partir de l'observation du mouvement des contours occultants, dans une séquence d'images calibrées.

Nous présentons plusieurs résultats auxquels nous avons abouti: Premièrement, nous introduisons un algorithme pyramidal de suivi de contours courbes et la correction des anomalies de contours dans une séquence d'images. Deuxièmement, une méthode originale de reconstruction de surfaces B-splines tridimensionnelles est développée à partir d'une séquence d'images calibrées. Finalement, nous proposons une approche de raccordement de différentes surfaces B-splines reconstruites en utilisant le maillage triangulaire et l'interpolation de surfaces. Des résultats sur des données synthétiques et des données réelles sont présentés.

Les applications potentielles de ce travail sont très importantes dans le cadre des problèmes de modélisation d'objets non polyédriques dans une séquence d'images calibrées.

Mots clés : Vision par ordinateur, Contours d'occultation, Objets non polyédriques, Suivi de contours courbes, Reconstruction de surfaces, Courbes et surfaces B-splines, Maillage triangulaire, Interpolation de surfaces.

Abstract

The work presented here falls into the general framework of representation and reconstruction of three dimensional surfaces of non polyhedral objects from series of images taken from a moving camera. This thesis is specifically dedicated to surface reconstruction using three dimensional B-spline surface patches based on the motions of occluding contours tracked through calibrated image sequences.

We present a hat full of results from this work: First we present a pyramid type algorithm for tracking and smoothing curve contours from an image sequence. Secondly an original method for reconstructing three dimensional surface by using B-spline surface patches from the recovered motion of occluding contours is presented. Finally we demonstrate an approach for merging overlapping B-spline surface patches using mesh triangulation and surface interpolation. Results for reconstruction based on synthetic and real data are presented.

The potential applications of this work are of great importance to the more general problem of non polyhedral objects reconstruction from calibrated image sequences.

Keywords : Computer vision, Occluding contours, Non polyhedral objects, Curve contours tracking, Reconstruction of three dimensional surfaces, B-spline curves and surface patches, Mesh triangulation, Surface interpolation.