



HAL
open science

Modélisation automatique d'objets non polyédriques par observation monoculaire

Emmanuel Arbogast

► **To cite this version:**

Emmanuel Arbogast. Modélisation automatique d'objets non polyédriques par observation monoculaire. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1991. Français. NNT: . tel-00339969

HAL Id: tel-00339969

<https://theses.hal.science/tel-00339969>

Submitted on 19 Nov 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.

THESE

présentée par

Emmanuel Arbogast

pour obtenir le grade de DOCTEUR

de l'INSTITUT NATIONAL POLYTECHNIQUE DE
GRENOBLE

(Arrêté ministériel du 23 novembre 1988)

spécialité: INFORMATIQUE

=====
Modélisation automatique d'objets non polyédriques
par observation monoculaire

=====
Date de soutenance: 12 Juillet 1991

Composition du jury:

Président: M. Philippe Cinquin

rapporteurs: M. Andrew Blake
M. Olivier Faugeras

examineurs: M. Radu Horaud
M. Roger Mohr

Thèse préparée au sein du laboratoire LIFIA-IMAG

REMERCIEMENTS

Je remercie la communauté européenne et le projet BRA FIRST pour le financement qui a rendu possible ma thèse.

Je remercie le PRC Communication Homme Machine et le groupement Orasis pour certaines des séquences d'images calibrées utilisées dans ce mémoire, et tout spécialement le laboratoire Irisa de Rennes qui a effectué les prises d'images.

Je remercie la société ITMI qui m'a fourni les autres séquences d'images calibrées .

Je remercie l'INPG et le laboratoire LIFIA pour m'avoir accueilli.

Je remercie l'équipe système pour m'avoir supporté pendant trois ans malgré toutes mes récriminations, et pour avoir constamment amélioré l'organisation du réseau du LIFIA.

Je remercie les équipes *voir* et *vision* pour avoir mis à ma disposition les matériels et logiciels indispensables à ce travail. Je remercie tout particulièrement Philippe Bobet, Jim Crowley, Yves Demazeau, Christophe Discours et Augustin Lux pour l'intérêt qu'ils ont porté à mes travaux et pour les discussions que nous avons eues.

Je tiens enfin surtout à remercier chaleureusement les personnes suivantes sans qui cette thèse ne serait pas :

Merci Pierre-Jean Laurent d'avoir pris le temps de me recevoir, et merci pour ces conseils qui se sont révélés extrêmement précieux.

Merci Radu Horaud pour les discussions fertiles que nous avons eues, et pour avoir mis à ma disposition le logiciel de traitement d'images FEAGRO d'où viennent toutes les données présentées dans ce mémoire.

Merci Patrick Gros, Luce Morin et Françoise Veillon d'avoir été là, de m'avoir écouté, corrigé et aidé dans les moments de doute.

De grands mercis à mon directeur de recherche et ami Roger Mohr pour les batailles acharnées pour régler mon problème de financement, pour les discussions scientifiques si constructives et enrichissantes, pour m'avoir enseigné la vraie voie de la Recherche Scientifique, faite de confiance et de probité intellectuelle, et pour la chaleur humaine et la compréhension d'autrui.

Merci à mes parents qui ont su me comprendre, me faire grandir en confiance, me permettre d'atteindre mes buts en me fournissant sans jamais faillir l'aide dont j'ai pu avoir besoin.

Merci enfin à ma femme Cynthia de m'avoir pardonné pour toutes les angoisses, les courtes nuits, les weekends inexistantes, enfin ... pour la vie de fous que je nous ai fait vivre, sans regret aucun, depuis le début de cette thèse.

Et merci à la Vie et aux challenges intellectuels extraordinaires qu'elle nous offre.

Table des matières

1	Introduction	1
2	Des images aux contours	5
2.1	Les caméras	5
2.2	La projection perspective	6
2.3	Les corrections géométriques du modèle caméra	8
2.4	Etalonnage de la caméra	9
2.5	Détection des points de contraste	10
2.6	Extraction des chaînes de points de contour	12
2.7	La sémantique des contours	12
2.7.1	Courbes de discontinuité	13
2.7.2	Marquages de surface	13
2.7.3	Courbes critiques	13
2.7.4	Autres types de courbes	14
2.8	La représentation des contours	16
2.8.1	Présentation générale des B_splines	17
2.8.2	L'ajustement des données aux moindres carrés	18
2.8.3	Ajustement avec régularisation	19
2.8.4	Application aux courbes du plan	19
2.8.5	Résultats	20
2.8.6	Estimation de la courbure	20
2.9	Post-traitement des chaînes de contours	23
2.9.1	Méthodes existantes	25
2.9.2	Rappels sur la détection des points de contraste	27
2.9.3	Détection des lieux de discontinuité de la courbure	28
2.9.4	Détection des discontinuités de tangence	29
2.9.5	Combinaison des critères de coupure	29
2.9.6	Résultats expérimentaux	30
2.9.7	Conclusion	33
2.10	Le suivi des contours	33

3	Reconstruction de surface	37
3.1	Introduction	37
3.2	Etat de l'art	38
3.3	Définitions	40
3.4	Les notations du problème	42
3.5	La reconstruction de surface	43
3.5.1	Principe	43
3.5.2	Théorie générale de la reconstruction pour une surface	45
3.5.3	Cas particulier d'une courbe dans l'espace	48
3.5.4	Remarque : indépendance de la paramétrisation en s	48
3.5.5	Estimation des courbures totales et moyennes	51
3.6	Implémentation	53
3.7	Résultats expérimentaux	54
3.7.1	Données synthétiques	55
3.7.2	Données réelles	59
3.7.3	Analyse de la précision	77
3.8	Conclusions	77
4	Autocalibration du mouvement	79
4.1	Introduction	79
4.1.1	Caméra mobile ou objet mobile	80
4.1.2	Calibration dynamique avec un système de mires	80
4.1.3	Repérage par rapport à des balises	80
4.2	Etat de l'art en autocalibration du mouvement	81
4.3	Principe de l'autocalibration	83
4.4	Analyse mathématique	84
4.4.1	La cinématique du solide : quelques rappels	85
4.4.2	Notations du problème	87
4.4.3	Les équations de l'autocalibration	87
4.4.4	La résolution	90
4.4.5	L'algorithme de résolution	91
4.4.6	Retour sur la reconstruction	93
4.5	Résultats expérimentaux	93
4.5.1	Données synthétiques	94
4.5.2	Données réelles	100
4.6	Conclusions	101
5	Conclusions	105
A	Présentation générale des B_splines	107
A.1	Détermination des fonctions de base des B_splines	107
A.2	L'ajustement des données aux moindres carrés	109
A.3	Ajustement avec terme de régularisation	110

B	La correction radiale du modèle sténopé	113
C	Le filtrage optimal par filtres récursifs	115
C.1	Filtrage optimal	115
C.2	Filtres récursifs	117
C.3	Lissage par filtres récursifs	118
C.4	Dérivation par filtres récursifs	118
D	Géométrie différentielle des courbes et surfaces	121
D.1	Courbes	121
D.2	Surfaces régulières	122
D.3	La première forme fondamentale	122
D.4	La deuxième forme fondamentale	123
D.4.1	Carte de Gauss	123
D.5	Classification des types de surface	125
E	Rappels de cinématique du solide	127
F	Comparaison avec le flot optique	129
F.1	Equivalence avec le flot optique	129
F.2	Conséquences	133
	Bibliographie	135

Chapitre 1

Introduction

“I had stopped at a florist on my way to his apartment and bought myself an extravagant red rose for my buttonhole. Now I removed this and handed it to him. He took it like a botanist or morphologist given a specimen, not like a person given a flower.

‘About six inches in length,’ he commented. ‘A convoluted red form with a linear green attachment.’

‘Yes,’ I said encouragingly, ‘and what do you think it *is*, Dr P.?’

‘Not easy to say.’ He seemed perplexed. ‘It lacks the simple symmetry of the Platonic solids, although it may have a higher symmetry of its own ... I think this could be an inflorescence or flower.’

‘Could be?’ I queried.

‘Could be,’ he confirmed.

‘Smell it,’ I suggested, and he again looked somewhat puzzled, as if I had asked him to smell a higher symmetry. But he complied courteously, and took it to his nose. Now, suddenly, he came to life.

‘Beautiful!’ he exclaimed. ‘An early rose...’ [Sac 87]

Extrait d’un dialogue entre le neuro-psychiatre américain Oliver Sacks et son patient, le Dr. P. .

Le Dr. P. est atteint d’un cancer du cortex visuel, et il n’est plus capable de reconnaître un objet “d’un coup d’œil” ; il lui faut l’explorer visuellement, et induire logiquement la réponse. Ainsi, il ne reconnaît ses amis que lorsqu’ils ont un trait caractéristique, etc. Le Dr. P. est cependant parfaitement opérationnel, et seules quelques aberrations de comportement révèlent son trouble physiologique.

Le comportement visuel du Dr. P. est semblable à celui d’un programme sophistiqué de vision par ordinateur d’aujourd’hui : il extrait la description la plus riche possible de l’objet perçu, et engendre un ensemble d’hypothèses d’où sort généralement la réponse quant au “quoi” (il n’a cependant aucun

problème quant au “où”). La composante humaine, caractérisée par la reconnaissance immédiate des objets familiers, a disparu. Il ressort que cette approche semble suffire pour une large gamme de comportements humains, mais en même temps qu’elle ne constitue qu’une faible part des capacités visuelles humaines.

Cet exemple est la preuve vivante que la vision par ordinateur telle qu’envisagée à ce jour est *possible*. La vision ne relève pas d’un processus global (ou magique !), mais bien d’une phase de description analytique de l’image, suivie ou accompagnée d’une phase de reconnaissance. C’est la démarche la plus courante étudiée dans le domaine de la vision par ordinateur.

Comme les chercheurs en vision ont à disposition un système de vision parfaitement fonctionnel, il est tentant de d’essayer de simuler certaines des capacités visuelles humaines.

Les expériences de psychologie expérimentale débordent d’exemples montrant que la vision humaine est sensible à la couleur, à la luminance, aux contours, à la direction des contours, aux ombres, aux reflets, au mouvement, à la stéréovision, bref à à peu près tout.

Les chercheurs en vision par ordinateur arrivent globalement aux mêmes conclusions : il est possible d’inférer des résultats riches de sens quant aux objets perçus, à partir de chacune de ces primitives¹.

La différence majeure entre le cerveau humain et les logiciels informatiques actuels tient en quelques mots : la vision humaine fonctionne bien ! Les algorithmes, eux, en sont encore au stade où ils ne fonctionnent qu’approximativement dans des conditions réalistes, et la collaboration entre plusieurs algorithmes est souvent impossible.

Compte tenu de cet état de l’art décevant, beaucoup d’efforts restent à consacrer au bas niveau de la vision pour améliorer les performances des algorithmes isolés, avant que de les faire collaborer entre eux.

La vision par ordinateur revient en effet de loin. Les efforts initiaux en vision ont consisté en l’analyse de scènes parfaites, et simples ; c’est la période du monde des blocs. Les données sont parfaites, les problèmes déjà difficiles à résoudre. La désillusion arrive, et vite, car il apparaît que ces données parfaites n’existent jamais dans les cas de prise d’images réelles, et que les algorithmes développés sont parfaitement incapables de gérer les erreurs. La remise en cause est douloureuse, et les chercheurs en vision par ordinateur jurent, mais un peu tard, qu’on ne les y prendra plus.

1. On pourra trouver un état de l’art dans des ouvrages généraux sur la vision par ordinateur comme [Bal 82] [Hor 86] [Coh 82]

Depuis une dizaine d'années, le travail en vision est reparti sur de nouvelles bases, en travaillant sur des données réelles. L'état de l'art avance peu à peu, en élargissant au fur et à mesure le domaine d'application des algorithmes et la précision des descriptions.

La valeur du signal de luminance étant peu fiable (les conditions d'éclairage peuvent changer), on s'est attaché à extraire des primitives robustes qui ne dépendent que de la variation du signal de luminance. Ces primitives dépendent de la structure des objets observés, et non de la nature ou de la puissance des éclairages utilisés. Les contours, ou points de maximum de contraste, sont des candidats privilégiés. Des points particuliers isolés, ou coins, ont d'abord été utilisés, puis les contours ont été approximés par des segments de droite, dans l'hypothèse d'un monde polyédrique. Ce n'est que récemment que la technique est devenue suffisamment mûre pour aborder la question des courbes et des surfaces directement, dans le cas des scènes non nécessairement polyédriques. Ce travail se situe dans cette perspective.

Nous nous proposons dans ce mémoire de présenter une méthodologie de travail dans le cas d'une vision monoculaire, avec mouvement, et pour des scènes non nécessairement polyédriques comportant des surfaces quelconques. En particulier, nous aborderons le problème de la représentation des contours, de la reconstruction de surface, et celui de la détermination du mouvement de la caméra par rapport à la scène par observation de lignes quelconques.

Le chapitre 2 aborde les questions de la représentation robuste des contours courbes et de leur segmentation en des points particuliers. On y montrera l'utilité des B-splines régularisées [Lau 87] pour la représentation des contours en vision, et l'utilisation originale de techniques standards de détection de contours [Der 87a, Can 86] pour la segmentation aux extrêmes de courbure ou aux discontinuités de tangence. Des résultats expérimentaux montrent l'intérêt de ces méthodes pour la description des images.

Le chapitre 2.10 couvre les techniques classiques de suivi de primitives le long d'une séquence d'images.

Le chapitre 3 traite de la reconstruction des lignes et surfaces dans le contexte d'une caméra en mouvement dont le déplacement est connu. Ce travail original est à rapprocher de techniques équivalentes développées indépendamment à Oxford par R. Cipolla [Bla 90] ou à Sophia Antipolis [Vai 89, Vai 90]. La technique utilisée est celle de la géométrie différentielle, et l'on montrera comment ce problème se ramène à celui, classique, de l'enveloppe de plans. Diverses expérimentations permettent de se rendre compte de la qualité des résultats, ainsi que des limites d'application de cette méthode.

La reconstruction de surface nécessite une connaissance précise des pa-

ramètres du mouvement de la caméra par rapport à l'objet observé. Pour suppléer à la faible précision des capteurs montés sur les robots, nous nous intéressons aux techniques d'autocalibration du mouvement. Le chapitre 4 propose une solution simple et nouvelle au problème de la détermination du mouvement à partir du suivi de contours courbes, améliorant les résultats existants dûs à O. Faugeras [Fau 90]. Du fait de l'absence sur ces courbes de points particuliers et de segments, les techniques d'autocalibration existantes sont en effet inapplicables.

Chapitre 2

Des images aux contours

La vision par ordinateur est un projet ambitieux qui doit permettre l'interprétation par ordinateur d'une scène perçue grâce à des capteurs de type visuel.

L'approche la plus courante consiste à modéliser la chaîne de mesure de façon à pouvoir formaliser mathématiquement les propriétés recherchées.

Nous nous restreindrons dans ce chapitre à la description succincte des parties de la chaîne de mesure qui nous serviront pour ce travail, c'est-à-dire la description des caméras utilisées, leur modèle idéal ainsi que les corrections géométriques couramment faites, la calibration de ces caméras, l'extraction des contours visibles dans les images, leur représentation et leur segmentation.

Le lecteur pressé et averti pourra commencer sa lecture directement au paragraphe 2.7, qui définit le vocabulaire utilisé et introduit le problème de la représentation et de la segmentation des contours.

2.1 Les caméras

Un capteur de type visuel est un capteur passif (par opposition aux capteurs actifs utilisant le laser ou le sonar) semblable dans le principe, sinon dans la forme, à un œil. Nous appellerons par la suite un tel capteur "caméra". Les caméras utilisées en vision par ordinateur sont du type CTD (Charge Transfer Device), et suivent le schéma de la figure 2.1.

On retrouve, comme dans un appareil photographique classique, un boîtier, un système de lentilles et une surface sensible. La particularité des caméras de vision par ordinateur réside dans le type de la surface sensible, qui consiste en une matrice plus ou moins dense de capteurs élémentaires, sensibles aux photons. Deux types de technologies cohabitent, dont la différence principale de principe est l'extraction des informations de la matrice :

- La technologie CCD (Charge-Coupled Device) extrait l'information de

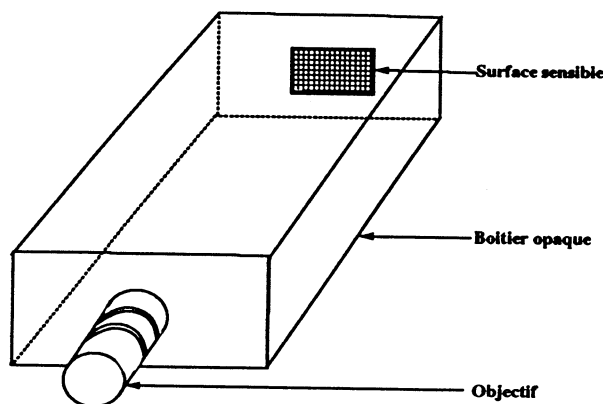


Figure 2.1. Schéma d'une caméra utilisée en vision par ordinateur

cette matrice ligne par ligne en engendrant un signal vidéo ; chaque ligne peut être considérée comme un registre à décalage. Cette technologie pose des problèmes délicats de synchronisation et de détection des capteurs élémentaires défectueux.

- Le technologie CID (Charge-Injection Devices) où l'adressage (X,Y) individuel de chaque capteur élémentaire se fait suivant des techniques similaires à celles utilisées pour les mémoires d'ordinateurs. La technologie CID est plus récente, et est la voie la plus prometteuse actuellement pour les capteurs de vision par ordinateur car les problèmes de synchronisation disparaissent, et que les capteurs élémentaires défectueux sont facilement décelables.

Les capteurs disponibles actuellement ont une surface sensible plane rectangulaire. De nombreuses raisons théoriques motivent le développement de "rétines sphériques" c'est-à-dire de surfaces sensibles sphériques, similaires aux rétines animales. Ces rétines sphériques simplifient la modélisation de la caméra.

2.2 La projection perspective

La modélisation la plus simple d'une caméra, et la plus communément employée dans la communauté "vision par ordinateur" est une approximation du modèle de lentille mince, dit "trou d'épingle", modèle sténopé, ou projection perspective pure. Les informations perçues par la caméra sont la

projection centrale de points dans la scène sur le plan image de la caméra (voir la figure 2.2).

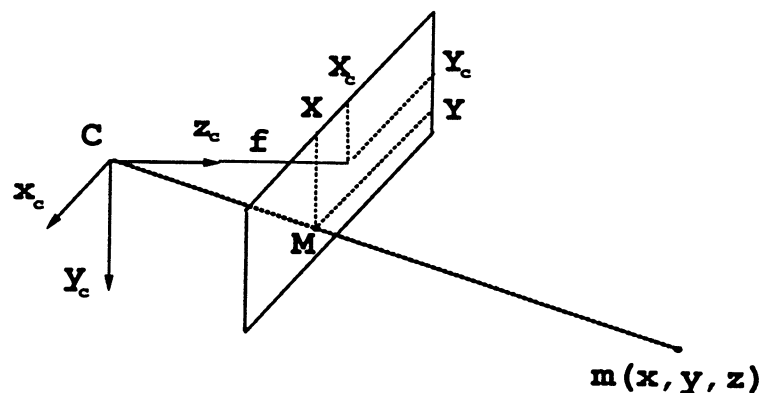


Figure 2.2. Modèle sténopé, ou projection perspective pure

La modélisation de la caméra suppose le plan image de la caméra orthogonal à l'axe optique de la lentille.

Le modèle de la caméra est exprimé dans le repère de référence de la caméra, dont l'origine est le centre focal C . (C, x_c) et (C, y_c) sont les axes parallèles au plan image, correspondant aux directions des lignes et colonnes des capteurs élémentaires de la surface sensible, et (C, z_c) est orienté vers le plan image et vers la scène. Le repère (C, x_c, y_c, z_c) est orthonormé direct.

Les paramètres intrinsèques permettant alors de modéliser la caméra, suivant les notations de F. Chaumette [Cha 90], sont :

- Les coordonnées (X_c, Y_c) en pixels de l'intersection de l'axe optique avec le plan image.
- La focale f de la lentille.
- La taille des pixels l_x et l_y suivant les deux axes x_c et y_c .

Dans le repère ainsi défini, un point $m(x,y,z)$ de la scène se projette en un point $M(X,Y)$ avec :

$$X = f \cdot \frac{x}{z} \quad , \quad Y = f \cdot \frac{y}{z}$$

Le repérage en pixels sur le plan image est défini comme dans la figure 2.3 :

Si l'on exprime M par sa position en pixels $M_p(X_p, Y_p)$ on obtient :

$$X_p = X_c + \frac{X}{l_x} \quad , \quad Y_p = Y_c + \frac{Y}{l_y}$$

L'ensemble de ces paramètres peut également être codé globalement sous la forme d'une matrice de projection perspective, qui peut être utilisée en l'absence de la connaissance individuelle des paramètres explicites et implicites.

$$X_d = X_p + K_d \cdot (X_p - X_c) \cdot [(X_p - X_c)^2 + (Y_p - Y_c)^2] \quad (2.1)$$

$$Y_d = Y_p + K_d \cdot (Y_p - Y_c) \cdot [(X_p - X_c)^2 + (Y_p - Y_c)^2] \quad (2.2)$$

Soit le modèle complet de la caméra :

$$X_d = X_c + F_x \cdot \frac{x}{z} + K_d \cdot F_x \cdot \frac{x}{z} \cdot \left[\left(F_x \cdot \frac{x}{z} \right)^2 + \left(F_y \cdot \frac{y}{z} \right)^2 \right] \quad (2.3)$$

$$Y_d = Y_c + F_y \cdot \frac{y}{z} + K_d \cdot F_y \cdot \frac{y}{z} \cdot \left[\left(F_x \cdot \frac{x}{z} \right)^2 + \left(F_y \cdot \frac{y}{z} \right)^2 \right] \quad (2.4)$$

avec $F_x = \frac{f}{l_x}$ et $F_y = \frac{f}{l_y}$.

On remarque que seuls les quotients $F_x = \frac{f}{l_x}$ et $F_y = \frac{f}{l_y}$ de la valeur de la focale et de la taille des pixels sont significatifs. En d'autres termes, on peut choisir arbitrairement $f=1$ et obtenir $F_x = \frac{1}{l_x}$ et $F_y = \frac{1}{l_y}$.

Le problème posé est souvent le problème inverse :

Soit un point observé M_d ; quel est le point M qui correspond à la transformation projection perspective sans déformation radiale ?

La solution, détaillée dans l'annexe B, est :

$$\begin{cases} X = \frac{X_r \cdot (1 + \epsilon_x)}{F_x} \\ Y = \frac{Y_r \cdot (1 + \epsilon_y)}{F_y} \end{cases}$$

avec :

$$\begin{cases} X_r = X_d - X_c \\ Y_r = Y_d - Y_c \\ \epsilon_x = \epsilon_y = -\frac{K_d \cdot R^2}{1 + 3K_d \cdot R^2} \\ R^2 = X_r^2 + Y_r^2 \end{cases}$$

La caméra utilisée pour les expérimentations qui seront présentées aux chapitres 3 et 4 est modélisée en prenant en compte la déformation radiale.

2.4 Etalonnage de la caméra

L'étalonnage de caméra consiste à prendre une caméra inconnue et à déterminer ses paramètres grâce à une expérimentation adéquate.

Le principe de l'expérimentation est résumé dans la figure 2.5.

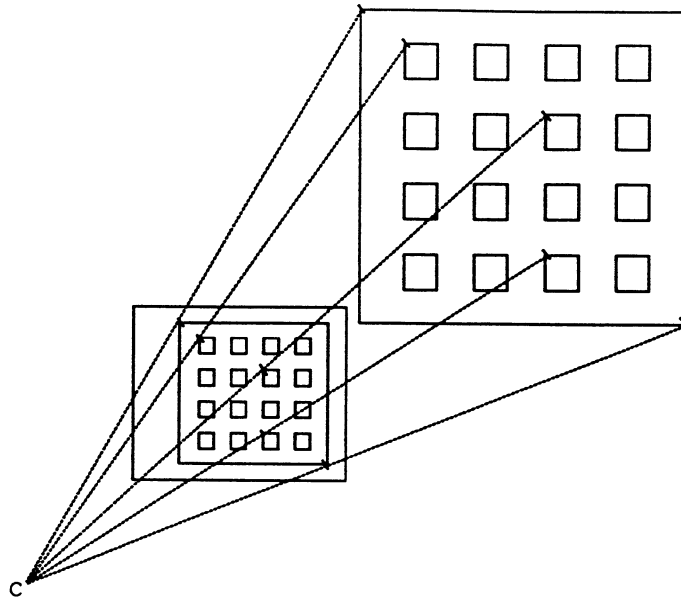


Figure 2.5. Dispositif expérimental pour la calibration

En plus des paramètres intrinsèques de la caméra interviennent ses paramètres extrinsèques, position et orientation de la caméra par rapport au repère de la scène. Une mire calibrée (tous les points de la mire ont une position parfaitement connue dans l'espace) est visualisée par la caméra. On cherche à minimiser la distance entre la projection prédite par le modèle et la position mesurée sur le plan image, pour l'ensemble des points de la mire. L'ensemble des paramètres qui minimisent cette distance détermine les paramètres intrinsèques et extrinsèques de la caméra. Le caractère non linéaire du critère à minimiser, dans le cas d'une caméra avec déformation radiale, impose l'utilisation d'une technique de minimisation non linéaire itérative. Le lecteur intéressé est invité à se reporter à [Cha 90] et [Fau 87b] pour les détails de la mise en équation et de la résolution.

2.5 Détection des points de contraste

L'expérimentation en vision par ordinateur montrent que le signal de luminance acquis par la caméra est notablement bruité, et la communauté scientifique, en conséquence, a cherché à extraire du signal les informations les plus robustes possibles. L'unanimité s'est faite pour considérer les points de contraste comme des primitives robustes. Les points de contraste sont définis informellement comme des points à variation maximale du signal de luminance. Ces points sont effectivement localisés le long des frontières entre les zones éclairées de manière homogène, indépendamment de la puis-

sance ou de la direction de l'éclairage. Cette indépendance aux conditions d'éclairage explique la stabilité de ces points.

La détection de ces points de contraste se fait par l'utilisation d'un filtre qui donne une réponse maximale en ces points. Du fait du bruit présent dans le signal, ce filtre doit être combiné à un filtre moyenneur. Des filtres ont été proposés depuis longtemps :

- Roberts [Rob 63], Prewitt et Sobel proposent des solutions qui s'apparentent toutes au calcul du gradient du signal de luminance, un point de contraste étant un point de gradient maximal.
- La technique du calcul du laplacien, ou calcul d'une dérivée seconde isotrope, définit le point de contraste comme le zéro du laplacien. Cette technique est équivalente d'un point de vue théorique à la technique du gradient pour la détection de la position du point de contraste dans un signal de type échelon. Par contre elle présente le défaut de perdre la direction du contour et celui de devoir calculer une dérivée seconde qui amplifie l'influence du bruit dans le signal.

Canny [Can 83] a pour la première fois présenté une formalisation théorique de la réponse d'un filtre à un signal de type échelon bruité, et déterminé le filtre dont la réponse est optimale au sens d'un critère ; l'annexe C décrit en détail le critère utilisé. Même si les images réelles s'éloignent des conditions théoriques qu'il suppose, son filtre optimal donne une réponse bien meilleure que les filtres proposés précédemment. D'autres filtres inspirés par cette formalisation de l'approche sont nés [Der 87a] [She 86], qui présentent tous cet aspect optimal selon un critère, et qui fournissent tous d'excellents résultats en pratique. Des progrès restent cependant à faire dans les zones où le gradient est particulièrement perturbé comme les coins où au moins trois contours se rencontrent, ou le long de contours non rectilignes.

La figure 2.6 représente un exemple d'image.

Le gradient du signal de luminance est généralement calculé pour chaque position des pixels de l'image. On obtient ainsi une tabulation de la fonction gradient. En chaque point, le gradient est composé de deux composantes suivant les axes du plan image. Le module de ce gradient est une information capitale concernant l'existence d'un point de contraste. La structure obtenue correspond à une structure d'image, où le signal est le module du gradient et non la luminance. La figure 2.7 visualise l'image obtenue en calculant le module du gradient en chaque pixel de l'image précédente.

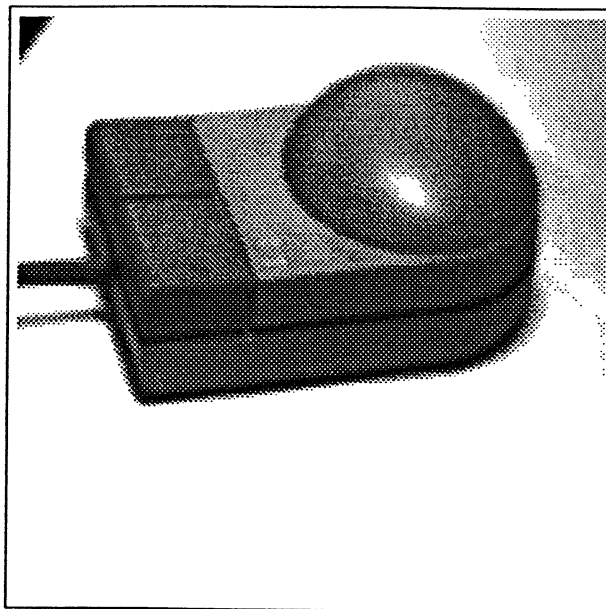


Figure 2.6. Exemple d'image

2.6 Extraction des chaînes de points de contour

A l'issue du filtrage (du type gradient ou laplacien), on est en présence d'une image dont on veut extraire et organiser les points de contraste, ou points de contour. Les points de contour sont organisés le long de chaînes de points de contour qui marquent les frontières de régions à gradient plus ou moins homogène. Il s'agit donc de déterminer ces chaînes de points de contour en suivant les maxima ou les zéros de la sortie du filtre.

La technique de seuillage par hysteresis a prouvé son utilité dans le cas du filtrage de type gradient, en présence de bruit : il s'agit de ne commencer à suivre une chaîne de contour que lorsque le gradient a une valeur supérieure à un seuil, et d'abandonner le chaînage lorsque le gradient tombe en dessous d'un autre seuil [Gir 87]. Cette technique a l'avantage d'éliminer de nombreux maxima du signal dûs au bruit.

Nous utilisons un filtrage optimal du type gradient [Der 87a] dans le logiciel FEAGRO développé au laboratoire LIFIA.

La figure 2.8 montre les chaînes de contours détectées.

2.7 La sémantique des contours

Les contours observés sur le plan image sont la projection de courbes particulières de la surface des objets observés.

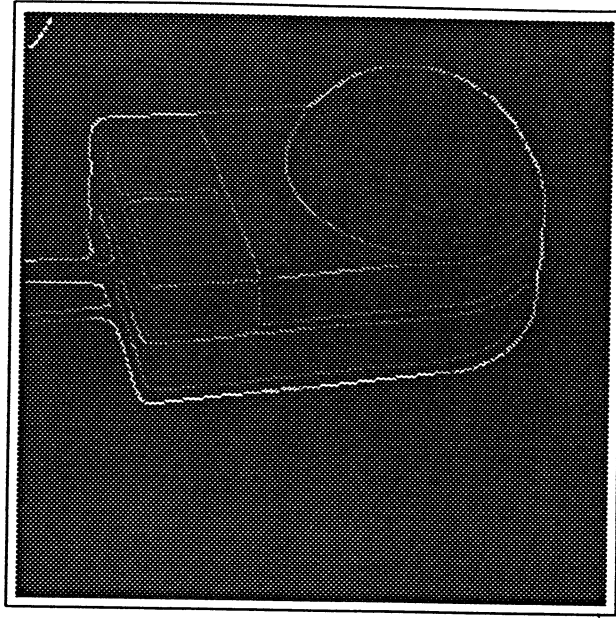


Figure 2.7. Gradient de l'image

Ces courbes peuvent être classées en plusieurs catégories dont les propriétés intrinsèques sont différentes, et qui par conséquent doivent être traitées de manière spécifique.

2.7.1 Courbes de discontinuité

On appelle courbe de discontinuité une courbe tracée sur la surface d'un objet telle que cette courbe est le lieu des points où le gradient de la surface est discontinu, soit une frontière entre deux surfaces C^1 . La figure 2.9 montre un exemple d'une telle ligne de discontinuité L , intersection de deux sphères.

Les arêtes des objets polyédriques appartiennent à cette catégorie.

Les courbes de discontinuité sont fixes relativement à l'objet.

2.7.2 Marquages de surface

Un autre type de courbe concerne les multiples marquages de surfaces, les textures présentes sur la surface, les inscriptions diverses, ..., courbes tracées sur une surface au moins C^1 , et fixes relativement à la surface, les rendant en cela indiscernables des courbes de discontinuité.

2.7.3 Courbes critiques

On appelle ligne de vue d'un point P de la scène la droite qui relie le centre focal C de la caméra au point P de la scène.

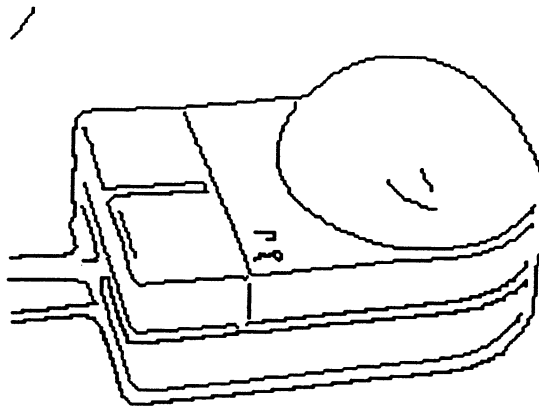


Figure 2.8. Chaînes de points de contour

On appelle courbe critique une courbe sur la surface d'un objet telle que la ligne de vue est tangente à la surface. La figure 2.10 montre deux exemples de courbes critiques, a) sur une sphère et b) sur un cylindre terminé par deux hémisphères.

Les courbes critiques perçues à la surface d'un objet dépendent de la surface de l'objet et de la position du centre optique de la caméra ; elles changent lorsque le point de vue change. Ce sont donc des courbes "virtuelles" de la surface de l'objet.

2.7.4 Autres types de courbes

Les trois types de courbes sus-cités partagent la propriété d'être intrinsèques à l'objet, ou à la position relative de l'objet par rapport à la caméra.

Il existe cependant deux autres types de courbes visibles à la surface d'un objet, nommément les ombres et les reflets portés sur la surface de l'objet, liés aux sources de lumière et à l'existence d'autres objets dans la scène.

Ces courbes sont mobiles sur la surface de l'objet, indépendamment de la géométrie de l'objet ou de la position de la caméra, et posent par conséquent des problèmes d'interprétation difficiles.

Le cas très particulier d'ombres ou de reflets fixes sur la surface de l'objet peut se ramener sans problèmes au cas précédent des marquages de surface. Il est supposé par la suite qu'ombres et reflets ne sont pas présents

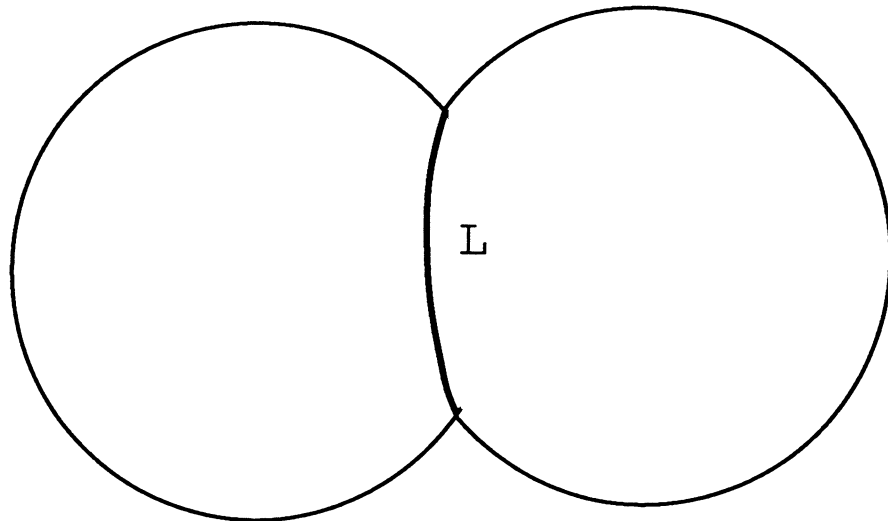
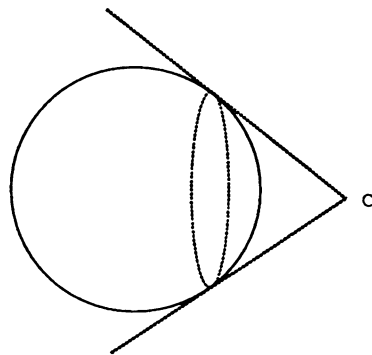
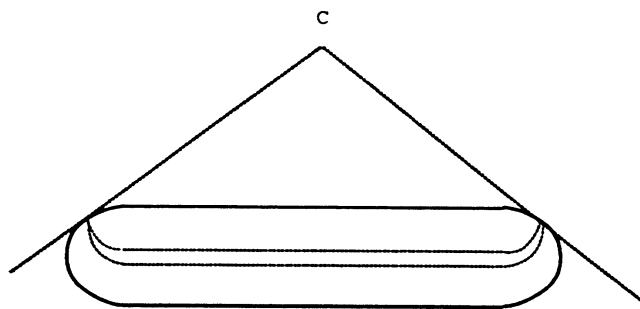


Figure 2.9. courbe de discontinuité : exemple



a)



b)

Figure 2.10. Courbes critiques : exemples

dans la scène observée, ou qu'ils sont filtrés par des préprocesseurs [Voo 87] [Kli 88] [Bla 85] adéquats qui les marqueront parmi l'ensemble des contours de l'image.

2.8 La représentation des contours

La généralisation de certaines techniques de vision au monde des objets non polyédriques passe par la représentation et la manipulation des contours curvilignes.

Les chaînes de points de contours, à l'état brut, présentent en effet des inconvénients majeurs :

1. La position des points de contours est bruitée, de par la discrétisation du plan image et de par les erreurs de détection.
2. La représentation est discrète, et l'on ne connaît que la position du contour en un nombre précis de points.
3. Le volume des données représentant les chaînes de points de contours est trop important, et on veut le réduire.

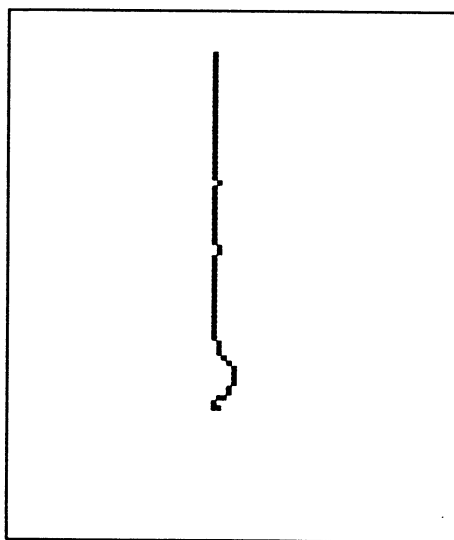


Figure 2.11. Une chaîne de points de contour représentant un segment de droite

La figure 2.11 est une illustration claire de ces deux problèmes. Le contour présenté *devrait* être un segment de droite, étant la projection d'une droite de l'espace.

Deux approches peuvent être envisagées pour corriger ces problèmes : l'approche filtrage et l'approche approximation fonctionnelle. Le filtrage consiste à lisser le signal discret par un filtre moyenneur, ce qui amoindrit dans une certaine mesure les problèmes liés au bruit, mais ne change

rien au caractère discret de la représentation. Cette approche doit être faite très soigneusement ainsi que le montre D. Lowe [Low 88], de façon à ne pas commettre de biais systématique sur la courbure des contours. L'approximation fonctionnelle se propose de représenter le contour par une ou plusieurs fonctions "lisses" (au moins C^1) ; cette proposition permet le filtrage du bruit car la fonctionnelle est lisse par définition, et permet d'obtenir une représentation continue du signal. La fonctionnelle permet entre autres choses une interpolation entre les pixels et la dérivation symbolique le long du contour.

L'application qui nous intéresse dans ce travail 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 donc été retenue.

Le choix de la fonctionnelle s'est tout naturellement porté sur les B_splines cubiques régularisées. En effet, ces fonctions possèdent de nombreux avantages :

- Le domaine des splines est un domaine bien étudié en théorie et bien dominé.
- Les B_splines permettent un ajustement très fin aux données. D'autres travaux récents corroborent cette conclusion [SM 90].
- Le degré 3 des B_splines cubiques utilisées est le degré minimum permettant d'obtenir des dérivées secondes continues, et donc une fonction de courbure continue (propriété dont l'utilité sera démontrée ultérieurement dans le paragraphe 2.9 sur la segmentation des contours).
- La régularisation (qui sera définie ultérieurement) permet de contrôler la forme de la B_spline vis-à-vis du bruit dans les données.
- L'algorithme d'ajustement de la fonctionnelle aux données a une complexité optimale (linéaire).

L'ajustement par B_splines cubiques régularisées a été formalisé et décrit en détail par P.J. Laurent [Lau 87]. La présentation qui suit en est directement inspirée.

2.8.1 Présentation générale des B_splines

Les B_splines doivent leur popularité à leur forte utilisation en CAO. On trouvera une présentation plus complète dans [Bar 87] et [BQ 89]. Il ne sera rappelé ici que les propriétés utilisées. Les notations utilisées sont

empruntées à [Lau 87], et le lecteur intéressé pourra trouver les détails en annexe A.

Les B_splines, formellement, sont des fonctions polynomiales par morceaux. Une B_spline de degré d est telle que les fonctions polynomiales la composant sont de degré d . Une B_spline de degré d a la propriété d'être C^{d-1} . Ainsi, on trouve couramment des B_splines linéaires (de degré 1), quadratiques (de degré 2), ou cubiques (de degré 3). Une B_spline est habituellement exprimée comme une combinaison linéaire de fonctions de base. Ces fonctions de base sont elles-mêmes polynomiales par morceaux.

Sans perte de généralité, on supposera une B_spline exprimée avec un paramètre t , $t \in [0, 1]$. On utilisera une subdivision régulière de $[0, 1]$ en n intervalles : $0 = u_0 < u_1 < \dots < u_n = 1$. On introduit des nœuds supplémentaires u_{-d}, \dots, u_{-1} et u_{n+1}, \dots, u_{n+d} de façon à pouvoir traiter les premiers et derniers intervalles comme dans le cas général. A chaque intervalle j est associé une fonction de base B_j , de degré d , dont le support est $[u_j, u_{j+d+1}]$. La forme générale de la B_spline Q est :

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

Les α_j sont appelés points de contrôle.

Les B_j sont les fonctions de base de la B_spline Q .

2.8.2 L'ajustement des données aux moindres carrés

Le problème de l'approche aux moindres carrés se pose ainsi :

soient $p + 1$ points (x_i, t_i) , $t_i \in [0, 1]$,

soit n le nombre d'intervalles réguliers dans $[0, 1]$,

soit Q , la B_spline de degré d recherchée :

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

L'approche aux moindres carrés consiste à minimiser le terme quadratique d'erreur suivant :

$$E = \sum_{i=0}^p (Q(t_i) - x_i)^2$$

Cette minimisation par rapport aux α_j se fait en annulant la dérivée partielle de E par rapport à chaque α_j , ce qui nous donne $n + d$ équations linéaires.

On obtient ainsi un système $M.X = B$, où X est le vecteur des α_j , et où M est une matrice $(n + d) \times (n + d)$ symétrique positive bande avec d sous-diagonales, ce qui permet d'utiliser des techniques numériques de résolution extrêmement efficaces, comme une variation de la méthode de Choleski pour des matrices bandes ; elle est linéaire par rapport au nombre de points de contrôle.

2.8.3 Ajustement avec régularisation

L'approche précédente souffre de deux problèmes :

1. il est nécessaire d'avoir au moins un point de donnée par intervalle, sans quoi le système est dégénéré.
2. il n'est pas possible de doser le degré de lissage voulu, ou d'imposer des contraintes supplémentaires sur la B_spline obtenue.

Il est possible, sans augmenter la complexité du système à résoudre, d'imposer une contrainte supplémentaire, permettant d'apporter une réponse aux deux points précédents [Lau 87] :

$$E' = \sum_{i=0}^p (Q(t_i) - x_i)^2 + \tau \int_0^1 (Q^{(d)}(t))^2 dt$$

La contrainte supplémentaire, pour une B_spline cubique, revient à minimiser la variation de la courbure, terme qui correspond parfaitement au lissage supplémentaire recherché ; les zones particulièrement bruitées sont en effet souvent caractérisées par une fonction de courbure très bruitée (la courbure est une fonction du deuxième ordre par rapport à la fonction d'origine, et le bruit est donc amplifié), donc à forte variation locale de courbure.

On dérive le terme supplémentaire par rapport aux α_j pour minimiser le nouveau E' obtenu. Le système à résoudre reste linéaire symétrique.

Le facteur τ est introduit pour doser le poids de ce terme régularisateur. On appellera τ le facteur de tension. Ce facteur de tension permet de passer continûment de la solution exacte aux moindres carrés à une solution dominée par ce terme régularisateur. On verra plus loin dans l'application comment on peut se servir de ce facteur de tension, et son importance pratique.

2.8.4 Application aux courbes du plan

Le cas précédent de courbes à valeurs réelles se généralise au cas de courbes dans le plan. La représentation paramétrique d'une courbe discrète du plan est un ensemble ordonné de points $(x_i(t_i), y_i(t_i))$. Pour approcher

cette courbe par B_spline, il suffit de considérer chacune des coordonnées séparément :

$$\begin{aligned}
 Q(t) &= \begin{cases} x(t) = \sum_{j=-d}^{n-1} X_j B_j(t) \\ y(t) = \sum_{j=-d}^{n-1} Y_j B_j(t) \end{cases} \\
 &= \sum_{j=-d}^{n-1} (X_j B_j(t), Y_j B_j(t)) \\
 &= \sum_{j=-d}^{n-1} V_j B_j(t)
 \end{aligned}$$

On notera que la solution reste très simple puisque le problème est séparable.

2.8.5 Résultats

La figure 2.12 est un exemple de B_splines appliquées aux courbes planes selon le principe présenté ci-dessus.

Elle montre l'influence de τ sur le lissage, et indique bien comment le choix de τ permet de contrôler la taille des détails recherchés, et par conséquent de contrôler le niveau de filtrage. Cette technique de filtrage est à rapprocher de filtrages multi-résolutions de type gaussien, en ce sens qu'il est possible de maintenir un ensemble de représentations de la courbe, à des résolutions de plus en plus fines. Les B_splines régularisées cependant présentent l'avantage sur le filtrage gaussien de ne pas déplacer outre mesure les centres de courbure [Low 88].

La figure 2.13 montre un second exemple de l'influence du terme régularisateur sur les contours, dans le cas d'une image plus complexe de la souris d'un micro-ordinateur.

La résolution de l'écran de visualisation ne permet pas cependant d'apprécier correctement la variation du lissage avec les valeurs de τ , mais permet tout de même de voir l'effet d'arrondi lié au lissage.

2.8.6 Estimation de la courbure

On étudie la stabilité de l'estimation de la courbure vis-à-vis du bruit de discrétisation. On observe un cercle synthétique de rayon 100 pixels, on fabrique la chaîne de contour correspondante, puis la spline régularisée qui l'approxime. On va étudier dans un premier temps l'importance du facteur de régularisation pour l'estimée de la courbure, puis l'importance du bruit. On rappelle que la courbure $C(t)$ est calculée symboliquement à partir de la spline par la formule:

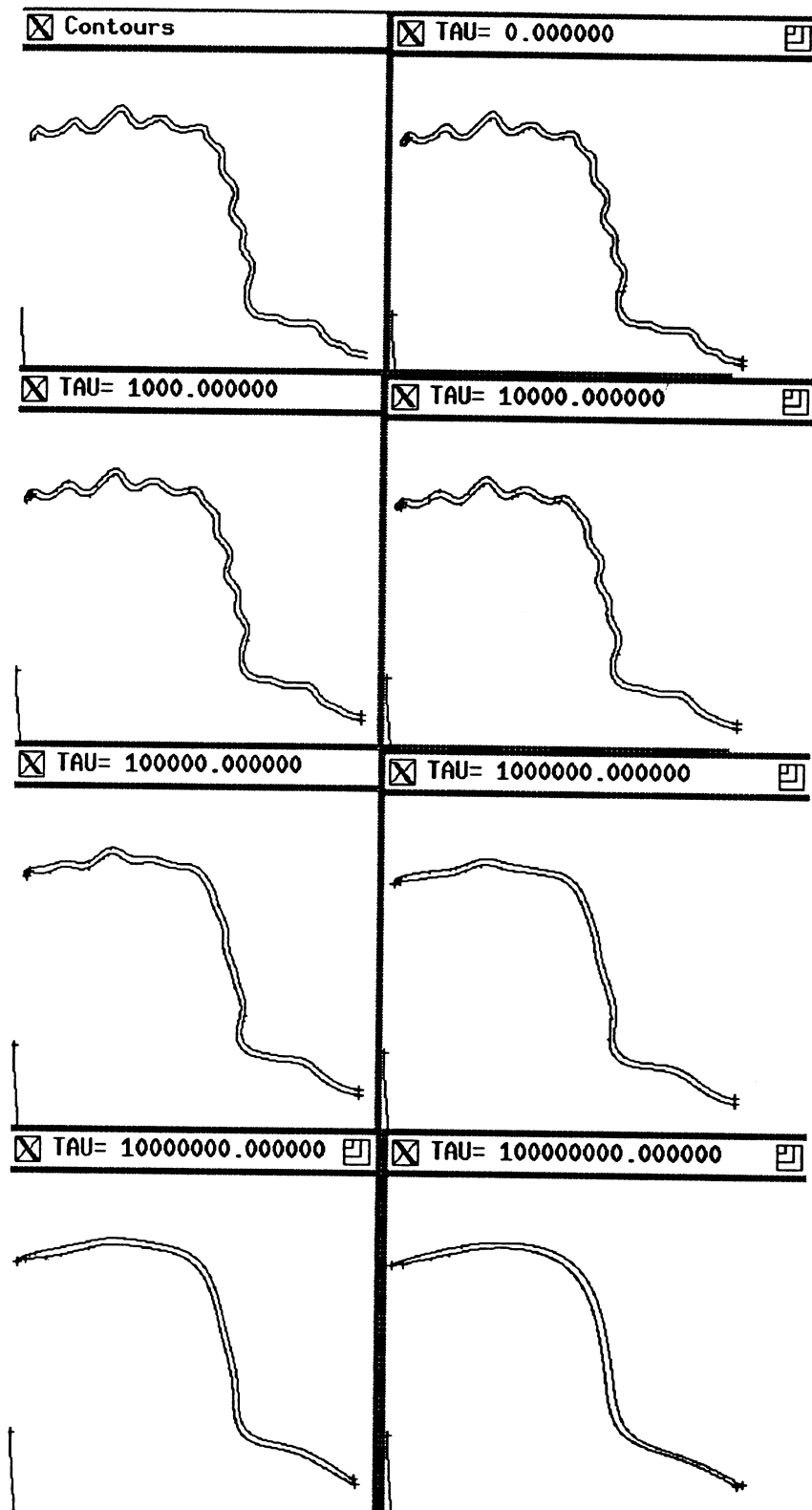


Figure 2.12. B-splines régularisées pour différentes valeurs de τ

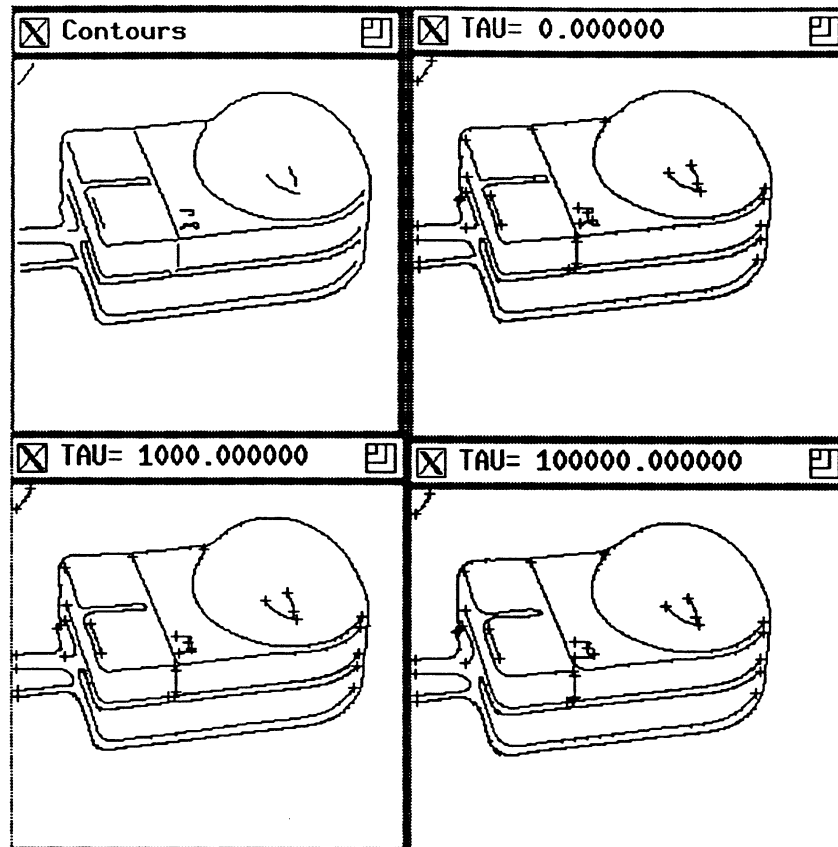


Figure 2.13. Lissage de la souris pour différentes valeurs de τ

$$C(t) = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}}$$

Influence du facteur de régularisation

Pour chacun test, on mesure la courbure moyenne avec et sans bruit, et l'écart-type des mesures avec et sans bruit.

Le bruit considéré est le seul bruit de discrétisation.

La table 2.1 nous montre bien l'effet réducteur de bruit du facteur de tension sur les données bruitées, en même temps que l'on voit l'effet de diminution de fidélité sur les données parfaites. On observe un optimum de 2,7% d'erreur pour $\tau = 1e7$ contre 16,8% d'erreur pour les B-splines classiques dans la colonne des données bruitées.

Influence du bruit et comparaison avec les splines simples

On compare les comportements des splines simples et des splines régularisées. Le bruit considéré dans cette expérimentation est un bruit blanc

τ	Sans bruit		Avec bruit	
	$\tilde{\rho}$	σ_ρ	$\tilde{\rho}$	σ_ρ
0	100.001	0.0988	100.068	16.8645
1	100.001	0.108	100.068	16.86
10	100.002	0.0883	100.064	16.842
1e2	100.001	0.098	100.035	16.65
1e3	100.004	0.143	99.84	15.20
1e4	100.011	0.054	99.519	10.20
1e5	100.034	0.22	99.71	5.894
1e6	100.111	0.60	99.687	2.874
1e7	100.36	1.587	100.066	2.781
1e8	101.156	4.206	100.723	4.59
1e9	103.699	11.22	103.42	11.516

Tableau 2.1. Courbure en fonction du facteur de tension

gaussien de moyenne nulle et d'écart-type croissant, exprimé en pixel, suivi d'une discrétisation. Le rayon est de 100 pixels et la tension pour les B_splines régularisées vaut 1e7.

σ_{bruit}	Splines simples		Splines régularisées	
	$\tilde{\rho}$	σ_ρ	$\tilde{\rho}$	σ_ρ
0	99.564	3.728	100.066	2.781
1	100.05	9.839	100.4	3.725
2	100.865	19.40	100.896	7.096
3	104.09	31.64	101.903	10.176
4	140.25	464.30	103.29	16.645
5	113.237	379.71	106.033	25.95
6	135.178	800.153	130.19	155.33

Tableau 2.2. Courbure en fonction du bruit pour splines simples et régularisées

La table 2.2 nous montre bien l'effet réducteur de bruit du facteur de tension sur les données bruitées.

2.9 Post-traitement des chaînes de contours

Le problème de la représentation des contours s'accompagne de celui de leur segmentation. En effet, les chaînes de contours obtenues ne correspondent pas aux contours attendus par les algorithmes ultérieurs d'analyse de la scène perçue :

- dans le cas classique des scènes polyédriques, les projections des arêtes sont souvent fusionnées en un contour unique, mais le découpage du contour en segments de droites permet facilement de segmenter correctement aux angles (sauf alignement accidentel de segments dans l'image).
- dans le cas de scène non nécessairement polyédriques, les contours de discontinuités et les contours critiques sont souvent fusionnés ; en effet, de manière courante, la variation de contraste entre le fond et l'objet fait que le contour extérieur de l'objet est extrait comme un seul contour, bien que puisse s'y enchaîner contours critiques et contours de discontinuités. La figure 2.14 montre un cas générique d'un tel contour. Le contour extérieur est généralement perçu comme un contour unique, alors qu'il est composé de portions de contours de discontinuité et de portions de contours critiques comme l'illustre la figure 2.15 où les contours de discontinuité sont en pointillés.

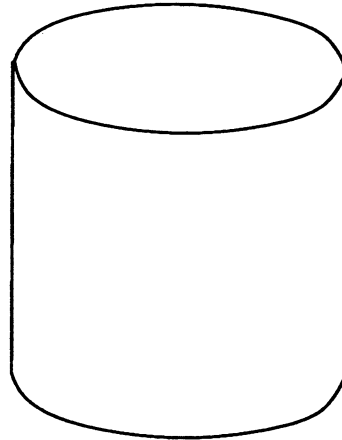


Figure 2.14. Contours de discontinuité et contours critiques sont fusionnés

L'image obtenue par projection des objets dont la surface est C^2 par morceaux est formée, en théorie, de contours sémantiquement différents qui se rejoignent en des points appelés jonctions.

La fonction de courbure d'une courbe paramétrique $(x(t), y(t))$ est définie de manière classique par :

$$C(t) = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}}$$

La figure 2.16 montre un catalogue des différentes jonctions existantes. La jonction en a) est une jonction avec extrême de courbure, sans discontinuité de courbure ni de tangence. La jonction en b) présente une discontinuité de tangence et de courbure. La jonction en c) présente une discontinuité de courbure sans discontinuité de tangence.

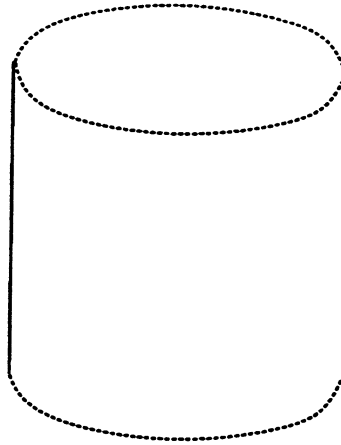


Figure 2.15. Contours de discontinuité (en pointillés)

Il est donc utile de segmenter les contours en sous-chaînes suivant des critères significatifs du point de vue sémantique pour l'objet, et du point de vue géométrique pour les contours. On pourra se référer à [Mal 87] pour une analyse complète des types de jonctions pouvant exister. A ce niveau, il suffit de noter que les discontinuités de tangence et les discontinuités de courbure sont des indices révélant une jonction, et donc un point de coupure de la chaîne.

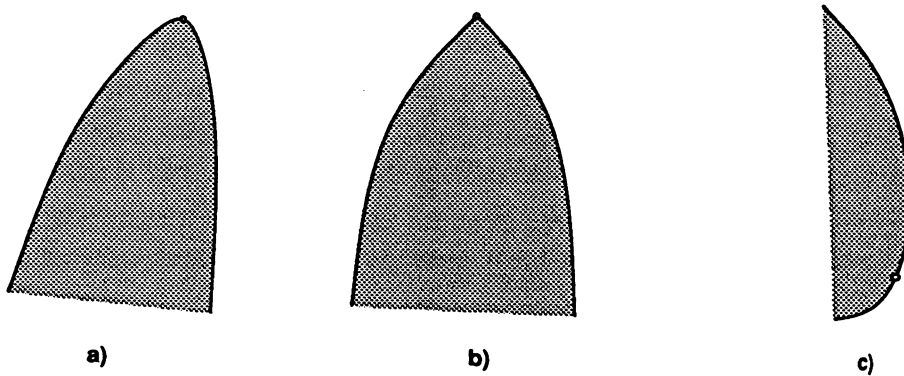


Figure 2.16. Catalogue des jonctions

2.9.1 Méthodes existantes

La détection des jonctions nécessite l'évaluation de la courbure le long du contour discret. La discrétisation ainsi que le bruit rend l'évaluation de la courbure très délicate, et impose un lissage préalable du signal.

Les approches existantes sont du type filtrage, du type approximation par une fonctionnelle ou du type morphologie mathématique.

Approches du type filtrage gaussien

Le problème spécifique du filtrage gaussien est qu'il déplace les maxima de courbure. Ainsi, pour un filtre qui lissera suffisamment le bruit des contours, on observe des déplacements des extrêmes très importants. L'approche brute est donc inacceptable pour nos besoins.

L'approche multi-résolution du type [Asa 86] ou avec correction de la courbure du type [Low 88] résolvent ce problème de manière satisfaisante, mais le coût du filtrage et de la correspondance inter-résolutions est important. Une autre approche de type filtrage permettant d'éviter le coût de la correspondance est décrite dans [SM 89]. Cependant, ces approches, meilleures techniques de type filtrage à notre connaissance à ce jour, souffrent de l'insuffisance de principe déjà mentionnée : la sortie du processus de filtrage est discrète.

Notre approche par B_splines régularisées présente l'avantage de fournir une représentation continue des contours lissés, permettant des dérivations symboliques, et des estimations inter-pixels. Par ailleurs, nos expérimentations montrent que le filtrage gaussien rend difficile la détection des discontinuités de courbure.

Autres approches pour la segmentation à partir de B_splines

D'autres approches du type B_spline existent [Gir 88] où les fonctions de base sont enrichies pour accueillir des fonctions C^0 comme des coins, ou même des fonctions discontinues, permettant la modélisation exacte de fonctions correspondant aux types de jonctions que nous avons décrits. Le problème consiste à trouver le nombre et la place de ces discontinuités. Cette approche très formelle fournit de bons résultats sur les exemples simples fournis, mais aucun exemple réaliste pour la vision par ordinateur n'est malheureusement montré. Il est cependant très clair que la complexité de l'algorithme est beaucoup plus grande que celle présentée dans ce mémoire. Nous verrons que notre approche à partir de B_splines "classiques" fournit de bons résultats, à coût moindre.

Morphologie mathématique

La dernière approche recensée est une approche de type morphologie mathématique [Lay 90] ; un filtrage gaussien faible est appliqué pour lisser la courbe et obtenir une fonction de courbure discrète. La fonction de courbure est séparée en ses composantes positives et négatives, et chaque composante est traitée par un opérateur morphologique : une ouverture est appliquée pour supprimer les pics de la courbure positive, alors qu'une fermeture supprime les creux de la courbure négative. Ces opérateurs sont appliqués pour différentes largeurs d'opérateur, et fournissent une représentation hié-

rarchique où ne subsistent à haut niveau que les extrêmes les plus significatifs. Cette approche permet une analyse fine des extrêmes de courbure en fonction du degré de détail recherché, sans entraîner les problèmes du filtrage gaussien en résolution multiple comme le déplacement des extrêmes de courbure. Par contre, il reste nécessaire d'effectuer un filtrage gaussien préalable, un seuillage à chaque niveau et les discontinuités de courbure ne sont bien détectées que si elles s'accompagnent d'un changement de signe de la courbure.

Nous sommes munis d'un outil de représentation des contours, les B-splines régularisées, et nous allons décrire notre réponse au problème de la détection des points où segmenter une chaîne de contour. Les points recherchés sont donc les lieux de discontinuité de tangence, et les lieux de discontinuité de la courbure. Nous allons pour cela utiliser une technique de filtrage optimal utilisé habituellement en vision pour la détection des points de contraste [Can 83] [Der 90] [Der 87b] [She 86].

2.9.2 Rappels sur la détection des points de contraste

Les performances des détecteurs de contours à base de gradient local se dégradent très vite avec le bruit présent dans les images. Modestino et Fries [Mod 77] introduisent la notion de filtrage récursif pour la détection des contours modélisés de manière stochastique. Canny dans son travail de thèse [Can 83] développe une formalisation précise de la détection et de la localisation des contours permettant une évaluation quantitative de la performance des détecteurs de contour. Il propose de plus un filtre récursif quasi-optimal selon ses critères. Le filtre est étudié pour un signal monodimensionnel, et étendu au cas bidimensionnel des images.

Le signal est modélisé comme un signal de type échelon additionné à un bruit blanc gaussien de variance connue. Le filtrage est effectué en convoluant le signal avec une fonction spatiale impaire à support borné. Le signal de sortie présente alors un maximum au point de contraste. La détection est évaluée en estimant la probabilité de ne pas détecter un vrai point de contraste, ou de détecter à tort un point non point de contraste. Ce critère est équivalent à maximiser le rapport signal/bruit. Le critère de localisation consiste à trouver le point de contour le plus près possible de sa position réelle. Ce critère est optimisé en minimisant la variance de la position du passage à zéro du Laplacien. Enfin le détecteur ne doit donner qu'un seul point de contraste en réponse à un point de contour unique, introduisant un troisième critère à vérifier. La combinaison de ces trois critères permet de déterminer le filtre optimal. Canny a trouvé que

la forme de ce filtre était proche de la première dérivée d'une gaussienne, permettant une implémentation par filtre récursif efficace.

Deriche [Der 87b] et Shen [She 86] proposent une autre approche plus simple utilisant un filtre antisymétrique à support infini.

Ces filtres permettent tous deux une implémentation par filtre récursif. On trouvera en annexe C un rappel concernant la définition et l'implémentation de ces filtres optimaux.

2.9.3 Détection des lieux de discontinuité de la courbure

Etant donnée une chaîne de contour, il est possible de calculer la B_spline régularisée ajustée à cette chaîne, et d'obtenir une fonctionnelle que nous pouvons dériver symboliquement pour obtenir les dérivées successives dont nous avons besoin. La courbure C se calcule en effet simplement à partir des dérivées par :

$$C(t) = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}}$$

La courbure C ainsi calculée est indépendante de la paramétrisation choisie, et représente une propriété intrinsèque de l'arc. x', y', x'' et y'' sont calculés par dérivation symbolique de la B_spline. La fonction $C(t)$ est une fonction C^0 , puisque x'' et y'' sont des fonctions réelles C^0 .

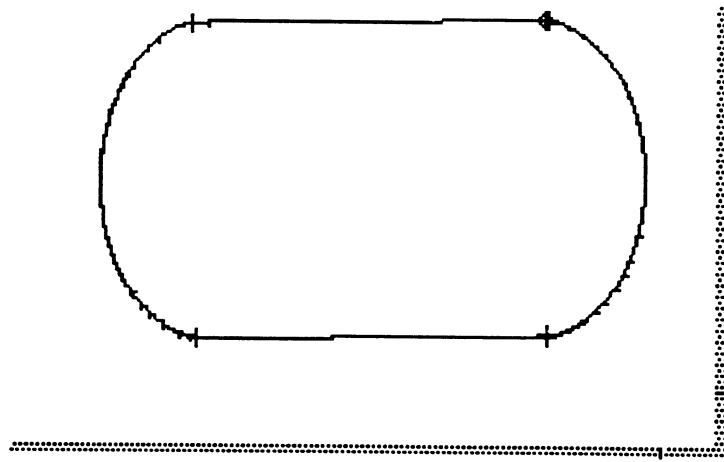


Figure 2.17. Discontinuité de courbure

On fait une analogie entre les points de contraste dans un signal de luminance et les discontinuités dans un signal de courbure, et nous les recherchons par conséquent comme les maxima du filtre de Deriche. Un seuillage est effectué pour éliminer les maxima dûs au bruit. La figure 2.17 montre l'extraction parfaite des points de discontinuité de la courbure pour le cas classique du contour extérieur d'un cylindre. Il est à remarquer que ces

points présentent une continuité de la tangente, ce qui les rend en général très délicats à détecter par les méthodes classiques. De plus, le lissage par B_spline qui a l'avantage de ne pas déplacer les points de discontinuité de courbure donne une localisation excellente car le lissage du filtre de Deriche est conservé à un niveau très faible.

2.9.4 Détection des discontinuités de tangence

Les discontinuités de tangence peuvent être détectés en cherchant les points de contraste sur le signal de l'angle.

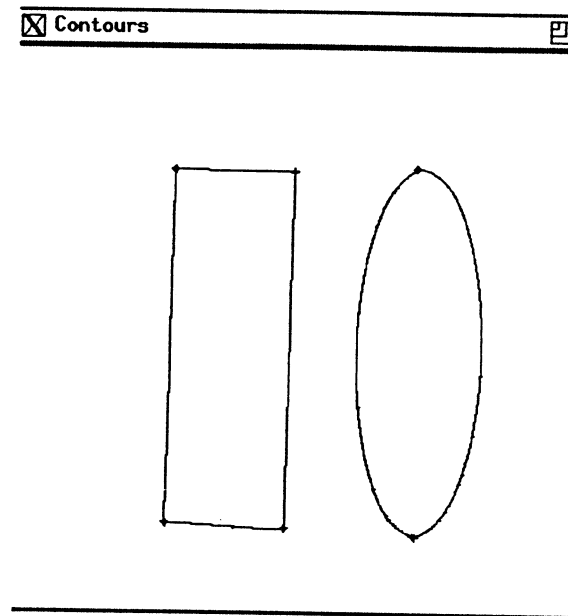


Figure 2.18. discontinuités de tangence

On remarquera que les maxima de courbures seront parfois parfois détectés comme des discontinuités de tangence, car ils correspondent à un signal échelon lissé pour le signal de l'angle. Les minima de courbure eux ne sont pas détectés. La figure 2.18 montre la détection des angles d'un polygône et des points sur le grand axe d'une ellipse.

Là encore, un seuillage est effectué pour éliminer les faux maxima liés au bruit.

2.9.5 Combinaison des critères de coupure

On reprend l'exemple de la souris pour illustrer la segmentation. Les segmentations sont indiquées par une croix à chaque extrémité d'un contour. Les débuts et fins des chaînes de contours d'origine sont indiqués par un rond.

La figure 2.19 montre la segmentation aux points d'extrêmes de courbure pour l'image de la souris, et la figure 2.20 montre les discontinuités de courbure sur la même image.

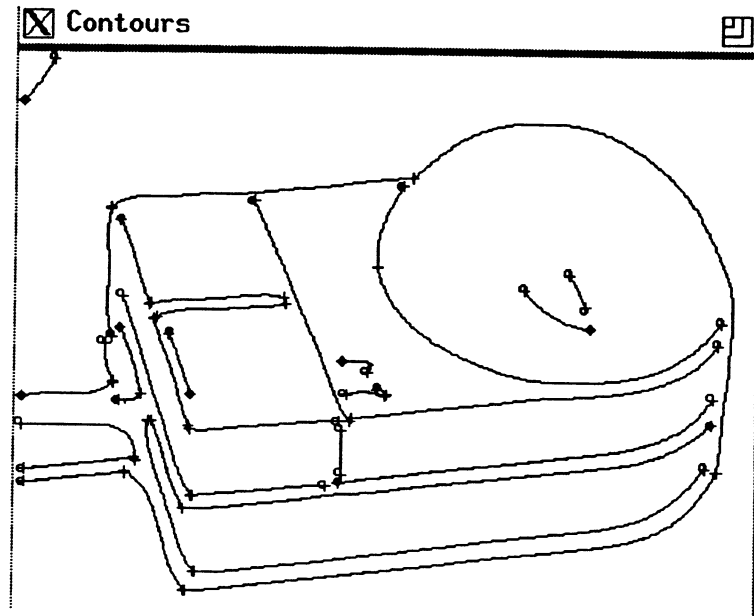


Figure 2.19. Segmentation de l'image de la souris

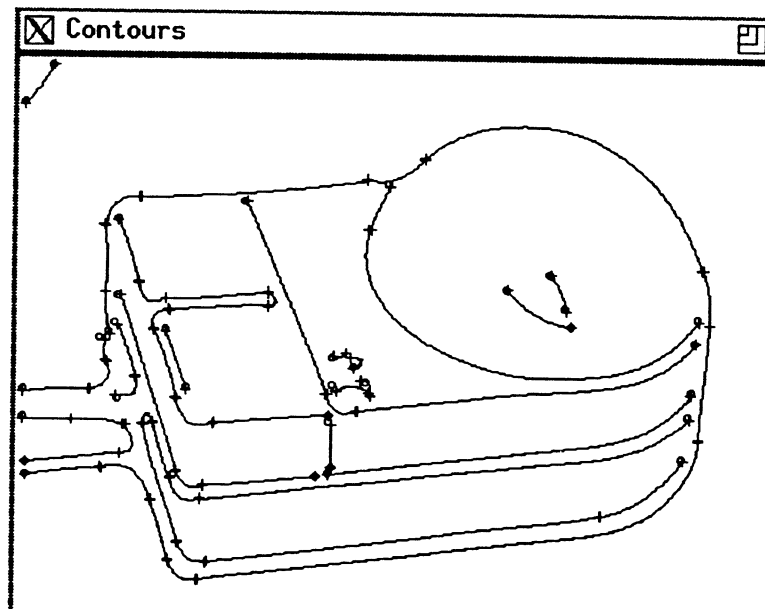


Figure 2.20. Discontinuités de courbure dans l'image de la souris

La combinaison des deux critères de coupure précédents est un problème difficile. En effet, si chaque critère répond bien à la question demandée, il répond malheureusement aussi dans d'autres cas. La figure 2.20 montre l'effet de la recherche des discontinuités de courbure dans le cas de l'image de la souris.

On s'aperçoit que, compte tenu du lissage, à chaque extrême de courbure correspond de part et d'autre deux points de discontinuités de courbure. Les deux algorithmes de segmentation appliqués l'un après l'autre demandent un travail d'interprétation supplémentaire pour l'étiquetage du type de discontinuité. Ce travail reste à faire et n'a pas été abordé.

2.9.6 Résultats expérimentaux

Les points de coupure utilisés sont exclusivement les points d'extrêmes de courbure.

Les chaînes de contours sont extraites dans un premier stade par l'extracteur de Deriche [Der 87b] [Can 83, Can 86] suivi par un chaînage de contour avec un seuillage par hysteresis. Les chaînes sont ensuite traitées avec notre algorithme. Elles sont donc approchées par une B_splines régularisée ; les points de coupure, extrêmes de la courbure, sont calculés, et chaque sous-chaîne obtenue est enfin approchée à son tour par une B_spline.

Des exemples supplémentaires sont présentés dans deux types de scène : entièrement polyédrique et à prédominance courbe.

Scènes polyédriques

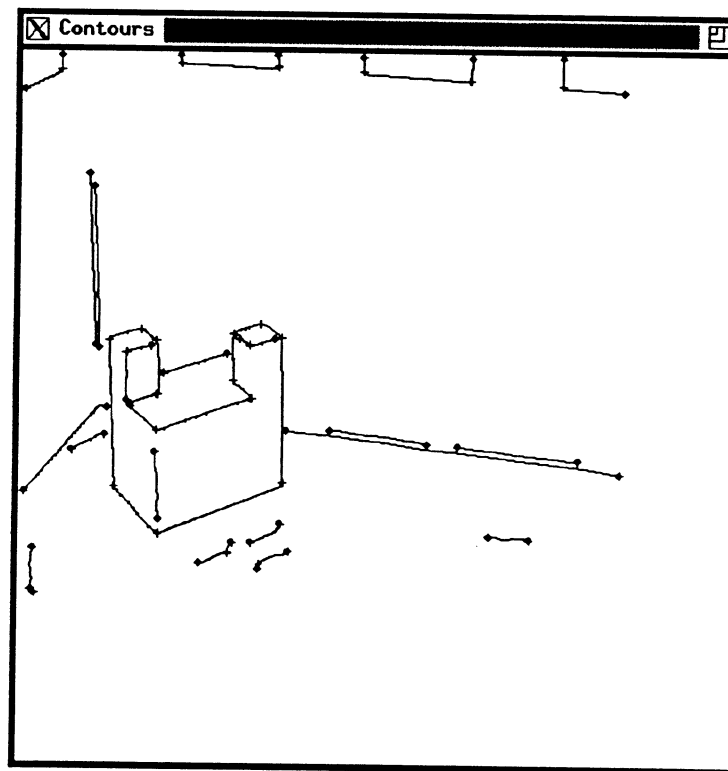


Figure 2.21. Segmentation dans une scène polyédrique

Un minimum pour toute technique de segmentation de contours est de bien détecter les angles dans une scène polyédrique. On peut voir dans la

figure 2.21 que la détection est très bonne. De plus, les coins ne sont jamais dégradés, comme souvent avec les approches de type découpage récursif où les coins sont écornés : le point de coupure correct est manqué, et l'on observe un petit segment en diagonale dans le coin comme illustré dans la figure 2.22.

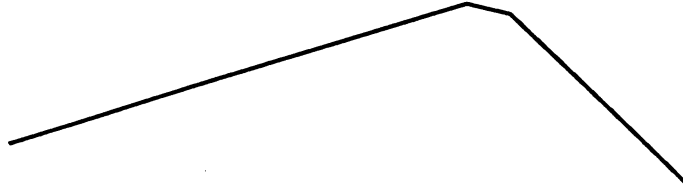


Figure 2.22. Découpage récursif : problème de segmentation dans les coins

Scènes à prédominance courbe

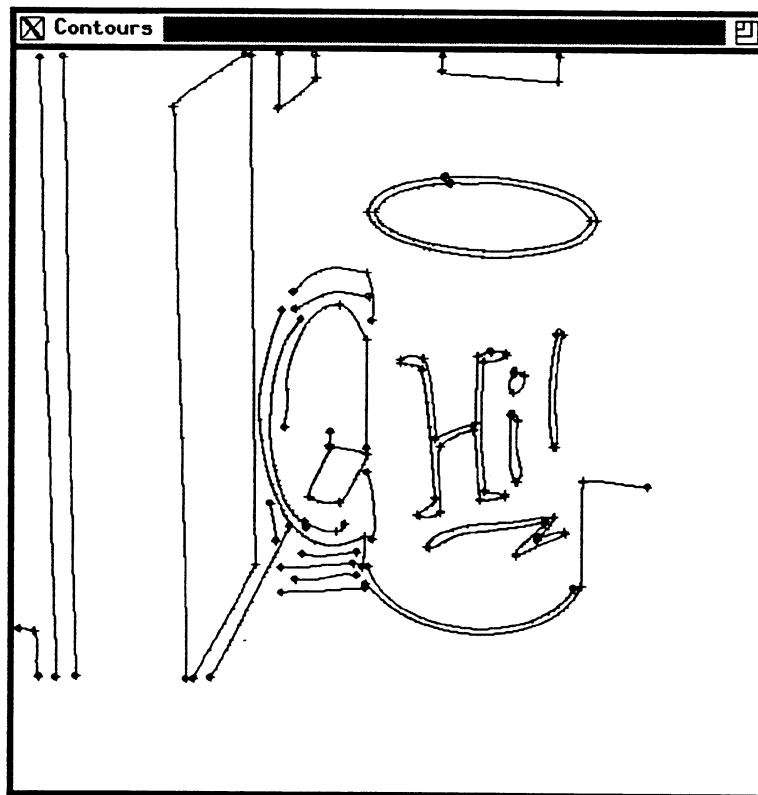


Figure 2.23. Segmentation dans une scène mixte

Là mieux qu'ailleurs on peut s'apercevoir que le comportement de l'algorithme est excellent pour ce type de scène comme on peut s'en apercevoir dans la figure 2.23.

Choix de n et de τ :

Le choix du nombre d'intervalles n pour l'approximation par B_spline dépend de la phase de l'algorithme : la première approximation de la chaîne pour la détection des points de coupure se fait avec un grand nombre d'intervalles, pour augmenter la précision de l'approximation. De même, pour éliminer les faux maxima, le facteur de tension est choisi très grand. Par contre, lors de la deuxième phase d'approximation, après coupure, le nombre d'intervalles est bien moindre, car les segments de chaîne restants sont par définition sans extrêmes de courbure, donc plus faciles à approcher. De même, le facteur de tension peut être choisi inférieur, pour améliorer la qualité de l'approximation. En pratique dans les exemples montrés, n est égal à un dixième du nombre de pixels de la chaîne, puis un vingtième dans la 2^{ème} phase. τ vaut 10^4 , puis 10^3 dans la deuxième phase.

2.9.7 Conclusion

Une autre approche, similaire à l'approche "multi-résolution", consiste à ne conserver les points extrêmes que s'ils se conservent à plus forte tension. Il faut tracker ces points des fortes tensions jusqu'aux tensions les plus faibles, de façon à ne pas perdre de précision sur la localisation des extrêmes. Cette approche a pour elle une excellente localisation des extrêmes, une bonne pertinence des extrêmes trouvés, le fait de ne pas avoir de seuil a priori de valeur de tension, mais le coût engendré par la multi-résolution est non justifié puisque les résultats obtenus par la solution du seuil sur la courbure nous satisfont pleinement.

Les B_splines cubiques régularisées nous fournissent un excellent outil de travail pour les contours curvilignes rencontrés en vision dans un univers non polyédrique : l'approximation est excellente, la représentation est continue, les points de coupure des chaînes sont détectables de manière fiable et rapide. C'est une base sûre de départ pour des travaux ultérieurs de vision de moyen niveau, comme la reconstruction, la stéréovision, l'autocalibration du mouvement avec des objets non nécessairement polyédriques.

2.10 Le suivi des contours

Les techniques de suivi de contours sont nécessaires pour les applications de reconstruction de surface et d'autocalibration qui seront décrites aux chapitres 3 et 4.

Le principe du suivi de contour est très simple : quand la caméra se déplace par rapport aux objets de la scène, chaque contour perçu se déplace

dans la séquence d'images ; le but du suivi de contour consiste à conserver l'identité d'un contour tout au long de son déplacement. Ainsi, tout processus ultérieur travaillant sur un même contour au cours du temps peut récupérer ses données très simplement.

Le suivi de contours est généralement envisagé comme une phase totalement indépendante, et aucune connaissance a priori n'est disponible. Seul est connu le fait que la séquence d'images est *dense*, c'est-à-dire que les points de vue de deux images successives sont rapprochés.

Plusieurs approches coexistent pour aborder ce problème :

1. Le flot optique permet d'obtenir une estimation du déplacement pour chaque pixel, ou plus particulièrement pour les pixels de contour, permettant une mise en correspondance a priori simple entre les contours¹.
2. Le filtrage optimal permet de modéliser le déplacement d'une primitive et de prédire sa position dans l'image suivante. Cette technique a été appliquée aux points et aux segments de droite [Cro 88] [Der 90], en utilisant le filtre de Kalman.

Une approche équivalente avec filtre de Kalman a été testée au LIFIA par Fabien Henry [Hen 90] pour suivre des contours quelconques ; n points régulièrement répartis sur une courbe sont choisis, et suivis au cours de la séquence. A chaque point est associée l'information de la direction de la tangente à la courbe. Cette expérience a effectivement permis de suivre les contours pendant une dizaine d'images, mais a montré les limites du procédé. En effet, si le déplacement normal est très bien estimé, le déplacement tangentiel l'est très mal, et les points suivis ont tendance à *sortir* des contours. Après dix ou quinze images, très peu de points sont encore présents et suivis.

3. Les "snakes" [Kaa 87] [Ber 90a] [Ber 90b] [Aek 90] ou contours actifs, sont attirés vers les extrêmes de contraste, tout en conservant une certaine rigidité. Un modèle de barre élastique est l'analogie mécanique utilisée pour formaliser la notion de rigidité. Le contour actif est initialisé près d'un contour de l'image, et il se déforme au fur et à mesure pour se coller au contour. Des techniques d'optimisation itérative permettent ainsi au contour actif de se déplacer progressivement de sa position dans une image vers sa position dans l'image suivante. Le problème de l'initialisation (nombre et position de ces courbes, création de nouveaux snakes au cours du temps) est cependant crucial et non encore correctement résolu.

1. On trouvera dans [Hor 80] et [Hor 77] une description des techniques de calcul du flot optique

4. Les techniques de corrélation, couramment utilisées en reconnaissance de formes, permettent également de résoudre le problème, pour la partie évaluation du déplacement. Une représentation pyramidale permet une implémentation efficace et robuste tout en s'affranchissant des limites de la taille de la fenêtre de recherche. Une telle approche est également en cours d'investigation au LIFIA.

Compte tenu des efforts engagés sur ce problème par d'autres chercheurs au LIFIA, nous n'avons pas approfondi la question. Les correspondances utilisées dans ce travail ont été établies soit à la main, soit en sélectionnant le meilleur candidat présent dans une fenêtre, sur la base d'une fonction de coût. La fonction de coût prend en compte longueur, orientation, courbure moyenne, et distance minimale. Cette très simple technique suffit en général pour les scènes simples étudiées.

Chapitre 3

Reconstruction de surface

Ce chapitre présente une méthode de géométrie différentielle pour la reconstruction de surface. La notion de surface spatiotemporelle est définie, puis la technique de reconstruction est introduite. Des résultats expérimentaux avec des données synthétiques et des données réelles sont enfin présentés.

3.1 Introduction

Aloimonos [Alo 88] a fait remarquer que la vision active continue permettait de bien poser un certain nombre de problèmes de vision, comme par exemple la reconstruction de surface à partir des ombrages. Ce travail est une contribution à cette approche “vision active” et porte sur la reconstruction des surfaces tridimensionnelles à partir de l’observation continue des contours critiques d’un objet par une caméra en mouvement. Nous verrons en effet que dans le contexte du continu, la reconstruction de la surface d’un objet est alors un problème bien posé.

C’est Koenderink [Koe 84] qui a mis en évidence la richesse d’informations portée par les contours critiques, notamment comment certaines mesures sur les contours perçus dans l’image permettaient d’inférer des résultats qualitatifs et quantitatifs sur la surface tridimensionnelle observée.

Les travaux de Bolles, Baker et Marimont [Bol 87] nous ont inspirés en montrant l’importance de la dimension “temporelle” avec la surface spatiotemporelle.

Giblin et Weiss [Gib 87] avaient posé le problème de la reconstruction de façon mathématique. Notre contribution est un prolongement direct de ce travail. D’abord nous étendons les résultats au cas de la projection perspective et du mouvement quelconque de la caméra au lieu de se limiter à celle de la projection orthogonale et translation simple. La discussion de ces équations nous permet de voir que les résultats sont applicables dans des cas bien plus généraux que celui des surfaces deux fois continûment différentiables : de fait il suffit que la surface soit différentiable par morceaux.

De plus cette méthode permet aussi la reconstruction tridimensionnelle de contours provenant de variation de luminance comme un trait de pinceau sur une tasse.

L'idée nouvelle que nous voulons introduire ici est celle de la reconstruction des surfaces par des méthodes différentielles qui évitent de se poser le problème de l'appariement au niveau des pixels. Brint et Brady [Bri 89], dans le contexte de la stéréovision sur des contours curvilignes, contournent ainsi le problème de l'appariement au niveau des pixels en établissant des correspondances courbes à courbes par des techniques différentielles ; cette approche permet de se passer de la connaissance précise de la géométrie épipolaire.

Ce travail est aussi proche de la perception du flot optique que nous ont donnée les chercheurs travaillant sur le mouvement à la suite des travaux initiaux de Horn [Hor 80] et Nagel [Nag 82], et nous verrons à la fin du chapitre les liens qui relient notre approche à celle du flot optique.

3.2 Etat de l'art

Nous nous posons donc le problème de la reconstruction des objets non nécessairement polyédriques par vision monoculaire, dans le cas où le mouvement de la caméra est connu. Le cas des objets polyédriques est déjà relativement bien compris, et l'on sait reconstruire points et droites à partir d'une séquence d'images [Fau 87a] [Xie 89] [Cro 90] par les équations, classiques maintenant, de la stéréovision.

Les objets non polyédriques posent de nouveaux problèmes :

- les surfaces observées n'étant plus planes, les contours observés ne sont pas nécessairement rectilignes, d'où un problème de représentation.
- les techniques de reconstruction les plus éprouvées [Fau 87a] [Aya 87] [Cro 90] se fondent sur les primitives segments de droite pour la reconstruction, et ces approches ne se généralisent pas simplement au cas des contours curvilignes (le cas des contours curvilignes n'est pas à confondre avec le cas des pixels de contour isolés où la topologie du contour est simplement ignorée).
- les objets non polyédriques imposent une autre limitation plus fondamentale aux techniques de reconstruction d'objets polyédriques : les contours critiques (ou occultants) pour plusieurs vues de l'objet ne sont pas les mêmes car les courbes critiques sur la surface de l'objet dépendent du point de vue. Il existe par contre une correspondance sémantique du fait qu'il s'agit d'un contour occultant de la même surface.

Ce travail a été directement inspiré des résultats de Giblin et Weiss [Gib 87] où la reconstruction est traitée dans le cas de la projection orthographique, et avec un mouvement rectiligne de caméra.

Nous avons fourni une première solution générale à ce problème [Arb 89, Arb 90] montrant comment la reconstruction des surfaces était possible dans le cas de la projection perspective et d'un mouvement quelconque de la caméra, à condition de connaître les paramètres du mouvement.

Les travaux de Roberto Cipolla [Cip 89] et de Régis Vaillant [Vai 89] sont des approches équivalentes menées indépendamment, et se distinguent essentiellement les unes des autres par le choix du dispositif expérimental et de la paramétrisation choisie pour représenter les contours.

Nous avons choisi dans [Arb 89, Arb 90] une paramétrisation des contours sous une forme fonctionnelle $y = f(x)$. Nous nous attachions essentiellement à obtenir une représentation des surfaces par rapport à la scène, indépendamment de l'obtention de caractéristiques précises de courbure de la surface.

R. Cipolla utilise une modélisation sphérique de la caméra, une paramétrisation très souple qui permet de simplifier les équations obtenues. Il met l'accent sur l'extraction des caractéristiques de courbure des surfaces observées. En particulier, il exhibe une technique robuste d'évaluation de la courbure radiale à partir du déplacement relatif entre un point d'un contour critique et un point d'un contour de discontinuité proche. Il calibre par ailleurs le mouvement de la caméra en utilisant la technique de calibration de caméra de Tsai [Tsa 87], après avoir éliminé les techniques d'auto-calibration [Fau 87a], faute de résultats corrects. S'il représente les contours par B-splines, il calcule les dérivées temporelles nécessaires par différences finies. Lorsqu'il évalue les courbures de surface, les différences finies ne permettent plus de mesures correctes des dérivées du deuxième ordre et il doit par conséquent utiliser une approximation par cercle osculateur.

R. Vaillant a choisi une représentation des surfaces par les angles d'Euler. Cette représentation l'amène à des équations complexes à manipuler. De plus, les dégénérescences aux pôles de la représentation le force à redéfinir des repères locaux pour ses estimations. L'originalité de son approche pour la reconstruction consiste à utiliser un développement limité de Fourier de la surface, de façon à s'affranchir des problèmes de discrétisation des positions de prises de vue. Son dispositif expérimental consiste en un système de stéréovision trinoculaire calibré. Il ne travaille pas par conséquent sur des séquences denses d'images comme R. Cipolla et nous-mêmes ; il s'affranchit ainsi du problème de l'estimation précise du mouvement.

Nous avons choisi ici de présenter le problème en utilisant la formalisation en coordonnées sphériques, empruntée à R. Cipolla.

3.3 Définitions

Surface spatio-temporelle Lors du mouvement relatif de la caméra par rapport à l'objet, les contours perçus par la caméra se déplacent sur le plan image. L'ensemble de ces contours qui se déplacent au cours du temps décrivent une surface appelée improprement surface spatio-temporelle ; le terme de spatio-temporel pourrait avantageusement être remplacé par celui de spatio-spatial puisque le déplacement des contours est dû au déplacement relatif caméra/objet et non au simple fait de l'écoulement du temps.

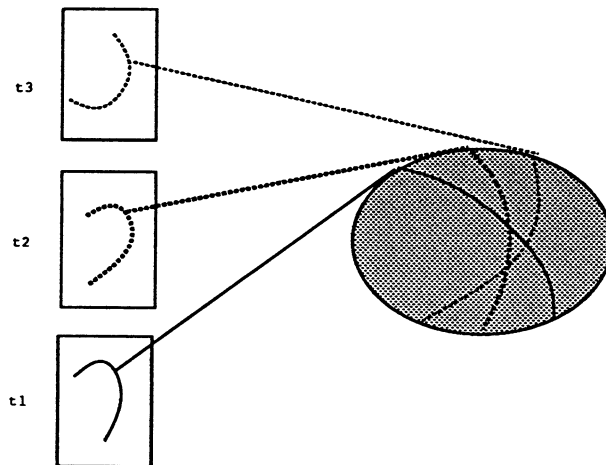


Figure 3.1. Contours perçus par la caméra en mouvement

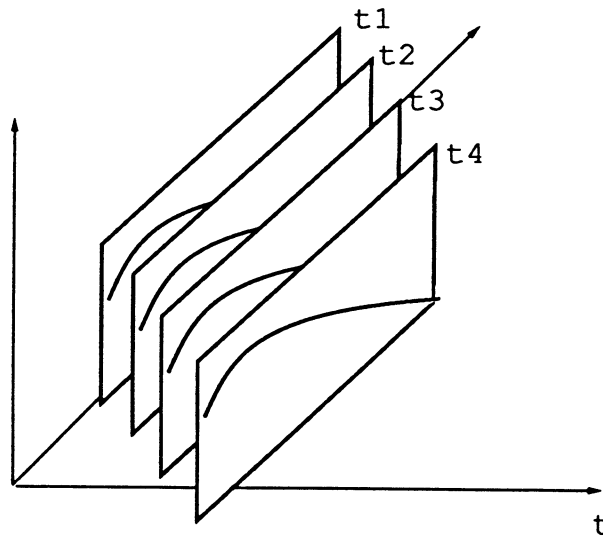


Figure 3.2. Formation d'une surface spatio-temporelle dans le repère caméra

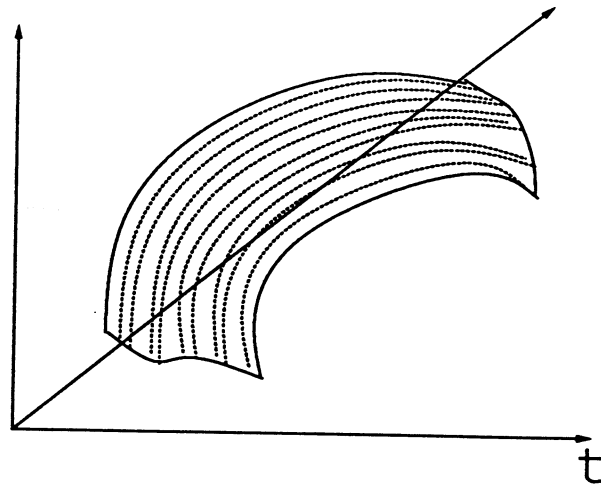


Figure 3.3. Surface spatio-temporelle: passage au continu

La figure 3.1 illustre la formation d'un contour sur le plan image, pour un certain nombre de positions de la caméra ; la figure 3.2 montre comment l'ensemble de ces contours est structuré temporellement, et la figure 3.3 la création de la surface spatio-temporelle, en passant "au continu". On verra que cette surface spatio-temporelle, ensemble des observations d'un objet donné au cours du temps, suffit en général à reconstruire la surface de l'objet si les paramètres du mouvement sont connus.

Modèle sphérique La définition de surface spatio-temporelle que nous utiliserons effectivement par la suite se déduit de la précédente en passant en géométrie sphérique : les contours du plan image sont projetés sur une sphère de centre le centre focal de la caméra, et de rayon 1. Cette transformation s'effectue de la façon simple suivante:

Soit un pixel observé P du plan image, localisé en $P(X_d, Y_d)$, en pixels dans le plan image. Ce point s'exprime dans le repère caméra par $P(x, y, f)$ grâce aux équations détaillées dans le chapitre 2. Le centre focal O de la caméra est par définition $O(0, 0, 0)$. Le point P est donc repéré dans le modèle sphérique par le vecteur $T = \frac{\vec{OP}}{|\vec{OP}|}$ | vecteur unitaire qui encode la direction sous laquelle est vue le pixel P depuis le centre focal.

Cette définition permet essentiellement de simplifier les calculs ultérieurs.

3.4 Les notations du problème

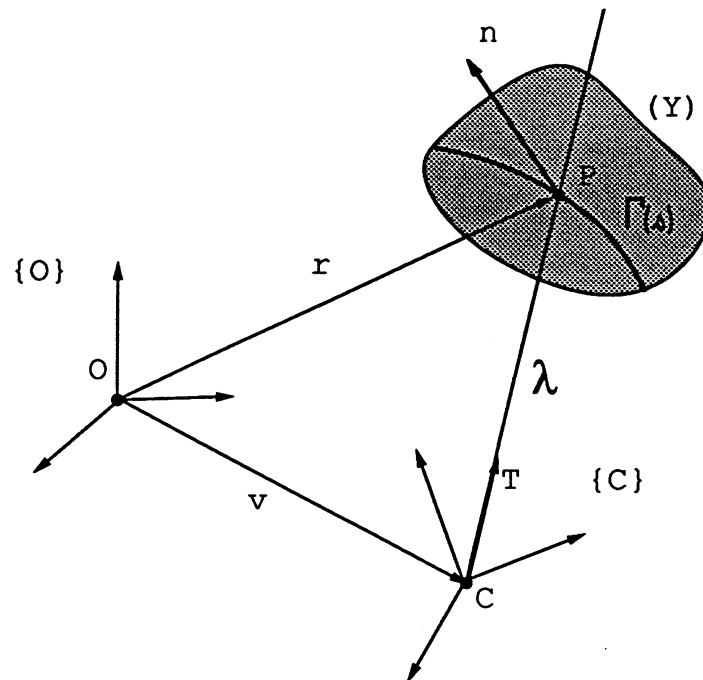


Figure 3.4. Les repères - les notations

La figure 3.4 résume les notations utilisées :

- **Les repères :** Tout vecteur X exprimé dans un repère $\{W\}$ sera noté ${}^w X$. De même, la dérivée temporelle de ce vecteur dans le repère $\{W\}$ sera notée ${}^w X_t$ (ou ${}^w \dot{X}$ si ce vecteur n'est fonction que du temps). En général, un vecteur Q fonction de u_1, u_2, \dots, u_n verra ses différentielles partielles dans $\{W\}$ notées ${}^w Q_{u_1}$ et ${}^w Q_{u_2}, \dots, {}^w Q_{u_n}$.
- Un objet (Y) , indéformable, est associé à un repère propre de référence $\{O\}$. L'objet (Y) est fixe dans le repère $\{O\}$, et le but de la reconstruction est d'obtenir la description de (Y) dans ce repère $\{O\}$.
- Une caméra, mobile par rapport à l'objet (Y) , est associée à son repère de référence $\{C\}$.
- Le référentiel $\{C\}$ est repéré par rapport à $\{O\}$ par une translation ${}^o v$ et par une matrice de rotation ${}^o R$. Nous noterons ${}^o \dot{v}$ la vitesse de $\{C\}$ dans $\{O\}$, par ${}^o \Omega_{c/o}$ le vecteur rotation instantanée, et de même ${}^o \ddot{v}$ et ${}^o \dot{\Omega}_{c/o}$ l'accélération de $\{C\}$ et la dérivée du vecteur rotation instantanée. ${}^o v, {}^o \dot{v}, {}^o \Omega_{c/o}, {}^o \ddot{v}$ et ${}^o \dot{\Omega}_{c/o}$ sont appelés les paramètres du mouvement de la caméra.
- Nous observons à t donné sur (Y) une courbe critique $\Gamma(s)$, où s est une paramétrisation de Γ .

- Un point particulier P sur Γ est considéré : nous noterons ${}^{\circ}r(s, t)$ le vecteur OP dans $\{O\}$: ${}^{\circ}r(s, t) = \Gamma(s)$
- Un point P de (Y) est repéré par r par rapport à O et par λT par rapport à C , où T est un vecteur unitaire et λ est la distance du point à C . T_t et T_s décrivent la différentielle de T en t et s . Le formalisme envisagé est celui de la géométrie sphérique où les contours perçus sont projetés sur une sphère de rayon 1 centré sur le centre de projection C de la projection perspective. Ce formalisme permet essentiellement une simplification des formules obtenues.
- La normale en P à la surface de (Y) est notée ${}^{\circ}n$.

3.5 La reconstruction de surface

3.5.1 Principe

On suppose que l'on observe une courbe critique d'une surface quelconque au moins C^1 . La projection de cette courbe critique engendre un contour critique sur le plan image de la caméra. Le suivi de ce contour grâce aux techniques de suivi évoquées précédemment mène à l'obtention d'une surface spatio-temporelle.

Les paramètres du mouvement (translation et rotation) étant connus, il est possible par changement de repère d'exprimer chaque contour dans le repère de référence $\{O\}$, et d'obtenir finalement notre surface spatio-temporelle dans le repère $\{O\}$.

On rappelle ici une propriété générale des courbes critiques, illustrée par la figure 3.5. Considérons un point P de la courbe critique observée ; la ligne de vue L et la tangente T_s au contour critique sont deux vecteurs non parallèles qui appartiennent tous deux au plan tangent Π à la surface en P . Ils définissent donc entièrement la position et l'orientation du plan tangent de la surface observée, en chaque point de la courbe critique.

La ligne de vue L et la tangente T_s au contour critique définissent le plan tangent à la surface en P :

Démonstration :

La ligne de vue L est représentée par son vecteur unitaire ${}^{\circ}T(s, t)$.

La position d'un point P de la surface de l'objet obéit à la relation suivante :

$${}^{\circ}\Gamma(s) = {}^{\circ}r(s, t) = {}^{\circ}v(t) + \lambda(s, t){}^{\circ}T(s, t) \quad (3.1)$$

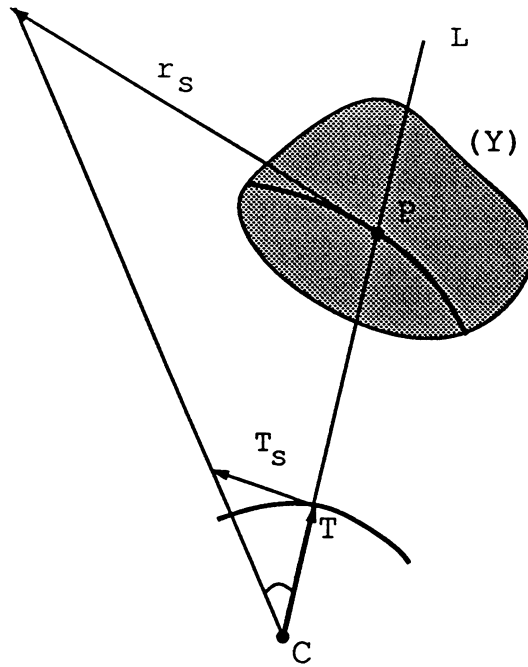


Figure 3.5. Propriété des courbes critiques

d'où en différentiant selon s , à t constant :

$${}^{\circ}r_s(s, t) = \lambda_s(s, t){}^{\circ}T(s, t) + \lambda(s, t){}^{\circ}T_s(s, t)$$

Deux vecteurs indépendants du plan tangent en P permettent de définir l'orientation du plan tangent. Nous savons déjà que la ligne de vue ${}^{\circ}T(s, t)$ est tangente à la surface puisque la courbe Γ est une courbe critique. ${}^{\circ}r_s(s, t)$ est la tangente en P à la courbe critique, et appartient donc également au plan tangent. En dehors du cas où ${}^{\circ}T(s, t)$ et ${}^{\circ}r_s(s, t)$ sont colinéaires, ${}^{\circ}T(s, t)$ et ${}^{\circ}r_s(s, t)$ définissent donc l'orientation du plan tangent.

La position du plan tangent est donnée du fait que la ligne de vue passe par le centre optique de la caméra ; le plan tangent passe donc également par le centre optique.

Le plan tangent en P est donc engendré par deux vecteurs particuliers : la ligne de vue ${}^{\circ}T(s, t)$ et la tangente à la courbe critique en P ${}^{\circ}r_s(s, t)$.

Puisque ${}^{\circ}r_s(s, t)$ est lui-même combinaison de ${}^{\circ}T(s, t)$ et de ${}^{\circ}T_s(s, t)$, alors ${}^{\circ}T_s(s, t)$ et ${}^{\circ}T(s, t)$ sont également générateurs du plan tangent. ${}^{\circ}T_s(s, t)$ et ${}^{\circ}T(s, t)$ sont toujours indépendants car ce sont deux vecteurs orthogonaux.

□

Le problème de la reconstruction se ramène donc au problème de retrouver la surface enveloppe de ses plans tangents.

3.5.2 Théorie générale de la reconstruction pour une surface

Tous les vecteurs sont exprimés dans le repère de référence objet $\{O\}$, puisque les paramètres du mouvement sont supposés connus.

La position d'un point P obéit à la relation suivante :

$${}^o\Gamma(s) = {}^o r(s, t) = {}^o v(t) + \lambda(s, t) {}^o T(s, t) \quad (3.2)$$

ou plus simplement, pour alléger les notations :

$${}^o r = {}^o v + \lambda {}^o T \quad (3.3)$$

La résolution consiste à éliminer les termes inconnus, pour ne conserver qu'une équation en λ et en T , puisque les T sont tous mesurables à partir des paramètres du mouvement.

Par différentiation de 3.3 en t , on trouve :

$${}^o r_t = {}^o \dot{v} + \lambda_t {}^o T + \lambda {}^o T_t \quad (3.4)$$

Elimination de λ_t :

La normale à la surface s'obtient à partir de ${}^o T$ et ${}^o T_s$:

$$si \quad {}^o n = \frac{{}^o T \wedge {}^o T_s}{|{}^o T \wedge {}^o T_s|} \quad alors \quad {}^o n \cdot {}^o T = 0 \quad (3.5)$$

De 3.4 et 3.5 on tire :

$${}^o r_t \cdot {}^o n = {}^o \dot{v} \cdot {}^o n + \lambda {}^o T_t \cdot {}^o n \quad (3.6)$$

Elimination de r_t :

Si la courbe Γ observée est une courbe critique, alors ${}^o r_t \cdot {}^o n = 0$ car ${}^o r_t$ appartient au plan tangent à la surface en P ; en effet, quand la caméra se déplace, le point observé à s constant sur la surface décrit une courbe sur (Y) qui passe par P . Sa dérivée en t appartient donc au plan tangent.

On a donc très simplement :

$$0 = {}^o \dot{v} \cdot {}^o n + \lambda {}^o T_t \cdot {}^o n \quad (3.7)$$

d'où :

$$\lambda = \frac{-{}^o \dot{v} \cdot {}^o n}{{}^o T_t \cdot {}^o n} \quad (3.8)$$

Cette équation est similaire à celle que nous avons obtenue dans [Arb 89], mais sa formalisation différente lui permet de ne dégénérer que dans le cas où ${}^o T_t \cdot {}^o n = 0$, ce qui correspond à un sous-cas qui s'analyse simplement :

Si ${}^o T_t \cdot {}^o n = 0$, on tire de l'équation 3.7 que :

$$0 = {}^o \dot{v} \cdot {}^o n$$

et donc le déplacement de la caméra s'effectue dans le plan tangent à la surface, ce qui ne fournit localement aucune information sur le point P et empêche donc la reconstruction en ce point. En pratique, il faudra évaluer ${}^o \dot{v} \cdot {}^o n$ et ne reconstruire que lorsque cette valeur est suffisamment éloignée de 0.

Puisque ${}^o \dot{v}$ nous est donné a priori, et que ${}^o T_t$ et ${}^o n$ se déduisent des données, on peut calculer λ , d'où une description de la surface de (Y) par :

$${}^o r(s, t) = {}^o v(t) + \lambda(s, t) {}^o T(s, t)$$

puisque ${}^o v(t)$ est connu à priori, et que ${}^o T(s, t)$ est une donnée.

Calcul pratique de ${}^o T$ et ${}^o T_s$

Les mesures sont effectuées dans le repère caméra $\{C\}$ et concernent la ligne de vue ${}^c T$ et la tangente au contour ${}^c T_s$. Les paramètres du mouvement étant connus, il est simple de transformer ${}^c T$ et ${}^c T_s$ dans le repère $\{O\}$ par simple changement de repère.

${}^o n$ est alors obtenu simplement par produit vectoriel.

Calcul pratique de ${}^o T_t$

Une fois ${}^o T$ connu, la surface spatio-temporelle est entièrement connue dans $\{O\}$. La description de la surface et sa manipulation demande de définir une paramétrisation de la surface.

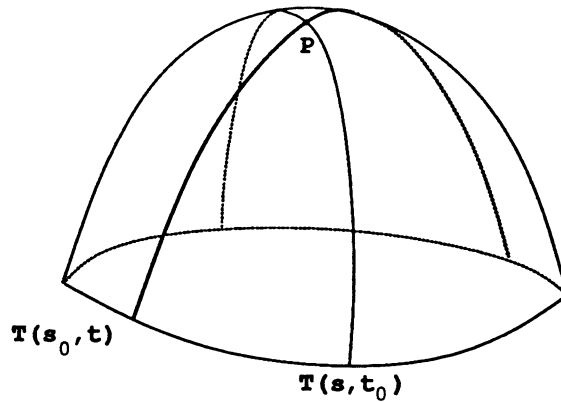


Figure 3.6. Paramétrisation de la surface spatio-temporelle

Chaque contour qui compose la surface spatio-temporelle possède sa propre paramétrisation ; il faut donc établir une correspondance inter-contour.

La figure 3.6 illustre cette notion en montrant la surface spatio-temporelle. Deux lignes sont tracées sur cette surface : il s'agit de $T(s, t_0)$, le contour observé à t_0 , et de $T(s_0, t)$, une correspondance arbitraire qui lie les contours à s_0 constant. 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. On prouvera plus loin que le choix de la mise en correspondance entre les contours, implicite dans la notation à s constant, peut être arbitraire. Puisque les paramètres du mouvement sont connus, il est naturel, mais non nécessaire, d'utiliser la géométrie épipolaire pour déterminer les correspondances, et le calcul de ces correspondances est particulièrement simple (voir paragraphe 3.6).

oT_t s'obtient alors simplement par différentiation à s constant. Le terme oT_t est intuitivement proche de la notion de disparité en stéréovision, et ${}^oT_t \cdot {}^on$ est la disparité projetée sur la direction normale au plan tangent.

Tous les termes de l'équation 3.8 sont donc bien issus directement des mesures et des paramètres connus du mouvement.

L'équation 3.8 est l'équation de la reconstruction, qui s'applique à tous les objets dont la surface est C^1 par morceaux. Ce résultat est strictement local puisque l'on voit qu'il suffit de connaître les dérivées partielles en s et

t au point considéré de la surface spatio-temporelle.

3.5.3 Cas particulier d'une courbe dans l'espace

Le contour observé sur le plan image n'est pas nécessairement un contour critique, mais peut-être un contour de discontinuité ou un marquage de surface. La formalisation de la reconstruction présentée ci-dessus s'adapte très simplement à ce cas particulier : en effet, ${}^o n$ n'est plus la normale à la surface, faute de surface, mais il est toujours défini à partir de ${}^o T$ et ${}^o T_s$, et l'on voit que la formule de reconstruction est encore valide dans ce cas dégénéré où l'enveloppe des plans se réduit à une courbe de l'espace. La courbe de discontinuité est reconstruite comme une surface, mais une analyse a posteriori de la surface reconstruite devrait permettre de différencier une surface réelle d'une courbe de discontinuité, par estimation de la courbure de la courbe ${}^o r(s_0, t)$: cette courbure devrait être infinie en théorie, alors qu'elle est finie pour toute véritable surface.

Un point de vue différent consiste à réaliser l'arbitraire d'un étiquetage " courbe de discontinuité " " courbe critique ". En effet, cet étiquetage dépend beaucoup de la résolution choisie: une courbure forte visualisée à faible résolution (ou de loin) apparaît comme une courbe de discontinuité alors que la même courbe vue à forte résolution (ou de plus près) peut apparaître comme une courbe critique. Un tel étiquetage paraît donc injustifié pour les besoins de la reconstruction. On verra cependant plus loin que cette estimation est délicate avec une approche directe, et que d'autres techniques semblent plus robustes. Nous ne les aborderons pas car elles s'éloignent sensiblement des principes utilisés dans ce travail.

3.5.4 Remarque : indépendance de la paramétrisation en s

La formule de reconstruction 3.8 demande d'établir une correspondance entre les contours de façon à définir une paramétrisation de la surface en s et t .

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.

En revanche, le paramètre s est libre, et plus encore les relations entre les contours successifs. 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 $T(s_0, t)$ (courbes à s constant) où la correspondance à s constant est pour l'instant non définie.

La formalisation induit que ce paramétrage est effectivement quelconque, et nous allons démontrer ici ce résultat quelque peu non intuitif :

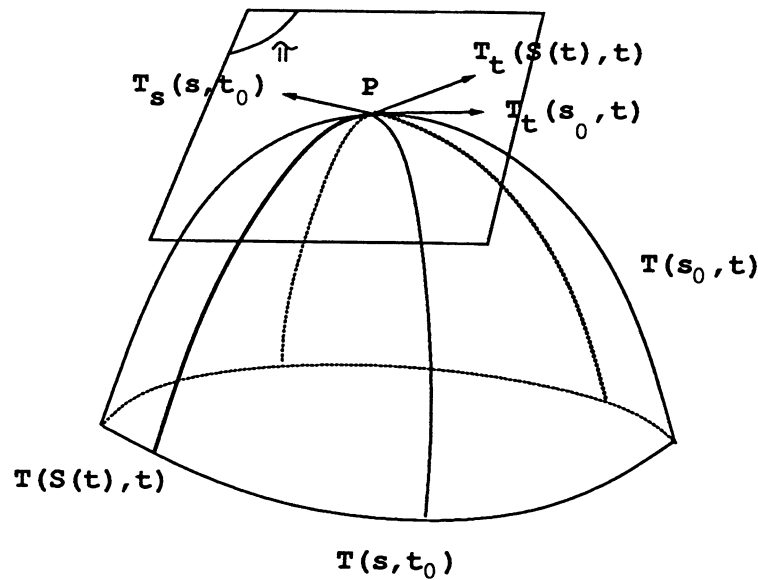


Figure 3.7. Deux paramétrisations de la surface spatio-temporelle

La nappe spatio-temporelle est définie intrinsèquement indépendamment d'une paramétrisation quelconque. La paramétrisation peut être explicitée en traçant sur la surface spatio-temporelle les courbes à "s constant". La figure 3.7 nous montre deux paramétrisations différentes de la surface spatio-temporelle, sous la forme de deux courbes passant par P_0 , respectant toutes deux le paramétrage temporel fixé. Le plan tangent π est représenté, et les deux différentielles temporelles des deux contours sont représentées :

1. une paramétrisation de référence avec une courbe $T(s_0, t)$.
2. une seconde paramétrisation "quelconque" avec une courbe $T(S(s_0, t_0, l), l)$.

Cette seconde paramétrisation est la forme la plus générale possible pour une paramétrisation locale de la surface spatio-temporelle qui respecte le paramétrage temporel et telle que $P_0 = T(s_0, t_0) = T(S(s_0, t_0, t_0), t_0)$. La fonction S vérifie $S(s_0, t_0, t_0) = s_0$ par construction.

En toute généralité, on note $H(s, t, l) = T(s, l)$ la courbe à s constant qui passe par le point $P = T(s, t)$ de la surface spatiotemporelle, et $G(s, t, l) = T(S(s, t, l), l)$ une autre courbe générale passant par P et respectant la paramétrisation temporelle. S vérifie $S(s, t, t) = s$.

On montre que la deuxième paramétrisation nous donne le même résultat pour la reconstruction :

Pour désambiguer les notations, on notera les dérivées partielles d'une fonction $K(u_1, u_2, \dots, u_r)$ $\partial_1 K, \partial_2 K, \dots, \partial_r K$.

On veut montrer que l'assertion suivante est vérifiée en P :

$$\begin{aligned} \lambda &= \frac{-\circ\dot{v}.\circ n}{\circ T_t.\circ n} \\ &= \frac{-\circ\dot{v}.\circ n}{\partial_3 G(s, t, t).\circ n} \\ &= \frac{-\circ\dot{v}.\circ n}{\partial_3 H(s, t, t).\circ n} \end{aligned}$$

c'est-à-dire que l'on a :

$$\begin{aligned} \partial_3 G(s, t, t).\circ n &= \left. \frac{\partial G(s, t, l)}{\partial l} \right|_{s, t, t} \cdot (T(s, t) \wedge \partial_1 T(s, t)) = \\ \partial_3 H(s, t, t).\circ n &= \left. \frac{\partial H(s, t, l)}{\partial l} \right|_{s, t, t} \cdot (T(s, t) \wedge \partial_1 T(s, t)) \end{aligned}$$

de façon à vérifier l'égalité des deux dénominateurs.

Par dérivation de G et H on obtient :

$$\begin{aligned} \frac{\partial G(s, t, l)}{\partial l} &= \partial_3 G(s, t, l) = \frac{dT(S(s, t, l), l)}{dl} \\ &= \partial_1 T(S(s, t, l), l) \cdot \partial_3 S(s, t, l) + \partial_2 T(S(s, t, l), l) \\ \frac{\partial H(s, t, l)}{\partial l} &= \partial_3 H(s, t, l) = \frac{dT(s, l)}{dl} = \partial_2 T(s, l) \end{aligned}$$

d'où :

$$\begin{aligned} \partial_3 G(s, t, l).\circ n &= (\partial_1 T(S(s, t, l), l) \cdot \partial_3 S(s, t, l) + \partial_2 T(S(s, t, l), l)) \cdot \\ &\quad (T(s, t) \wedge \partial_1 T(s, t)) \end{aligned}$$

au point $P(s, t)$ considéré on a :

$$\begin{aligned} \partial_2 T(S(s, t, t), t) &= \partial_2 T(s, t) \\ \partial_1 T(S(s, t, t), t) &= \partial_1 T(s, t) \end{aligned}$$

d'où au point $P(s, t)$ considéré :

$$\begin{aligned} \partial_3 G(s, t, t).\circ n &= \partial_2 T(S(s, t, t), t) \cdot (T(s, t) \wedge \partial_1 T(s, t)) \\ &= \partial_2 T(s, t) \cdot (T(s, t) \wedge \partial_1 T(s, t)) \\ &= \partial_3 H(s, t, t).\circ n \end{aligned}$$

D'où le résultat annoncé.

□

Il ressort de cette propriété fondamentale qu'il est possible de différentier en t la surface spatio-temporelle suivant une direction quelconque, ce qui sous-entend qu'il n'est nul besoin d'avoir une correspondance point à point spécifique. Toute correspondance est valable et la valeur du résultat est indépendante de la paramétrisation choisie.

3.5.5 Estimation des courbures totales et moyennes

La description intrinsèque des surfaces (à un déplacement de l'espace près) dépend de deux formes quadratiques notées I_P et II_P , et les coefficients de ces formes quadratiques doivent vérifier les trois équations de Gauss et Codazzi-Mainardi. On pourra trouver une preuve dans [Car 76].

La première forme fondamentale I_P permet de définir la métrique sur la surface.

La deuxième forme fondamentale II_P décrit comment la surface s'éloigne de son plan tangent, et est relative aux notions de courbure.

On pourra trouver dans l'annexe D un rappel de géométrie différentielle qui définit formellement les notions et notations utilisées ici.

Les endomorphismes associés aux deux formes fondamentales en $P = r(s, t)$ s'écrivent :

$$I_P = \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} {}^o r_t \cdot {}^o r_t & {}^o r_t \cdot {}^o r_s \\ {}^o r_s \cdot {}^o r_t & {}^o r_s \cdot {}^o r_s \end{bmatrix}$$

et

$$II_P = \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} {}^o r_{tt} \cdot {}^o n / |{}^o n| & {}^o r_{ts} \cdot {}^o n / |{}^o n| \\ {}^o r_{st} \cdot {}^o n / |{}^o n| & {}^o r_{ss} \cdot {}^o n / |{}^o n| \end{bmatrix}$$

Le type de surface en un point dépend du signe de la courbure totale (ou gaussienne) κ_{gauss} et de la courbure moyenne \mathcal{H} en ce point. Ces courbures dépendent des formes quadratiques, et se calculent indépendamment de la paramétrisation choisie (les notations sont celles rappelées dans l'annexe D) :

$$\begin{aligned} \mathcal{H} &= \frac{GL + EN - 2FM}{LN - M^2} \\ \kappa_{gauss} &= \frac{2\mathcal{H}^2}{H^2} \end{aligned} \quad (3.9)$$

avec :

$$H = EG - F^2$$

Calcul pratique de κ_{gauss} et de \mathcal{H}

$$\begin{cases} {}^o r_t \cdot {}^o n = 0 \\ {}^o r_s \cdot {}^o n = 0 \end{cases}$$

d'où par différentiation en s et t :

$$\begin{cases} {}^o r_{tt} \cdot {}^o n = -{}^o r_t \cdot {}^o n_t \\ {}^o r_{ss} \cdot {}^o n = -{}^o r_s \cdot {}^o n_s \\ {}^o r_{st} \cdot {}^o n = -{}^o r_s \cdot {}^o n_t \end{cases}$$

Cette manipulation utilisant les propriétés des contours critiques permet de passer d'une expression avec des dérivées troisièmes des mesures à une expression avec des dérivées secondes seulement.

Après quelques simples manipulations on obtient :

$$\begin{cases} {}^o r_t &= {}^o \dot{v} + \lambda_t {}^o T + \lambda {}^o T_t \\ {}^o r_s &= \lambda_s {}^o T + \lambda {}^o T_s \\ \lambda_s &= \frac{({}^o T_t \cdot {}^o n) \cdot ({}^o \dot{v} \cdot {}^o n_s) - ({}^o \dot{v} \cdot {}^o n) \cdot ({}^o T_{ts} \cdot {}^o n + {}^o T_t \cdot {}^o n_s)}{({}^o T_t \cdot {}^o n)^2} \\ \lambda_t &= \frac{({}^o T_t \cdot {}^o n) \cdot ({}^o \dot{v} \cdot {}^o n_t + {}^o \ddot{v} \cdot {}^o n) - ({}^o \dot{v} \cdot {}^o n) \cdot ({}^o T_{tt} \cdot {}^o n + {}^o T_t \cdot {}^o n_t)}{({}^o T_t \cdot {}^o n)^2} \\ {}^o r_{tt} \cdot {}^o n &= -{}^o \ddot{v} \cdot {}^o n - \lambda [{}^o T_{tt} \cdot {}^o n + 2 \cdot {}^o T_t \cdot ({}^o T \wedge {}^o T_{st})] - 2 \cdot {}^o \dot{v} \cdot {}^o n_t \\ {}^o r_{ss} \cdot {}^o n &= -\lambda {}^o T_s \cdot ({}^o T \wedge {}^o T_{ss}) = \lambda {}^o T_{ss} \cdot {}^o n \\ {}^o r_{st} \cdot {}^o n &= -{}^o \dot{v} \cdot {}^o n_s - \lambda {}^o T_t \cdot {}^o n_s \end{cases}$$

où tous les termes sont exprimés en fonctions des paramètres connus du mouvement et des mesures sur la surface spatio-temporelle. On peut donc calculer les premières et deuxièmes formes fondamentales en fonction des mesures sur la surface spatio-temporelle, de leurs dérivées premières et secondes et des paramètres du mouvement. L'expérimentation montre malheureusement que cette formulation reste extrêmement sensible aux erreurs de mesure, et ne peut en aucun cas fournir de résultats corrects en pratique. On pourra se rapporter à [Vai 90] ou [Bla 90] pour des techniques discrètes permettant une évaluation plus robuste des courbures de la surface.

3.6 Implémentation

Le chapitre 2 a présenté l'extraction des chaînes de contours à partir des images, leur segmentation aux points d'extrême de courbure, et leur représentation continue sous forme de B_splines.

Le chapitre 2.10 a présenté le suivi des contours entre les images successives de la séquence.

On possède maintenant un contour, suivi pendant n images successives, ainsi que les paramètres du mouvement qui correspondent.

La paramétrisation de la surface spatio-temporelle est obtenue en utilisant la géométrie épipolaire. La figure 3.8 présente cette géométrie graphiquement :

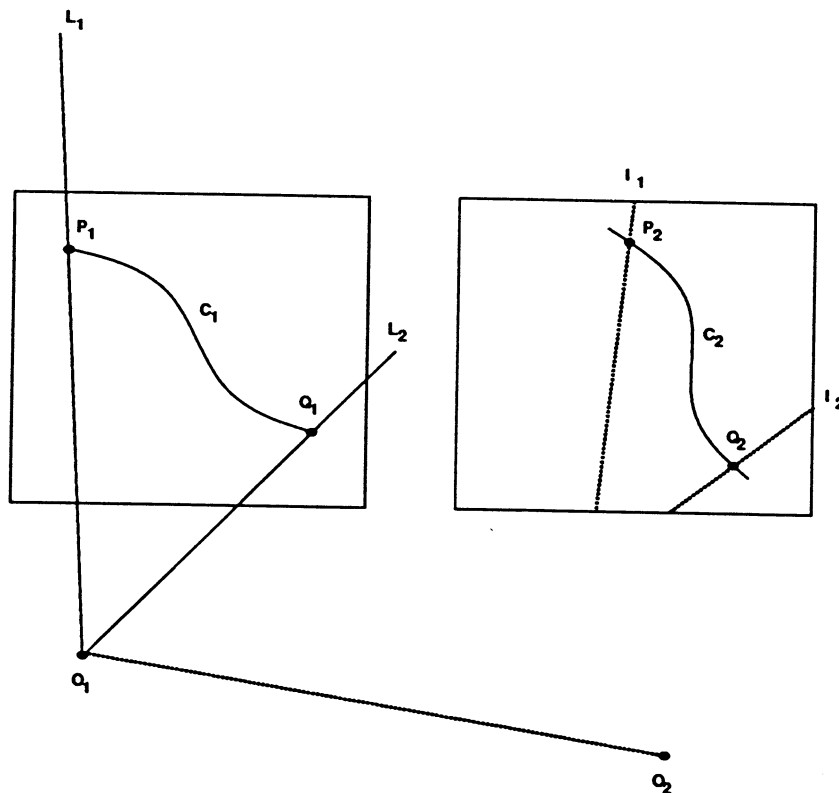


Figure 3.8. Géométrie épipolaire

Le contour C_1 de l'image prise au temps t doit être mis en relation avec le contour C_2 de l'image prise au temps $t + 1$. Un point P_1 de C_1 appartient à la ligne de vue L_1 . La projetée de la ligne de vue sur l'image prise au temps $t + 1$ est l_1 , couramment appelée épipôle de L_1 . L'intersection de l_1 avec C_2 définit le point P_2 . Cette construction permet de définir le correspondant sur C_2 d'un point de C_1 . Si P_1 à t a pour abscisse s sur C_1 ,

alors on définira l'abscisse de P_2 à $t + 1$ comme étant s également. On a vu que la reconstruction était indépendante de la paramétrisation ; ce principe permet de faciliter le calcul effectif des correspondances entre contours. Les extrémités P_1 et Q_1 de C_1 sont mis en correspondance avec P_2 et Q_2 de C_2 , et les points R_1 entre P_1 et Q_1 sont mis en correspondance avec les points R_2 entre P_2 et Q_2 par interpolation linéaire sur l'abscisse curviligne. Cette correspondance est établie sur les contours dans le plan image, mais les calculs ultérieurs sont faits, on le rappelle, en géométrie sphérique.

Les correspondances entre contours étant définies, on construit la surface spatio-temporelle $T(s, t)$ dans le repère $\{O\}$ en exprimant la direction de la ligne de vue dans $\{O\}$ pour chaque point R des contours, et en normalisant le vecteur obtenu.

La formule de reconstruction 3.8 nécessite l'évaluation de $T(s_0, t_0)$, $T_s(s_0, t_0)$ et de $T_t(s_0, t_0)$.

L'évaluation des courbures totales et moyennes nécessite de plus l'évaluation de $T_{ss}(s_0, t_0)$, $T_{st}(s_0, t_0)$ et de $T_{tt}(s_0, t_0)$.

Pour ce faire, chaque contour $T(s, t_0)$ est approximé par B_spline régularisée, ce qui permet une évaluation fine de $T(s_0, t_0)$, de $T_s(s_0, t_0)$ et de $T_{ss}(s_0, t_0)$.

Ensuite, les points autour de $T(s_0, t_0)$ sont recherchés le long de $T(s_0, t)$ et sont alors approchés par B_spline régularisée, ce qui permet une évaluation fine de $T_t(s_0, t_0)$ et de $T_{tt}(s_0, t_0)$.

En ces points autour de $T(s_0, t_0)$ le long de $T(s_0, t)$ sont calculées les dérivées $T_s(s_0, t)$ qui sont alors approximées par B_spline régularisée, ce qui permet une évaluation fine de $T_{st}(s_0, t_0)$.

${}^{\circ}\dot{v} \cdot {}^{\circ}n$ est évalué et si sa valeur est supérieur à un seuil donné (les vecteurs ${}^{\circ}\dot{v}$ et ${}^{\circ}n$ doivent faire un angle différent d'au moins 10 degrés d'un angle droit), alors la profondeur λ et les courbures moyenne et totale sont calculées.

Remarque concernant la dérivation spatiale Soit $T(s, t_0)$ un contour au temps t_0 , et $K(S, t_1)$ (avec $t_1 = t_0 + 1$) le contour suivant. La correspondance linéaire entre les contours impose une relation entre S et s : $S = a * s + b$. La dérivée de K par rapport à s s'écrit donc trivialement: $K_s = \frac{d K(a*s+b, t_1)}{d s} = a * d(K(S, t_1))/dS$

3.7 Résultats expérimentaux

Deux séries de résultats sont présentées, l'une avec des données synthétiques, l'autre en données réelles. Les données synthétiques permettent à la fois de vérifier la validité des algorithmes (et de leur implémentation) et

de tester la sensibilité à un certain nombre de facteurs. Les données réelles montrent la validité du principe en face des incertitudes et erreurs présentes dans la réalité.

3.7.1 Données synthétiques

Deux types de données sont présentées ci-dessous: une courbe de discontinuité dans l'espace, en l'occurrence une simple droite, et une sphère dans l'espace.

Droite dans l'espace

La figure 3.9 montre une séquence de six images synthétiques d'un segment de l'espace. La projection du segment est bruitée puis digitalisée. Plus précisément, chaque point considéré de la droite dans l'espace est projeté sur le plan image; les coordonnées de la projection sont perturbées par un bruit additif, de moyenne nulle et de variance donnée. Les coordonnées sont alors arrondies à l'entier le plus proche.

Le mouvement de la caméra consiste en une translation uniforme de 0.1 mètre par unité de temps selon les deux axes Y et Z du plan parallèle au plan image et en une rotation de 0.1 radian par unité de temps selon l'axe optique Z.

Variance bruit en pixels	erreur moyenne	erreur maxi	erreur mini	Ecart-type
0	0.024305	0.404977	0.005878	0.058557
1	0.017484	0.181759	0.000381	0.023403
2	0.032886	0.182845	0.000266	0.030402
3	0.047908	0.168593	0.001014	0.024262
4	0.062663	0.261031	0.012367	0.025969
5	0.086245	0.137115	0.035951	0.022693

Tableau 3.1. Erreurs en Z de reconstruction du segment

Le segment est en fait reconstruit comme une surface, donc chaque contour observé est reconstruit indépendamment dans l'espace. On obtient donc autant de reconstructions du segment de l'espace que d'images. Chaque segment reconstruit est alors projeté sur chaque image, comme illustré dans la figure 3.10. Nous appelons ce procédé la rétroprojection de la reconstruction.

On observe un recouvrement visuellement parfait, et les données numériques montrent que la reconstruction est exacte à quelques dixièmes de pourcent près. La reconstruction n'est pas notablement affectée par le bruit gaussien sur les contours, grâce à la qualité du lissage par B_spline. Il convient de noter cependant que le bruit engendré est éloigné des bruits réels

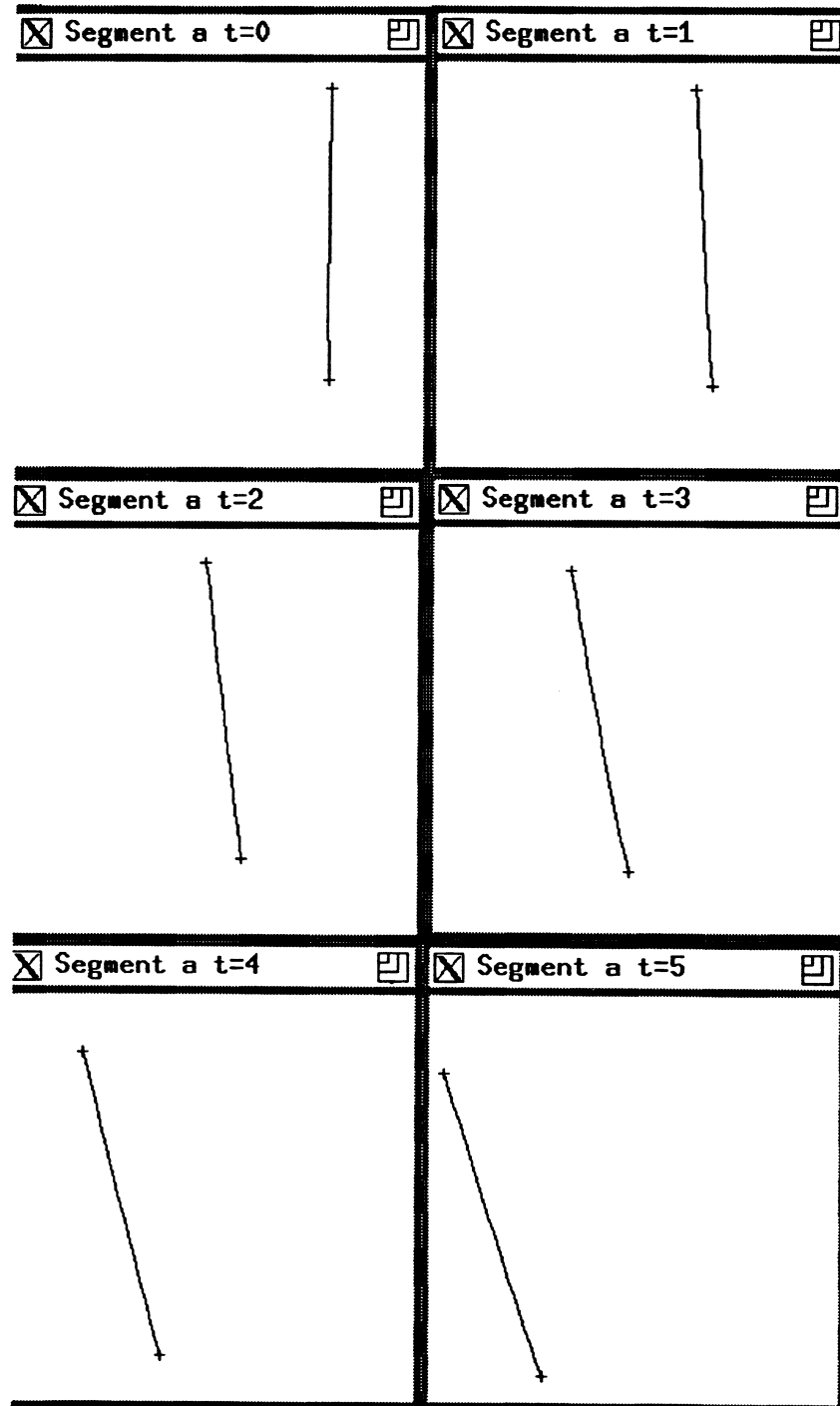


Figure 3.9. Séquence du segment dans l'espace

rencontrés dans une image, car les bruits réels sont souvent à moyenne non nulle. La table 3.1 récapitule l'évolution de l'erreur moyenne absolue et de l'erreur maximale absolue (en mètre) de reconstruction sur la profondeur du segment, en fonction du bruit ajouté le long des contours; la profondeur théorique du contour est de 6 mètres. On notera que la valeur du coefficient de régularisation τ a été conservée constante et égale à la valeur habituellement utilisée pour les images réelles. L'erreur maximale absolue a systématiquement été observée aux bords des contours suivis, et est due aux problèmes de lissage spatial aux extrémités des contours.

Sphère dans l'espace

La figure 3.12 montre une séquence de 4 images d'une sphère dans l'espace. Les images synthétiques sont obtenues par "lancé de rayon" et sont donc affectées par un bruit de discrétisation. Comme précédemment, le lissage est conservé à ses valeurs habituelles, et n'a pas été ajusté de manière optimale. Le mouvement de la caméra consiste en une translation simple du centre optique C le long de l'axe Y , sans aucune rotation (voir figure 3.11).

La rétroprojection de la reconstruction sur chaque image de la séquence n'est pas présentée, car peu lisible : les contours rétroprojetés se superposent avec le contour extérieur de la sphère, et aucune observation significative n'est possible.

La figure 3.13 montre par contre les contours reconstruits, vus en perspective depuis trois points de vue différents. La première vue correspond à une des positions de la caméra. La deuxième est prise depuis l'axe Y , mais d'une position plus excentrée que la dernière image prise. La dernière correspond à une vue latérale, pratiquement dans le plan des cercles critiques vus sur la sphère, seule position où les "erreurs en Z " sont clairement visibles. Les pôles de la sphère ne sont pas reconstruits car ils correspondent aux zones où le déplacement de la caméra se fait dans le plan tangent à la sphère ; l'algorithme marque donc ces points comme non reconstructibles.

Il ressort de ces résultats que la reconstruction est excellente relativement aux données : la rétroprojection des données sur les images montre une cohérence parfaite. Par contre, la dernière vue latérale révèle clairement les défauts de reconstruction non visibles depuis les positions correspondant aux prises de vue. Il ressort donc de cette observation l'importance de ne pas se fier à la rétroprojection pour juger de la "qualité" de la reconstruction, résultat bien connu pour les reconstructions de stéréovision.

La "qualité" de la reconstruction va être évaluée en calculant la distance du point reconstruit au centre théorique de la sphère. La sphère observée

est une sphère de 0.1 mètre de rayon observée à 1 mètre de distance.. La table 3.2 récapitule les moyenne, maximum, minimum et écart type de la distance du point reconstruit au centre de la sphère.

	Moyenne	Valeur maxi	Valeur mini	Ecart-type
Rayon	0.100007	0.100647	0.099699	0.000138

Tableau 3.2. Valeurs des rayons pour la sphère reconstruite

Un autre facteur mérite d'être étudié : il s'agit de la discrétisation temporelle. L'influence du pas de déplacement de la caméra influe fortement sur la qualité de reconstruction :

en effet, plus les prises de vue sont rapprochées, plus on se rapproche des hypothèses du continu qui sous-tendent notre approche différentielle de la reconstruction, mais plus le rapport signal/bruit des mesures différentielles chute. Les déplacements dans le plan image sont en effet très faibles, et de plus en plus proches du bruit de mesure.

Plus les prises de vue sont éloignées, plus on s'éloigne du continu, mais meilleur est le rapport signal/bruit des mesures car la valeur du bruit devient faible par rapport à la valeur du déplacement.

Ce fait est illustré par l'expérience suivante : la caméra se déplace toujours linéairement le long de l'axe Y en observant la même sphère que précédemment. On fait varier le pas du déplacement entre les prises de vue. Pour chaque valeur du pas, on prend une séquence complète d'images, et l'on reconstruit la sphère. On remarque l'existence d'une valeur optimale pour un pas de 0.1 m dans la table 3.7.1 qui montre l'évolution des mesures en fonction du pas.

Pas = 0.01	Moyenne	Valeur maxi	Valeur mini	Ecart-type
Rayon	0.102841	0.137345	0.099790	0.005415
Pas = 0.05	Moyenne	Valeur maxi	Valeur mini	Ecart-type
Rayon	0.100257	0.110117	0.099637	0.000954
Pas = 0.1	Moyenne	Valeur maxi	Valeur mini	Ecart-type
Rayon	0.100007	0.100647	0.099699	0.000138
Pas = 0.15	Moyenne	Valeur maxi	Valeur mini	Ecart-type
Rayon	0.100060	0.101582	0.099401	0.000218
Pas = 0.18	Moyenne	Valeur maxi	Valeur mini	Ecart-type
Rayon	0.100148	0.100823	0.099431	0.000241

Tableau 3.3. Influence du pas de discrétisation temporel

L'influence du bruit sur les mesures est clairement visible pour les petits pas de déplacement, et conduit à des résultats très approximatifs (jusqu'à

37% d'erreur pour un pas de 0.01 m , pour l'erreur maximale). Les déplacements plus grands se dégradent beaucoup moins grâce au lissage de la surface spatio-temporelle.

3.7.2 Données réelles

Les séquences utilisées sont présentées avec 4 images pour des raisons de place. Les calculs de reconstruction sont menés eux sur les séquences complètes (10 à 15 images). En effet, le lissage de la surface spatiotemporelle est d'autant meilleur que le nombre d'images pris en compte est grand.

Séquence du dessin plat

La figure 3.14 montre une séquence de quatre images d'un dessin plat. Le mouvement de la caméra correspond à une translation simple.

La figure 3.15 correspond à la rétroprojection de la reconstruction des contours. Les contours blancs correspondent à la projection des contours reconstruits sur chacune des images de la séquence. Le contour reconstruit est le contour visible en gras sur la figure 3.14. La surface reconstruite apparaît qualitativement correcte sur les projections.

La figure 3.16 nous montre une vue latérale de la surface reconstruite. Le plan image où est projeté la surface reconstruite est un plan parallèle au plan (Y, Z) de la scène. L'objet de la scène étant plat, nous nous attendons à observer une seule ligne horizontale. Les erreurs de reconstruction sont visibles au mieux dans cette figure car elles sont des erreurs de profondeur dans la direction Z .

La figure 3.17 nous montre une vue générale de la surface reconstruite, prise latéralement et au dessus de la surface.

Plus quantitativement, nous allons faire des mesures sur le contour reconstruit. Nous allons en particulier vérifier la planarité du contour reconstruit, et ses dimensions dans le plan.

Moyenne	Valeur maxi	Valeur mini	Ecart-type
929.5	933.1	920.8	1.4

Tableau 3.4. Valeurs des coordonnées en Z pour le dessin plat

La table sur laquelle est posée la feuille de papier est orthogonale à l'axe Z de la caméra. Les coordonnées en Z des points reconstruits du contour sont résumées dans la table 3.4 ; on y trouve la valeur moyenne de la coordonnée en Z , l'écart type, la valeur maximale et la valeur minimale.

La figure 3.18 nous montre deux dimensions caractéristiques H et L du contour, que nous pouvons mesurer sur le contour reconstruit pour chaque position envisagée. La table 3.5 résume le résultat de ces mesures :

	H	L
théorique	55	7
image 0	57.4	8.5
image 1	55	7.9
image 2	56.1	7.6
image 3	56.7	7.7

Tableau 3.5. Valeurs de H et L pour le dessin plat

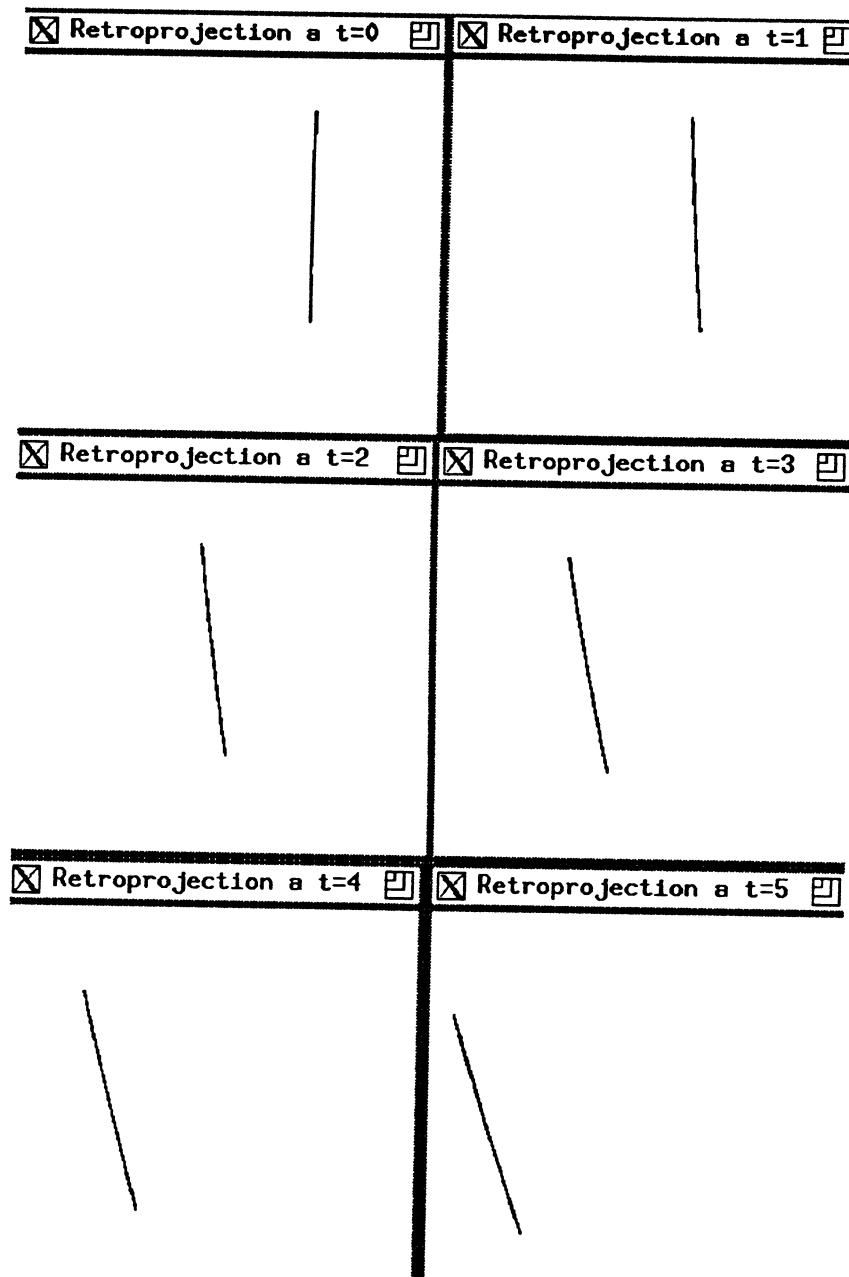


Figure 3.10. Rétroprojection du segment reconstruit

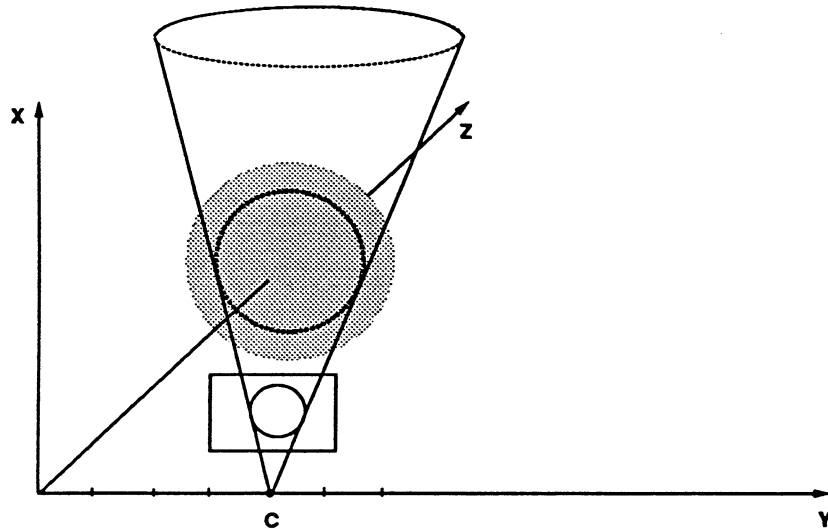


Figure 3.11. Mouvement de la caméra pas rapport à la sphère

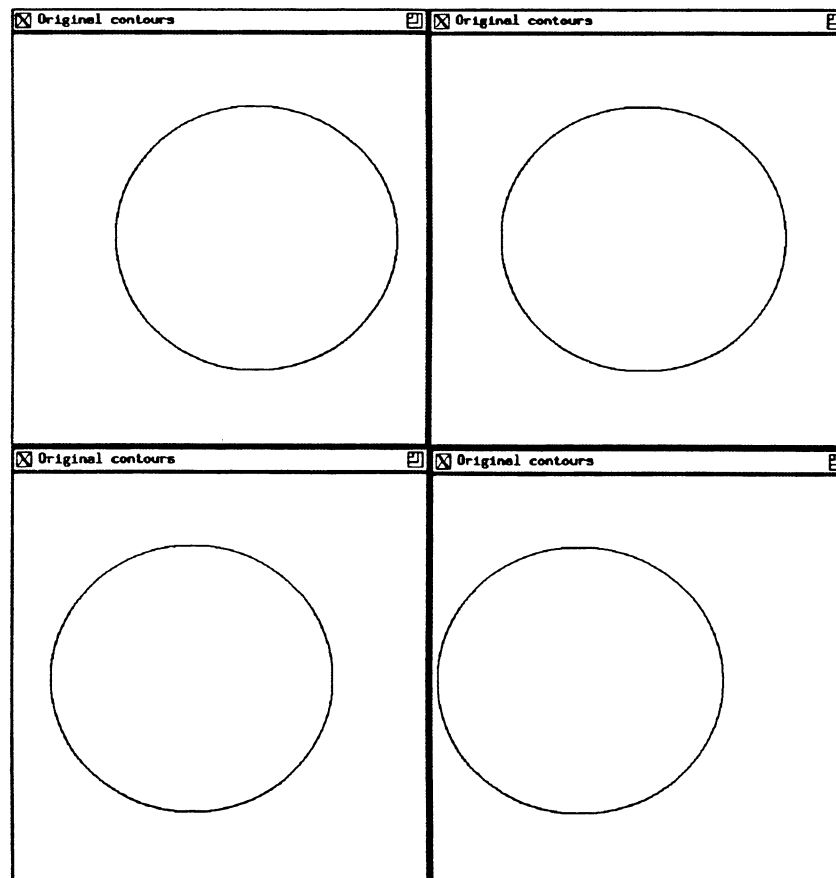


Figure 3.12. Séquence de la sphère dans l'espace

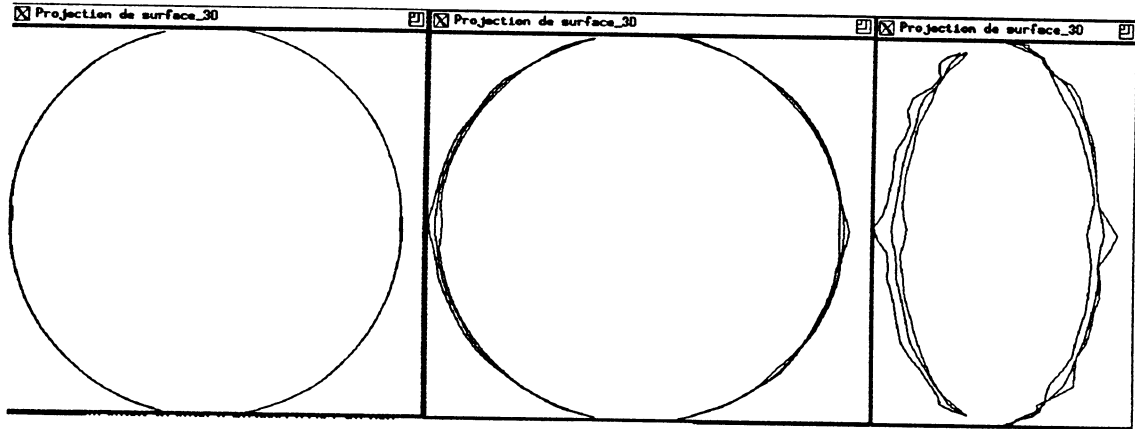


Figure 3.13. Séquence de la sphère dans l'espace

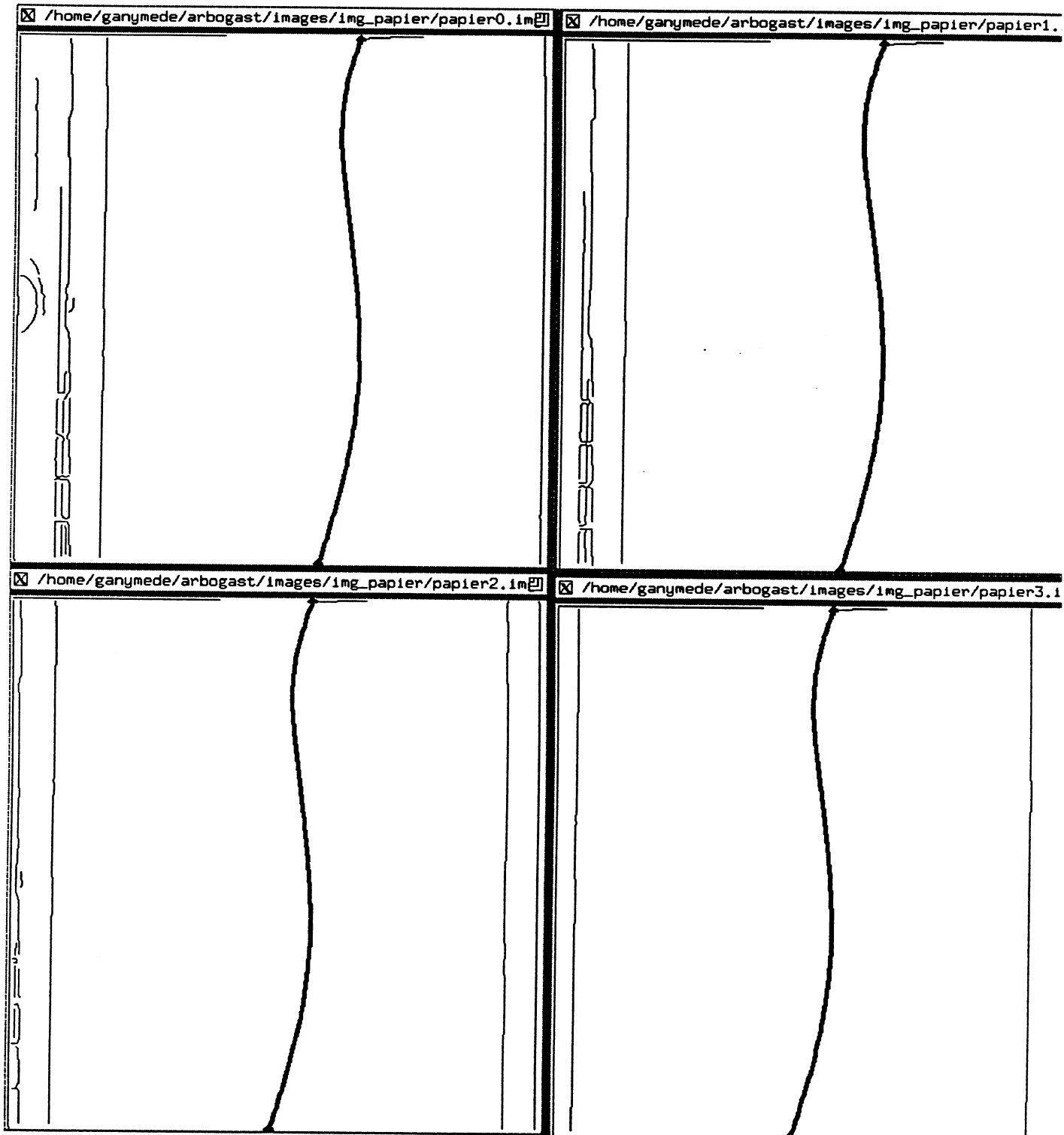


Figure 3.14. Séquence du dessin plat

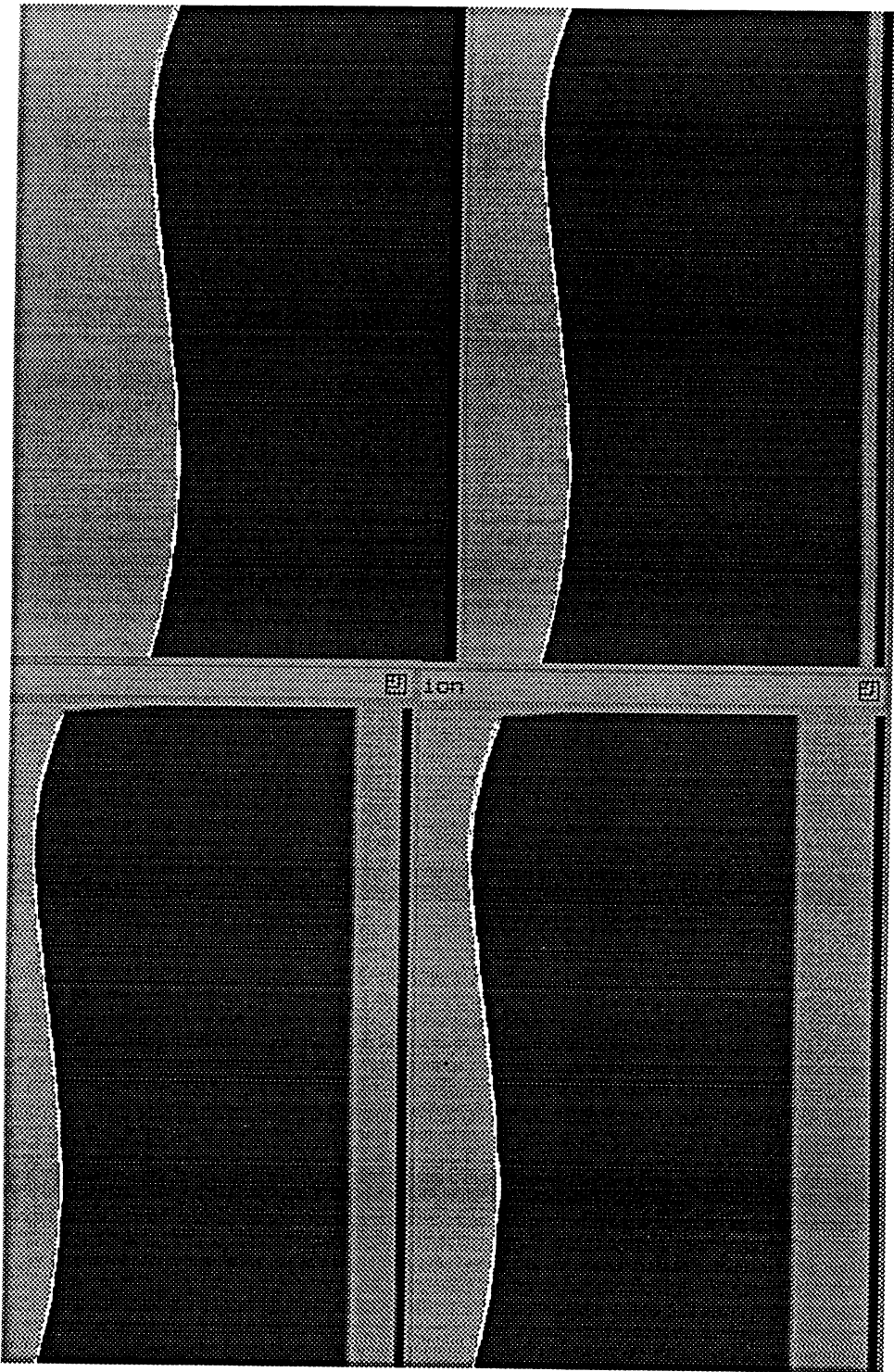


Figure 3.15. Reconstruction d'un contour de la séquence du dessin plat

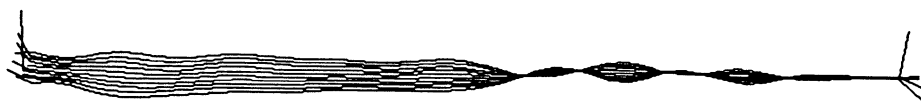


Figure 3.16. Vue latérale de la reconstruction de la séquence du dessin plat

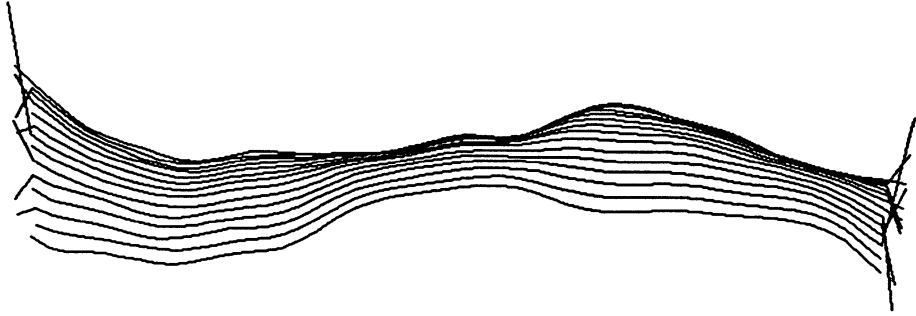


Figure 3.17. Vue générale de la reconstruction de la séquence du dessin plat

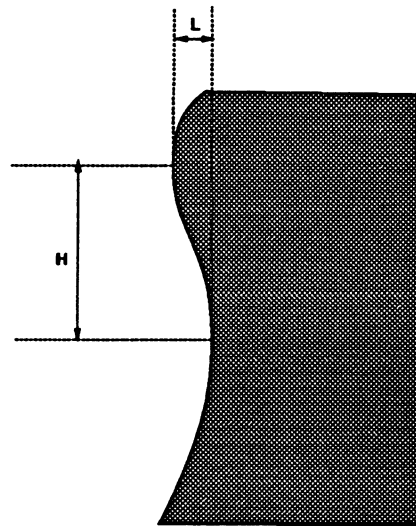


Figure 3.18. Dimensions du dessin

Séquence de l'ampoule électrique

La figure 3.19 montre une séquence de quatre images d'une ampoule électrique. Le mouvement de la caméra correspond à une translation simple.

La figure 3.20 correspond à la rétroprojection de la reconstruction des contours. Les contours blancs correspondent à la projection des contours reconstruits sur chacune des images de la séquence. Le contour reconstruit est le contour visible en gras sur la figure 3.19. La surface reconstruite apparaît qualitativement correcte sur les projections.

La figure 3.21 nous montre une vue latérale de la surface reconstruite. Le plan image où est projeté la surface reconstruite est un plan parallèle au plan (Y, Z) de la scène. Les erreurs de reconstruction sont visibles au mieux dans cette figure car elles sont des erreurs de profondeur dans la direction Z .

La figure 3.22 nous montre une vue générale de la surface reconstruite, prise latéralement et au dessus de la surface.

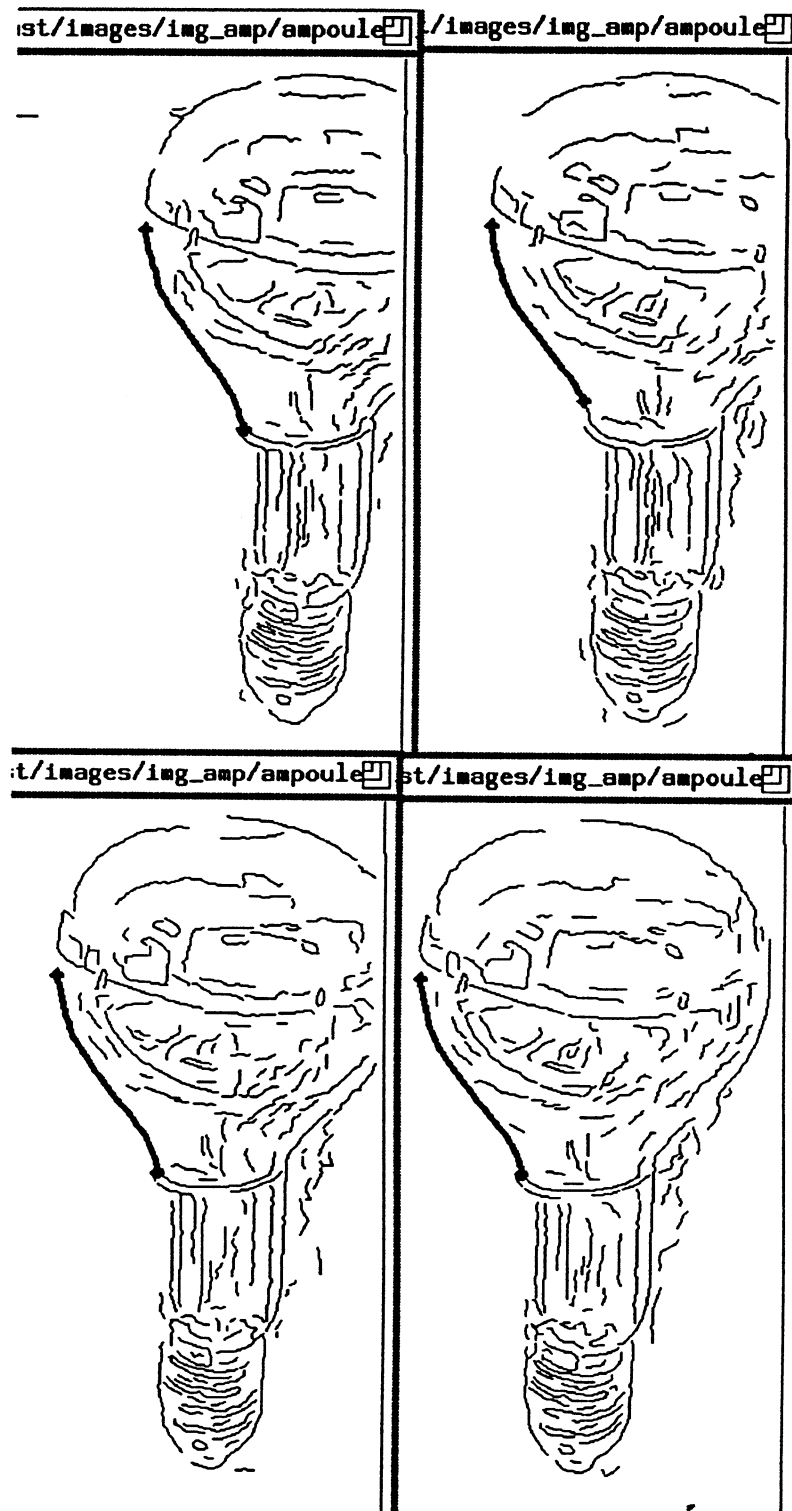


Figure 3.19. Séquence de l'ampoule électrique

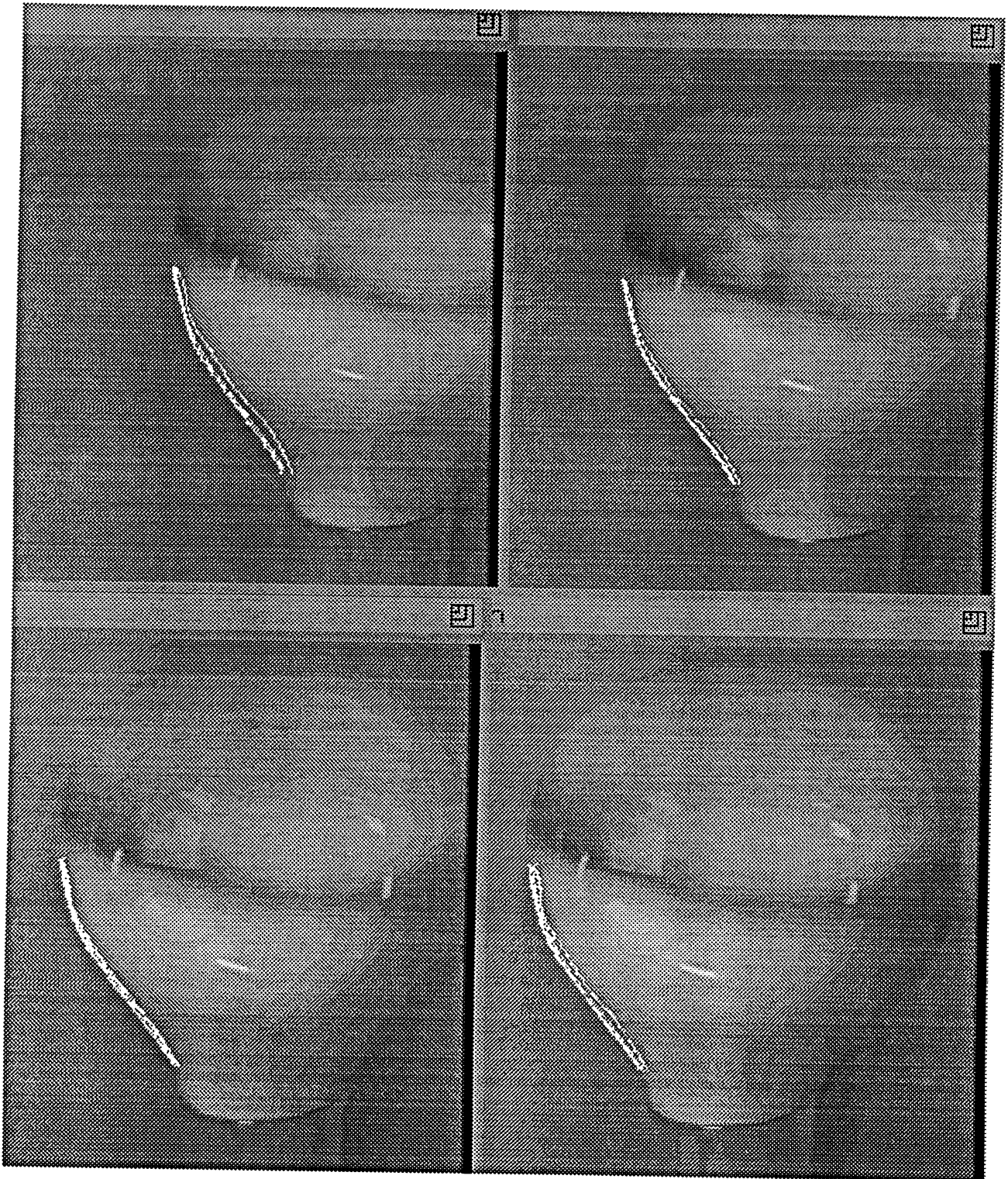


Figure 3.20. Reconstruction de la séquence de l'ampoule électrique

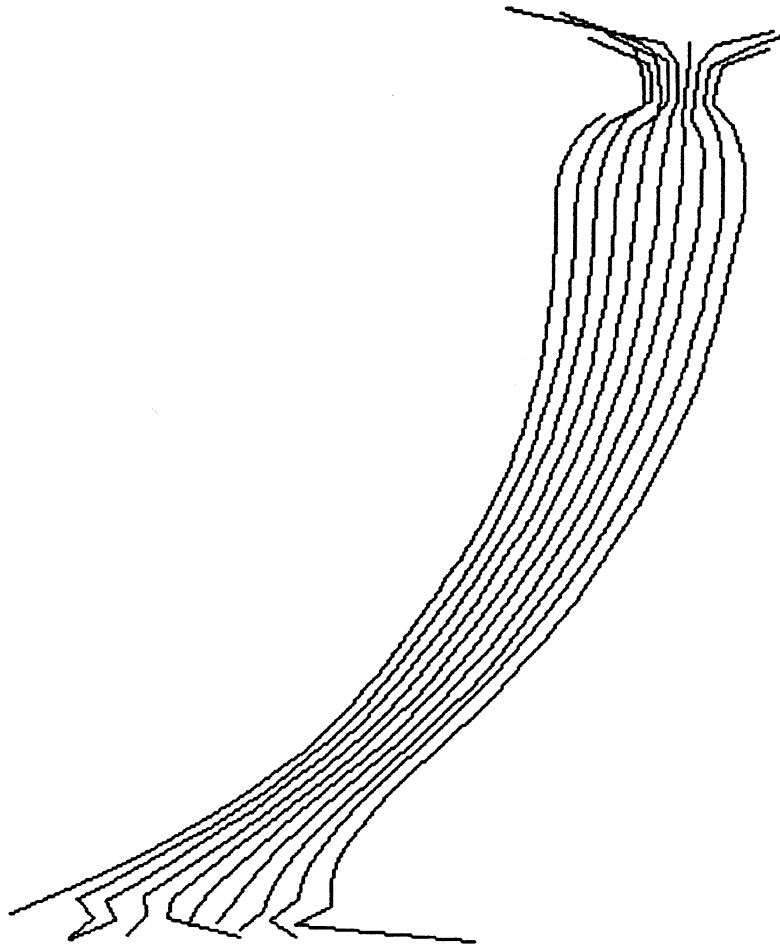


Figure 3.21. Vue latérale de la reconstruction de la séquence de l'ampoule

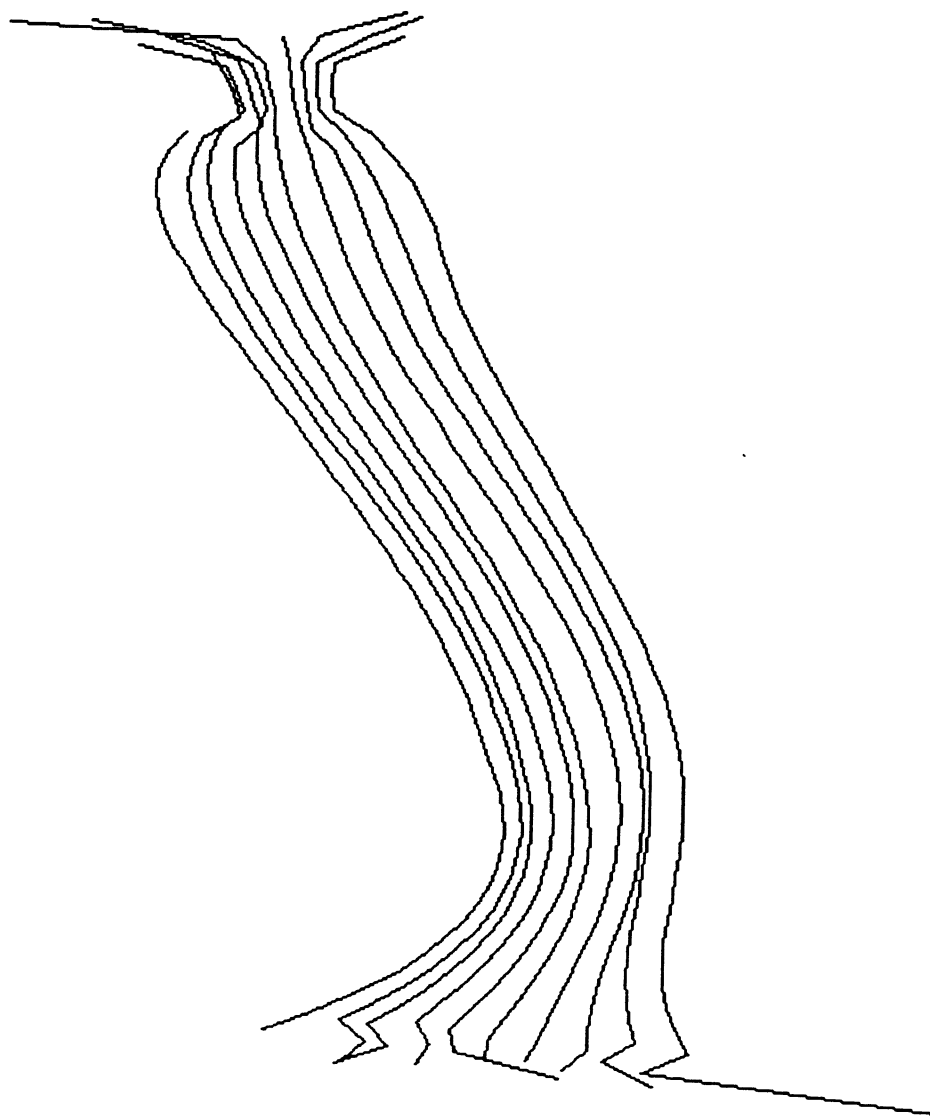


Figure 3.22. Vue générale de la reconstruction de la séquence de l'ampoule

Séquence du verre

La figure 3.23 montre une séquence de quatre images d'un verre. Le mouvement de la caméra correspond à une translation simple.

La figure 3.24 correspond à la rétroprojection de la reconstruction des contours. Les contours blancs correspondent à la projection des contours reconstruits sur chacune des images de la séquence. Le contour reconstruit est le contour visible en gras sur la figure 3.23. La surface reconstruite apparaît qualitativement correcte sur les projections.

La figure 3.25 nous montre une vue latérale de la surface reconstruite. Le plan image où est projeté la surface reconstruite est un plan parallèle au plan (Y, Z) de la scène. Les erreurs de reconstruction sont visibles au mieux dans cette figure car elles sont des erreurs de profondeur dans la direction Z .

La figure 3.26 nous montre une vue générale de la surface reconstruite, prise latéralement et au dessus de la surface.

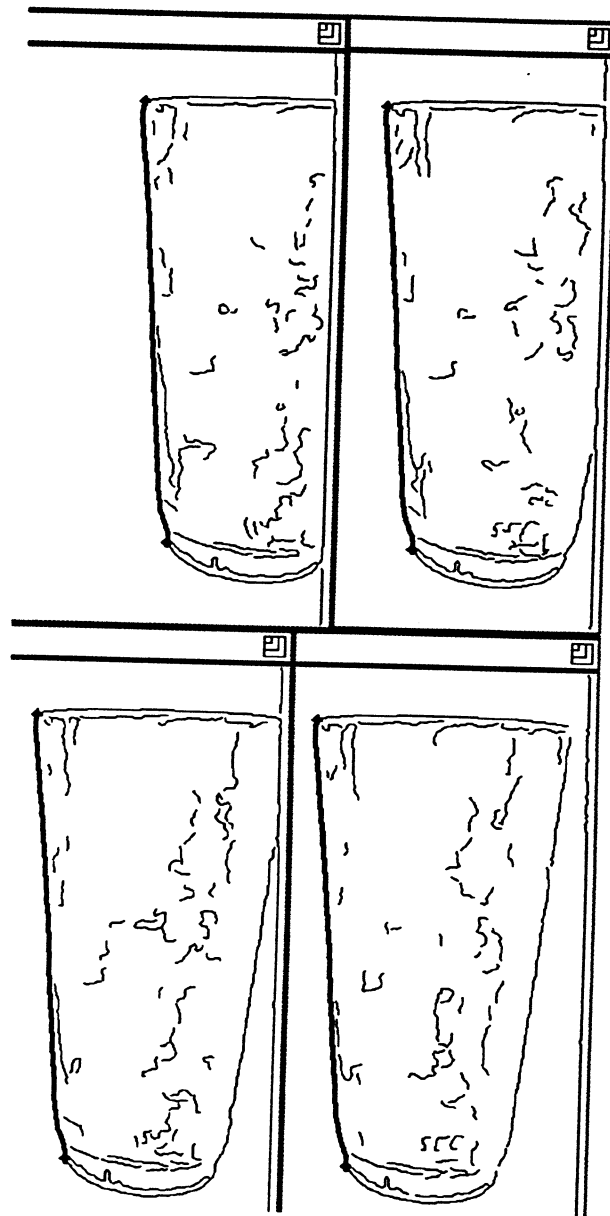


Figure 3.23. Séquence du verre

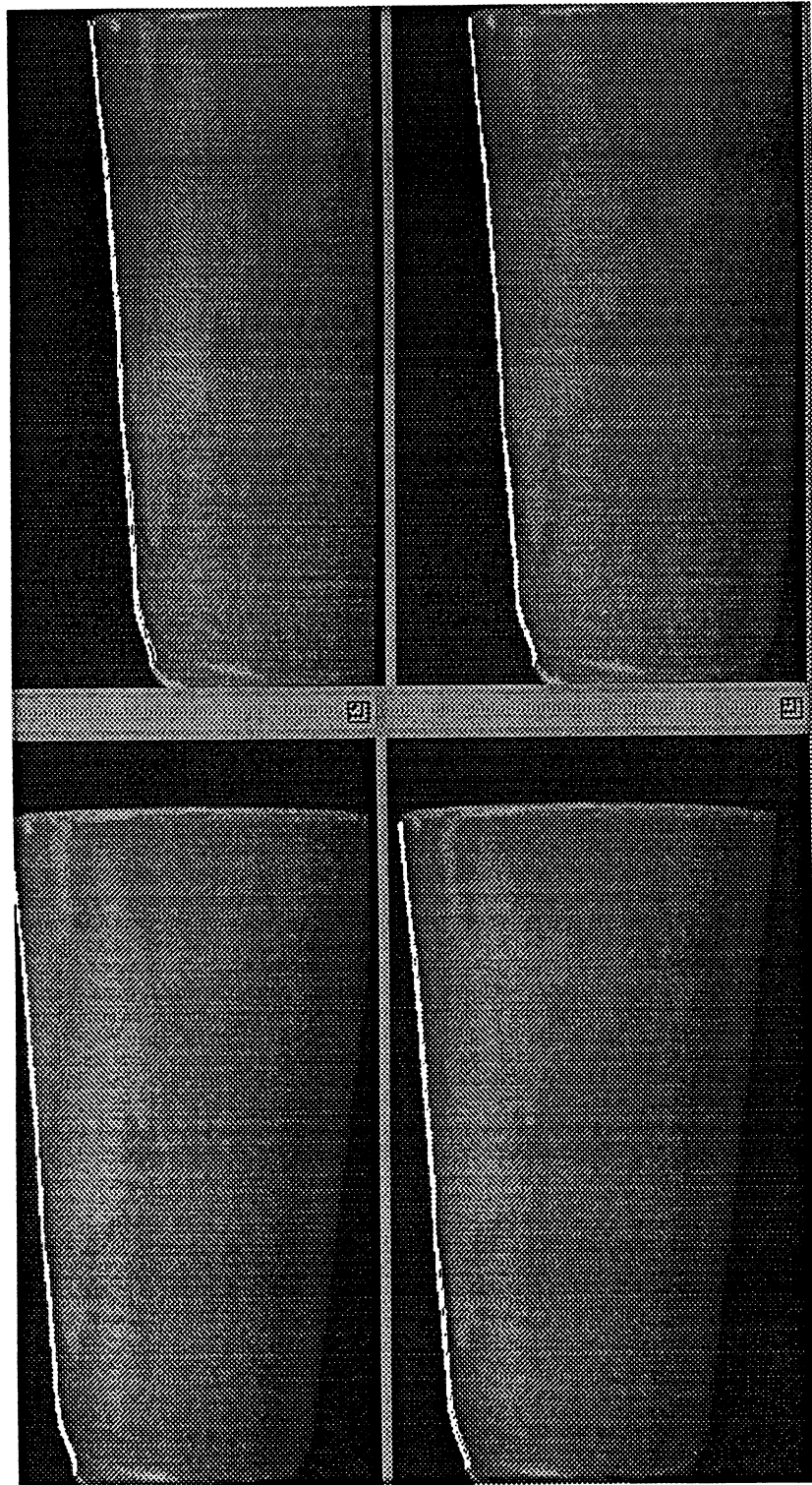


Figure 3.24. Reconstruction d'un contour de la séquence du verre

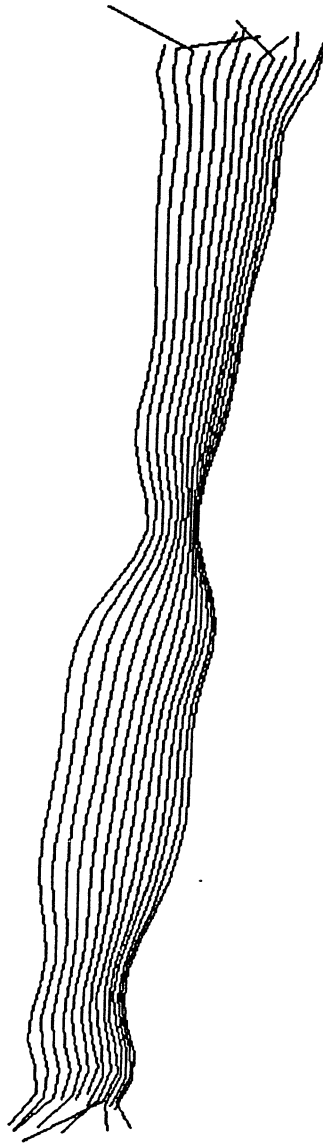


Figure 3.25. Vue latérale de la reconstruction de la séquence du verre

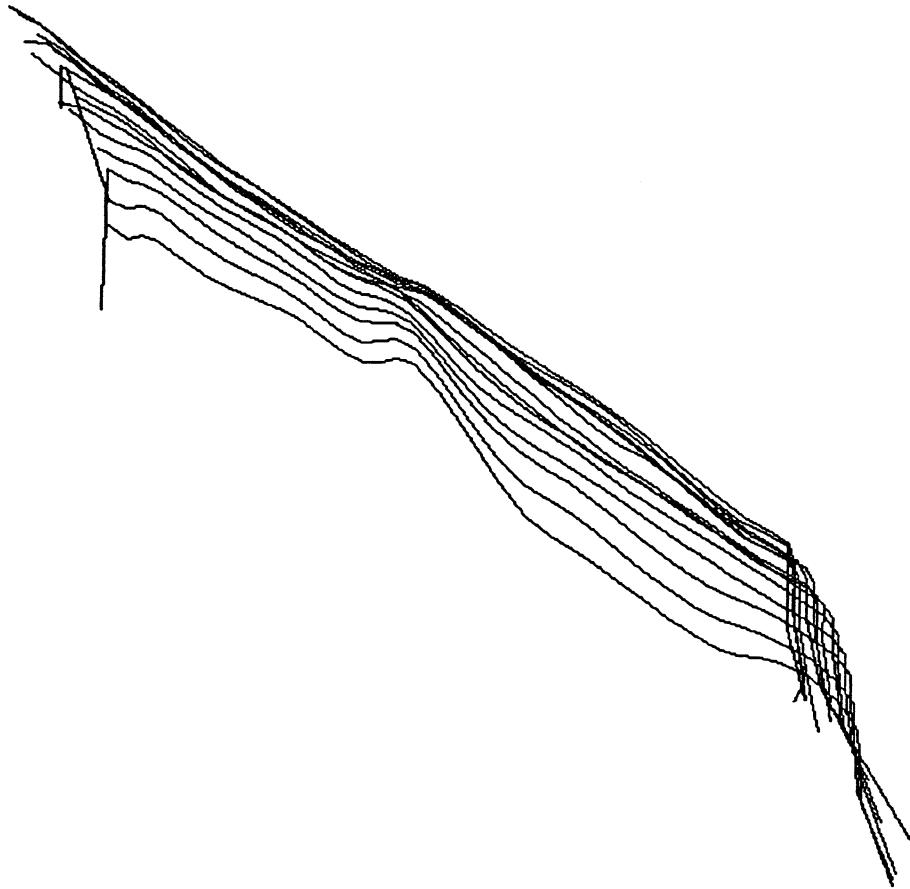


Figure 3.26. Vue générale de la reconstruction de la séquence du verre

3.7.3 Analyse de la précision

Deux facteurs essentiels influent sur la précision des résultats de reconstruction.

Tout d'abord, il s'agit d'une méthode différentielle, et elle est donc sensible aux erreurs de mesure, car ces erreurs sont amplifiées par la dérivation. Il s'agit donc de soigner l'estimation des dérivées par un filtrage soigneux des mesures. De plus, l'estimation dépend de manière cruciale des paramètres du mouvement, or cette détermination est difficile à faire précisément.

Ensuite, le choix du déplacement de la caméra est important : on a vu que le déplacement des contours devait être suffisant pour garantir un rapport signal/bruit correct. De plus, pour optimiser la précision de la reconstruction en un point, il faut déplacer la caméra orthogonalement au plan tangent en ce point, comme le montre l'observation de la formule de reconstruction.

3.8 Conclusions

Nous avons présenté une technique de reconstruction de surface à partir des contours critiques. Cette méthode permet d'unifier la reconstruction des surfaces et la reconstruction des lignes de discontinuité, donnant une grande robustesse vis-à-vis des erreurs de segmentation des contours. On notera que les surfaces planes et concaves ne présentent aucun contour critique à l'observateur, et qu'elles ne peuvent donc être ni décelées, ni reconstruites par cette technique. Cette méthode de reconstruction devra donc être envisagée en collaboration avec d'autres méthodes de façon à fournir une description complète de la surface des objets.

Le sentiment de l'auteur est que cette technique de reconstruction est très propre d'un point de vue théorique et sensible au bruit sur la position des contours. Cependant, la technique de détection de contours utilisée [Der 87a] et le lissage par B-splines régularisées [Lau 87] sont des outils permettant de s'affranchir du problème du bruit, en permettant des estimations fiables des dérivées le long des contours.

Malgré l'aspect différentiel de la méthode, nous montrons que la reconstruction est possible avec des déplacements assez grands de la caméra par rapport à l'objet.

La précision demandée sur les paramètres du mouvement est importante, mais atteignable en pratique avec des robots de manipulation bien calibrés.

Les autres approches citées [Cip 89] [Vai 89] souffrent du même problème, et nécessitent un calibrage fin préalable à toute manipulation, avec un dispositif expérimental particulier.

Cette technique profitera sans aucun doute des apports d'approches sans calibrage comme celles de R. Mohr, L. Morin et E. Grosso [Moh 91b, Moh 91a].

On trouvera en annexe F une comparaison entre cette technique de reconstruction et les techniques fondées sur le flot optique. La lecture de cette annexe nécessite d'avoir lu au préalable le chapitre 4 suivant sur l'autocalibration.

Chapitre 4

Autocalibration du mouvement

Ce chapitre aborde le problème de l'estimation du mouvement de la caméra par observation des courbes de discontinuité de la scène.

Une description intuitive de la solution sera suivie par la formalisation mathématique.

Un algorithme de résolution simple est enfin présenté et ses performances sont évaluées sur des données synthétiques et des données réelles..

4.1 Introduction

Au chapitre précédent, nous avons abordé le problème de la reconstruction des objets non nécessairement polyédriques par vision monoculaire, dans le cas où le mouvement de la caméra est connu.

Nous nous intéressons maintenant au cas où le mouvement de la caméra par rapport à l'objet observé est inconnu, et le problème consiste à déterminer le mouvement à partir d'informations visuelles.

Le cas des objets polyédriques est déjà relativement bien compris, et l'on sait déterminer les paramètres du mouvement de la caméra à partir du suivi de points ou de droites dans une séquence d'images.

Dans le contexte d'objets non nécessairement polyédriques, la problématique est différente en ce sens que les contours de discontinuité ne sont plus rectilignes. Nous nous intéressons donc ici au problème de la détermination du mouvement à partir de l'observation de contours quelconques en mouvement dans l'image. Cette détermination peut nous servir ultérieurement à la reconstruction telle qu'envisagée dans le chapitre 3 précédent.

En dehors de l'autocalibration, les paramètres du mouvement peuvent provenir de plusieurs sources.

4.1.1 Caméra mobile ou objet mobile

Dans le cas où la caméra (ou l'objet) est monté sur un support mobile dont on connaît les paramètres du mouvement, il est possible d'utiliser directement les données capteur de la base mobile.

Dans le cas des robots d'assemblage où la précision de déplacement est maximale, on s'aperçoit que la précision finale obtenue, excellente en relatif, est moins bonne en absolu, mais suffisante pour les besoins de la reconstruction.

Par contre cette méthode n'a plus aucun sens dès que c'est l'objet qui bouge avec un mouvement inconnu, ou dans le cas de plusieurs objets mobiles dans la scène.

4.1.2 Calibration dynamique avec un système de mires

Une autre approche consiste à calibrer le mouvement en utilisant une caméra. Il s'agit donc d'installer dans le champs de vision une mire parfaitement connue, de bouger mire ou caméra suivant le même mouvement que celui désiré lors de la phase d'expérimentation réelle, d'utiliser un calibrage dynamique pour les paramètres extrinsèques de la caméra [Len 87] [Tsa 87], puis de remplacer la mire par l'objet inconnu pour l'expérimentation. Cette approche, satisfaisante dans le cadre de la modélisation d'un objet en laboratoire [Cip 89], est inapplicable dans le cadre plus général d'exploration d'un environnement inconnu, ou dans le cas où plusieurs objets se déplacent avec des mouvements indépendants dans la scène.

4.1.3 Repérage par rapport à des balises

Si la scène contient des balises dont la position 3D est connue, il est alors possible d'envisager une toute autre approche, permettant d'ignorer totalement le problème de calibrage de la caméra. Ainsi, si l'on se réfère à [Moh 90], on s'aperçoit qu'il est possible, si l'on connaît quelques balises présentes dans l'image, de déterminer directement les paramètres dont nous avons besoin sans rien connaître sur la caméra ni sur son mouvement. Il est possible de plus de reconstruire "qualitativement" dans le cas où les balises utilisées ne sont pas connues en 3D, en leur affectant des positions arbitraires. On reconstruit ainsi de manière générique, relativement "à la position des balises". En outre, ce positionnement relatif par rapport au référentiel des balises est plus robuste que le positionnement dans le repère caméra, comme montré par L. Morin, E. Grosso et R. Mohr [Moh 91b].

La dernière technique utilisable est celle qui nous intéresse, c'est-à-dire l'autocalibration du mouvement.

4.2 Etat de l'art en autocalibration du mouvement

Le problème discret de l'autocalibration du mouvement consiste à retrouver la translation et la rotation de la caméra entre deux prises de vue (on suppose la scène statique).

Un certain nombre d'auteurs ont abordé le problème en supposant connu un certain nombre de points (ou de droites) dans la scène [Hor 87].

On suppose maintenant qu'aucun point dans la scène n'est connu a priori, et que seules sont disponibles n correspondances entre points sur les plans images de plusieurs vues [Fau 87a] [Wen 89] [Tos 87] [Tsa 84] [LH 81] [Hor 86]. Parmi cette deuxième famille de travaux (qui ne nécessite aucune connaissance a priori sur les objets observés), on peut discerner deux sous classes :

- Trouver le nombre de points minimum pour obtenir les paramètres extrinsèques de la caméra (position et orientation). Ainsi, Prazdny [Pra 80], Nagel et Neumann [Nag 81] montrent que le problème peut être résolu avec 5 correspondances de points, fournissant trois équations non linéaires. Faugeras et Maybank [Fau 89b] montrent de plus qu'il existe jusqu'à 10 solutions à ce système. Longuet-Higgins [LH 81] et Tsai et Huang [Tsa 84] montrent qu'une solution linéaire existe si l'on se donne 8 points en correspondance. Des cas dégénérés où aucune solution n'existe apparaissent dans le cas de 4 points alignés, 7 points dans un plan, 6 sur un hexagone ou 8 sur les coins d'un cube.
- Trouver un algorithme efficace et stable permettant d'obtenir ces paramètres avec un maximum de fiabilité en utilisant un maximum de points de mesure. Ces approches [Fau 87a] [Har 87] [Jer 90], dérivées des précédentes en introduisant une technique de résolution au sens des moindres carrés, permettent une meilleure résistance au bruit sur les mesures.

Ces techniques, qui supposent la poursuite de points dans une séquence d'images, sont effectivement sensibles au bruit de digitalisation et aux erreurs de segmentation. Ainsi R. Cipolla [Cip 89] montre que les solutions trouvées par les algorithmes d'optimisation non linéaires utilisés [Fau 87a] [Har 87] peuvent être fausses ! Il convient donc de posséder une bonne approximation de la solution recherchée pour initialiser le processus d'optimisation.

Liu et Huang [Liu 86] et Faugeras, Lustman et Toscani [Fau 87a] formalisent le problème en appariant les segments de droite, diminuant ainsi

d'un ordre de grandeur ambiguïtés d'appariement et erreurs de localisation. Trois vues sont nécessaires pour la résolution.

Le problème continu de l'autocalibration du mouvement consiste à retrouver les paramètres du vissage instantané du mouvement. Cette approche s'impose lorsque les images sont prises depuis des points de vue très proches l'un de l'autre, comme dans le cas des séquences denses d'images qui nous intéressent.

Subbarao [Sub 86] et Waxman, Kamgar-parsi et Subbarao [Wax 87] présentent une approche du type flot optique, où le champ de vitesse normal est connu le long de contours arbitraires. Ils utilisent les différentielles du flot jusqu'au deuxième ordre. Ils proposent une solution exacte dans le cas de contours plans, et une solution au premier ordre pour les surfaces. Ils montrent également que plusieurs solutions existent, notamment trois quand la surface est une surface hyperbolique spéculaire de courbure moyenne -1 , deux lorsque la surface est hyperbolique ou parabolique de courbure moyenne -1 et que le mouvement est parallèle au plan image, et deux également lorsque la translation, la normale à la surface et la ligne de vue sont coplanaires.

Broida et Chellappa [Bro 86] supposent connu le suivi de points rigidement liés à la surface de l'objet, et proposent une résolution par filtrage récursif de Kalman pour retrouver les paramètres du vissage.

Faugeras, Deriche et Navab [Fau 89a] et T. Viéville et Faugeras [Vi 90] supposent connu le suivi de droites au cours de la séquence d'images.

L'approche qui nous concerne est celle du continu. Nous n'avons malheureusement à notre disposition ni suivi de points ni suivi de droites. Nous nous dirigeons donc vers une solution sans appariement de points, similaire dans l'esprit à la technique de reconstruction présentée ci-dessus, où seule la correspondance contour à contour est nécessaire, et où l'on utilise les propriétés différentielles des surfaces et des contours pour déterminer les paramètres du mouvement.

Dans cet esprit, O. Faugreras [Fau 90] pose, en pionnier, le problème général de la détermination du mouvement et de la déformation d'une courbe, quand la courbe conserve sa longueur (courbe parfaitement flexible mais parfaitement non extensible). Il en conclut que cette détermination est impossible, mais que par contre l'approche est soluble dans le cas de courbes rigides, où le mouvement se réduit à un mouvement de solide.

L'analyse qui suit se propose également de résoudre le problème du mouvement d'une courbe de discontinuité, avec une formalisation mathématiquement simple, et en suivant une approche "intuitivement" claire.

4.3 Principe de l'autocalibration

Cette approche consiste à calibrer le mouvement en utilisant une caméra, sans dispositif expérimental particulier. Il s'agit de déterminer les paramètres du mouvement exclusivement à partir d'informations visuelles, sans connaître quoi que ce soit sur la scène perçue. Ce problème dans sa généralité n'a pas de solution [Fau 90]. Il est par contre possible de le résoudre partiellement dans le cas où une information a priori est connue quant aux contours observés.

Ainsi, on va présenter une solution dans le cas de contours de discontinuité, quand les paramètres intrinsèques de la caméra sont connus. Cette résolution n'est possible qu'à un facteur d'échelle près bien entendu, car l'observation monoculaire est soumise à une ambiguïté échelle/vitesse : l'objet observé peut être loin et se déplacer vite, ou proche et se déplacer lentement. Cette ambiguïté n'affecte que la translation et la rotation peut être entièrement déterminée.

La seule contrainte imposée est que les contours observés soient des contours de discontinuité ou des marquages de surface, c'est-à-dire qu'ils soient rigides et fixes dans leur repère de référence $\{O\}$. Le mouvement de la caméra par rapport à l'objet sera déterminé relativement à ce repère de référence $\{O\}$.

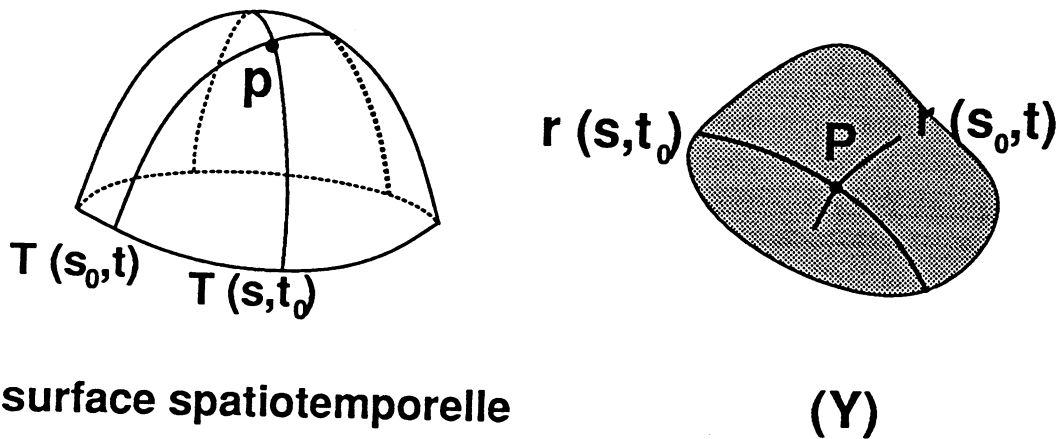


Figure 4.1. Courbe à $s=\text{constante}$ dans le cas d'un contour critique

Cas d'une courbe critique :

Lorsque la caméra se déplace par rapport à une surface (Y) , on observe un ensemble de contours qui forment la surface spatio-temporelle. Une courbe $T(s_0, t)$ à $s_0 = \text{constante}$ passant par un point $p = T(s_0, t_0)$ de cette surface spatio-temporelle correspond à une courbe $r(s_0, t)$ à $s_0 = \text{constante}$

passant par $P = r(s_0, t_0)$ sur la surface de l'objet, ainsi que l'illustre la figure 4.1.

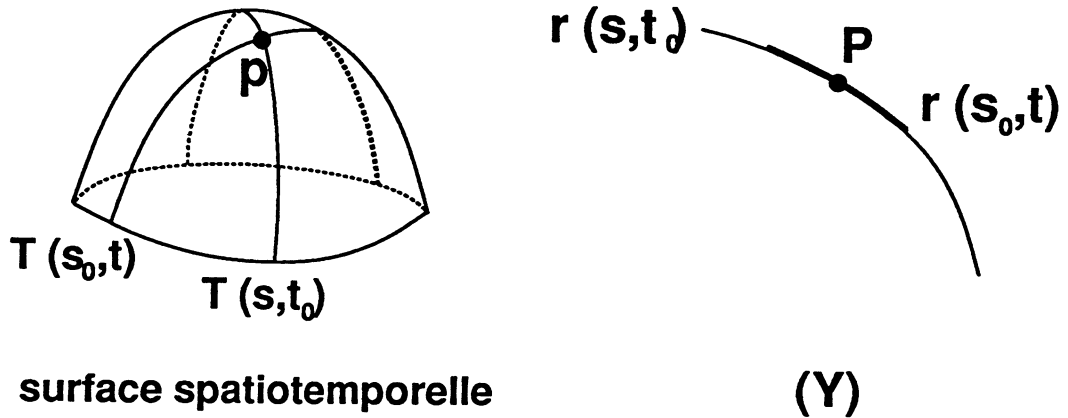


Figure 4.2. Courbe à $s=\text{constante}$ sur la courbe de discontinuité (Y)

Cas d'une courbe de discontinuité :

Dans le cas d'une courbe de discontinuité, la configuration générale est celle de la figure 4.2 où l'objet (Y), courbe de discontinuité, se réduit à l'arc correspondant à $r(s, t_0)$, et où la courbe $r(s_0, t)$ est un sous-ensemble de cet arc : en effet, pour une paramétrisation arbitraire de la surface spatio-temporelle, la courbe $T(s_0, t)$ correspond à un arc $r(s_0, t)$ nécessairement sur la courbe. On a donc une propriété fondamentale qui permet l'autocalibration :

Les deux courbes $r(s, t_0)$ et $r(s_0, t)$ passent toutes deux localement en P , et correspondent au même arc. Leurs différentielles en s et t au point $r(s_0, t_0)$ sont donc parallèles :

$${}^{\circ}r_s \wedge {}^{\circ}r_t = 0 \quad (4.1)$$

Cette contrainte, exprimée dans $\{C\}$, permet d'expliciter comme on va le voir plus loin une équation qui ne fait intervenir que les mesures et les paramètres du mouvement, permettant ainsi de résoudre le problème.

4.4 Analyse mathématique

Le problème de l'autocalibration est à cheval entre la cinématique du solide et la géométrie différentielle. Des rappels de cinématique du solide suivent, avant d'aborder le cœur du problème. En effet, la contrainte 4.1 est exprimée dans le repère $\{O\}$, et doit en fait être exprimée dans le repère où sont faites les mesures, c'est-à-dire dans $\{C\}$.

4.4.1 La cinématique du solide : quelques rappels

Dérivation dans un repère mobile

Les quelques notions de cinématique nécessaires concernent la dérivation des vecteurs dans des repères mobiles, et les notations utilisées dérivent de [Cra 86].

On suppose l'existence de deux repères mobiles l'un par rapport à l'autre : $\{O\}$, et $\{C\}$ mobile par rapport à $\{O\}$.

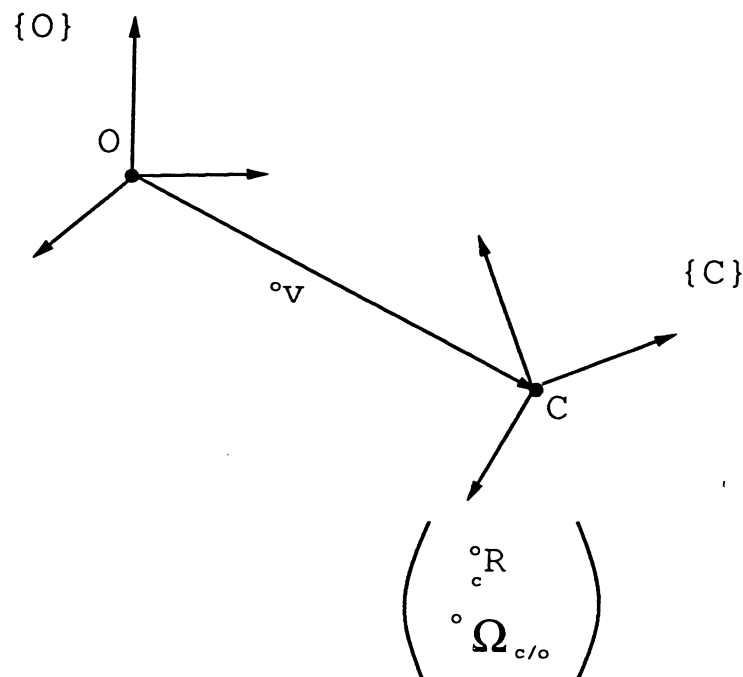


Figure 4.3. Les repères et leurs relations

La figure 4.3 montre que le mouvement de $\{C\}$ par rapport à $\{O\}$ est repéré par :

1. une translation ${}^o v$, vecteur coordonnées de l'origine C de $\{C\}$ exprimé dans $\{O\}$.
2. une vitesse de translation ${}^o \dot{v}$ et une accélération ${}^o \ddot{v}$, dérivées première et seconde de ${}^o v$ exprimées dans $\{O\}$.
3. une vitesse de rotation instantanée ${}^o \Omega_{c/o}$, vecteur vitesse de rotation instantanée du mouvement de vissage entre $\{C\}$ et $\{O\}$, exprimant la rotation de $\{C\}$ par rapport à $\{O\}$, exprimée dans $\{O\}$. Ce vecteur a pour support l'axe de la rotation, et pour module la vitesse de rotation.
4. une accélération de rotation instantanée ${}^o \dot{\Omega}_{c/o}$, dérivée du vecteur vitesse de rotation instantanée, exprimée dans $\{O\}$.

5. La matrice de rotation entre le repère $\{C\}$ et le repère $\{O\}$ est notée : cR , et permet de passer des coordonnées d'un vecteur exprimé dans $\{C\}$ à des coordonnées exprimées dans $\{O\}$.

Soit un vecteur U , fonction de t , mobile dans $\{C\}$. On note cU_t la dérivée de U dans le repère $\{C\}$, et oU_t la dérivée de U dans le repère $\{O\}$. On note de même ${}^c({}^oU_t)$ la dérivée de U dans le repère $\{O\}$, exprimée dans le repère $\{C\}$.

Les deux équations clé de la cinématique concernant la dérivation de U dans le repère $\{O\}$ au premier et deuxième ordre sont :

$$\begin{aligned} {}^oU_t &= {}^cR {}^cU_t + {}^o\Omega_{c/o} \wedge {}^oU \\ {}^oU_{tt} &= {}^cR {}^cU_{tt} + 2{}^o\Omega_{c/o} \wedge {}^cR {}^cU_t + {}^o\Omega_{c/o} \wedge ({}^o\Omega_{c/o} \wedge {}^oU) + {}^c\dot{\Omega}_{c/o} \wedge {}^oU \end{aligned}$$

Ces mêmes équations, exprimées dans le repère $\{C\}$, prennent la forme simplifiée suivante (on remarquera que la matrice de passage disparaît) :

$${}^c({}^oU_t) = {}^cU_t + {}^c\Omega_{c/o} \wedge {}^cU \quad (4.2)$$

$${}^c({}^oU_{tt}) = {}^cU_{tt} + 2{}^c\Omega_{c/o} \wedge {}^cU_t + {}^c\Omega_{c/o} \wedge ({}^c\Omega_{c/o} \wedge {}^cU) + {}^c\dot{\Omega}_{c/o} \wedge {}^cU \quad (4.3)$$

On pourra trouver une démonstration de ces résultats classiques dans l'annexe E.

Relations entre matrice rotation et vecteur rotation

Nous utiliserons également l'équivalence matrice rotation et vecteur rotation. Une rotation quelconque peut en effet être notée sous la forme d'un vecteur dont la direction est l'axe de rotation, et dont le module est l'angle de rotation.

Si K est le vecteur rotation de module θ et si $\hat{K} = (k_x, k_y, k_z)$ est le vecteur unitaire associé à K , alors la matrice rotation directe associée s'écrit :

$$Rot(\hat{K}, \theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix}$$

avec $c\theta = \cos \theta$, $s\theta = \sin \theta$ et $v\theta = 1 - \cos \theta$.

Soit une matrice rotation R quelconque d'angle différent de 0 et Π , le vecteur rotation \hat{K} et l'angle θ se déduisent de R par :

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

$$\theta = \arccos \left(\frac{r_{11} + r_{22} + r_{33} - 1}{2} \right)$$

$$\hat{K} = \frac{1}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

4.4.2 Notations du problème

Nous reprenons les notations introduites au paragraphe 3.4 sur la reconstruction (voir la figure 4.4).

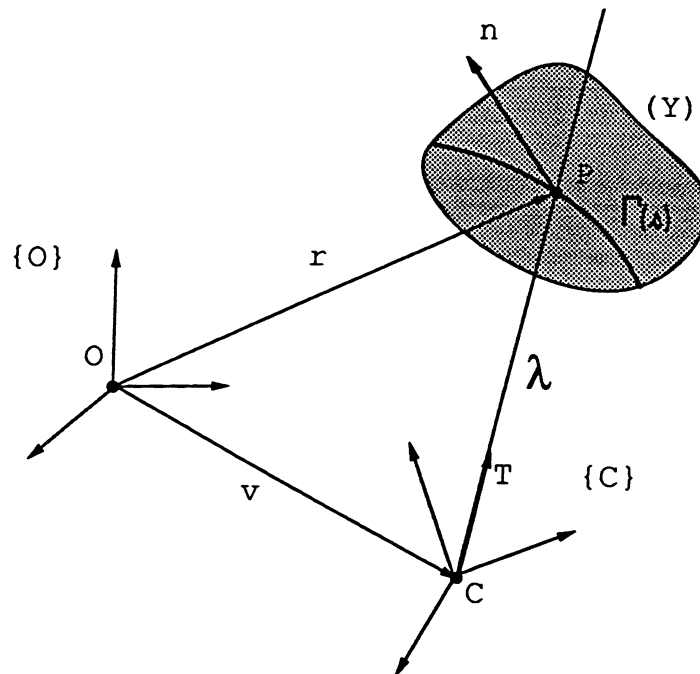


Figure 4.4. Les repères - les notations

4.4.3 Les équations de l'autocalibration

On rappelle l'équation clé de l'autocalibration 4.1 exprimant le fait que la courbe observée est fixe dans l'espace repéré par le repère $\{O\}$:

$${}^o r_s \wedge {}^o r_t = 0$$

ce qui permet de déduire deux équations scalaires pour chaque coordonnée du produit vectoriel, ou une équation scalaire unique concernant le module, et ce pour chaque point considéré.

Nous présentons ci-dessous le calcul du module :

$$|{}^o r_s \wedge {}^o r_t|^2 = ({}^o r_s \wedge {}^o r_t) \cdot ({}^o r_s \wedge {}^o r_t) \quad (4.4)$$

$$= ({}^o r_t \cdot {}^o r_t)({}^o r_s \cdot {}^o r_s) - ({}^o r_t \cdot {}^o r_s)^2 \quad (4.5)$$

$$= 0$$

Remarque : calcul de la courbure gaussienne.

On pourra trouver dans l'annexe D un rappel de géométrie différentielle

La courbure gaussienne, produit des deux courbures principales de la surface est un invariant indépendant de la paramétrisation de la surface. L'expression générale de la courbure gaussienne est de la forme :

$$\kappa_{gauss} = \frac{|II_P|}{|I_P|} \quad (4.6)$$

où $|I_P|$ et $|II_P|$ désignent les déterminants des matrices des endomorphismes associées aux deux formes fondamentales.

Le cas d'un cylindre généralisé dont le rayon tend vers 0 est tel que l'une de ses courbures principales reste inchangée, tandis que l'autre tend vers l'infini (car le rayon de courbure tend vers 0). La courbure gaussienne, produit des courbures principales, tend donc également vers l'infini. Cette propriété implique d'après 4.6 que le déterminant de la première forme fondamentale tend vers 0 (à condition que le déterminant de la deuxième forme fondamentale reste borné) :

$$|I| = 0 \iff ({}^o r_t \cdot {}^o r_t)({}^o r_s \cdot {}^o r_s) - ({}^o r_t \cdot {}^o r_s)^2 = 0$$

On retrouve l'équation 4.5, donnant une interprétation sous forme de courbure de la surface à la contrainte 4.5.

□

L'équation 4.5 est exprimée dans $\{O\}$, et on peut expliciter chaque terme dans $\{C\}$:

$$\begin{aligned} {}^o r_t &= {}^o \dot{v} + \lambda_t {}^o T + \lambda {}^o T_t \\ {}^o r_s &= \lambda_s {}^o T + \lambda {}^o T_s \end{aligned}$$

L'équation 4.5 exprimée dans le repère $\{C\}$ permet d'expliciter les paramètres du mouvement.

Détaillons les différents termes dont nous aurons besoin :

${}^c({}^o T)$, ${}^c({}^o T_s)$, ${}^c({}^o T_{ss})$, ${}^c({}^o n)$ et ${}^c({}^o n_s)$: par définition, on a :

$$\begin{aligned} {}^c({}^o T) &= {}^c T \\ {}^c({}^o T_s) &= {}^c T_s \\ {}^c({}^o T_{ss}) &= {}^c T_{ss} \\ {}^c({}^o n) &= {}^c({}^o T) \wedge {}^c({}^o T_s) \\ {}^c({}^o n_s) &= {}^c({}^o T) \wedge {}^c({}^o T_{ss}) \end{aligned}$$

${}^c({}^o T_t)$ et ${}^c({}^o T_{st})$: d'après la propriété 4.2 :

$$\begin{aligned} {}^c({}^o T_t) &= {}^c T_t + {}^c \Omega_{c/o} \wedge {}^c T \\ {}^c({}^o T_{st}) &= {}^c T_{st} + {}^c \Omega_{c/o} \wedge {}^c T_s \end{aligned}$$

${}^c({}^o n_t)$:

$${}^c({}^o n_t) = {}^c({}^o T_t) \wedge {}^c({}^o T_s) + {}^c({}^o T) \wedge {}^c({}^o T_{st})$$

et l'on utilise les formules ci-dessus pour exprimer chaque terme dans $\{C\}$.

${}^c({}^o T_{tt})$: D'après la propriété 4.3 :

$${}^c({}^o T_{tt}) = {}^c T_{tt} + {}^c \dot{\Omega}_{c/o} \wedge {}^c T + 2 {}^c \Omega_{c/o} \wedge {}^c T_t + {}^c \Omega_{c/o} \wedge ({}^c \Omega_{c/o} \wedge {}^c T)$$

λ_s et λ_t :

$$\begin{aligned} \lambda_s &= \frac{\partial \lambda}{\partial s} \\ &= - \frac{({}^o T_t \cdot {}^o n) \cdot ({}^o \dot{v} \cdot {}^o n_s) - ({}^o \dot{v} \cdot {}^o n) \cdot ({}^o T_{ts} \cdot {}^o n + {}^o T_t \cdot {}^o n_s)}{({}^o T_t \cdot {}^o n)^2} \\ \lambda_t &= \frac{\partial \lambda}{\partial t} \\ &= - \frac{({}^o T_t \cdot {}^o n) \cdot ({}^o \dot{v} \cdot {}^o n_t + {}^o \ddot{v} \cdot {}^o n) - ({}^o \dot{v} \cdot {}^o n) \cdot ({}^o T_{tt} \cdot {}^o n + {}^o T_t \cdot {}^o n_t)}{({}^o T_t \cdot {}^o n)^2} \end{aligned}$$

et l'on utilise de même les formules ci-dessus pour exprimer chaque vecteur dans $\{C\}$.

${}^c({}^o\dot{v})$ et ${}^c({}^o\ddot{v})$: Ces deux termes sont des inconnues de notre problème, et nous les garderons sous la forme ${}^c({}^o\dot{v})$ et ${}^c({}^o\ddot{v})$.

Le cas de nullité des dénominateurs ${}^oT_t \cdot {}^on = 0$ correspond comme dans le cas de la reconstruction à d'un déplacement dans le plan tangent à la surface. On ne peut tirer aucune information sur le mouvement dans ce cas.

L'équation 4.5 pourrait donc être réécrite dans le repère $\{C\}$ en substituant les termes ci-dessus dans 4.5.

Remarque : L'équation 4.5 est l'expression du module de $r_s \wedge r_t$; sa valeur est donc positive ou nulle. Le cas de nullité qui nous intéresse intervient lorsque ces vecteurs sont colinéaires. Le cas où r_t est nul correspond au cas de la géométrie épipolaire, et est contenu dans le cas général de l'équation 4.5.

4.4.4 La résolution

L'équation 4.5 est une équation différentielle dont les inconnues, correspondent aux quatre vecteurs ${}^c({}^o\dot{v})$, ${}^c({}^o\ddot{v})$, ${}^c\Omega_{c/o}$ et ${}^c\dot{\Omega}_{c/o}$.

La forme de cette équation est une fonction en ses douze inconnues $P = P(y, x_1, \dots, x_{12}) = 0$ (avec y le point considéré de la surface spatio-temporelle), et va être résolue numériquement au sens des moindres carrés : en effet, il est possible d'exprimer cette contrainte en chaque point de la surface spatio-temporelle y_i et d'obtenir n équations pour les n points y_i de la surface ($P_i = P(y_i, x_1, \dots, x_{12})$). Cette fonction étant par construction toujours positive, il suffit de minimiser par rapport aux inconnues la somme de ces équations $\sum_i P_i$.

L'ambiguïté échelle/vitesse fait que ces inconnues ne peuvent être toutes déterminées sans autre information, et l'on peut choisir a priori une contrainte supplémentaire, pour permettre la résolution, comme de fixer le module de la vitesse initiale du repère $\{C\}$ à une valeur arbitraire \dot{v}_0 (estimation connue par ailleurs, ou valeur parfaitement arbitraire).

Si l'on veut résoudre le problème globalement (par une technique d'éléments finis par exemple) pour tous les pas de temps, alors l'approche présentée schématiquement ci-dessus est fautive car elle considère les douze inconnues comme indépendantes, or des relations différentielles les relient puisque ${}^c({}^o\ddot{v})$ est la dérivée de ${}^c({}^o\dot{v})$, et que ${}^c\dot{\Omega}_{c/o}$ est la dérivée de ${}^c\Omega_{c/o}$. La solution que l'on obtient ne prend pas en compte ces contraintes, et n'est pas correcte.

Si l'on résoud les équations localement en un seul pas de temps, l'hypothèse peut cependant être admise, mais les résultats expérimentaux que nous avons obtenus montrent une grande instabilité.

On peut se ramener classiquement à une expression correcte et stable en faisant une résolution par différences finies : à chaque pas de temps i on peut écrire, par approximation au premier ordre :

$$\begin{aligned} {}^c({}^o\ddot{v})_i &= {}^c({}^o\dot{v})_i - {}^c({}^o\dot{v})_{i-1} \\ ({}^c\dot{\Omega}_{c/o})_i &= ({}^c\Omega_{c/o})_i - ({}^c\Omega_{c/o})_{i-1} \end{aligned}$$

La forme de P_i devient $P_i = P(y_i, x_1, \dots, x_6)$, fonction en 6 inconnues, et l'on minimise toujours $\sum_i P_i$. Le problème est maintenant formulé de manière satisfaisante. La résolution demande de connaître une valeur initiale de $({}^c\Omega_{c/o})_0$ et de ${}^c({}^o\dot{v})_0$. Le choix de la valeur initiale de la vitesse de translation est peut se faire à l'échelle prêt, mais la direction du vecteur doit être bonne. Le choix de la vitesse de rotation initiale est également sensible.

En fait, P est un quotient de polynômes, et l'on peut ramener la contrainte à une contrainte polynomiale du $10^{\text{ème}}$ degré en ses 6 inconnues, somme des carrés de trois polynômes du $5^{\text{ème}}$ degré (4.5 est en effet l'expression du carré du module de $r_s \wedge r_t$). Il suffit pour cela d'annuler le numérateur du quotient de polynôme ; ce faisant, on peut introduire des solutions supplémentaires, d'où la nécessité de connaître une approximation de la solution pour converger vers la solution réelle.

Ce résultat pour l'autocalibration est à comparer avec celui de O. Faugeras [Fau 90] où les contraintes obtenues sont deux polynômes du $9^{\text{ème}}$ degré soit une équation scalaire unique du $18^{\text{ème}}$ degré en prenant la somme des carrés. Le degré inférieur de la contrainte obtenue garantit a priori une meilleure stabilité de la solution car le nombre de minimum locaux existant est moindre, et le choix de la solution initiale nécessaire à la résolution est moins crucial. Des résultats récents de O. Faugeras [Fau 91] montrent un résultat équivalent au nôtre, c'est-à-dire une contrainte polynomiale du $5^{\text{ème}}$ degré.

Le problème adressé dans [Fau 90] est relatif à la notion de flot optique, et à son utilisation. O. Faugeras s'attache à démontrer que le flot optique ne permet pas de déterminer les paramètres du mouvement dans le cas d'une courbe infiniment flexible mais inextensible, mais par contre que la même approche permet la résolution dans le cas plus simple d'une courbe rigide.

4.4.5 L'algorithme de résolution

La résolution se fait par pas de temps, et à chaque pas, la résolution se fait en plusieurs étapes :

Obtention des mesures : Pour chaque point de la surface spatio-temporelle sélectionné, calculer cT , cT_s , cT_t , ${}^cT_{ss}$, ${}^cT_{tt}$, ${}^cT_{st}$.

Calcul de P : Pour chaque point, calcul de r_t et r_s , permettant le calcul du terme P_i et calcul de la fonctionnelle à minimiser $\sum_i P_i$.

Optimisation : Optimisation sur l'ensemble des mesures par une technique d'optimisation non linéaire itérative du type Powell (sans dérivées partielles) [Pre 88]. On obtient donc, à chaque pas de temps i , $({}^c\Omega_{c/o})_i$ et ${}^c({}^o\dot{v})_i$.

Calcul de la matrice de passage oR : La matrice de passage entre $\{C\}$ et $\{O\}$ oR_i s'obtient au pas de temps i en utilisant la matrice au temps $i-1$ ${}^oR_{i-1}$ et le vecteur rotation instantané au pas de temps $i-1$ $({}^o\Omega_{c/o})_{i-1}$.

Tous les variables au temps $i-1$ sont connues, donc ${}^o\Omega_{c/o}$ s'obtient par :

$$({}^o\Omega_{c/o})_{i-1} = {}^oR_{i-1} ({}^c\Omega_{c/o})_{i-1}$$

On note :

$${}^oR_k = ({}^oe_{1,k}, {}^oe_{2,k}, {}^oe_{3,k})$$

d'où :

$${}^oe_{j,i} = {}^oe_{j,i-1} + ({}^o\Omega_{c/o})_{i-1} \wedge {}^oe_{j,i-1}, \quad 1 \leq j \leq 3$$

En pratique, compte tenu des pas de temps non infinitésimaux considérés, cette formule ne permet pas de conserver des vecteurs ${}^oe_{j,i}$ normés ; on préférera utiliser la formule de Rodrigues pour calculer l'image des vecteurs ${}^oe_{j,i-1}$ par une rotation d'axe l'axe de ${}^o\Omega_{c/o}$, et d'angle le module de ${}^o\Omega_{c/o}$ (pour un pas de temps de 1, l'angle de rotation effectué, si la rotation instantanée est constante, est égal au module de ${}^o\Omega_{c/o}$ fois 1).

avec ${}^oR_0 = I$: la position d'origine de la caméra définit le repère de référence $\{O\}$.

Calcul de $({}^o\Omega_{c/o})_i$: $({}^c\Omega_{c/o})_i$ est obtenu par l'optimisation. oR_i étant connu, $({}^o\Omega_{c/o})_i$ s'obtient très simplement par :

$$({}^o\Omega_{c/o})_i = {}^oR_i ({}^c\Omega_{c/o})_i$$

Calcul de $({}^o\dot{v})_i$: ${}^c({}^o\dot{v})_i$ est obtenu par l'optimisation. oR_i étant connu, $({}^o\dot{v})_i$ s'obtient très simplement par :

$$({}^o\dot{v})_i = {}^oR_i ({}^c({}^o\dot{v})_i)$$

Initialisation L'initialisation peut se faire en prenant les valeurs estimées a priori au temps 0, ce qui fixe l'échelle pour la détermination de la translation.

On peut également choisir de fixer la vitesse de translation à une valeur arbitraire, mais les valeurs de la vitesse de translation calculées ultérieurement sont alors à une échelle sans rapport avec la réalité.

4.4.6 Retour sur la reconstruction

On rappelle que la formule de reconstruction 3.8 présentée au paragraphe 3.5 était :

$$\lambda = \frac{-{}^o\dot{v} \cdot {}^o n}{{}^o T_t \cdot {}^o n}$$

avec :

$${}^o r = {}^o v + \lambda {}^o T$$

Lorsque le mouvement de la caméra est obtenu comme ci-dessus par autocalibration, les formules de reconstruction sont les suivantes :

Calcul de λ : On réexprime l'équation dans $\{C\}$:

$$\lambda = - \frac{{}^c({}^o\dot{v}_i) \cdot ({}^c T \wedge {}^c T_s)}{({}^c T_t + {}^c \Omega_{c/o_i} \wedge {}^c T) \cdot ({}^c T \wedge {}^c T_s)}$$

Calcul de ${}^o v$: ce terme se calcule par intégration :

$${}^o v_i = \int_0^i {}^o R {}^c ({}^o \dot{v}) dt = {}^o v_{i-1} + {}^o R_i {}^c ({}^o \dot{v})_i$$

4.5 Résultats expérimentaux

Une première partie de l'expérimentation est effectuée en données synthétiques, de façon à valider l'approche théorique et l'implémentation. La deuxième partie traite les données réelles, et l'on pourra analyser la sensibilité du processus aux erreurs.

L'unité de temps est inconnue, et les vitesses sont donc exprimées "par unité de temps". Nous utiliserons cependant parfois par commodité la seconde comme unité de temps arbitraire.

4.5.1 Données synthétiques

Le contour test utilisé pour valider l'algorithme est un arc d'ellipse en mouvement par rapport à la caméra. Un mouvement de translation pure sera d'abord étudié, puis la rotation sera ajoutée. L'ellipse d'axes de longueurs respectives 2 mètres et 2,6 mètres est au départ à une distance de 8 mètres par rapport à la caméra. La figure 4.5 illustre la situation de l'ellipse par rapport au repère de référence de la scène.

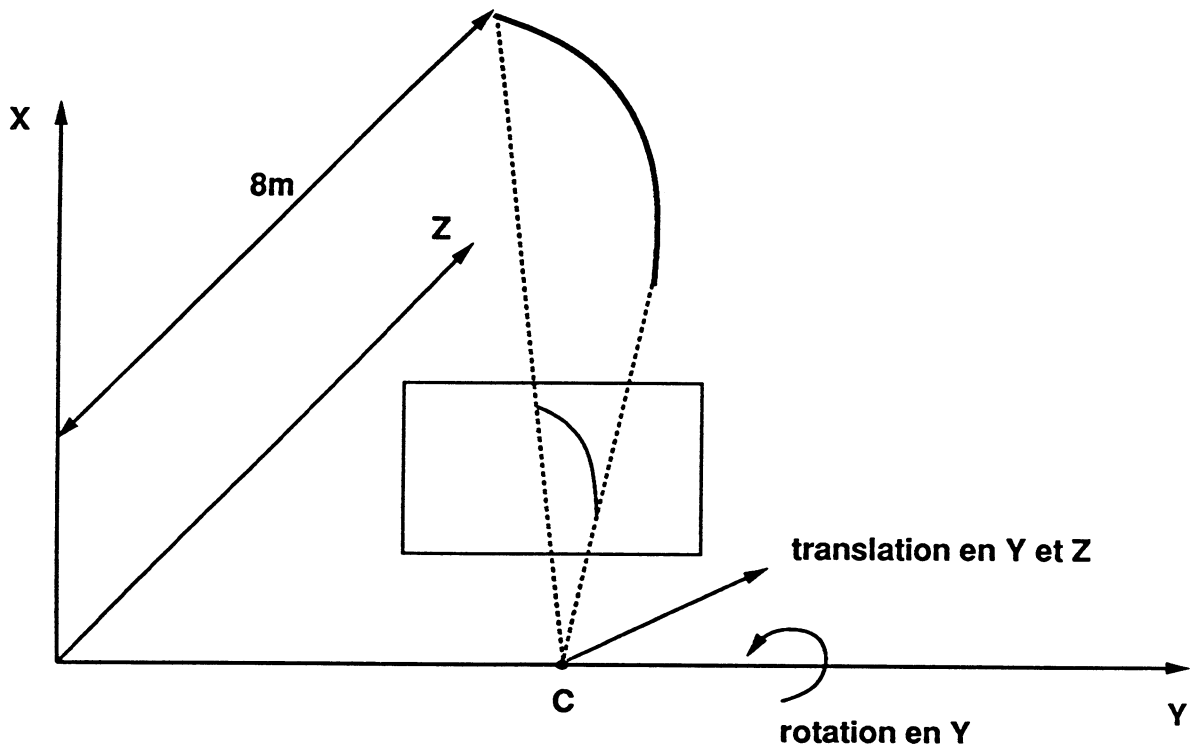


Figure 4.5. Situation de l'ellipse par rapport à la caméra

Mouvement de translation pure

Le mouvement est une translation uniforme, égale en Y et Z , où Z est la direction de l'axe optique de la caméra et Y la direction horizontale dans le plan image (la rotation est nulle dans ce cas). On va étudier la sensibilité des paramètres calculés de translation vis-à-vis de la vitesse de translation. Une série de séquence de contours est engendrée à vitesse croissante, et la vitesse est calculée pour chaque séquence.

La figure 4.6 montre la séquence de 4 images pour une vitesse de 0.4 mètres par unité de temps.

La résolution par différences finies impose de connaître des valeurs initiales au temps $t = 0$. Les estimations a priori du mouvement à $t = 0$ sont

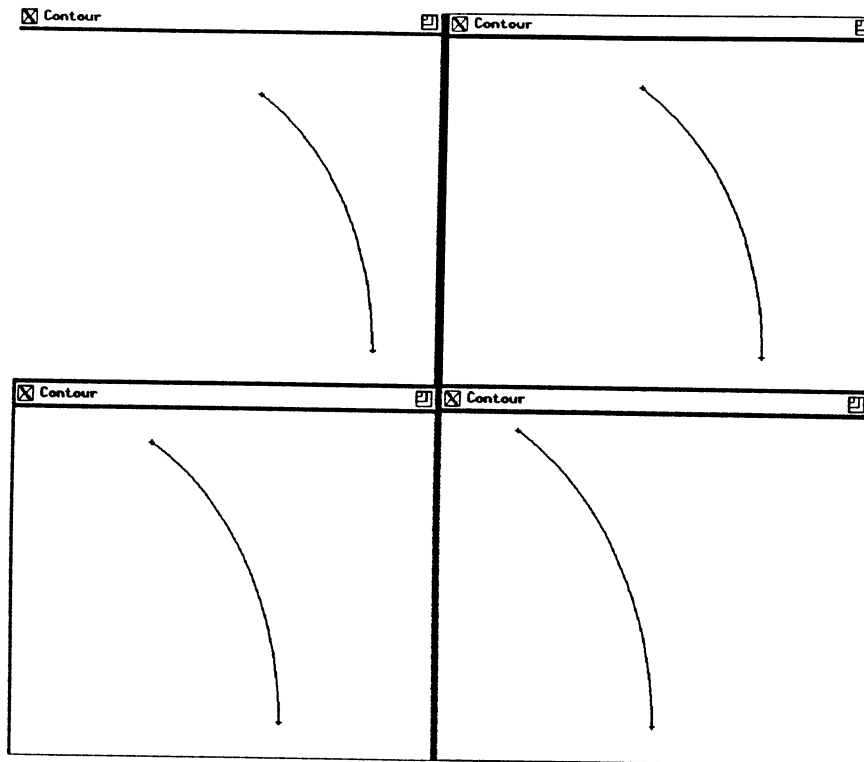


Figure 4.6. Séquence de l'arc d'ellipse pour $v = 0.4m/s$

utilisées pour l'initialisation à $t = 0$. On estime alors vitesse de translation et vitesse de rotation instantanée pour la deuxième position à $t = 1$.

La table 4.1 résume les résultats. Les unités sont le mètre par unité de temps et le radian par unité de temps. La correspondance utilisée est la correspondance épipolaire, en utilisant l'estimation a priori du mouvement.

Vitesse théorique		Vitesse de translation calculée			Vitesse de rotation calculée		
selon y	selon z	selon x	selon y	selon z	selon x	selon y	selon z
0.01	0.01	-4.38e-07	0.01	0.01	8.227e-11	3.635e-08	-2.966e-07
0.05	0.05	-2.09e-06	0.0500	0.0500	-1.202e-07	7.490e-07	8.719e-11
0.2	0.2	-0.00058	0.2003	0.2007	-2.377e-05	5.265e-05	0.00015
0.4	0.4	-0.0019	0.4023	0.409	-0.00026	0.00034	-0.00017
0.8	0.8	-0.011	0.812	0.902	-0.000369	0.00035	0.00018

Tableau 4.1. Valeurs de la vitesse calculée

Pour une vitesse de 0.4 mètre par unité de temps, le déplacement du contour sur le plan image est de l'ordre de 100 pixels, c'est-à-dire que l'on s'éloigne considérablement des hypothèses de mouvement infinitésimal.

Pour les mêmes valeurs de mouvement, mais pour une correspondance totalement arbitraire, en l'occurrence selon une direction orthogonale au contour, on obtient les résultats de la table 4.2. Les résultats sont moins bons que les précédents et prouvent la nécessité d'utiliser une correspondance la moins arbitraire possible pour améliorer la fiabilité des résultats. Si le mouvement de la caméra est approximativement connu, il est donc bon de l'utiliser. Si le mouvement du contour dans le plan image peut être estimé, s'en servir pour établir la correspondance.

Vitesse théorique		Vitesse de translation calculée			Vitesse de rotation calculée		
selon y	selon z	selon x	selon y	selon z	selon x	selon y	selon z
0.01	0.01	1.928e-06	0.0100	0.0100	-2.51e-08	4.7e-07	6.386e-10
0.05	0.05	0.0002	0.0501	0.0507	-3.426e-06	-2.101e-05	-9.64e-05
0.2	0.2	0.00117	0.202	0.222	-0.00024	-5.45e-05	-0.002
0.4	0.4	0.021	0.4037	0.479	-0.0011	-0.0015	-0.0084
0.8	0.8	0.1282	0.844	0.916	0.0081	4.33e-05	-0.0327

Tableau 4.2. Valeurs pour une correspondance orthogonale

Mouvement quelconque

Le mouvement est maintenant compliqué en ajoutant une rotation à la translation. La rotation est suivant l'axe Y (l'axe horizontal du plan image pour la position de départ à $t = 0$). L'ellipse d'axes de longueurs respectives 2 mètres et 2,6 mètres est toujours au départ à une distance de 8 mètres par rapport à la caméra.

On va étudier la sensibilité des paramètres calculés de rotation vis-à-vis de la vitesse de rotation. Une série de séquences de contours est engendrée à vitesse de rotation croissante : pour chaque vitesse de rotation, une séquence est engendrée et la vitesse de rotation est calculée. La vitesse de translation en Y et Z est choisie fixe et égale à 0.2 m/s .

La figure 4.7 montre la séquence de 4 images pour une vitesse de rotation de 0.05 radians par unité de temps.

Les estimations a priori du mouvement à $t = 0$ sont utilisées pour l'initialisation à $t = 0$. On estime alors vitesse de translation et vitesse de rotation instantanée pour la deuxième position à $t = 1$. La table 4.3 résume les résultats. Les unités sont le mètre par unité de temps et le radian par unité de temps. La correspondance utilisée est la correspondance épipolaire, en utilisant l'estimation a priori du mouvement.

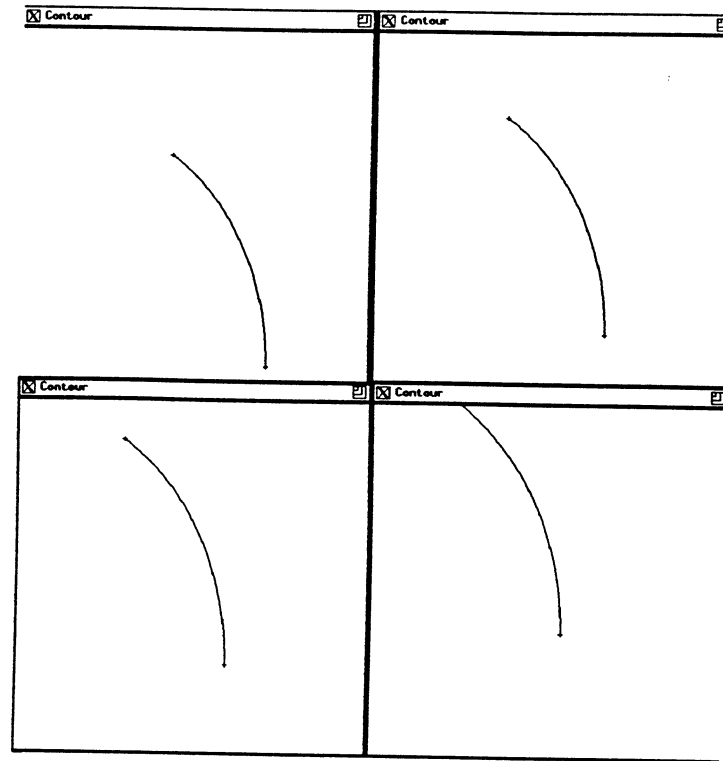


Figure 4.7. Séquence de l'arc d'ellipse pour $v = 0.2 \text{ m/s}$ et $\Omega = 0.05 \text{ rad/s}$

Pour les mêmes valeurs de mouvement, mais pour une correspondance selon une direction orthogonale au contour, on obtient les résultats de la table 4.4 :

On s'aperçoit qu'en l'absence totale d'information sur le mouvement dans le plan image, les résultats sont assez approximatifs, et dégèrent rapidement. En particulier, les vitesses supérieures à 0.2 rad/s ne sont pas montrées dans la table car les valeurs trouvées sont incohérentes.

Vitesse théorique de rotation	Vitesse de translation calculée			Vitesse de rotation calculée		
	en x	en y	en z	en x	en y	en z
0.01	-0.0002	0.2001	0.2007	-1.953e-06	0.0100	0.0001
0.05	-0.0004	0.2004	0.197	-8.180e-06	0.05010	1.591e-05
0.1	-0.004	0.2004	0.205	-3.39e-05	0.100	-2.33e-05
0.2	-0.023	0.2029	0.215	-0.0001	0.20004	-7.871e-05
0.4	-0.00156	0.1893	0.202	0.00045	0.368	-0.00193

Tableau 4.3. Valeurs de la vitesse de rotation calculée

Vitesse théorique de rotation	Vitesse de translation calculée			Vitesse de rotation calculée		
	en x	en y	en z	en x	en y	en z
0.01	0.0212	0.202	0.2403	6.434e-05	0.00925	-0.00574
0.05	0.0243	0.205	0.241	-5.738e-05	0.0499	2.68e-06
0.1	-0.2176	-0.133	3.934	-0.0325	0.1202	-0.00442

Tableau 4.4. Valeurs de la vitesse de rotation pour une correspondance orthogonale

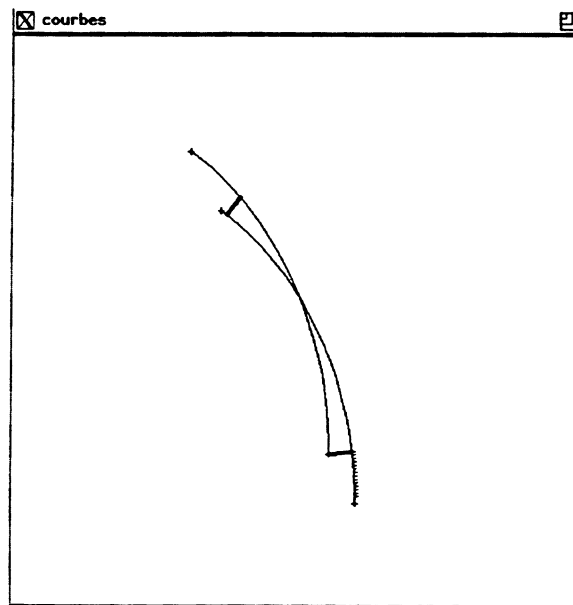


Figure 4.8. Recouvrement pour $v = 0.2 \text{ m/s}$ et $\Omega = 0.05 \text{ rad/s}$

La figure 4.8 montre le recouvrement utilisé pour $\Omega = 0.05 \text{ rad/s}$ et montre bien la raison de telles erreurs¹ : le recouvrement utilisé est beaucoup trop loin de la réalité, et trop loin des hypothèses différentielles, donc l'interprétation trouvée consiste en des translations supplémentaires pour compenser les déplacements apparents du contour induits par la correspondance arbitraire. Cette remarque prouve l'importance d'une correspondance fine entre les contours pour la qualité de l'autocalibration : en particulier, la correspondance dans le plan image doit tenir compte de la forme du contour. Il est clair en effet que les contours à courbure constante posent des ambiguïtés au niveau de la correspondance qui empêchent ce processus

1. Deux contours successifs sont superposés sur une même image, et les deux traits en gras montrent les correspondances entre les débuts et fins des contours ; les correspondances intermédiaires sont obtenues par interpolation linéaire sur l'abscisse curviligne

d'autocalibration de fournir des résultats cohérents.

On peut enfin étudier la sensibilité au bruit sur les contours de la séquence ; on reprend le cas d'une vitesse de translation constante en Y et Z de 0.2 m/s et d'une vitesse de rotation constante de $\Omega = 0.05 \text{ rad/s}$ autour de Y , mais en ajoutant un bruit gaussien de moyenne nulle sur la position des pixels du contour. La table 4.5 montre l'influence des valeurs de translation et rotation calculées en fonction de la variance du bruit. L'unité du bruit est le pixel.

Variance en x et y	Vitesse de translation calculée			Vitesse de rotation calculée		
	en x	en y	en z	en x	en y	en z
0	-0.00147	0.20016	0.198	-5.699e-06	0.05006	1.19e-05
0.5	-0.00145	0.2002	0.1979	-2.416e-06	0.05007	-3.767e-07
1	-0.00133	0.20008	0.198	-2.216e-05	0.05006	-1.365e-05
2	-0.00136	0.20012	0.1979	-2.025e-05	0.05006	-2.829e-06
4	-0.00113	0.2003	0.198	8.127e-06	0.05002	-1.462e-06
8	-0.00151	0.199	0.198	-3.985e-05	0.05005	2.249e-05
16	-0.00115	0.2003	0.198	-4.282e-07	0.05003	-1.091e-06

Tableau 4.5. Influence du bruit sur l'autocalibration

On s'aperçoit du peu d'influence du bruit gaussien de moyenne nulle sur les paramètres du mouvement.

Enfin étudions la stabilité de la résolution par différences finies. Cette technique algorithmique de coût faible se paie en effet par une précision réduite, et surtout par une accumulation d'erreurs. Une solution précise pour ce type de problème nécessite en effet l'emploi de techniques "lourdes" du type éléments finis.

La validité expérimentale de l'approche par différences finies va consister à étudier la stabilité du système au cours du temps, pour des valeurs croissantes du mouvement. On se place dans le cas générique précédent avec une vitesse de translation constante en Y et Z et une vitesse de rotation constante autour de Y . On choisira les modules des vitesses de translation en Y et Z et de rotation autour de Y égaux. Pour chaque valeur d'un module de vitesse, on engendre une séquence de 10 images, et l'on calcule le mouvement pour la 10^{ième} position. La table 4.6 résume les valeurs du mouvement pour différentes vitesses.

On remarque une dégradation rapide de l'évaluation de la vitesse de translation en z , composante toujours la plus sensible car elle n'engendre que de très petits déplacements dans le plan image. La dégradation globale progressive était prévisible puisque la matrice rotation, orientation de la

Vitesse théorique	Vitesse de translation calculée			Vitesse de rotation calculée		
	en x et y	en x	en y	en z	en x	en y
0.01	6.323e-05	0.0100	0.00998	4.271e-0	0.0100	-5.645e-06
0.02	0.000505	0.0199	0.0197	-9.639e-06	0.0200	-8.158e-05
0.04	0.00592	0.03916	0.036	-5.638e-05	0.0398	-0.00061
0.06	0.0116	0.05048	0.0423	0.000154	0.0603	-0.00285
0.08	0.0294	0.0701	0.0353	-0.000717	0.0768	-0.00289

Tableau 4.6. Stabilité des paramètres du mouvement après 10 images

caméra par rapport à un repère fixe, est obtenue de manière incrémentale, et que les erreurs s'accumulent. Les valeurs calculées plus localement (sur 4 à 5 images) donnent cependant des valeurs correctes pour les paramètres du mouvement.

4.5.2 Données réelles

Séquence de la poire

La figure 4.9 présente les images entières de la séquence de la poire. Les contours en gras sont les contours utilisés pour estimer les paramètres du mouvement.

image	vitesse	Vitesse de translation			Vitesse de rotation		
		en x	en y	en z	en x	en y	en z
1	a priori	-0.0129	-0.0266	-0.00038	-0.04506	0.02007	0.02044
	calculée	-0.0133	-0.0267	0.00199	-0.0444	0.0231	0.0193
2	a priori	-0.01112	-0.02727	0.000487	-0.04649	0.01687	0.020327
	calculée	-0.0115	-0.0271	0.0054	-0.0437	0.0234	0.0197

Tableau 4.7. Valeurs calculées par rapport aux valeurs connues

La table 4.7 présente les résultats de l'autocalibration sur cette séquence. On y trouve les résultats a priori, valeurs données par les capteurs du robot sur le déplacement de la caméra, et les résultats calculés, valeurs issues de l'autocalibration.

Séquence du masque

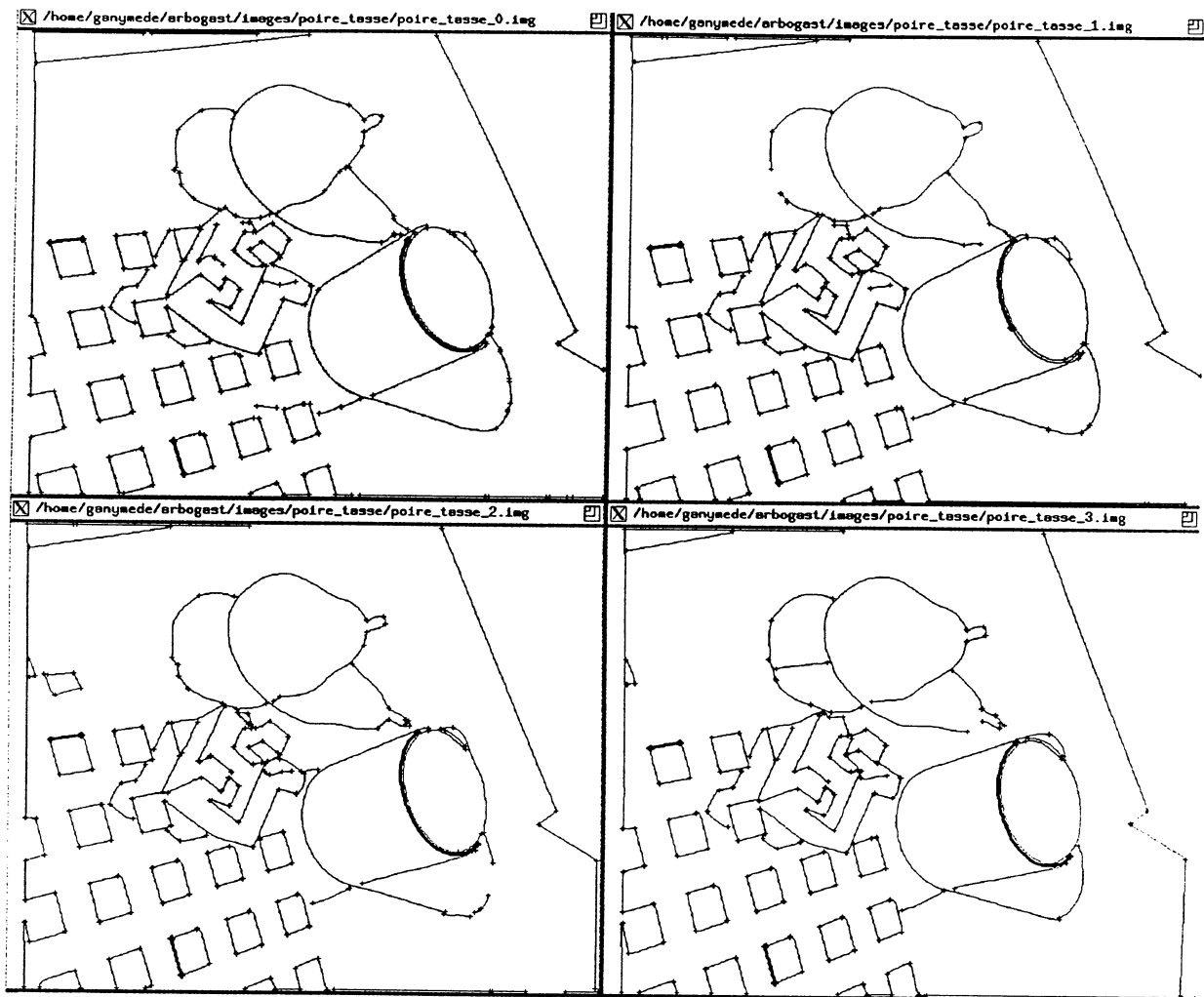


Figure 4.9. Séquence de la poire

La figure 4.10 présente les images de la séquence du masque. Le contour en gras est le contour utilisé pour estimer les paramètres du mouvement.

La table 4.8 présente les résultats de l'autocalibration sur cette séquence. On y trouve les résultats a priori, valeurs données par les capteurs du robot sur le déplacement de la caméra, et les résultats calculés, valeurs issues de l'autocalibration.

4.6 Conclusions

Nous avons présenté une technique de détermination des paramètres du mouvement de la caméra par rapport à l'objet, valable lorsqu'aucune correspondance de points ni aucune correspondance de droites n'est disponible. Cela étend le cadre des techniques d'autocalibration avec l'utilisation d'une

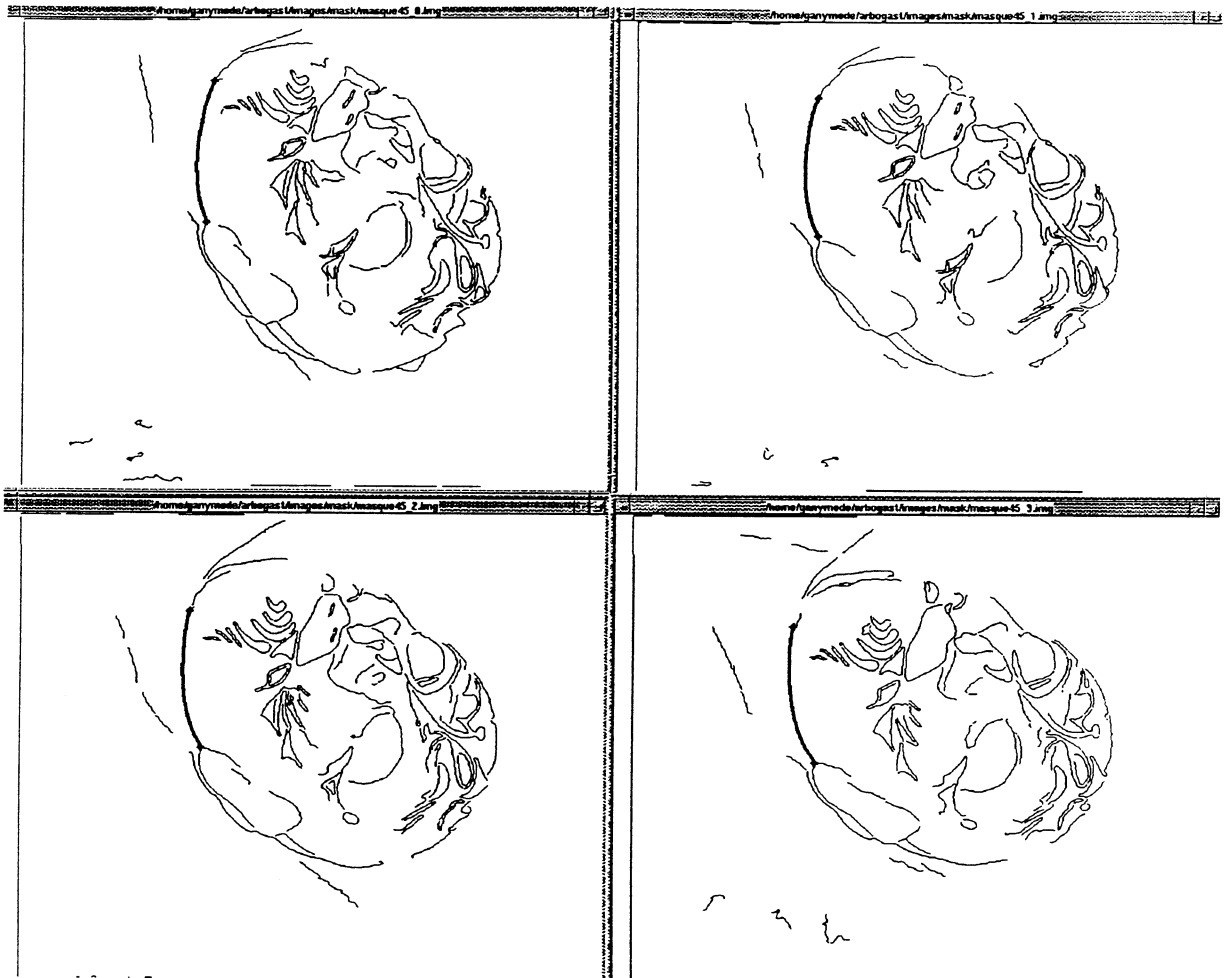


Figure 4.10. Séquence du masque

primitive très générale, applicable dans le cas de scènes non polyédriques.

La détermination est faite à l'aide d'une technique simple de différences finies et fournit rapidement de bonnes estimations des paramètres du mouvement de la caméra.

Cette technique est rendue robuste par l'emploi d'une résolution aux moindres carrés, peu sensible aux erreurs de mesure.

Les expérimentations en données synthétiques et données réelles montrent la validité de l'approche. Les différences finies ne permettent cependant pas une résolution sur une longue séquence d'images, car les erreurs s'accumulent et les résultats tendent à diverger.

On peut noter que tous les contours de discontinuité de l'objet peuvent être pris en compte par cette technique d'autocalibration, en construisant autant de surfaces spatio-temporelles que de contours. La robustesse du résultat peut être ainsi améliorée.

Si plusieurs objets rigides se déplacent indépendamment dans la scène, il

		Vitesse de translation			Vitesse de rotation		
image	vitesse	en x	en y	en z	en x	en y	en z
1	a priori	-0.0237	-0.0283	-0.00037	2.328e-05	1.683e-05	0.0873
	calculée	-0.0237	-0.0287	-0.00453	-0.000119	0.0001329	0.0870
2	a priori	-0.0212	-0.0301	0.000488	-3.785e-5	-4.787e-5	0.0872
	calculée	-0.0198	-0.03164	-0.0103	-0.000891	-0.000995	0.087

Tableau 4.8. Valeurs calculées par rapport aux valeurs connues

est important de calculer chaque mouvement séparément. Il faut pour cela segmenter l'image en régions où le mouvement tridimensionnel associé est homogène. Ce problème est difficile mais des éléments de solution peuvent être trouvés dans [Fra 90].

Il serait intéressant de se pencher sur des techniques plus sophistiquées de résolution du type éléments finis, tout en sachant que le coût en terme de temps de traitement est important.

Chapitre 5

Conclusions

Ce travail s'est articulé autour du thème de la reconstruction de surface en vision par ordinateur, et a apporté des contributions dans des domaines variés.

Nous avons tout d'abord proposé une réponse au problème de la représentation des courbes avec les B-splines régularisées. Cette représentation permet tout à la fois :

- une grande fidélité par rapport aux contours bruts issus du traitement d'images.
- une bonne résistance au bruit et un contrôle vis-à-vis du bruit grâce à l'influence du terme de régularisation.
- une interpolation entre pixels grâce à la forme continue de la représentation.
- Une expression stable des dérivées le long du signal grâce à un traitement symbolique.

Cette approche est de plus efficace en temps de calcul.

Le problème de la reconstruction des courbes et des surfaces est traité dans le cadre d'une observation monoculaire avec une caméra en mouvement. Nous nous sommes attaché à évaluer l'intérêt de la méthode dans des conditions réalistes de fonctionnement réel, c'est-à-dire en utilisant les données brutes des capteurs de déplacement de la caméra. Ce travail a prouvé la validité de l'approche et montré que l'on pouvait obtenir des résultats de reconstruction fiables et précis avec une caméra et un robot bien calibrés.

Les données tridimensionnelles obtenues peuvent être utilisées par exemple en robotique de manipulation pour la manipulation d'objets inconnus, en robotique mobile pour l'évitement d'obstacles, ou afin de générer automatiquement des descriptions d'objets.

La détermination des paramètres du mouvement de la caméra par rapport à l'objet observé représente enfin la dernière contribution importante de ce mémoire. Il est démontré que l'observation du déplacement d'un contour de discontinuité permet de déterminer avec précision le mouvement relatif entre la scène et la caméra. Cette détermination est faite à l'échelle près pour la translation de la caméra, et nous avons testé nos algorithmes sur des données réelles.

Ces données sur le mouvement peuvent aider la navigation des robots mobiles, en complément des capteurs de mouvement de type odométrique, accéléromètre, ou centrale inertielle.

Ce travail a révélé de nombreuses perspectives de recherche qui n'ont malheureusement pas pu être couvertes pendant cette thèse. Les problèmes suivants méritent d'être approfondis :

1. le suivi des contours le long de la séquence et l'établissement des correspondances.
2. l'analyse de la précision des résultats. En particulier, nous pensons qu'il est important d'avoir des mesures quantitatives sur la qualité de la reconstruction et de s'intéresser au problème du choix des points pris en compte pour l'autocalibration.
3. une technique de résolution pour l'autocalibration, plus sophistiquée que les différences finies : la faisabilité est maintenant démontrée à l'aide de cette méthode simple, et il est intéressant de voir les précisions qui peuvent être atteintes en résolvant les équations différentielles avec une meilleure précision que l'ordre 0.
4. enfin, l'apport de techniques sans calibration pour faire du positionnement relatif nous paraît fondamental, et nous espérons que notre travail sera repris dans cet esprit, pour comparer la qualité des résultats avec les nôtres.

Annexe A

Présentation générale des B_splines

Sans perte de généralité, on supposera une B_spline exprimée avec un paramètre t , $t \in [0, 1]$. On utilisera une subdivision régulière de $[0, 1]$ en n intervalles : $0 = u_0 < u_1 < \dots < u_n = 1$. On introduit des nœuds supplémentaires u_{-d}, \dots, u_{-1} et u_{n+1}, \dots, u_{n+d} de façon à pouvoir traiter les premiers et derniers intervalles comme dans le cas général. A chaque intervalle j est associé une fonction de base B_j , de degré d , dont le support est $[u_j, u_{j+d+1}]$. La forme générale de la B_spline Q est :

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

Les α_j sont appelés points de contrôle.

Les fonctions de base B_j sont définies comme suit.

A.1 Détermination des fonctions de base des B_splines

Une B_spline de degré d est une fonction C^{d-1} , polynomiale par morceaux, combinaison linéaire de fonctions de base B_j . Cela impose aux fonctions de base B_j d'être polynomiales de degré d par morceaux, et d'être C^{d-1} . On choisit ces fonctions de telle façon qu'elles aient un support limité, ou alternativement qu'elles valent 0 en dehors de leur support. On montre facilement que les fonctions B_j doivent avoir un support de $d+1$ intervalles pour vérifier toutes les contraintes : continuité à l'ordre $d-1$ des dérivées de B_j en chacun des points frontières des intervalles.

Ainsi, (voir figure A.1) une fonction de base de B_spline linéaire est définie sur deux intervalles, est linéaire sur chaque intervalle, et est C^0 aux frontières des intervalles. De même pour les B_splines cubiques, les fonctions de base sont définies sur 4 intervalles et sont C^3 aux frontières (voir

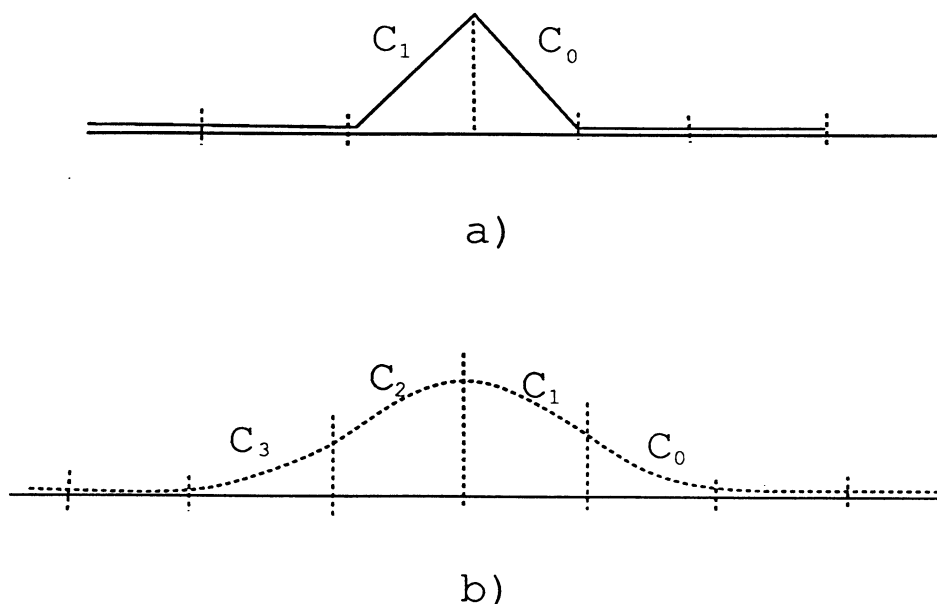


Figure A.1. Fonctions de base pour les B_splines a) linéaires b) cubiques

figure A.1).

Un calcul très simple¹ montre que ces contraintes sont insuffisantes, et qu'il faut ajouter une contrainte supplémentaire arbitraire pour la résolution.

La contrainte supplémentaire choisie est $\sum_{i=0}^d C_i(t') = 1$, $\forall t' \in [0, 1]$, car elle donne une propriété intéressante aux points de contrôle :

Pour un t donné, seules $d + 1$ fonctions de base participent réellement au calcul de $Q(t)$ et $Q(t) = \sum_{i=-d}^{n-1} \alpha_i B_i(t) = \sum_{i=j-d}^j \alpha_i B_i(t) = \sum_{i=j-d}^j \alpha_i C_{i-j+d}(t)$ peut alors être interprété comme le fait que pour l'intervalle u_j concerné, le point $Q(t)$ est le barycentre des points de contrôle α_i concernés ($\alpha_{j-d}, \dots, \alpha_j$) affectés des poids $B_i(t)$, où que les $B_i(t)$ sont les coordonnées barycentriques de $Q(t)$ relativement aux points de contrôle.

1. La fonction de base d'une B_spline de degré d est une fonction polynomiale par morceaux de degré d , définie sur $d + 1$ intervalles : en effet, aux frontières des intervalles, la continuité est C^{d-1} , et l'on a donc d contraintes à chaque point frontière. Il y a $d + 2$ points frontière, soit $d \times (d + 2)$ contraintes. Or, il y a $(d + 1) \times (d + 1)$ paramètres de polynômes à déterminer, soit un paramètre de trop, d'où le besoin d'adjoindre une contrainte supplémentaire.

En revanche, on voit de même qu'une fonction de base avec un support de d intervalles ne suffirait pas puisque l'on aurait $d + 1$ points frontières soit $d \times (d + 1)$ contraintes pour $d \times (d + 1)$ paramètres de polynômes ; la résolution de nos contraintes nous fournirait une seule solution qui serait le polynôme nul sur chaque intervalle, or cette solution triviale ne nous intéresse pas.

On peut montrer rapidement comment calculer ces fonctions de base dans le cas des B_splines linéaires : les contraintes sont la continuité d'ordre 0 à chaque frontière d'intervalle. On note C_0 et C_1 les deux morceaux de polynôme de degré 1 de B , et l'on se ramène à un paramétrage entre 0 et 1 sur chaque intervalle. Ainsi on a :

$$\begin{cases} C_0(1) = 0 \\ C_1(0) = 0 \\ C_0(0) = C_1(1) \end{cases}$$

C_0 et C_1 sont définis par :

$$\begin{cases} C_0(t) = a_0t + b_0 \\ C_1(t) = a_1t + b_1 \end{cases}$$

d'où :

$$\begin{cases} b_0 = 0 \\ a_0 = b_1 \\ a_0 + a_1 = 0 \end{cases}$$

La contrainte supplémentaire choisie est $C_0(0) + C_1(0) = 1$.

On en tire finalement :

$$\begin{cases} a_0 = 1 & b_0 = 0 \\ a_1 = -1 & b_1 = 1 \end{cases}$$

La même approche est valable pour les fonctions de base des B_splines de degré supérieur.

Le calcul pour les B_splines cubiques est trop long pour être explicité ici, mais on obtient, en notant $p = t$ et $q = 1 - t$:

$$\begin{cases} C_0(t) = q^3/6 \\ C_1(t) = (3q(pq + 1) + 1)/6 \\ C_2(t) = (3p(pq + 1) + 1)/6 \\ C_3(t) = p^3/6 \end{cases}$$

A.2 L'ajustement des données aux moindres carrés

Le problème de l'approche aux moindres carrés se pose ainsi :

soient $p + 1$ points (x_i, t_i) , $t_i \in [0, 1]$,

soit n le nombre d'intervalles réguliers dans $[0, 1]$,

soit Q , la B_spline de degré d recherchée :

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

L'approche aux moindres carrés consiste à minimiser le terme quadratique d'erreur :

$$\begin{aligned} E &= \sum_{i=0}^p (Q(t_i) - x_i)^2 \\ &= \sum_{i=0}^p \left(\left(\sum_{j=-d}^{n-1} \alpha_j B_j(t_i) \right) - x_i \right)^2 \end{aligned}$$

Cette minimisation par rapport aux α_j se fait en annulant la dérivée partielle de E par rapport à chaque α_j , ce qui nous donne les $n+d$ équations linéaires suivantes :

Pour k de $-d$ à $n-1$:

$$\frac{\partial E}{\partial \alpha_k} = 0 \Rightarrow \sum_{j=-d}^{n-1} \alpha_j \left(\sum_{i=0}^p B_j(t_i) B_k(t_i) \right) = \sum_{i=0}^p x_i B_k(t_i)$$

Les seules valeurs de j pour lesquelles il faut calculer $B_j(t_i)$ sont $j = k-d, \dots, k$, si $[u_k, u_{k+1}]$ est l'intervalle concernant t_i . On obtient ainsi un système $M.X = B$, où X est le vecteur des α_j , et où M est une matrice $(n+d) \times (n+d)$ symétrique positive bande avec d sous-diagonales, ce qui permet d'utiliser des techniques numériques de résolution extrêmement efficaces, comme une variation de Choleski pour matrice bande, linéaire par rapport au nombre de points de contrôle.

A.3 Ajustement avec terme de régularisation

Il est possible, sans augmenter la complexité du système à résoudre, d'imposer une contrainte supplémentaire.

$$E' = \sum_{i=0}^p (Q(t_i) - x_i)^2 + \tau \int_0^1 (Q^{(d)}(t))^2 dt$$

On dérive le terme supplémentaire par rapport aux α_j pour minimiser le nouveau E' obtenu. Tout calcul fait, l'apport de cette contrainte supplémentaire revient à rajouter à M les termes de régularisation, pour chaque intervalle et chaque j et k :

Pour k de $-d$ à $n-1$:

$$\begin{aligned}
\frac{\partial}{\partial \alpha_k} \left[\int_0^1 (Q^{(d)}(t))^2 dt \right] &= \int_0^1 \sum_{j=-d}^{n-1} \alpha_j B_j^{(d)}(t) B_k^{(d)}(t) dt \\
&= \sum_{i=0}^{n-1} \int_{u_i}^{u_{i+1}} \sum_{j=-d}^{n-1} \alpha_j B_j^{(d)}(t) B_k^{(d)}(t) dt
\end{aligned}$$

Les B_j étant polynomiaux de degré d par morceaux, $B_j^{(d)}(t)$ est constant sur chaque intervalle, d'où l'ajout à M , pour chaque intervalle, d'une matrice symétrique $(d+1) \times (d+1)$.

Annexe B

La correction radiale du modèle sténopé

On reprend les notations introduites au paragraphe 2.2.

Si l'on exprime M par sa position en pixels $M_p(X_p, Y_p)$ on obtient :

$$X_p = X_c + \frac{X}{l_x} \quad , \quad Y_p = Y_c + \frac{Y}{l_y}$$

Un coefficient unique K_d , coefficient de déformation radiale, capture la déformation observée dans de nombreuses caméras.

Le point réellement observé dans le plan image n'est pas M_p mais M_d avec :

$$\begin{cases} X_d = X_p + K_d \cdot (X_p - X_c) \cdot [(X_p - X_c)^2 + (Y_p - Y_c)^2] \\ Y_d = Y_p + K_d \cdot (Y_p - Y_c) \cdot [(X_p - X_c)^2 + (Y_p - Y_c)^2] \end{cases} \quad (\text{B.1})$$

Soit le modèle complet de la caméra :

$$\begin{cases} X_d = X_c + F_x \cdot \frac{x}{z} + K_d \cdot F_x \cdot \frac{x}{z} \cdot [(F_x \cdot \frac{x}{z})^2 + (F_y \cdot \frac{y}{z})^2] \\ Y_d = Y_c + F_y \cdot \frac{y}{z} + K_d \cdot F_y \cdot \frac{y}{z} \cdot [(F_x \cdot \frac{x}{z})^2 + (F_y \cdot \frac{y}{z})^2] \end{cases} \quad (\text{B.2})$$

avec $F_x = \frac{f}{l_x}$ et $F_y = \frac{f}{l_y}$.

Le problème posé est souvent le problème inverse soit :

Soit un point observé M_d , quel est le point M correspondant qui obéit à la transformation projection perspective sans déformation radiale ?

On peut réécrire B.1 et B.1 sous la forme :

$$\begin{cases} X_d = X_c + F_x \cdot X + K_d \cdot F_x \cdot X \cdot [(F_x \cdot X)^2 + (F_y \cdot Y)^2] \\ Y_d = Y_c + F_y \cdot Y + K_d \cdot F_y \cdot Y \cdot [(F_x \cdot X)^2 + (F_y \cdot Y)^2] \end{cases} \quad (\text{B.3})$$

Par élimination du terme $[(F_x \cdot X)^2 + (F_y \cdot Y)^2]$ dans B.3, on obtient une première relation très simple :

$$F_y \cdot Y = F_x \cdot X \cdot \frac{Y_d - Y_c}{X_d - X_c} \quad (\text{B.4})$$

Par substitution de B.4 dans B.3 on obtient :

$$\begin{cases} X_d - X_c - F_x \cdot X - K_d \cdot \left[1 + \left(\frac{Y_d - Y_c}{X_d - X_c} \right)^2 \right] \cdot (F_x \cdot X)^3 = 0 \\ Y_d - Y_c - F_y \cdot Y - K_d \cdot \left[1 + \left(\frac{X_d - X_c}{Y_d - Y_c} \right)^2 \right] \cdot (F_y \cdot Y)^3 = 0 \end{cases}$$

On sait que X_d et Y_d sont des approximations de solutions pour X_p et Y_p , donc on effectue les changements de variable suivants :

$$\begin{cases} X_r = X_d - X_c \\ Y_r = Y_d - Y_c \\ \varepsilon_x = \frac{F_x \cdot X}{X_r} - 1 = \frac{X_p - X_d}{X_d - X_c} \\ \varepsilon_y = \frac{F_y \cdot Y}{Y_r} - 1 = \frac{Y_p - Y_d}{Y_d - Y_c} \end{cases}$$

où ε_x et ε_y représentent l'erreur relative faite en prenant X_p et Y_p comme solution au lieu de X_d et Y_d , par rapport à X_c et Y_c . ε_x et ε_y sont des infiniment petits du premier ordre.

On tire :

$$\begin{cases} \frac{\varepsilon_x}{X_r^2} + K_d \cdot \left[1 + \left(\frac{Y_r}{X_r} \right)^2 \right] (1 + \varepsilon_x)^3 = 0 \\ \frac{\varepsilon_y}{Y_r^2} + K_d \cdot \left[1 + \left(\frac{X_r}{Y_r} \right)^2 \right] (1 + \varepsilon_y)^3 = 0 \end{cases} \quad (\text{B.5})$$

ε_x et ε_y sont petits devant 1, donc le système B.5 se ramène par approximation au premier ordre à :

$$\varepsilon_x = \varepsilon_y = -\frac{K_d \cdot R^2}{1 + 3K_d \cdot R^2}$$

avec $R^2 = X_r^2 + Y_r^2$.

L'égalité de ε_x et ε_y montre bien l'aspect radial de la correction géométrique.

On obtient finalement la solution :

$$\begin{cases} X = \frac{X_r \cdot (1 + \varepsilon_x)}{F_x} \\ Y = \frac{Y_r \cdot (1 + \varepsilon_y)}{F_y} \end{cases}$$

Annexe C

Le filtrage optimal par filtres récurrents

C.1 Filtrage optimal

Le signal considéré $I(x)$ dans le cas monodimensionnel de la recherche de points de contraste est un signal de type échelon $U(x)$ d'amplitude A , bruité par un bruit blanc gaussien $n(x)$ de variance connue n_0^2 :

$$I(x) = A.U(x) + n(x)$$

Les critères de Canny [Can 83] [Can 86] du filtre f pour la détection et la localisation de points de contraste dans un signal de type échelon, bruité par un bruit blanc gaussien, sont au nombre de trois :

Bonne détection La probabilité de ne pas détecter un véritable point de contraste, ou de détecter à tort un faux point de contraste doit être minimisée. Ce critère correspond au maximum du rapport signal/bruit SNB défini comme le quotient de la réponse à un signal de type échelon non bruité sur la racine carrée de la moyenne du carré de la réponse au signal bruité :

$$SNB = \frac{A}{n_0} \cdot \frac{\int_{-\infty}^0 f(x) dx}{\left[\int_{-\infty}^{\infty} f^2(x) dx \right]^{1/2}}$$

Bonne localisation Les points de contraste détectés doivent être les plus proches possible des véritables points de contraste. Ce critère correspond à minimiser la variance de la position du point de contraste, ou à trouver le maximum du critère suivant :

$$L = \frac{A}{n_0} \cdot \frac{|f'(0)|}{\left[\int_{-\infty}^{\infty} f^2(x) dx \right]^{1/2}}$$

Une réponse unique Le filtre devrait enfin fournir une réponse unique à un unique point de contraste dans le signal. La distance x_{max} entre les pics de la réponse au signal bruité est une fraction k de la largeur W de l'opérateur f :

$$x_{max} = k.W = 2.\Pi. \left[\frac{\int_{-\infty}^{\infty} f'^2(x)dx}{\int_{-\infty}^{\infty} f''^2(x)dx} \right]^{1/2}$$

Canny définit le filtre optimal comme le filtre qui maximise le produit $SNB.L$ sous la contrainte du troisième critère. La solution f pour le filtre obéit à l'équation différentielle suivante :

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

Dans le cas d'un filtre antisymétrique à support fini $[0, W]$, la solution est de la forme :

$$f(x) = a_1.e^{\alpha x}.sin(\omega.x) + a_2.e^{\alpha x}.cos(\omega.x) + a_3.e^{-\alpha x}.sin(\omega.x) + a_4.e^{-\alpha x}.sin(\omega.x) + C$$

avec les conditions aux bornes :

$$f(0) = 0 \quad f(W) = 0 \quad f'(0) = S \quad f'(W) = 0$$

Le filtre est étendu au domaine $[-W, +W]$ par $f(x) = -f(-x)$.

Canny a remarqué que son opérateur avait à peu près la forme de la première dérivée d'une gaussienne :

$$f(x) = -\frac{x}{\sigma^2}.e^{-x^2/2.\sigma^2}$$

Ce filtre permet une extension simple au cas bi-dimensionnel, mais est sous-optimal.

Deriche [Der 90] [Der 87b] propose l'utilisation d'un filtre antisymétrique à support infini, qui vérifie la même équation différentielle, avec les conditions aux bornes suivantes :

$$f(0) = 0 \quad f(+\infty) = 0 \quad f'(0) = S \quad f'(+\infty) = 0$$

D'où une solution plus simple que le filtre de Canny :

$$f(x) = -c.e^{-\alpha.|x|}.sin(\omega.x)$$

Ce filtre est approximativement équivalent sous certaines conditions à :

$$f(x) = -c.x.e^{-\alpha|x|}$$

C.2 Filtres récurrents

Soit $f(n)$ la fonction discrète correspondant à $f(x)$, le filtre à réaliser. Soit $F(Z)$ sa transformée en Z .

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

$f(n)$ est réécrite sous la forme de la somme de deux fonctions causales :

$$\left| \begin{array}{l} f(n) = f_-(n) + f_+(n) \text{ , pour } n = -\infty, \dots, \infty \\ f_-(n) = \begin{cases} 0 & n \geq 0 \\ \alpha_1(p_2)^n + \alpha_1^*(p_2^*)^n & n < 0 \end{cases} \\ f_+(n) = \begin{cases} 0 & n < 0 \\ \alpha_1(p_2)^n + \alpha_1^*(p_2^*)^n & n \geq 0 \end{cases} \\ \text{avec } \alpha_1 = \frac{-c}{2.i} \text{ , } p_1 = e^{-\alpha+i.\omega} \text{ , } p_2 = e^{\alpha+i.\omega} \end{array} \right.$$

où * représente le conjugué complexe.

Chaque terme $\alpha_i.(p_i)^n$ a pour transformée en Z la fonction $\frac{\alpha_i}{(1-p_i.Z^{-1})}$, d'où :

$$\left| \begin{array}{l} F(Z) = F_-(Z) + F_+(Z) \\ F_+(Z^{-1}) = \frac{a.Z^{-1}}{1+b_1.Z^{-1}+b_2.Z^{-2}} \\ F_-(Z) = \frac{-a.Z}{1+b_1.Z+b_2.Z^2} \\ a = -c.e^{-\alpha}.\sin(\omega) \text{ , } b_1 = -2.e^{-\alpha}.\cos(\omega) \text{ , } b_2 = e^{-2.\alpha} \end{array} \right.$$

Ces deux transformées en Z correspondent à deux fonctions de transfert rationnelles du deuxième ordre récurrentes de gauche à droite pour F_+ et de droite à gauche pour F_- .

La réponse y d'un tel système à une entrée x se calcule récursivement de la façon suivante :

$$\left| \begin{array}{l} y^+(m) = x(m-1) - b_1.y^+(m-1) - b_2.y^+(m-2) \text{ , pour } m = 1, \dots, M \\ y^-(m) = x(m+1) - b_1.y^-(m+1) - b_2.y^-(m+2) \text{ , pour } m = M, \dots, 1 \\ y(m) = a.(y^+(m) - y^-(m)) \text{ , pour } m = 1, \dots, M \end{array} \right.$$

c est déterminé par la contrainte de normalisation suivante :

$$\sum_0^{+\infty} f(n) = - \sum_{-\infty}^0 f(n) = -1$$

d'où :

$$c = \frac{1 - 2.e^{-\alpha}.\cos(\omega) + e^{-2.\alpha}}{e^{-\alpha}.\sin(\omega)}$$

C.3 Lissage par filtres récursifs

La convolution du filtre optimal avec le signal est équivalente à la dérivée de la convolution du signal avec l'intégrale du filtre. L'intégrale du filtre optimal est un filtre lisseur.

Deriche [Der 90] propose le filtre lisseur discret suivant :

$$S(n) = k(\alpha |n| + 1)e^{-\alpha|n|}$$

Le filtre récursif du deuxième ordre qui réalise cette fonction pour un signal discret x de M valeurs est le filtre y de la forme :

$$\left\{ \begin{array}{ll} k & = \frac{(1-e^{-\alpha})^2}{1+2\alpha e^{-\alpha}-e^{-2\alpha}} \\ x(0) & = 0 \\ y_1(0) & = y_1(-1) = 0 \\ y_1(n) & = k [x(n) + e^{-\alpha}(\alpha - 1)x(n - 1)] \\ & \quad + 2e^{-\alpha}y_1(n - 1) - e^{-2\alpha}y_1(n - 2) \quad , \text{ pour } n = 1, \dots, M \\ x(M + 1) & = x(M + 2) = 0 \\ y_2(M + 1) & = y_2(M + 2) = 0 \\ y_2(n) & = k [e^{-\alpha}(\alpha + 1)x(n + 1) + e^{-2\alpha}x(n + 2)] \\ & \quad + 2e^{-\alpha}y_2(n + 1) - e^{-2\alpha}y_2(n + 2) \quad , \text{ pour } n = M, \dots, 1 \\ y(n) & = y_1(n) + y_2(n) \quad , \text{ pour } n = 1, \dots, M \end{array} \right.$$

Shen [She 86] propose un filtre lisseur sous-optimal implémentable sous la forme d'un filtre récursif du premier ordre :

$$S(n) = k_0 e^{-\alpha|n|}$$

avec :

$$\left\{ \begin{array}{ll} k_0 & = \frac{1-e^{-\alpha}}{1+e^{-\alpha}} \\ y_1(n) & = x(n) + e^{-\alpha}y_1(n - 1) \quad , \text{ pour } n = 1, \dots, M \\ y_2(n) & = e^{-\alpha}(x(n + 1) + y_2(n + 1)) \quad , \text{ pour } n = M, \dots, 1 \\ y(n) & = k_0(y_1(n) + y_2(n)) \quad , \text{ pour } n = 1, \dots, M \end{array} \right.$$

L'opérateur de Shen est plus efficace que celui de Deriche car les deux passes pour le calcul de y_1 et y_2 peuvent être effectuées en parallèle et parce que le nombre d'additions et de multiplications est moindre. Par contre le rapport signal/bruit de l'opérateur de Shen est moins bon.

C.4 Dérivation par filtres récursifs

Plutôt que de lisser puis de dériver, il est possible de combiner les deux phases en un filtre unique, dérivée du filtre lisseur, et d'obtenir ainsi directement une implémentation par filtre récursif du deuxième ordre du filtre optimal de Deriche.

Le filtre a la forme suivante :

$$D(n) = k.n.e^{-\alpha|n|}$$

Le filtre récursif y du deuxième ordre qui réalise cette fonction est le suivant :

$$\left| \begin{array}{l} k = -\frac{(1-e^{-\alpha})^2}{e^{-\alpha}} \\ y_1(n) = x(n-1) + 2.e^{-\alpha}.y_1(n-1) - e^{-2\alpha}.y_1(n-2) \quad , \text{ pour } n = 1, \dots, M \\ y_2(n) = x(n+1) + 2.e^{-\alpha}.y_2(n+1) - e^{-2\alpha}.y_2(n+2) \quad , \text{ pour } n = M, \dots, 1 \\ y(n) = k.e^{-\alpha} [y_1(n) - y_2(n)] \quad , \text{ pour } n = 1, \dots, M \end{array} \right.$$

La dérivée du filtre de Shen a la forme suivante :

$$D(n) = \begin{cases} -\frac{(1-e^{-\alpha})}{e^{-\alpha}}.e^{-\alpha n} & , \text{ pour } n > 0 \\ 0 & , \text{ pour } n = 0 \\ \frac{(1-e^{-\alpha})}{e^{-\alpha}}.e^{\alpha n} & , \text{ pour } n < 0 \end{cases}$$

Cette fonction est réalisée par le filtre récursif y du premier ordre suivant :

$$\left| \begin{array}{l} y_1(n) = x(n-1) + e^{-\alpha}.y_1(n-1) \quad , \text{ pour } n = 1, \dots, M \\ y_2(n) = -x(n+1) + e^{-\alpha}.y_2(n+1) \quad , \text{ pour } n = M, \dots, 1 \\ y(n) = -(1-e^{-\alpha}) [y_1(n) + y_2(n)] \quad , \text{ pour } n = 1, \dots, M \end{array} \right.$$

Annexe D

Géométrie différentielle des courbes et surfaces

Ces rappels généraux sur la géométrie différentielle des surfaces peuvent être trouvés dans [Car 76].

D.1 Courbes

Une courbe paramétrée différentiable est une application $\alpha : I \rightarrow \mathbb{R}^3$ d'un ouvert I de \mathbb{R} dans \mathbb{R}^3 .

La paramétrisation α est dite régulière si $\alpha'(t) \neq 0$ pour tout $t \in I$.

L'abscisse curviligne s à partir de $t_0 \in I$, associée à α , est définie par :

$$s(t) = \int_{t_0}^t |\alpha'(t)| dt$$

Nous nous plaçons maintenant dans le cas où la courbe est exprimée par son abscisse curviligne.

$\alpha'(s) = T(s)$ est le **vecteur tangent** en s à la courbe.

$|\alpha''(s)| = \kappa(s)$ est la **courbure** en s à la courbe. Si $\kappa(s) \neq 0$ alors $1/\kappa(s)$ est appelé **rayon de courbure** en s à la courbe.

$N(s)$: Si la courbure en s est non nulle, alors le vecteur $N(s)$ tel que $\alpha''(s) = \kappa(s)N(s)$ est appelé **vecteur normal** en s à la courbe.

$B(s)$: Le vecteur $B(s) = T(s) \wedge N(s)$ est appelé **vecteur binormal**.

$\Theta(s)$: $\Theta(s)$ tel que $B'(s) = \Theta(s)N(s)$ est appelé **torsion** en s à la courbe.

Les trois vecteurs $T(s)$, $N(s)$ et $B(s)$ forment un repère orthonormé appelé repère de Frénet, comme illustré sur la figure D.1.

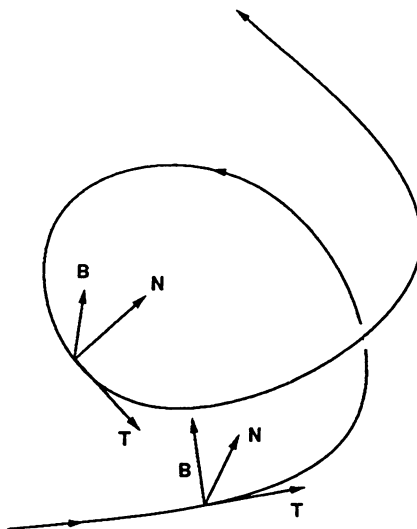


Figure D.1. Repère de Frénet

D.2 Surfaces régulières

Un sous-ensemble (S) de \mathbb{R}^3 est une surface régulière si, pour tout point P de (S) , il existe un voisinage V de \mathbb{R}^3 et une application $X : U \rightarrow V \cap (S)$ d'un ouvert $U \subseteq \mathbb{R}^2$ dans $V \cap S \subseteq \mathbb{R}^3$ tel que :

1. X est un homéomorphisme, c'est-à-dire que X est continue et a une inverse X^{-1} continue.
2. X est différentiable.
3. la différentielle $dX_q : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ de X en q pour $q \in U$ est une injection.

D.3 La première forme fondamentale

Le produit scalaire de \mathbb{R}^3 induit un produit scalaire restreint sur le plan tangent T_p d'une surface régulière (S) noté \langle, \rangle_p . Si w_1 et w_2 sont deux vecteurs du plan tangent T_p alors :

$$\langle w_1, w_2 \rangle_p = w_1 \cdot w_2$$

où “ \cdot ” désigne le produit scalaire naturel de \mathbb{R}^3 .

La forme quadratique $I_p(w)$ est définie par :

$$I_p(w) = \langle w, w \rangle_p = w \cdot w = |w|^2 \geq 0$$

La forme quadratique $I_p(w)$ est appelée la première forme fondamentale de la surface S en P .

La première forme fondamentale permet de faire des mesures sur la surface (longueur de courbes, angles de vecteurs, surfaces de régions).

Soit une paramétrisation X de la surface (S) :

$$\begin{aligned} X : U &\rightarrow (S) \\ (u, v) &\mapsto (x(u, v), y(u, v), z(u, v)) \end{aligned}$$

Un vecteur w du plan tangent en P s'exprime dans la base $\left\{\frac{\partial X}{\partial u}, \frac{\partial X}{\partial v}\right\}$ notée $\{X_u, X_v\}$:

$$w = \lambda X_u + \mu X_v$$

d'où :

$$I_p(w) = \langle w, w \rangle_p = \lambda^2 E + 2\lambda\mu F + \mu^2 G$$

avec :

$$E = \langle X_u, X_u \rangle_p$$

$$F = \langle X_u, X_v \rangle_p$$

$$G = \langle X_v, X_v \rangle_p$$

La normale à la surface en P est N avec :

$$N = \frac{X_u \wedge X_v}{H}$$

avec :

$$H = \sqrt{EG - F^2} = |X_u \wedge X_v|$$

H est le déterminant de la matrice de l'endomorphisme associé à la première forme fondamentale, et est indépendant de la paramétrisation.

D.4 La deuxième forme fondamentale

D.4.1 Carte de Gauss

Soit (S) une surface d'orientation N . L'application $N : S \rightarrow \mathbb{R}^3$ qui prend ses valeurs sur la sphère unité est appelée carte de Gauss.

La différentielle $dN_p : T_p(S) \rightarrow T_p(S)$ permet de définir la deuxième forme fondamentale II_p :

$$II_p(v) = - \langle dN_p(v), v \rangle_p$$

La deuxième forme fondamentale est une notion reliée à la courbure normale.

Soit une surface (S) et un vecteur w du plan tangent en un point P de la surface. Le plan contenant la normale à la surface et le vecteur w coupe la surface le long d'un arc sur la surface de (S) . La courbure de cet arc est appelée courbure normale κ_n .

Toutes les courbes tracées à la surface de (S) , passant par P , et tangentes à un vecteur w du plan tangent en P ont la même courbure normale. Une illustration de cette construction est présentée figure D.2.

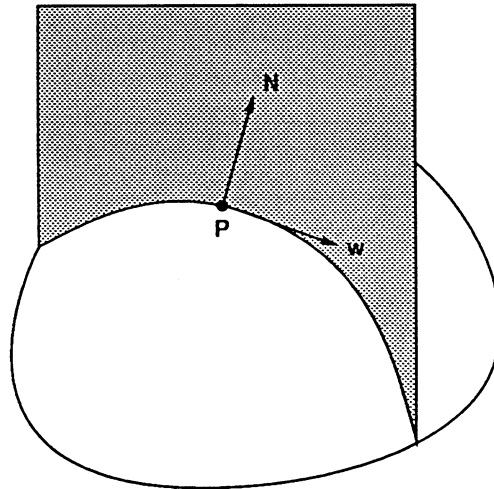


Figure D.2. Courbure normale

L'interprétation géométrique de la deuxième forme fondamentale est liée à la courbure normale :

$$II_p(w) = \kappa_n$$

Si l'on exprime w dans la base $\{X_u, X_v\}$ alors on obtient :

$$w = \lambda X_u + \mu X_v$$

$$II_p(w) = \lambda^2 L + 2\lambda\mu M + \mu^2 N$$

avec :

$$L = \frac{\partial^2 X}{\partial u^2} \cdot N_{S(P)}$$

$$M = \frac{\partial^2 X}{\partial u \partial v} \cdot N_{S(P)}$$

$$N = \frac{\partial^2 X}{\partial v^2} \cdot N_{S(P)}$$

Les directions principales de la surface (S) sont les vecteurs propres de l'endomorphisme associé à la deuxième forme fondamentale, et leurs coordonnées dans la base $\{X_u, X_v\}$ sont solutions de :

$$(FL - EM)\lambda^2 + (GL - EN)\lambda\mu + (GM - FN)\mu^2 = 0$$

Les courbures principales en sont les valeurs propres. Elles sont solution de :

$$(EG - F^2)\rho^2 - (LG + EN - 2FM)\rho + (LN - M^2) = 0$$

La courbure totale (ou gaussienne) κ_{gauss} est le produit des deux courbures principales. La courbure moyenne \mathcal{H} en est la demi-somme.

Soit :

$$\mathcal{H} = \frac{\rho_1 + \rho_2}{2} = \frac{GL + EN - 2FM}{2H^2}$$

$$\kappa_{gauss} = \rho_1 \rho_2 = \frac{LN - M^2}{H^2}$$

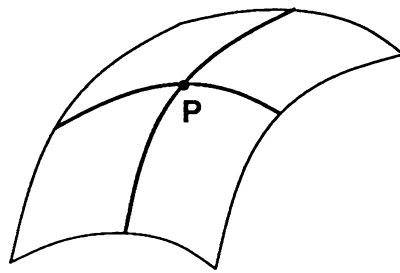
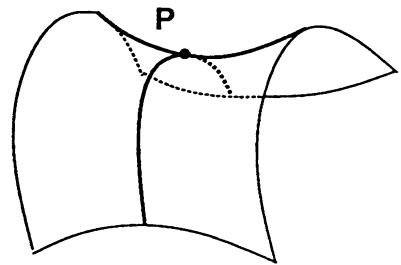
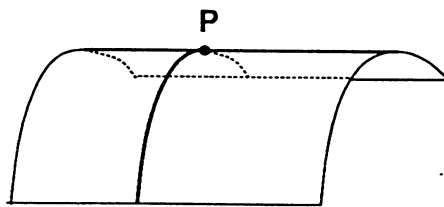
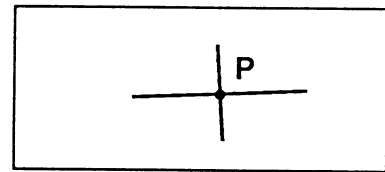
D.5 Classification des types de surface

La courbure gaussienne et la courbure moyenne sont à l'origine d'une classification des surfaces:

Un point P d'une surface (S) est:

- **elliptique** si $\kappa_{gauss} > 0$
- **hyperbolique** si $\kappa_{gauss} < 0$
- **parabolique** si $\kappa_{gauss} = 0$ et $\mathcal{H} \neq 0$.
- **plan** si $\kappa_{gauss} = 0$ et $\mathcal{H} = 0$.

La figure D.3 résume ces différents cas graphiquement, en montrant en gras les courbures principales.

**Elliptique****Hyperbolique****Parabolique****Plan****Figure D.3.** Courbures principales pour les classes de surface

Annexe E

Rappels de cinématique du solide

On suppose l'existence de deux repères mobiles l'un par rapport à l'autre : $\{O\}$, et $\{C\}$ mobile par rapport à $\{O\}$.

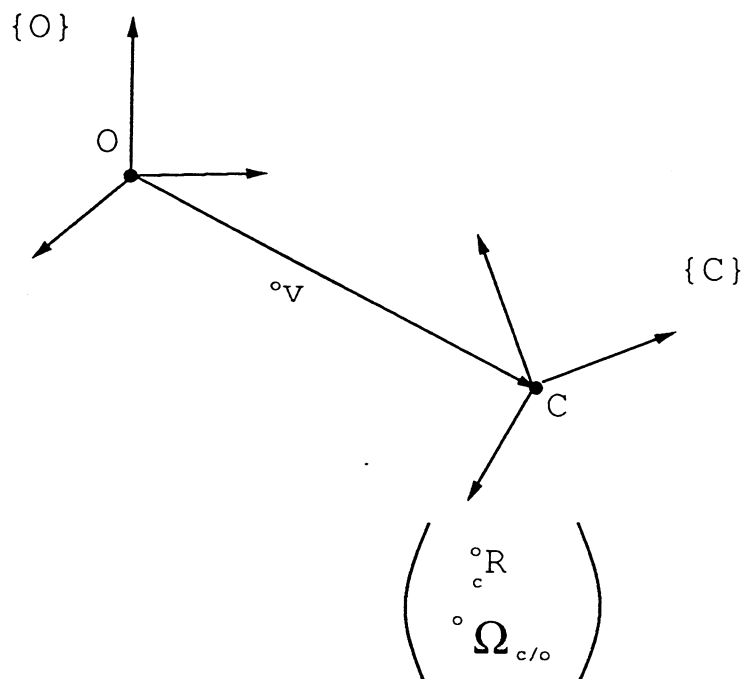


Figure E.1. Les repères et leurs relations

La figure E.1 montre que le mouvement de $\{C\}$ par rapport à $\{O\}$ est repéré par :

1. une translation ov , vecteur coordonnées de l'origine C de $\{C\}$ exprimé dans $\{O\}$.
2. une vitesse de translation ${}^o\dot{v}$ et une accélération ${}^o\ddot{v}$, dérivées première et seconde de ov exprimées dans $\{O\}$.

3. une vitesse de rotation instantanée ${}^o\Omega_{c/o}$, vecteur vitesse de rotation instantanée du mouvement de vissage entre $\{C\}$ et $\{O\}$, exprimant la rotation de $\{C\}$ par rapport à $\{O\}$, exprimée dans $\{O\}$. Ce vecteur a pour support l'axe de la rotation, et pour module la vitesse de rotation.
4. une accélération de rotation instantanée ${}^o\dot{\Omega}_{c/o}$, dérivée du vecteur vitesse de rotation instantanée, exprimée dans $\{O\}$.
5. La matrice de rotation entre le repère $\{C\}$ et le repère $\{O\}$ est notée : oR , et permet de passer des coordonnées d'un vecteur exprimé dans $\{C\}$ à des coordonnées exprimées dans $\{O\}$.

Soit un vecteur ${}^cU(U_1, U_2, U_3)$, fonction de t , mobile dans $\{C\}$. On note cU_t la dérivée de U dans le repère $\{C\}$, et oU_t la dérivée de U dans le repère $\{O\}$. On note de même ${}^c({}^oU_t)$ la dérivée de U dans le repère $\{O\}$, exprimée dans le repère $\{C\}$. Le repère $\{C\}$ est représenté par la matrice oR qui est composée des trois vecteurs E_1, E_2, E_3 , vecteurs de base du repère $\{C\}$ exprimés dans le repère $\{O\}$.

Le torseur cinématique de la caméra exprimé en C a pour moment ${}^o\dot{v}$ et pour vecteur ${}^o\Omega_{c/o}$.

On sait alors que le champs des vitesses d'un vecteur L lié au repère $\{C\}$ par rapport au repère $\{O\}$ s'exprime par :

$$\frac{dL}{dt} = {}^o\Omega_{c/o} \wedge L$$

Pour un vecteur U quelconque mobile dans le repère $\{C\}$ on a :

$$\begin{aligned} {}^oU &= \sum_i U_i E_i \\ {}^o\frac{dU}{dt} &= {}^o\dot{U} = \frac{d\sum_i U_i E_i}{dt} = \sum_i \frac{dU_i}{dt} E_i + \sum_i U_i \frac{dE_i}{dt} \\ &= {}^o\left({}^c\frac{dU}{dt}\right) + \sum_i U_i \cdot {}^o\Omega_{c/o} \wedge E_i \\ &= {}^o\left({}^c\frac{dU}{dt}\right) + {}^o\Omega_{c/o} \wedge U \\ &= {}^o({}^c\dot{U}) + {}^o\Omega_{c/o} \wedge U \end{aligned}$$

Le calcul de la dérivée seconde se fait en dérivant cette relation une deuxième fois :

$$\begin{aligned} {}^o\frac{d^2U}{dt^2} &= \frac{d}{dt} \left(\sum_i \frac{dU_i}{dt} E_i \right) + \frac{d}{dt} ({}^o\Omega_{c/o} \wedge U) \\ &= \sum_i \frac{d^2U_i}{dt^2} E_i + \sum_i \frac{dU_i}{dt} {}^o\Omega_{c/o} \wedge E_i + {}^o\dot{\Omega}_{c/o} \wedge U + \Omega \wedge ({}^o({}^c\dot{U}) + {}^o\Omega_{c/o} \wedge U) \\ &= {}^o({}^c\ddot{U}) + 2{}^o\Omega_{c/o} \wedge ({}^c\dot{U}) + {}^o\dot{\Omega}_{c/o} \wedge U + {}^o\Omega_{c/o} \wedge ({}^o\Omega_{c/o} \wedge U) \end{aligned}$$

Annexe F

Comparaison avec le flot optique

Les travaux initiaux de Longuet-Higgins [LH 80] Horn [Hor 80] [Neg 87] et Nagel [Nag 82] sur le flot optique ont des liens étroits avec la technique que nous venons d'exposer, et mettent en lumière de nombreuses propriétés intéressantes. Dans les deux cas en effet sont étudiées des séquences denses d'images à niveaux de gris par des techniques différentielles. Nous allons montrer l'équivalence des formules obtenues dans le cas de la reconstruction, en explicitant les paramètres du mouvement.

F.1 Equivalence avec le flot optique

Soit une séquence dense d'images de niveaux de gris $E(x, y, t)$.

Soit un plan image à une distance focale arbitraire 1 (ce qui ne nuit pas à la généralité du résultat).

Soit $p = (x, y, 1)$ l'image sur le plan optique d'un point de la scène $P = (X, Y, Z)$.

Soient \dot{v} et Ω la vitesse de translation instantanée et la vitesse de rotation instantanée des points de la scène par rapport au repère de la caméra.

Les hypothèses de Horn et Schunck [Hor 80] qui supposent négligeables les variations de luminance dues au changement de l'orientation des surfaces observées et aux variations de l'éclairage permettent de conclure que la dérivée totale de la luminance¹ est nulle :

$$\frac{d E}{d t} = \frac{\partial E}{\partial t} + \frac{\partial E}{\partial x} \times \frac{d x}{d t} + \frac{\partial E}{\partial y} \times \frac{d y}{d t} = E_t + x_t \times E_x + y_t \times E_y = 0 \quad (\text{F.1})$$

En intégrant les contraintes du mouvement, on peut réécrire l'équation F.1 sous la forme suivante, due à Horn et Negahdaripour [Neg 87] :

1. On considère ici le flot réel, alors qu'il est notoirement connu que seul le flot apparent est mesurable [Fau 90]

$$E_t + a.\Omega + \frac{\dot{v}.b}{Z} = 0 \quad (\text{F.2})$$

où les vecteurs a et b sont définis comme suit :

$$a = \begin{pmatrix} -E_x \\ -E_y \\ xE_x + yE_y \end{pmatrix}$$

$$b = \begin{pmatrix} E_y + y(xE_x + yE_y) \\ -E_x - x(xE_x + yE_y) \\ yE_x - xE_y \end{pmatrix}$$

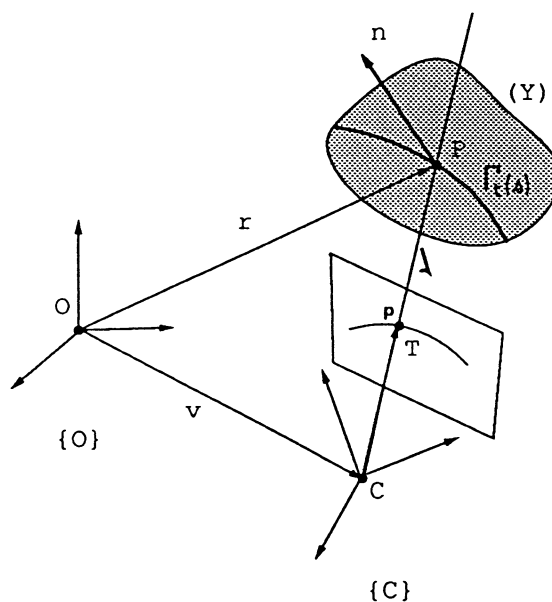


Figure F.1. Comparaison reconstruction/flot optique

Pour comparer cette formulation avec la nôtre, il convient de démontrer un résultat au préalable : il faut noter que l'équation de la reconstruction 3.8 est inchangée lorsque le vecteur ligne de vue T devient le vecteur reliant le centre optique au point p du plan image. Le module de T n'est plus unitaire mais cela n'affecte pas la forme de l'équation de reconstruction ; seule la signification de λ est changée. λ devient le facteur de proportionnalité entre le vecteur T et le vecteur CP , comme illustré dans la figure F.1. On a donc encore :

$$\lambda = \frac{{}^o\dot{v}.{}^on}{{}^oT_t.{}^on} \quad (\text{F.3})$$

La figure F.2 montre les notations respectives de chacune des approches reconstruction et flot optique.

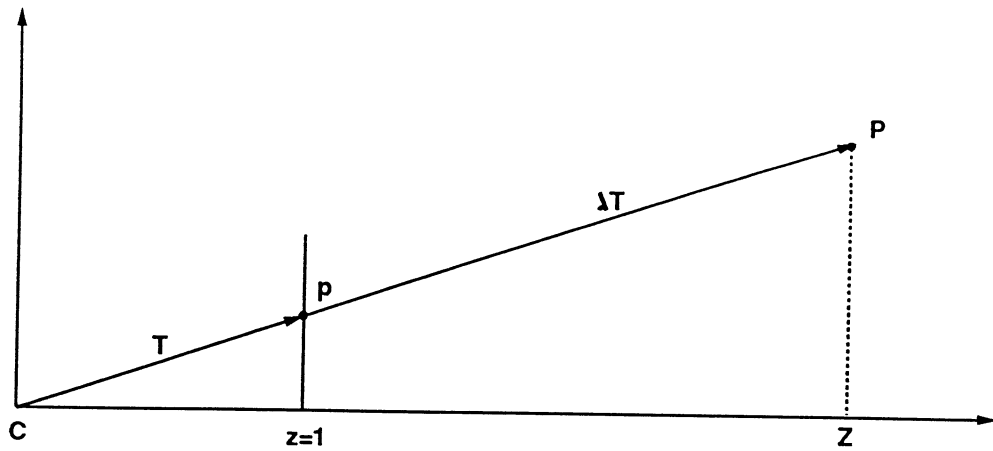


Figure F.2. Comparaison reconstruction/flot optique

On peut alors réécrire l'équation F.2 sous une forme semblable à celle de l'équation de la reconstruction F.3 :

$$Z = -\frac{a \cdot \dot{v}}{E_t + b \cdot \Omega} \quad (\text{F.4})$$

Z et λ sont manifestement égaux par application directe du théorème de Thalès : $(CP) \div (Cp) = (CZ) \div (Cz) = Z = \lambda$.

Les équations F.3 et F.4 mettent alors en évidence que numérateurs et dénominateurs doivent être égaux (ou proportionnels) pour que l'équivalence soit vraie.

Equivalence des numérateurs

Si l'on admet que le long d'un contour, la tangente au contour est orthogonale au gradient de la fonction de luminance de l'image, un vecteur tangent T_s au contour en un point donné sur le plan image peut s'exprimer sous la forme :

$$T_s = \begin{vmatrix} -E_y \\ E_x \\ 0 \end{vmatrix}$$

Le vecteur normal n s'écrit alors comme :

$$T = \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

$$n = T \wedge T_s = \begin{vmatrix} -E_x \\ -E_y \\ xE_x + yE_y \end{vmatrix} = a$$

D'où l'égalité des numérateurs.

En pratique, avec les opérateurs courants de détection de contours, l'assertion *la tangente au contour est orthogonale au gradient de la fonction de luminance de l'image* n'est pas nécessairement vérifiée.

□

Equivalence des dénominateurs

Puisque le repère caméra $\{C\}$ est en mouvement par rapport au repère de référence $\{O\}$, on doit réexprimer oT_t dans son repère local :

$${}^oT_t = {}^cT_t + \Omega \wedge {}^oT$$

Si Ω est :

$$\Omega = \begin{vmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{vmatrix}$$

Alors oT_t vaut :

$${}^oT_t = \begin{vmatrix} x_t + \omega_2 - \omega_3 y \\ y_t + \omega_3 x - \omega_1 \\ \omega_1 y - \omega_2 x \end{vmatrix}$$

d'où :

$${}^oT_t \cdot {}^o n = -E_x(x_t + \omega_2 - \omega_3 y) - E_y(y_t + \omega_3 x - \omega_1) + (xE_x + yE_y) \cdot (\omega_1 y - \omega_2 x)$$

De même en développant $E_t + b \cdot \Omega$:

$$\begin{aligned} E_t + b \cdot \Omega &= -x_t E_x - y_t E_y + b \cdot \Omega \\ &= -E_x(x_t + \omega_2 - \omega_3 y) - E_y(y_t + \omega_3 x - \omega_1) \\ &\quad + (xE_x + yE_y) \cdot (\omega_1 y - \omega_2 x) \\ &= {}^oT_t \cdot {}^o n \end{aligned}$$

□

D'où l'équivalence de la formule F.3 de la reconstruction et de la formule F.4 dérivée du flot optique.

F.2 Conséquences

On peut tirer deux conséquences principales de ce résultat :

1. On pourrait imaginer de reconstruire intégralement les surfaces en suivant les isophotes entre les images successives, au même titre que les lignes de discontinuité. L'instabilité notoire de ces courbes rend caduque cette approche.
2. La formule issue du flot optique est également correcte pour les contours critiques.

Bibliographie

- [Aek 90] F. Van Aeken et Y. Demazeau. Processing elastic patterns the Massively Parallel Way. *IAPR Workshop on machine vision applications*, pages 409–412, Tokyo, Novembre 1990.
- [Alo 88] J. Aloimonos, I. Weiss, et A. Bandyopadhyay. Active Vision. *International Journal of Computer Vision*, pages 333–356, 1988.
- [Arb 89] E. Arbogast et R. Mohr. Reconstruction de surfaces à partir des contours critiques dans un contexte multi-images. Rapport no. RR-94, LIFIA-IMAG, Grenoble, juin 1989.
- [Arb 90] E. Arbogast et R. Mohr. 3D structures inference from images sequences. H.S. Baird, éditeur, *Proceedings of the Syntactical and Structural Pattern Recognition Workshop*, pages 21–37, Murray-Hill, 1990.
- [Asa 86] H. Asada et M. Brady. The Curvature Primal Sketch. *IEEE Transactions on PAMI*, 8(1):2–14, 1986.
- [Aya 87] N. Ayache et O.D. Faugeras. Building, Registrating, and Fusing Noisy Visual Maps. *Proceedings of 1st International Conference on Computer Vision*, pages 73–82, Londres, 1987.
- [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.
- [Ber 90a] M.O. Berger. Snake growing. *Proc. of the first European Conference on Computer Vision*, pages 570–572, avril 1990.
- [Ber 90b] M.O. Berger et R. Mohr. Towards autonomy in active contour models. *Proceedings of the 10th International Conference on Pattern Recognition*, pages 847–851, 1990.

- [Bla 85] A. Blake. Specular Stereo. *Proceedings of 8th International Joint Conference on Artificial Intelligence*, pages 973–976, 1985.
- [Bla 90] A. Blake et R. Cipolla. Robust estimation of surface curvature from deformation of apparent contours. O. Faugeras, éditeur, *Proc. of the first European Conference on Computer Vision*, pages 465–474, Antibes, April 1990. Springer Verlag.
- [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.
- [BQ 89] Su Bu-Qing et Liu Ding-Yuan. *Computational Geometry - Curve and surface Modeling*. Harcourt Brace Jovanovich, Pu., 1989.
- [Bri 89] A.T. Brint et M. Brady. Stereo matching of curves by least deformation. *International workshop on Intelligent Robots and systems*, pages 163–170, Septembre 1989.
- [Bro 86] T.J. Broida et R. Chellappa. Kinematics and structure of a rigid object from a sequence of noisy images. *Workshop on motion: representation and analysis*, pages 95–100, 1986.
- [Can 83] J.F. Canny. Finding Edges and Lines in Images. Rapport, MIT Artificial Intel. Laboratory, Cambridge, MA, 1983.
- [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.
- [Cha 90] F. Chaumette. *La relation vision-commande: théorie et application à des tâches robotiques*. Thèse de Doctorat, Université de Rennes I, juillet 1990.
- [Cip 89] Roberto Cipolla. Active Visual Inference of Qualitative Geometry. Rapport no. OUEL 1795/89, University of Oxford, University of Oxford, Department of engineering science, Park Road, Oxford OX1 3PJ, U.K., July 1989.
- [Coh 82] P. R. Cohen et E.A. Feigenbaum. *The handbook of artificial intelligence*, volume 3. William Kaufmann, Inc, 1982.
- [Cra 86] John J. Craig. *Introduction to robotics. Mechanics and control*. Addison-Wesley, 1986.

- [Cro 88] J.L. Crowley, C. Discours, et P. Stelmaszyk. Measuring Image Flow by Tracking Edges Lines. *Proceedings of 2nd International Conference on Computer Vision*, Tarpon Springs, dec. 1988.
- [Cro 90] J.L. Crowley et P. Stelmazyk. Measurement and Integration of 3D structures by tracking edges lines. O. Faugeras, éditeur, *Proc. of the first European Conference on Computer Vision*, pages 269–280. Springer Verlag, April 1990.
- [Der 87a] 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 87b] R. Deriche et J.P. Cocquerez. Extraction de composantes connexes basée sur une détection optimale des contours. *Actes du congrès MARI-87*, pages 1–9, Tome 2, Paris, 1987.
- [Der 90] R. Deriche et O. Faugeras. Tracking line segments. *Proc. of the first European Conference on Computer Vision*, pages 259–267, Antibes, April 1990. Springer Verlag.
- [Fau 87a] O.D. Faugeras, F. Lustman, et G. Toscani. Motion and structure from point and line matches. *Proc. Int. Conf. Computer Vision*, June 1987.
- [Fau 87b] O.D. Faugeras et G. Toscani. Camera Calibration for 3D Computer Vision. *Proceedings of International Workshop on Machine Vision and Machine Intelligence*, Tokyo, 1987.
- [Fau 89a] O. Faugeras, N. Deriche, et N. Navab. From optical flow of lines to 3D motion and structure. *International Workshop on Intelligent Robots and Systems*, pages 646–649, 1989.
- [Fau 89b] 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, March 1990.
- [Fau 91] O. Faugeras. unknown. DARPA ESPRIT, éditeur, *Applications of invariance in computer vision*, page unknown, Reykjavik, Iceland, March 1991.
- [Fra 90] E. Francois et P. Bouthemy. The derivation of qualitative information in motion analysis. O. Faugeras, éditeur, *First European Conference on Computer Vision*, pages 226 – 230, Antibes, April 1990. INRIA, Springer-Verlag.

- [Gib 87] P. Giblin et R. Weiss. Reconstruction of Surfaces from Profiles. *First Int. Conf. on Computer Vision*, pages 136–144, London, 1987.
- [Gir 87] G. Giraudon. An Efficient Edge Following Algorithm. *Proceedings of 5th Scandinavian Conference on Image Analysis*, pages 547–554, Stockholm, 1987.
- [Gir 88] D. Girard. Détection de discontinuités dans un signal (ou une image) par Inf-Convolution Spline et validation croisée: un algorithme rapide non paramétré. Rapport no. RR 702 I M, IMAG, Grenoble, France, janvier 1988.
- [Har 87] C.G. Harris. Determination of ego-motion from matched points. *3rd Alvey vision Conference*, pages 189–192, 1987.
- [Hen 90] F. Henry. Poursuite de contours en vue d'une reconstruction 3D. Rapport, LIFIA, 1990.
- [Hor 77] B.K.P. Horn. Understanding Image Intensities. *Artificial Intelligence*, 8:201–231, 1977.
- [Hor 80] B. Horn et B. Schunk. Determining Optical Flow. Ai-memo 572, MIT, 1980.
- [Hor 86] B.K.P. Horn. *Robot Vision*. The MIT Press, 1986.
- [Hor 87] R. Horaud. New Methods for Matching 3D objects with Single Perspective Views. *IEEE Transactions on PAMI*, 9(3):401–412, 1987.
- [Jer 90] C. Jerian et R. Jain. Polynomials methods for structure from motion. *IEEE Transactions on PAMI*, 12(12):1150–1165, 1990.
- [Kaa 87] M. Kaas, A. Witkin, et D. Terzopoulos. Snakes: active contours models. *Proceedings of 1st International Conference on Computer Vision*, pages 259–268, june 1987.
- [Kli 88] G.J. Klinker, S.A. Shafer, et T. Kanade. The Measurement of Highlights in Color Images. *International Journal of Computer Vision*, 2(1):7–32, 1988.
- [Koe 84] J.J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.
- [Lau 87] P. J. Laurent. Courbes ouvertes ou fermées par B-splines régularisées. Rapport no. RR 652-M-, IMAG, Grenoble, mars 1987.

- [Lay 90] F. Laymarie. *Tracking and describing deformable objects using active contour models*. Thèse de Doctorat, McGill, McGill Research Center for Intelligent Machines, 3480 University Street, Montreal, Quebec, Canada H3A 2A7, février 1990.
- [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. *IEEE International Conference on Robotics and Automation*, pages 68–75, Raleigh, 1987.
- [LH 80] Longuet-Higgins et Prasadny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London*, pages 385–397, 1980.
- [LH 81] H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. *Nature*, volume 293, pages 133–135. XX, Sept 1981.
- [Liu 86] Y. Liu et T.S. Huang. Estimation of rigid body motion using straight line correspondences, further results. *Proceedings of 8th International Conference on Pattern Recognition*, pages 306–307, octobre 1986.
- [Low 88] D.G. Lowe. Organization of Smooth Image Curves. *Proceedings of 2nd International Conference on Computer Vision*, pages 558–567, Tampa, 1988.
- [Mal 87] J. Malik. Interpreting Line Drawings of Curved Objects. *International Journal of Computer Vision*, 1:73–103, 1987.
- [Mod 77] J.W. Modestino et R.W. Fries. Edge detection in noisy images using recursive digital filtering. *Comput. Graphics image processing*, 6:409–433, 1977.
- [Moh 90] R. Mohr et E. Arbogast. It can be done without camera calibration. *Pattern Recognition Letters*, 12:39–43, 1990.
- [Moh 91a] R. Mohr et L. Morin. Relative positioning from geometric invariants. *Proceedings of the Conf. on Computer Vision and Pattern Recognition*, June 1991.
- [Moh 91b] R. Mohr, L. Morin, et E. Grosso. Relative Positioning with poorly calibrated cameras. *Applications of Invariance in Computer Vision*, Reykjavik - Iceland, 25 - 28 March 1991. DARPA - ESPRIT Workshop.

- [Nag 81] H.H. Nagel et B. Neumann. On 3-d reconstruction from two perspective views. *Proc. of the 7th Joint Conference an Artificial Intelligence*, pages 661–663, 1981.
- [Nag 82] H. Nagel. Displacement Vectors from the Second Order Variations in Image Sequences. *Computer Vision, Graphics and Image Processing*, 21:85–117, 1982.
- [Neg 87] Negahdaripour et Horn. Direct passive navigation. *IEEE Transactions on PAMI*, 1(9):168–176, janvier 1987.
- [Pra 80] K. Prazdny. Egomotion and relative depth map from optical flow. *Biological cybernetics*, 36:87–102, 1980.
- [Pre 88] W.H. Press, B.P. Flannery, S.A. Teukolsky, et W.T. Vetterling W.T. *Numerical recipes in C*. Press syndicate of the university of Cambridge, 1988.
- [Rob 63] L.G. Roberts. Machine perception of three-dimensionnal solids. *Optical and Electro-optical Information Processing*, pages 159–197. MIT press, Cambridge, MA, 1963.
- [Sac 87] O. Sacks. *The man who mistook his wife for a hat, and other clinical tales*. Harper and Row, 1987.
- [She 86] J. Shen et S. Castan. An optimal linear operator for edge detection. *Proc. of the conf. on Computer Vision and Pattern Recognition*, Miami, 1986.
- [SM 89] P. Saint-Marc, J.S. Chen, et G. Medioni. Adaptative smoothing: a general tool for early vision. *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 618–624, San Diego, California, 1989.
- [SM 90] P. Saint-Marc et G. Medioni. B-spline contour representation and symmetry detection. *Computer vision - ECCV 90*, pages 604–606. Institute for Robotics and Intelligent Systems, USC, Ca, Springer Verlag, 1990.
- [Sub 86] M. Subbarao. Interpretation of image motion fields: a spatio-temporal approach. *Workshop on motion: representation and analysis*, pages 157–165, mai 1986.
- [Tos 87] G. Toscani et O.D. Faugeras. Mouvement par reconstruction et reprojection. *11ème Colloque sur le Traitement du signal et des images (GRETSI)*, pages 535–538, Nice, 1987.

- [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*, volume 3, pages 323–344. IEEE, 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 90] R. Vaillant. *Géométrie différentielle et vision par ordinateur : détection et reconstruction des contours d'occultation de surfaces*. Thèse de Doctorat, Univ. de Paris-Sud, Dec. 1990.
- [Vi 90] T. Viéville et O. Faugeras. Feed-forward recovery of motion and structure from a sequence of 2D-lines matches. *Proceedings of Third International Conference on Computer Vision*, pages 517–520, decembre 1990.
- [Voo 87] H. Voorhees et T. Poggio. Detecting texon and texture boundaries in natural images. *Proceedings of 1st International Conference on Computer Vision*, pages 250–258, June 1987.
- [Wax 87] A.M. Waxman, B. Kamgar-Parsi, et M. Subbarao. Closed-form solutions to image flow equations for 3D structure and motion. *International Journal of Computer Vision*, volume 1, pages 239–258. Kluwer academic Publishers, 1987.
- [Wen 89] J. Weng, T.S. Huang, et N. Ahuja. Motion and structure from two perspective views: algorithms, error analysis and error estimation. *IEEE Transactions on PAMI*, 11(5):451–476, may 1989.
- [Xie 89] M. Xie. *Contribution à la vision dynamique : reconstruction d'objets 3D polyédriques par une caméra mobile*. Thèse de Doctorat, Université de Rennes 1, Juin 1989.

Liste des Figures

2.1	Schéma d'une caméra utilisée en vision par ordinateur	6
2.2	Modèle sténopé, ou projection perspective pure	7
2.3	Repère local en pixels du plan image	8
2.4	Déformation radiale	8
2.5	Dispositif expérimental pour la calibration	10
2.6	Exemple d'image	12
2.7	Gradient de l'image	13
2.8	Chaînes de points de contour	14
2.9	courbe de discontinuité : exemple	15
2.10	Courbes critiques : exemples	15
2.11	Une chaîne de points de contour représentant un segment de droite	16
2.12	B_splines régularisées pour différentes valeurs de τ	21
2.13	Lissage de la souris pour différentes valeurs de τ	22
2.14	Contours de discontinuité et contours critiques sont fusionnés	24
2.15	Contours de discontinuité (en pointillés)	24
2.16	Catalogue des jonctions	25
2.17	Discontinuité de courbure	28
2.18	discontinuités de tangence	29
2.19	Segmentation de l'image de la souris	30
2.20	Discontinuités de courbure dans l'image de la souris	31
2.21	Segmentation dans une scène polyédrique	31
2.22	Découpage récursif : problème de segmentation dans les coins	32
2.23	Segmentation dans une scène mixte	32
3.1	Contours perçus par la caméra en mouvement	40
3.2	Formation d'une surface spatio-temporelle dans le repère caméra	40
3.3	Surface spatio-temporelle: passage au continu	41
3.4	Les repères - les notations	42
3.5	Propriété des courbes critiques	44
3.6	Paramétrisation de la surface spatio-temporelle	47
3.7	Deux paramétrisations de la surface spatio-temporelle	49
3.8	Géométrie épipolaire	53
3.9	Séquence du segment dans l'espace	56

3.10	Rétroprojection du segment reconstruit	61
3.11	Mouvement de la caméra pas rapport à la sphère	62
3.12	Séquence de la sphère dans l'espace	62
3.13	Séquence de la sphère dans l'espace	63
3.14	Séquence du dessin plat	64
3.15	Reconstruction d'un contour de la séquence du dessin plat .	65
3.16	Vue latérale de la reconstruction de la séquence du dessin plat	65
3.17	Vue générale de la reconstruction de la séquence du dessin plat	66
3.18	Dimensions du dessin	66
3.19	Séquence de l'ampoule électrique	68
3.20	Reconstruction de la séquence de l'ampoule électrique	69
3.21	Vue latérale de la reconstruction de la séquence de l'ampoule	70
3.22	Vue générale de la reconstruction de la séquence de l'ampoule	71
3.23	Séquence du verre	73
3.24	Reconstruction d'un contour de la séquence du verre	74
3.25	Vue latérale de la reconstruction de la séquence du verre . .	75
3.26	Vue générale de la reconstruction de la séquence du verre . .	76
4.1	Courbe à s =constante dans le cas d'un contour critique . . .	83
4.2	Courbe à s =constante sur la courbe de discontinuité (Y) . .	84
4.3	Les repères et leurs relations	85
4.4	Les repères - les notations	87
4.5	Situation de l'ellipse par rapport à la caméra	94
4.6	Séquence de l'arc d'ellipse pour $v = 0.4m/s$	95
4.7	Séquence de l'arc d'ellipse pour $v = 0.2 m/s$ et $\Omega = 0.05 rad/s$	97
4.8	Recouvrement pour $v = 0.2 m/s$ et $\Omega = 0.05 rad/s$	98
4.9	Séquence de la poire	101
4.10	Séquence du masque	102
A.1	Fonctions de base pour les B_splines a) linéaires b) cubiques	108
D.1	Repère de Frénet	122
D.2	Courbure normale	124
D.3	Courbures principales pour les classes de surface	126
E.1	Les repères et leurs relations	127
F.1	Comparaison reconstruction/flot optique	130
F.2	Comparaison reconstruction/flot optique	131

Liste des Tableaux

2.1	Courbure en fonction du facteur de tension	23
2.2	Courbure en fonction du bruit pour splines simples et régularisées	23
3.1	Erreurs en Z de reconstruction du segment	55
3.2	Valeurs des rayons pour la sphère reconstruite	58
3.3	Influence du pas de discrétisation temporel	58
3.4	Valeurs des coordonnées en Z pour le dessin plat	59
3.5	Valeurs de H et L pour le dessin plat	60
4.1	Valeurs de la vitesse calculée	95
4.2	Valeurs pour une correspondance orthogonale	96
4.3	Valeurs de la vitesse de rotation calculée	97
4.4	Valeurs de la vitesse de rotation pour une correspondance orthogonale	98
4.5	Influence du bruit sur l'autocalibration	99
4.6	Stabilité des paramètres du mouvement après 10 images . .	100
4.7	Valeurs calculées par rapport aux valeurs connues	100
4.8	Valeurs calculées par rapport aux valeurs connues	103

Résumé

Nous proposons une méthodologie de travail dans le cadre d'une vision monoculaire en mouvement pour des scènes non nécessairement polyédriques.

Nous montrons en particulier l'intérêt de la géométrie différentielle et de la surface spatio-temporelle dans le problème de reconstruction de surfaces à partir d'images de contours et dans celui de l'estimation du mouvement propre de la caméra à partir de l'observation de contours rigides.

La technique de reconstruction est présentée, qui unifie la reconstruction des lignes courbes et des surfaces à partir du mouvement. Observant un contour dans une séquence d'images, il nous est possible de reconstruire soit la courbe dans l'espace correspondant au contour tridimensionnel si ce dernier est une courbe fixe de l'espace, soit la surface sur laquelle s'appuie le contour observé dans le cas d'un contour d'occultation, frontière d'une surface.

Cette technique nécessite une connaissance précise du mouvement de la caméra. Cette information peut elle même être déterminée par l'observation de lignes courbes inconnues de la scène.

Ces méthodes différentielles nécessitent un excellent lissage des données, tout en ayant une précision de type sub-pixel sur la position des contours. Les B-splines régularisées constituent notre réponse à ce problème : elles offrent des représentations qui autorisent des segmentations et des dérivations stables, tout en faisant une approximation aux moindres carrés des données observées.

Des exemples synthétiques et réels illustrent la faisabilité de ces techniques.