



AVERTISSEMENT

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

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

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

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

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

http://www.cfcopies.com/V2/leg/leg_droi.php

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

[M] 1993 CHABBI, H.

Construction de facettes 3D par stéréovision intégrant des principes de géométrie projective

THÈSE

présentée le 17 février 1993

pour l'obtention du

Doctorat de l'Institut National Polytechnique de Lorraine
(Spécialité Informatique)

par

Houda CHABBI



Composition du jury :

Président : Roger MOHR

Rapporteurs : Serge CASTAN
Jean-Paul HATON
Marc RICHTIN

Examineurs : Yves DEMAZEAU
Gérald MASINI

Remerciements

Je tiens tout d'abord à exprimer mes remerciements aux membres de mon jury pour l'honneur qu'ils me font :

Monsieur Roger Mohr, Professeur à l'Institut Polytechnique de Grenoble, d'avoir présidé ce jury et pour les différentes discussions que nous avons eu au sujet de mon travail,

Monsieur Serge Castan, Professeur à l'université Paul-Sabatier de Toulouse, Monsieur Marc Richetin, Professeur à l'université Blaise Pascal de Clermont-Ferrand et Monsieur Jean-Paul Haton, Professeur de l'université de Nancy I, pour avoir accepté d'être les rapporteurs de cette thèse,

Monsieur Yves Demazeau, chargé de recherche CNRS au Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle à Grenoble pour avoir accepté d'examiner mon mémoire et pour l'intérêt qu'il a porté à ce travail,

Monsieur Gérard Masini, chargé de recherche CNRS au Centre de Recherche en Informatique de Nancy et directeur de mes travaux, pour le constant intérêt qu'il a porté à ce travail.

Je suis heureuse aujourd'hui de remercier tous ceux qui m'ont conseillé et aidé à réaliser ce travail dans de bonnes conditions et plus précisément , j'exprime mes remerciements chaleureux à Stéphane Paris et à Marie-Odile Berger de l'équipe de Vision du CRIN à qui je dois beaucoup pour les nombreuses et fructueuses discussions que nous avons eues ensemble au sujet de mon travail.

Je tiens également à exprimer mes remerciements à Brigitte Wrobel-Dautcourt pour avoir relu ma thèse avec autant de soin.

Enfin, merci de tout cœur à ma famille et à tous mes amis en particulier Anja Habacha, Julian Anigbogu, Sylvie Machicote, Brigitte Piquet et tous les autres pour leur sympathie et leur soutien amical.

Résumé

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la vision artificielle et l'objectif recherché est de développer un processus de reconstruction stéréoscopique des facettes 3D des scènes observées.

Le principe fondamental que nous avons suivi pour réaliser la construction de facettes 3D a été d'éviter dans la mesure du possible, tout calcul direct sur les segments 3D. Nous avons alors proposé un protocole d'utilisation de l'information région et l'information contour pour construire des indices 2D : *facettes 2D* par l'intermédiaire desquels nous inférons l'existence d'un plan 3D. Utilisant ensuite des techniques de géométrie projective, nous vérifions en 2D la validité de l'hypothèse de planarité relative à chacun de ces indices. La recherche des paramètres de chacune des facettes 3D de la scène utilise alors toute l'information acquise en 2D concernant les propriétés 3D à vérifier par ces facettes à construire. Cette reconstruction permet ainsi de maintenir la propriété de planarité ou encore de coplanarité entre facettes améliorant la localisation 3D de ces primitives.

Table des matières

1	Introduction	5
2	Le constructeur de facettes 3D : son contexte et ses principes	9
2.1	Acquisition des données : les capteurs	9
2.2	Bas niveau : extraction des indices	11
2.3	Niveau intermédiaire : recherche d'indices "évoluées"	12
2.4	Haut niveau	15
2.5	Les principes du constructeur de facettes 3D	16
2.5.1	Quelle méthodologie pour la reconstruction?	16
2.5.2	Quelles hypothèses 2D engendrer ?	21
2.6	Architecture du constructeur de facettes	23
3	La construction de facettes 2D	27
3.1	Quelques définitions	27
3.1.1	Définition d'une facette 2D	28
3.1.2	A propos des segments	29
3.1.3	La correspondance région-segment	29
3.1.4	La correspondance région-facette	30
3.2	Module d'initialisation	31
3.2.1	Prétraitement sur les segments	31
3.2.2	Construction des premières facettes incertaines	31
3.3	Module de construction de polygones 2D	32
3.3.1	Fonction de recherche d'un segment candidat	33
3.3.2	Phase-1 : étude des facettes incertaines	34
3.3.3	Phase-2 : recherche des facettes à partir des régions	38
3.4	Le module de contrôle de la demande des cartes régions	40
3.4.1	Présentation de l'outil de segmentation en régions	41
3.4.2	Heuristique adoptée pour la gestion des demandes de cartes régions	42
3.4.3	Étape-1 : Calculs des contrastes pertinents d'un segment	45
3.4.4	Étape-2 : Calculs des seuils pertinents pour la segmentation	50
3.5	Résultats expérimentaux	51
3.6	Discussion	56
3.6.1	Introduction de l'information point de fuite	56
3.6.2	Introduction des résultats d'un détecteur de coins	59
3.6.3	Autres travaux sur la reconstruction de facettes 2D	60
3.7	Conclusion	61

4	Mise en correspondance des facettes 2D	63
4.1	La mise en correspondance en stéréovision	63
4.1.1	Les primitives images à apparier	64
4.1.2	Les contraintes en stéréovision	65
4.1.3	Les techniques de mise en correspondance	67
4.2	Appariement de facettes 2D	69
4.2.1	Les contraintes de mise en correspondance de facettes	69
4.2.2	L'algorithme de mise en correspondance	71
4.3	Résultats expérimentaux	74
4.4	Discussion	76
5	Géométrie projective et test de coplanarité	79
5.1	Utilisation des invariants projectifs en vision	80
5.1.1	Pourquoi utiliser les invariants projectifs?	80
5.1.2	Et dans notre cas?	81
5.2	Rappels sur la géométrie projective	82
5.2.1	Espace projectif, coordonnées homogènes et homographies	82
5.2.2	L'invariant de base de la géométrie projective: le birapport	83
5.2.3	Base projective	85
5.2.4	Coordonnées projectives	85
5.3	Système de bases projectives en correspondance	87
5.3.1	Notion de plan de référence	88
5.3.2	La transformation $T_{i \rightarrow j}^{\Pi}$	88
5.4	Trace d'une droite 3D sur un plan de référence	91
5.4.1	Définition de la trace d'une droite	91
5.4.2	Construction de la trace d'une droite	91
5.5	Contrainte de coplanarité d'une droite et d'un plan de référence	94
5.6	Caractérisation de la coplanarité de points 3D	95
5.6.1	Trace d'un ensemble de points 3D	95
5.6.2	Contrainte de coplanarité de points 3D	96
5.6.3	Un algorithme de recherche de coplanarité un ensemble de points	97
5.6.4	Comment tester la coplanarité de deux ensembles de points?	98
5.7	A propos d'une contrainte de coplanarité d'un ensemble de droites	99
5.7.1	Et si on prenait plus d'un plan de référence?	100
5.7.2	Un algorithme de recherche de coplanarité d'un ensemble de droites	102
5.7.3	Une méthode de construction des plans de références	103
5.8	Autre approche pour détecter la coplanarité de points	104
5.8.1	Utilisation de la géométrie épipolaire	105
5.8.2	Discussion	105
5.9	En résumé	106
6	Utilisation du test de coplanarité	109
6.1	Processus composant l'étape de prétraitement	109
6.2	Construction des systèmes de bases projectives	110
6.3	A propos des erreurs sur les traces	112
6.3.1	Origine et interprétation de l'erreur	112
6.3.2	La prise en compte des erreurs	114

6.3.3	Construction de l'image d'un isobarycentre	115
6.4	Processus de vérification de la planarité	117
6.4.1	Test de planarité	119
6.4.2	Traces caractéristiques	119
6.5	Processus de recherche des facettes coplanaires	120
6.6	D'autres propriétés exploitables en 2D	126
6.6.1	Position relative d'une primitive par rapport à un plan	126
6.6.2	Applications utiles de la propriété de positionnement	127
6.7	Reconstruction de facettes 3D	129
6.7.1	Etape de recherche du plan support d'une facette	129
6.7.2	Etape de recherche des bords d'une facette	130
6.8	Résultats expérimentaux	130
6.8.1	Trace de segments parallèles au plan de référence	131
6.8.2	Résultats du processus du calcul de la trace	132
6.8.3	Résultats de reconstruction 3D	135
6.9	Discussion	139
7	Conclusion	143
7.1	Bilan	143
7.2	Perspectives	145
7.3	Conclusion générale	149
A	Algorithme de recherche des contrastes pertinents pour un segment	151
B	Calculs des coordonnées	153
B.1	Coordonnées d'un point sur une droite	153
B.2	Coordonnées d'un point dans un plan	153
B.2.1	Coordonnées projectives à partir des coordonnées cartésiennes	153
B.2.2	Coordonnées cartésiennes à partir des coordonnées projectives d'un point	154
C	Construction de la droite à l'infini d'un plan	155
D	Autres résultats du constructeur de facettes	157
	Bibliographie	159
	Index	167

1

Introduction

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la vision artificielle et l'objectif recherché est de développer un processus de reconstruction des surfaces planes appelées facettes 3D qui composent la scène observée et ce à partir d'images stéréoscopiques. Ces images sont délivrées par le robot mobile de l'Inria, qui se déplace dans des scènes d'intérieur.

Il est généralement admis qu'un système de vision dispose de différents modules qui extraient des informations pertinentes : *indices*, des images décrivant de la scène observée. Ces informations sont ensuite mises en correspondance avec des modèles des objets enregistrés par le système auparavant. Ces modèles représentent l'ensemble des connaissances que le robot possède a priori sur son environnement et des connaissances qu'il acquiert au fur et à mesure de son fonctionnement. Ils lui permettent de réaliser deux tâches qui sont l'identification et la localisation des objets de la scène.

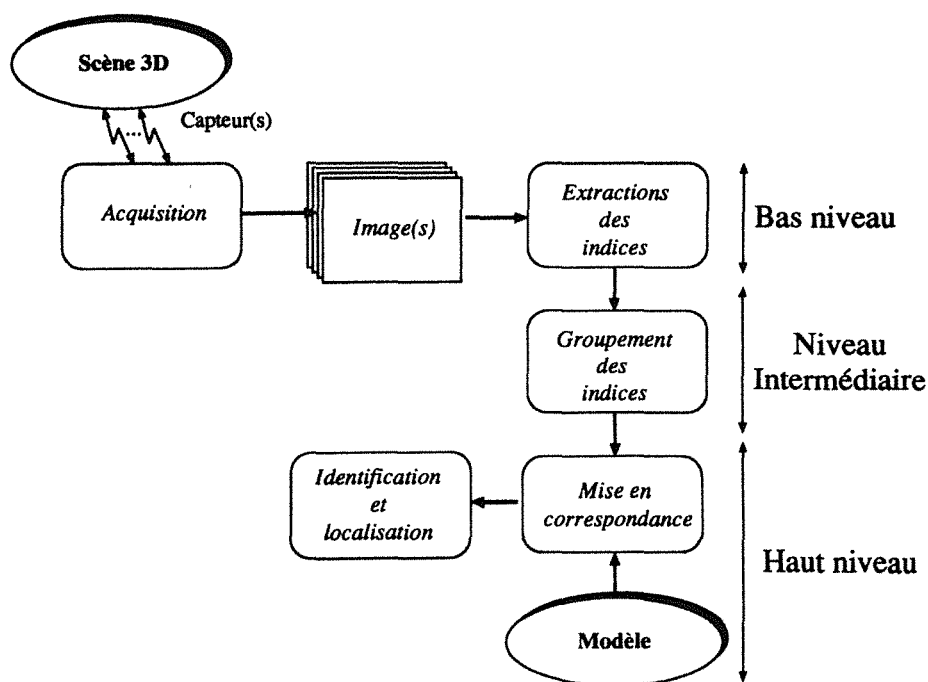


Figure 1.1. Architecture simplifiée d'un système de vision artificielle.

Les différents modules composant un système de vision peuvent être décomposés en trois catégories suivant les tâches accomplies. Dans la Figure 1.1 nous présentons, à titre indicatif, une architecture très simplifiée d'un système de vision mettant en évidence les différents niveaux qui le composent. Ce système est conçu sous forme d'une chaîne complète de vision et doit être capable d'extraire des informations 3D, de construire automatiquement les modèles des objets et de l'environnement perçus et de mettre en correspondance les informations extraites avec une partie de ces modèles. Dans notre équipe, nous nous intéressons au développement d'un tel système de vision, le système MAGIC¹, pour un robot mobile muni d'un banc stéréoscopique trinoculaire et se déplaçant en intérieur. Dans un premier temps, les scènes traitées sont supposées être essentiellement composées d'objets polyédriques. Cette supposition permet de considérer que la description des objets peut être faite à partir d'assemblage de primitives tridimensionnelles : *polygones plans (appelés facettes) et segments*.

Les indices 2D élémentaires délivrés par le bas niveau tels *les contours* ou *les régions* à partir des images à niveau de gris fournies par les caméras CCD du robot ont l'inconvénient d'être hétérogènes et bien que complémentaires, d'être extraits indépendamment les uns des autres. C'est pourquoi le niveau intermédiaire de ce système de vision est orienté vers une recherche de structure de l'environnement fondée sur une utilisation combinée des différents indices délivrés par le bas niveau. Le constructeur de facettes que nous avons développé se situe dans ce niveau du système. Dans la version actuelle de ce constructeur de facettes nous avons envisagé une utilisation combinée des informations régions et segments pour aboutir à la construction des facettes 3D. Présentons brièvement les motivations de ce choix et signalons que la justification complète de tous nos choix est donnée dans le prochain chapitre (Section 2.5). En théorie, les concepts de contours extraits d'une image correspondent à l'information duale des régions. Cependant, en pratique et du fait du bruit inhérents aux images nous remarquons que d'une part cette dualité théorique entre les résultats délivrés par les techniques de segmentation en contour et en régions n'est pas maintenue et que d'autre part ces deux types d'outils ne présentent pas la même stabilité. En effet, si les détecteurs de contours présentent une segmentation presque équivalente entre deux images stéréoscopiques les outils de segmentation en régions, quant à eux, ne présentent pas cette même qualité, fragilisant ainsi les systèmes stéréoscopiques fondée sur l'information région [WD88]. Pourtant, en dépit de ses défauts nous n'avons pas complètement délaissé l'information région au profit de l'information contours car c'est elle qui se rapproche le plus de la notion de facettes. Et afin de combler les imperfections de l'une avec les points forts de l'autre, nous avons proposé un protocole d'utilisation combinée de l'information région et de sa duale théorique : l'information contour. Cette utilisation combinée des informations extraites par le bas niveau : les segments et les régions, nous permet de construire des indices 2D : *facettes 2D* par l'intermédiaire desquels nous inférons l'existence d'un plan 3D. Utilisant ensuite des techniques de géométrie projective, nous vérifions la validité de l'hypothèse de planarité relative à chacun de ces indices. La recherche des paramètres de chacune des facettes 3D de la scène, est réalisée après l'étude des images 2D, et est effectuée en fin de traitement. Cette reconstruction 3D est fondée sur l'utilisation de toute l'information acquise en 2D concernant les propriétés

1. Un système Multi-AGents pour l'Interprétation et la Compréhension de scènes.

3D vérifiées par les facettes à construire : comme par exemple la propriété de planarité ou encore la coplanarité d'un ensemble de facettes 3D à reconstruire.

La partie haut niveau du système est censée être munie de modèles géométriques des objets. Pour ce faire, une phase d'apprentissage doit permettre de construire automatiquement ces modèles à partir de séquences d'images saisissant chaque objet sous tous ses angles. Les facettes et segments 3D extraits de chaque triplet d'une séquence sont recoupés pour former, d'une part, la description de la structure de l'objet correspondant et, d'autre part, les différents aspects 3D stables sous lesquels l'objet peut être vu. Ces derniers servent à construire le graphe d'indexation [Par91] qui permet d'accéder au plus vite à la description d'un objet à partir d'un ensemble de groupements perceptuels qui le caractérisent et qui peuvent être extraits à faible coût.

Donnons à présent le plan de ce mémoire. Mise à part l'introduction générale ce mémoire comporte sept chapitres et quatre annexes sont dédiés à présenter des techniques de calculs utiles à l'implantation ainsi que des résultats d'exécution. Voici brièvement les centres d'intérêts de chacun de ces chapitres :

- *le Chapitre 2* comporte deux parties. L'une est réservée aux définitions du vocabulaire et des notions utilisées dans ce mémoire, l'autre situe le cadre de notre travail et décrit les principes généraux utilisés pour la construction des facettes 3D.
- *le Chapitre 3* est dédié à la présentation de la construction de la structure facette 2D dans une image, fondée sur l'information région et contour. Des résultats et une discussion terminent ce chapitre.
- *le Chapitre 4* présente le module d'appariement de la structure facette 2D entre les images d'un triplet stéréoscopique. La méthode mise en jeu s'inspire de l'algorithme de stéréovision trinoculaire présenté par Lustman et Ayache [AL87].
- *le Chapitre 5* est consacré à une étude théorique fondée sur la géométrie projective, concernant l'établissement d'une contrainte de coplanarité liée à un ensemble de points ou de droites homologues. Cette étude permet de dégager une méthode de validation en 2D de la planarité des facettes à reconstruire en 3D.
- les résultats introduits au chapitre précédent sont appliqués dans *le Chapitre 6*. Nous y présentons également un algorithme concernant la détection des facettes coplanaires à partir de la donnée des triplets de facettes homologues. Le processus de calcul des paramètres 3D des facettes est détaillée en fin de chapitre.
- dans *le Chapitre 7* nous concluons et présentons les perspectives de notre travail.

2

Le constructeur de facettes 3D : son contexte et ses principes

Avant de présenter en détail le cadre du travail ainsi que l'approche adoptée pour réaliser un extracteur de surfaces planes 3D, nous commençons par présenter différentes définitions de notions relatives à la vision artificielle et utilisées dans ce manuscrit. S'inspirant de l'architecture présentée dans la Figure 1.1 ; nous réservons à chacun des trois niveaux composant un système de vision artificielle une section explicative. Ayant introduit la terminologie ainsi que le cadre de travail, nous passons dans la section 2.5 à la présentation de l'objectif de cette thèse et des principes que nous avons adopté pour réaliser notre travail. Pour ceci nous commençons par présenter une étude bibliographique sur les travaux relatifs à la recherche de plans 3D, puis nous introduisons notre propre approche.

2.1 Acquisition des données : les capteurs

La vision artificielle exige l'utilisation de capteurs visuels spécifiques, faisant partie de la machinerie du robot. Généralement, les capteurs visuels sont classés en deux groupes suivant le type d'interaction qu'ils ont avec leur environnement : nous avons ainsi les capteurs actifs et les capteurs passifs :

- Les capteurs actifs sont de type laser, ultrason ou tactiles par exemple. Ce sont des capteurs qui disposent d'un émetteur et d'un récepteur et qui acquièrent une information spatiale. Il existe deux techniques d'analyse : la triangulation et le temps de vol [Jar83]. Dans le cas de la triangulation le principe se fonde sur la connaissance de la distance H entre le récepteur et l'émetteur. La distance D du point qui réfracte l'onde est alors directement déduite de la direction du signal émis et réfracté. Dans le cas de du temps de vol (respectivement du décalage de phase) la mesure du temps mis entre l'émission et la réception du signal (respectivement du déphasage entre la lumière émise et réfléchi) permet de calculer la distance du point observé. Dans tous les cas de figures, l'information acquise par un capteur actif de la scène se présente sous forme d'une carte de distances. Aucune information de type couleur ou texture des surfaces observées n'est présente dans cette représentation.

- Les capteurs passifs sont des caméras CCD entre autres. Dans le cas d'une caméra, les données acquises sur l'environnement sont sous forme d'une matrice 2D, appelée *image numérique*, qui renferme une information de luminance (niveau de gris ou couleur) émise ou réfléchi par l'environnement. L'information contenue dans chaque élément de cette matrice, appelé *pixel*, est ambiguë car elle dépend de la lumière, de l'orientation et de la distance des objets de la scène à la caméra. Ainsi, à l'encontre des données acquises avec des capteurs actifs, la notion de profondeur ne ressort pas immédiatement d'une image.

Pour notre part, nous nous intéressons exclusivement à des systèmes de vision munis de caméras. Différentes dénominations désignent un système de vision passif suivant le nombre de caméras dont il est pourvu. Lorsque le système de vision comporte une seule caméra on parle de *vision monoculaire*. Alors que le terme de *système de stéréovision* désigne les systèmes constitués de plus d'une caméra. Dans notre cas nous disposons d'un système de vision à trois caméras dit *système de stéréovision trinoculaire*.

Détaillons à présent le modèle géométrique d'une caméra [BB82] [Fau88]. Le modèle le plus souvent adopté est celui dit du *sténopé* (ou *pinhole* en anglais) [Tos87]. Ce modèle décrit une caméra comme étant formée par un *centre optique*¹ O appartenant à un *axe optique* et par un *plan image* Im perpendiculaire à l'axe optique, situé à une distance f , appelée *la distance focale*, du centre optique. L'image p d'un point de l'espace P est le résultat d'une projection perspective parfaite de centre O sur le plan Im (Figure 2.1). Une telle projection s'inscrit dans le cadre des transformations projectives ce qui permet d'appliquer plusieurs résultats de la géométrie projective pour résoudre certains problèmes de vision [MA90]; nous y reviendrons par la suite.

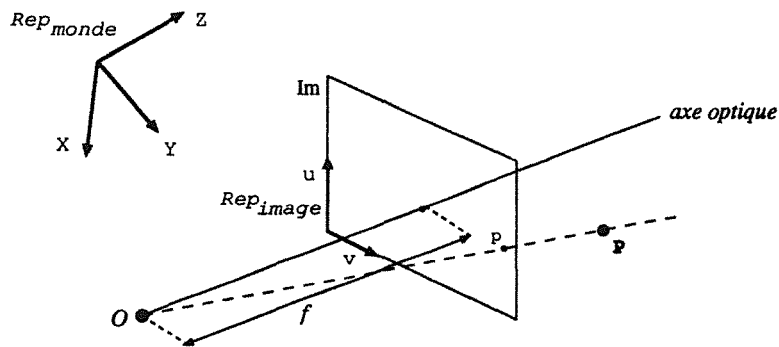


Figure 2.1. Modèle à sténopé

Dans la Figure 2.1 nous adoptons les notations suivantes :

1. Rep_{monde} : repère absolu dans la scène. Les coordonnées de tous les points des objets 3D sont exprimées dans ce repère.
2. Rep_{image} : repère lié au plan image. Tous les points images sont exprimés dans ce repère.

1. Le terme *point principal* peut également être utilisé.

La transformation qui lie les coordonnées d'un point $P = (X, Y, Z)_{Rep_{monde}}$ à son projeté $p = (u, v)_{Rep_{image}}$ s'écrit sous la forme suivante :

$$\begin{cases} u = \frac{a_{11}X + a_{12}Y + a_{13}Z + a_{14}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \\ v = \frac{a_{21}X + a_{22}Y + a_{23}Z + a_{24}}{a_{31}X + a_{32}Y + a_{33}Z + a_{34}} \end{cases}$$

Les coefficients a_{ij} ne sont autres que les coefficients de la matrice M 4×3 : la *matrice de transformation perspective* associée à la caméra.

$$M = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

La détermination de la valeurs des coefficients de la matrice M constitue la phase de *calibrage* [Tos87].

Remarquons que la transformation perspective n'est pas bijective. En effet, la *projection inverse* associée à tout point p du plan image l'ensemble des points de la demi-droite (pP). Ceci explique l'ambiguïté interdisant de retrouver directement l'information tridimensionnelle à partir d'une seule image.

2.2 Bas niveau : extraction des indices

Les images acquises par les capteurs passifs ne sont pas exploitables à l'état brut pour la compréhension de la scène. Il s'agit donc de procéder à l'extraction d'indices pertinents. Les premiers indices² sont extraits à partir de l'étude du signal qui a provoqué la formation de l'image : c'est la phase de segmentation. Deux types de segmentations sont souvent envisagées :

- *la détection de contours*. Il s'agit de retrouver les points de discontinuité de l'intensité lumineuse, qui correspondent très souvent à une information pertinente. En effet, les points de contours correspondent à différents phénomènes physiques. Certains proviennent des arêtes des objets (ou des bords tangentiels pour les surfaces courbes), d'autres sont produits par les limites d'ombres ou de reflets d'une surface.

Différents outils ont été développés pour extraire ces points. Dans notre cas nous avons utilisé le détecteur de Deriche [Der87] (intégré dans Visilog) ainsi que celui de Castan *et al.* [CSZ91] : ISEF. Comme le segment est la structure de base utilisées pour décrire les scènes polyédriques, les points de contours obtenus sont chaînés entre eux et approximés par des segments de droites [Gir87].

- *la segmentation en régions*. L'objectif est de partitionner topologiquement l'image en zones vérifiant des critères données d'homogénéité. Il s'agit souvent de rechercher les zones de luminosité homogène [WD88]. C'est un travail dual à celui de l'extraction des points de discontinuités. Chaque zone extraite représente une

2. Egalement appelés primitives.

région. Bien que l'idée sous-jacente à cette segmentation soit de retrouver les surfaces des objets de la scène, la segmentation en régions est loin de répondre à cet objectif. Dus à l'instabilité des outils de segmentation, des problèmes de choix de paramètres adéquats sont toujours présents.

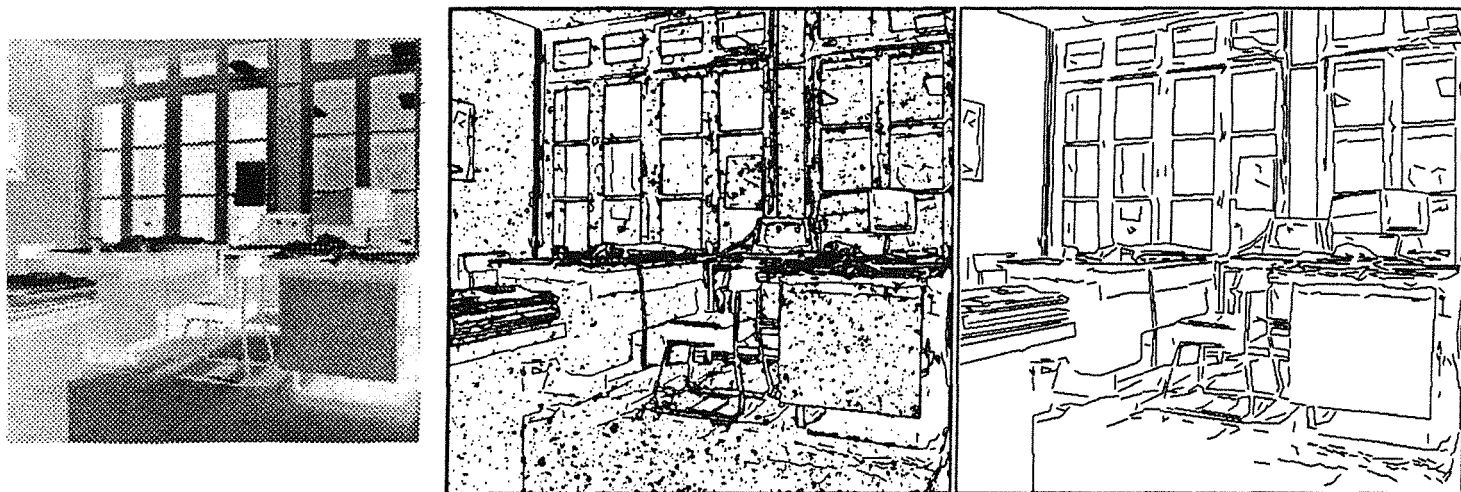


Figure 2.2. Exemple d'extraction d'indices. A gauche : l'image en niveau de gris. Au milieu : les contours des régions. A droite : les segments.

D'autres caractéristiques peuvent également être extraites d'une image, comme par exemple la texture. Dans le cas où l'image observée est celle d'un tas de grains de café, par exemple, ni l'extraction de régions ni celle des contours n'est adéquate pour faire ressortir des indices pertinents (le grain de café dans l'exemple) permettant de décrire l'image. La description de l'image se fait au moyen de la détermination du modèle du motifs répétitif de l'image dit *texel* [BB82].

2.3 Niveau intermédiaire : recherche d'indices "évolués"

Tous les traitements qui sont effectués par le *bas niveau* visent à faciliter la tâche de perception du robot. Mais chaque indice pris à part n'est porteur que d'une partie de l'information significative, car trop ponctuelle, pour être exploitable. Il est donc primordial de développer des processus qui à partir de ces indices pauvres et isolés en construisent d'autres plus évolués renfermant une information globale et pertinente. Ce sont des processus de ce type qui forment le *niveau intermédiaire*.

Dans sa théorie [Mar82], Marr propose de bien séparer la phase d'extraction des indices de l'image, de celle du calcul des données tri-dimensionnelles relatives à la scène. Il s'agit de la représentation *2,5 sketch* centrée sur l'observateur. Plusieurs processus ont été développés pour réaliser cette tâche. Certains opèrent sur une image (monoculaire) alors que d'autres opèrent sur plus d'une image (stéréovision ou mouvement).

Notons que les méthodes décrites par la suite et indicées par (A) et (B) sont dites passives alors que toutes les autres, (C) (D) et (E), sont regroupées sous le terme de méthodes actives. Cette distinction vient essentiellement de ce que les premières méthodes analysent les images obtenues par les caméras alors que les autres conditionnent les configurations des prises de vue en modifiant les caractéristiques de la caméra (distance focale ou position) ou encore exigent un éclairage spécial de la scène.

A- Méthodes monoculaires : *shape-from-X*

Toutes ces méthodes développent des heuristiques permettant de retrouver une représentation de type *2,5 sketch* à partir :

- des contours (*shape from contour*) [BY84] : il s'agit de retrouver l'orientation et la normale des surfaces visibles dans une image à partir des contours qui en sont extraits. Cette technique donne de bons résultats quand elle est exclusivement appliquée aux contours provenant d'une discontinuité physique entre deux surfaces. Mais comme les points contours extraits d'une image ne peuvent être classés selon leur origine physique, cette technique demeure peu fiable.
- de la texture (*shape from texture*) [BB82] : ces méthodes supposent que la texture de la surface observée est composée d'une primitive régulière générique de base connue. Ainsi tout changement dans l'image de cette primitive en taille ou forme est directement lié à la disposition de la surface dont elle provient. La mesure des gradients de ces changements permet d'estimer l'orientation de la surface ou encore la distance de la surface à l'observateur. Le point faible de cette approche réside au niveau de l'extraction des textures qui est problème qui n'est pas encore entièrement résolu.
- de l'ombrage (*shape from shading*) [Hor86] [BB82] : quand une surface ayant des propriétés complètement définies telles la position, l'orientation, la réflectance . . . , est illuminée avec une ou plusieurs sources de lumière, de propriétés parfaitement connues (position . . .), l'image obtenue par une caméra calibrée s'exprime alors en fonction des paramètres de la surface observée, de la lumière et de la caméra utilisée. Ainsi, connaissant parfaitement les conditions de prise de vue concernant la lumière et les paramètres des caméras, il est possible d'extraire des images certaines propriétés (orientation) concernant les surfaces observées. Signalons que toutes les contraintes exigées par cette technique et qui sont rarement vérifiées en réalité qui rendent son utilisation peu fiable.

Dans l'état actuel de la recherche, ces méthodes sont cependant difficilement utilisables et aboutissent à une reconstruction peu stable et peu fiable compte tenu des inconvénients cités pour chacune d'elles.

B- La stéréovision

La stéréovision constitue un des moyens les plus utilisés en vision passive pour rechercher les données 3D de l'environnement observé. La reconstruction de facettes 3D, qui constitue notre travail utilise des techniques de stéréovision. Le principe de base de la stéréovision [BB82] [Aya90], consiste à utiliser plusieurs images d'une même scène

prises sous des angles légèrement différents, parfaitement connus (banc stéréoscopique calibré). Le processus de stéréovision est constitué des deux étapes suivantes :

1. la première étape consiste à retrouver dans les images les points 2D qui proviennent d'un même point 3D. Cette étape est connue sous le nom de *mise en correspondance* et fait l'objet d'une étude au Chapitre 4.
2. la deuxième étape consiste à déterminer la profondeur de chaque point d'un objet par *triangulation*, c'est à dire par intersection des projections inverses provenant de chaque image de ce point.

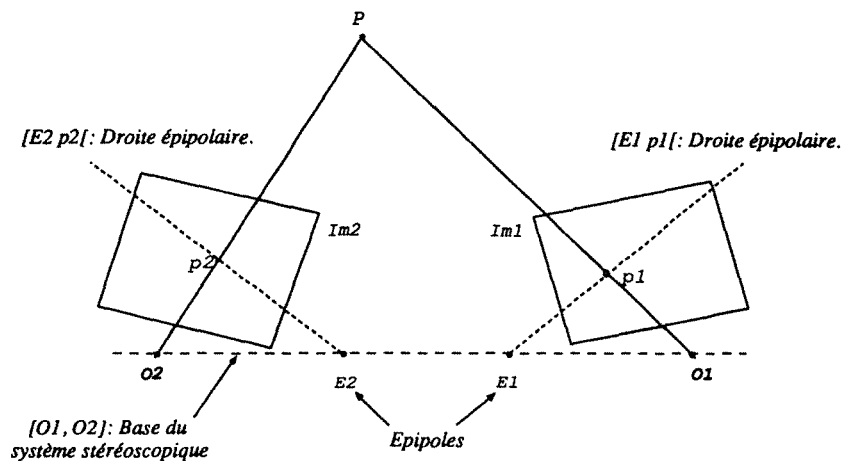


Figure 2.3. Géométrie épipolaire

Un système stéréoscopique est régi par les propriétés bien connues de la *géométrie épipolaire* (Figure 2.3). Prenons le cas d'un système stéréoscopique binoculaire et présentons les différentes propriétés géométriques relatives à ce système. Un point P de la scène se projette sur les deux images en deux points (p_1, p_2) appelé couple de points *homologues*. Il est facile de vérifier que tout couple de points homologues appartient à un couple de droites bien spécifiques. Chacune de ces droites provient de l'intersection du plan image considéré et du plan, dit *plan épipolaire*, passant par le centre optique de chacune des caméras O_1, O_2 et par le point P . Ces couples de droites définissent des *droites épipolaires conjuguées*. Il est également vérifié que les droites épipolaires d'un même plan image forment un faisceau de droites passant par un point E_i particulier appelé *épipôle*. Ce point est en fait l'image du centre optique de O_i relativement à la caméra i . Enfin, les points (O_1, O_2, E_1, E_2) définissent une droite dite *base du système stéréoscopique*.

La connaissance parfaite la géométrie épipolaire d'un système de stéréovision donné permet la reconstruction 3D. En effet, après avoir déterminé les couples de points homologues, il est alors possible de reconstruire les points 3D correspondant par triangulation.

C- Le zooming

Cette approche est également dénommée *stéréovision axiale*. L'idée consiste à utiliser un objectif à focale variable (zoom), monté sur une caméra fixe. Par la suite,

différentes images de la même scène sont prises avec des valeurs de focales différentes. Après la mise en correspondance des primitives, une technique de triangulation permet de réaliser la reconstruction 3D des points de la scène observée. Le *zooming* peut-être vu comme étant une translation du centre optique selon l'axe optique, mais une modélisation trop simpliste de ce système peut engendrer des erreurs incontrôlables [LRD91].

D- Le mouvement

Nous dénombrons deux types d'approches dans ce cas :

- aborder le problème de manière *discrète* comme le fait par exemple Lustman [Lus87]. Dans ce cas il y a utilisation d'une séquence de vues prise par une caméra mobile qui filme un environnement statique. Une phase de mise en correspondance, similaire à celle utilisée en stéréovision, est nécessaire entre les segments extraits des différentes images. C'est à partir de ces ensembles de segments homologues qu'il y a estimation des valeurs des inconnues du problème qui sont les points de prise de vue et le mouvement observé.
- aborder le problème de manière *différentielle* [Bou88]. Il s'agit des méthodes dites de calcul de flot optique ou du champ apparent de déplacement. Dans le cas où des objets de la scène ou la caméra se déplacent, la fonction intensité $I(x, y)$ devient une fonction dépendante du temps $I(x, y, t)$. Ainsi, prenant deux images prises à des instants t et $t + \delta t$, proches sur l'échelle du temps, tout changement de niveau de gris entre les images d'un même point est uniquement imputé au changement de la position de l'objet par rapport à la caméra. L'étude de ces changements permet de mettre en évidence des équations contraignant le mouvement. Le flot optique correspond à des dérivés premières spatiales ou temporelles de l'image. La recherche de ce flot optique permet de retrouver le mouvement 3D des objets et par la suite la reconstruction de la structure de l'environnement.

E- Illumination de la scène

Le principe consiste à illuminer la scène avec une lumière étalonnée et d'en prendre des images par des caméras. La recherche de la profondeur des objets de la scène se fait à partir de la caractérisation, dans les images 2D, des transformations subies par la lumière étalon. Certains éclairent la scène par des plans de lumière laser [Jar83] alors que d'autres projettent, à l'aide d'un projecteur de diapositives, l'image d'une grille dont les caractéristiques sont parfaitement connues [GPC91]. Une caméra CCD est utilisée pour capter l'image de la scène. L'étude de cette image permet de retrouver la trace de la grille sur la scène. Puis partant de l'espacement 3D connu et des déformations obtenues dans l'image de cette grille, il est devenu alors possible de restituer les informations 3D de la scène observée par triangulation.

2.4 Haut niveau

Le haut niveau est constitué des processus de reconnaissance et d'interprétation du robot. Ceci lui permet par la suite de se localiser, de se mouvoir et de manipuler les

objets d'une manière "autonome". L'action de reconnaissance comporte deux notions :

1. *identifier* un objet. Cette fonction répond à la question "*Que voit-on?*".
2. *localiser* un objet donné en le situant dans la scène observée. Cette fonction répond à la question "*Quelle est la position de tel objet?*".

Afin de réaliser cette tâche, le robot dispose d'une base de connaissances contenant les modèles des scènes observées. Un modèle comporte différentes informations pouvant aller de la description de multiples objets ou de classes d'objets qui composent une scène, à la description topologique de l'environnement dans lequel il évolue [Thi89]. La mise en correspondance entre une partie du modèle et la description de l'environnement fournie par le niveau intermédiaire permet d'étiqueter la scène et ainsi de procéder à la reconnaissance. Cette tâche de reconnaissance est d'autant plus difficile à mener que le nombre de modèle de la base est important. Pour ceci, un index est souvent adjoint à base de connaissances permettant ainsi un accès plus rapide aux modèles [Par91] [PB92].

D'autre part, afin de mener à bien sa tâche de perception et d'être complètement autonome, le système de vision doit être capable d'enrichir sa base de connaissances en mémorisant tout nouvel objet qui se présente. Cette fonction doit être réalisée par les processus dits *d'apprentissage*.

2.5 Les principes du constructeur de facettes 3D

Notre travail se situe dans la partie niveau intermédiaire du système MAGIC (Chapitre 1). L'objectif est de réaliser un regroupement d'indices facilitant la tâche de perception. La description de la structure de l'environnement est réalisée au moyen de facettes (polygones plans 3D) et de segments 3D. Nous rappelons que nous utilisons un système stéréoscopique trinoculaire et que les scènes traitées sont supposées être essentiellement composées d'objets polyédriques. Les segments 3D peuvent par exemple être obtenus grâce à l'algorithme de stéréovision développé par Ayache et Lustman [AL87], qui délivre un ensemble de donnée 3D denses. Reste le problème de la reconstruction de facettes 3D. Avant de présenter la démarche et les principes pour lesquels nous avons optées, nous allons commencer par présenter certains travaux liés à la construction de facettes en stéréovision pour en faire une synthèse et y situer notre travail.

2.5.1 Quelle méthodologie pour la reconstruction?

La littérature offre plusieurs approches possibles au problème de reconstruction de plans utilisant la stéréovision en scènes d'intérieur. Globalement nous classons ces approches en deux groupes : d'une part, celles qui se fondent sur l'information segment pour la recherche des plans de la scène et d'autre part, celles qui se fondent sur l'information région, duale de la première.

A- Reconstruction fondée sur les segments

Dans ce cas, la recherche des plans 3D s'effectue directement sur l'ensemble des segments 3D obtenus à partir de la stéréovision. Néanmoins, notons que les segments 3D

engendrés par stéréovision sont entachés d'erreurs dont l'amplitude est proportionnelle à la profondeur. La Figure 2.4 illustre ce résultat : la vue de face des segments 3D donne l'illusion d'une reconstruction parfaite alors que la vue de côté permet d'estimer l'ampleur de l'erreur.

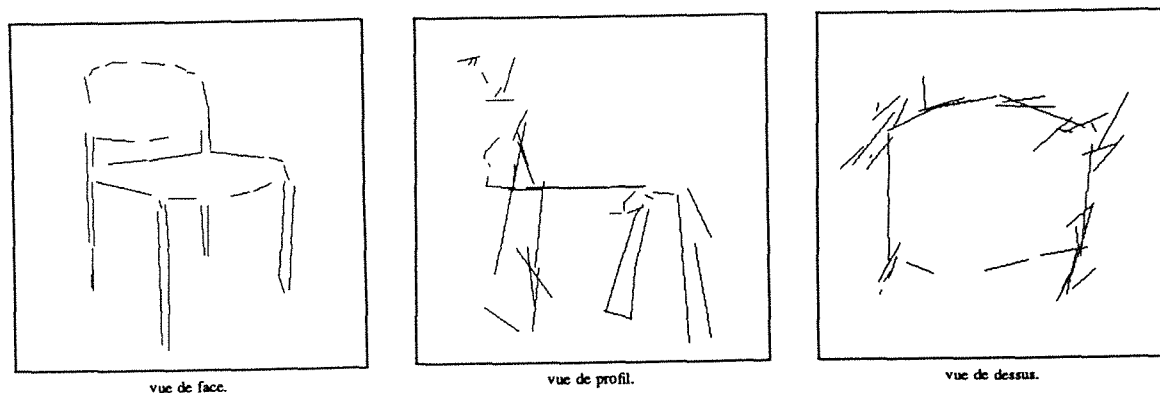


Figure 2.4. Exemple des segments 3D d'une scène [Mar91]

- *Prédiction et Vérification des Hypothèses (PVH)*. Robert de St Vincent [Vin86] recherche uniquement les plans verticaux de l'espace. La prédiction de l'existence d'un plan vertical se fait en utilisant des indices locaux (recherche de segments voisins orthogonaux) au voisinage des segments 3D obtenus par stéréovision. Le plan est ensuite construit au sens des moindres carrés. Mise à part la restriction qui porte sur le type de plans recherchés, la gestion de l'erreur est inexistante. Quand on sait que celle-ci peut être de l'ordre de 30 à 50 cm pour les segments situés entre 5 et 8m de la caméra, il paraît peu réaliste d'essayer de retrouver directement des plans à partir de ces mesures. L'auteur propose d'essayer d'améliorer les résultats en utilisant un modèle des incertitudes attachées aux segments 3D comme celui décrit dans [AF87] et qui permettrait de tenir compte des erreurs.

Un autre travail de type PVH est décrit par Skordas [Sko88] pour la recherche des plans de l'espace. Dans ce cas, aucune restriction n'est émise sur la nature des plans à rechercher. La phase de prédiction est amorcée en 2D : l'idée est d'effectuer un choix judicieux d'un ensemble initial de segments 2D susceptibles d'appartenir au même plan. Deux types de configurations de segments sont retenues : ceux qui ont une extrémité commune et ceux qui sont quasi-parallèles (dans le cas où l'effet de perspective est faible). A chacune de ces configurations est associé, par triangulation, un plan 3D. Chacun de ces plans est enrichi par l'adjonction de tout segment 3D dont la direction s'en approche. Cet enrichissement permet de réajuster les paramètres du plan. Il serait encore peut être plus intéressant d'étendre la phase de génération d'hypothèses 2D à celle de recherche de structures de types polygones 2D ce qui éviterait d'avoir à rechercher le reste

des segments faisant partie du plan dans l'ensemble des segments 3D augmentant ainsi la combinatoire. Peu de résultats expérimentaux sont présentés dans ce travail; il nous est donc difficile de faire une évaluation de la méthode. Remarquons que l'idée d'amorcer la recherche en 2D rejoint celle introduite par Mohan et Neviata dans [MN89]. Ces derniers estiment qu'il est préférable de former des hypothèses en 2D, réduisant ainsi considérablement l'espace de recherche et facilitant les étapes de mise en correspondance. Ceci nous semble constituer une bonne argumentation et que nous avons retenu pour notre propre approche.

- *Système expert.* L'introduction d'une information sémantique et d'une information *a priori* sur l'environnement permet de réaliser l'approche système expert. Cette approche introduite par Thonnat [Tho86] consiste en une phase de génération d'hypothèses suivie d'une phase de classification pour éliminer les mauvaises hypothèses et structurer la description de l'environnement. La génération des hypothèses plans est fondée sur une fusion de plans 3D obtenus à partir de tous les couples de segments 3D coplanaires. Par la suite, l'utilisation d'un système expert permet d'étiqueter les différents plans trouvés suivant leur orientation. Par exemple un plan vertical à une distance maximale est étiqueté comme étant le mur du fond de la scène. Cette phase d'étiquetage permet de rejeter certaines hypothèses non plausibles. Nous remarquons que dans cette technique, mis à part le problème combinatoire du à la génération de tous les types de plans à partir des segments 3D, une connaissance bien détaillée de la scène observée est nécessaire pour permettre l'étiquetage de la scène. Cette méthode ne peut donc être envisagée dans notre cadre de travail.
- *Triangulation de Delaunay.* Les travaux [BFBM87] [BMSFB88] fondés sur une reconstruction par triangulation de Delaunay utilisent directement les segments 3D délivrés par stéréovision. L'idée est de paver l'espace de tétraèdres (de Delaunay) dont tous les segments 3D sont les arêtes. Un critère de visibilité permet ensuite de marquer les tétraèdres vides. La réunion de ces derniers permet de retrouver les espaces libres dans une scène. Ceci permet d'obtenir une information intéressante pour la navigation. Cette approche peut être utilisée pour retrouver les surfaces des objets. Cependant, comme elle utilise directement les segments 3D, nous avons tout lieu de croire que cette technique peut décomposer la surface plane d'un objet en tétraèdres non coplanaires basés sur des arêtes qui ont été mal reconstruites.

B- Reconstruction fondée sur les régions

La recherche des surfaces planes 3D d'une scène peut également être faite en supposant que les régions de l'image, décrivant la scène, proviennent de ces plans. Dans ce cas l'hypothèse de l'existence du plan est faite en 2D et elle est associée à l'information région.

- *Approcher les surfaces 3D par un plan calculé aux moindres carrés.* Dans les travaux présentés dans [WD88] [Mon88], l'approche adoptée est de considérer que toute région 2D d'une image provient de la projection d'une surface 3D plane.

A chaque couple de régions homologues est associé un ensemble de segments homologues. Cet ensemble provient de la mise en correspondance des segments qui bordent ces régions. La reconstruction 3D est alors réalisée par triangulation des segments. L'inconvénient de cette méthode provient du fait que ces segments sont reconstruits indépendamment les uns des autres et que les plans sont ensuite approchés par la méthode des moindres carrés en utilisant les segments 3D provenant d'une même région.

- *Utilisation de la contrainte planaire* Dans le cas d'un banc stéréoscopique rectifié (lignes épipolaires parallèles), Remion *et al.* [RMK87] définissent une transformation affine T_P , faisant passer les coordonnées image gauche des points d'un même plan P aux coordonnées image droite. Ainsi, pour tout couple de point (p_g, p_d) provenant d'un couple de régions homologues (R_g, R_d) nous avons la contrainte suivante:

$$\begin{cases} x_d = \alpha * x_g + \beta * y_g + \gamma \\ y_d = y_g \end{cases} \quad \text{où} \quad \begin{cases} p_d = (x_d, y_d) \\ p_g = (x_g, y_g) \end{cases} \quad (2.1)$$

Les paramètres (α, β, γ) ne dépendent que des paramètres (a, b, c, d) du plan P dont proviennent les régions (R_g, R_d) . En utilisant cette contrainte de planarité deux types de travaux ont été menés.

Dans les travaux de Gagalowicz et de Vezien [Gag89] [VG91], toute région R_i est associée à un vecteur attribut V_{R_i} , comprenant la taille de la région, la position de son centre de gravité et son centre d'inertie d'ordre 2. Ils définissent alors une transformation vectorielle, qui découle de la Contrainte (2.1), et qui permet, pour toute région 3D R plane de plan support $P : z = a * x + b * y + c$, de prédire le vecteur attribut de l'image gauche R_g à partir des paramètres du plan (a, b, c) et de la donnée de l'image droite R_d . Ceci s'écrit alors sous la forme suivante :

$$T_{(a,b,c)}(V_{R_d}) = V_{R_g}$$

Ainsi, étant donné des couples (R_g, R_d) de régions homologues, la minimisation de l'erreur quadratique entre les vecteurs attributs $T_{(a,b,c)}(V_{R_d})$ et V_{R_g} permet de retrouver les valeurs des paramètres du plan support de la région 3D qui leur correspondent.

Si l'intérêt de cette méthode semble être son extension possible à des vecteurs attributs plus riches en information, son désavantage à notre avis est de se servir d'attributs très sensibles aux erreurs de segmentation et aux occultations.

Bonnin [Bon91] utilise directement les résultats présentés par Remion *et al.* [RMK87]. Après avoir effectué une segmentation en régions (qui tient compte des points contours) sur des couples d'images stéréoscopique, il passe à la phase de mise en correspondance des régions qui lui permet d'avoir un appariement de leurs sommets. Utilisant alors la Contrainte 2.1 sur ces couples de points (p_d, p_g) , et en minimisant l'erreur entre les coordonnées des points p_d et de leur prédits $T_P(p_g)$, il détermine les paramètres du plan associé à la région 3D. Il est vrai que pour avoir des résultats intéressants par cette méthode, les couples de points

appariés doivent être fiables ce qui semble être le cas dans le travail proposé par Bonnin.

C- Synthèse

Au regard de ces différentes techniques de reconstruction de facettes planes, nous voyons se dégager deux philosophies :

- (a) l'une qui privilégie la recherche des plans directement dans l'espace 3D (toutes celles qui utilisent directement les segments 3D)
- (b) la deuxième qui génère des hypothèses dans l'espace 2D (à partir des régions ou des segments 2D) et procède à leur validation en recherchant le plan correspondant à ces hypothèses.

Ce dilemme entre ces deux types de raisonnement se retrouve souvent en vision. Une réflexion autour de ce sujet se trouve dans l'article de Kanatani [Kan89]. Il souligne que le passage en 3D par projection inverse ne peut qu'accentuer les erreurs de l'image dues aux bruits. De ce fait, certains calculs en 3D sont complètement faussés et une gestion de l'erreur est nécessaire. Ainsi, s'il est possible d'exprimer des contraintes 3D en terme de primitives 2D et réaliser des mesures uniquement en 2D il est préférable de repousser la reconstruction 3D, et donc l'identification réelle de l'objet en 3D, à une étape ultérieure. Cette réflexion que nous partageons nous semble constituer une bonne argumentation en faveur de la recherche de contraintes de planarité en 2D et non en 3D.

L'idée introduite par Skordas [Sko88] ou encore celle fondée sur les régions semblent être une bonne idée pour la génération d'hypothèse 2D, pour la détermination de plans 3D. En effet, la recherche de polygones 2D permet d'avoir une hypothèse fiable et complète quant à l'existence d'un plan en 3D. Les méthodes proposées par Gagalowicz [Gag89] et Vézien [VG91] ou Bonnin [Bon91] permettent de déterminer en 2D les paramètres des plans 3D par une reconstruction sous contrainte de planarité.

Il est cependant essentiel de souligner que dans les différentes approches citées ci-avant, il n'y a pas réellement de phase de validation de la planarité en 2D puis passage à la reconstruction du plan 3D. Ces deux phases sont confondues en une seule ce qui ne permet pas de récupérer un résultat intermédiaire permettant comme dans notre cas de rechercher, par exemple, la coplanarité de facettes en 2D. En effet, nous proposons pour notre part, une méthode qui sépare la phase de recherche des structures planes validées en 2D et la phase de reconstruction 3D. Les hypothèses de la recherche des structures planes dans l'image sont réalisées à partir d'inférences sur les primitives 2D (ce point est approfondi dans le prochain paragraphe). Des vérifications purement 2D permettent contrairement aux autres méthodes, sans rechercher explicitement les paramètres du plan support de la facette, de prouver la planarité de la structure 2D. De plus, nous prouvons que la méthode proposée permet de décrire un algorithme de recherche de facettes coplanaires et de facettes parallèles de la scène observée en restant toujours dans le cadre de calculs 2D.

Reste alors la question : comment engendrer l'hypothèse de l'existence d'une structure plane de la scène dans une certaine zone de l'image?

2.5.2 Quelles hypothèses 2D engendrer ?

L'étude du système visuel humain a permis de mettre en évidence que le regroupement perceptuel, qui consiste à inférer ou éliminer des contours, afin de faire ressortir les structures caractéristiques de l'image, joue un rôle important dans le processus d'interprétation. Le regroupement perceptuel, qui permet de réorganiser les primitives de l'image en groupe de primitives comportant une information sémantique plus riche, a été étudié par les psychologues de Gestalt dès les années 1920. Dans leur rapport [MN88], Mohan et Nevatia présentent le problème de l'extraction de structures 2D significatives de l'image. Ils notent que les processus de segmentation ont un champ d'action trop local peu dépendant des structures composant la scène, les rendant trop sensibles aux variations intempestives de l'intensité. Ils proposent donc d'extraire de l'image des groupements d'indices qui répondent aux critères suivants :

1. ces groupements doivent permettre de décrire la scène ;
2. ils doivent être des invariants visuels : ne dépendant pas du point de prise de vue ;
3. il doit y avoir une relation entre ces groupements et les entités physiques de la scène (par exemple : faces images \leftrightarrow faces objets).

Ayant choisi de travailler en 2D, nous avons adopté ces critères pour effectuer la recherche des groupements d'indices pertinents. Ces derniers doivent permettre d'inférer l'existence d'une surface plane d'un objet de la scène. Partant du fait que la projection d'un polygone 3D est un polygone 2D, nous nous proposons de retrouver dans les images les polygones 2D qui existent. Cette structure 2D permet par la suite d'inférer l'hypothèse de l'existence de surfaces d'objets plans dans la scène.

Se pose alors la question de l'approche à adopter pour rechercher cette structure 2D. Deux démarches sont possibles. La première consiste à réaliser une segmentation qui crée une forte corrélation entre les entités surfacique de la scène et les entités régions obtenues par les algorithmes de segmentation alors que la seconde consiste à réaliser un *regroupement perceptuel* des primitives segments. En vision artificielle, ce concept de regroupement a été présenté par Marr [Mar82] et consiste à regrouper les primitives du bas niveau au moyen de contraintes topologiques, ce qui permet d'organiser ces primitives images en groupe d'indices caractéristiques se rapprochant beaucoup du regroupement fait dans le cas de la vision humaine. Mais, la faiblesse de chacune de ces approches provient des problèmes inhérents à la primitive utilisée :

1. les outils de segmentation en régions sont très sensibles aux variations d'intensité dans les images. Ainsi, deux surfaces distinctes, ayant des contrastes relativement faibles peuvent être fusionnées en une seule. Alors qu'une surface qui nous semble homogène peut être décomposée en plusieurs petites régions. Cette décomposition est généralement due à l'existence de petites variations d'intensité imperceptibles à l'œil mais affectant le bon déroulement des outils de segmentation. Ainsi, vu les imperfections de ces segmentations, il semble inadéquat de prendre directement les régions comme étant des projections de surfaces d'objets de la scène.

Pour améliorer la segmentation en régions, certains proposent de tenir compte de l'information points de contours. Ainsi Wrobel et Monga [MBD87] proposent de procéder en deux étapes. Une segmentation initiale est réalisée de tel sorte

qu'aucun point de contour ne fasse partie des régions initiales. Par la suite, une procédure de fusion de régions est réalisée. Le critère de regroupement dépend de la variance de la région totale et du pourcentage de points contours sur la frontière commune aux deux régions à fusionner.

Afin d'améliorer les résultats de la segmentation et à des fins de stéréovision, Gagalowicz *et al.* [Gag89] [RG91], proposent de réaliser une segmentation hiérarchique sur un couple d'images stéréoscopiques. Ces segmentations utilisent également des prédicats concernant l'homogénéité et le nombre de points contours sur la frontière. Les résultats des différentes segmentations sont stockées sous forme d'un arbre gardant trace des relations entre les régions d'un niveau et ceux du niveau suivant. La mise en correspondance entre les deux arbres relatifs respectivement à chacune des images permet à la fin d'établir une segmentation en régions de chacune des images. Le choix des seuils pour effectuer les différentes segmentations utilisent des techniques qui s'inspirent des travaux de Kohler [Koh81] et Ohlander [OPR78]. Tous deux proposent d'utiliser une approche hiérarchique de la segmentation en régions générant différentes cartes, puis utilisent des heuristiques, généralement fondées sur l'information contour pour se débarrasser des régions "parasites" et tendre vers un partitionnement de l'image où chaque surface d'objet est représentée par une région. Cette segmentation récursive est contrôlée par le choix automatique des valeurs des paramètres de l'outil segmentation permettant de transformer l'information région petit à petit.

Quant à Bonnin [Bon91], il propose d'effectuer initialement d'une part une segmentation en régions très fine (provoquant une sur-segmentation de l'image traitée) et d'autre part une recherche de points de contours. Par la suite, il propose de fusionner les régions utilisant essentiellement l'information nombre de point de contours sur les frontières communes aux régions et éventuellement un critère d'homogénéité des régions à fusionner en cas d'indétermination. Cette opération de fusion a pour effet d'effacer les points de contours qui sont sur la frontière commune des régions fusionnées et de fermer les contours entourant les régions finales.

Toutes ces améliorations introduites par les différents auteurs, permettent d'aboutir à une segmentation régions plus pertinente mais engendrent le problème de gestion d'un nombre imposant de paramètres (les différents seuils utilisés dans chacune des méthodes).

2. la recherche d'indices regroupant les segments 2D suivant des critères de proximité et de colinéarité pour décrire les images à souvent été utilisé avec succès [Low85] [Qua89] [Sko88]. Cependant la recherche des suites fermées de segments connexes est plus difficile à réaliser car toutes les jonctions dans l'image ne sont pas détectées. Ceci vient du fait que les techniques de détection de contours, bien que plus stables que celles de segmentation en régions, se heurtent aux problèmes de détection de coins et perdent souvent toute trace de connexions entre des segments faisant partie du pourtour d'une même surface.

Synthèse

Nous venons de voir que pour réaliser la recherche des polygones 2D dans l'image, que d'une part il est insuffisant d'utiliser que l'information segment 2D, puisque les points de jonctions sont souvent perdus et que d'autre part l'information région est une information précaire. Cependant, comme la surface d'un objet présente souvent une zone d'intensité presque homogène (sauf dans le cas où la surface est texturée) il semble raisonnable d'utiliser l'information région pour localiser dans l'image certaines surfaces représentant la projection des surfaces d'objets. Nous proposons alors une méthode de recherche d'indices 2D : *les facettes 2D*, qui utilise conjointement ces deux informations duales pour réaliser une description de l'image. Cette utilisation combinée de ces informations duales permet de contrebalancer les défauts de l'une et de l'autre. afin de mieux parer aux imperfections de la segmentation en régions, le module de construction de facettes 2D utilise différentes cartes régions générées selon ses besoins dans une approche de type *multiseuillage* qui nous permet de traiter en premier lieu les facettes les plus contrastées, donc les plus sûres à trouver, avant de passer à celles qui le sont moins.

Cette coopération région/contour n'est pas limitative et pourrait s'étendre à l'introduction d'informations telles que les coins ou encore la texture. Ceci répond à l'optique du niveau intermédiaire que nous voulons développer et qui doit être constitué de processus qui puissent utiliser différentes sources d'informations pour en créer une plus complète.

2.6 Architecture du constructeur de facettes

Nous allons à présent présenter l'architecture du système que nous avons développé pour la construction des facettes 3D. Mais auparavant, récapitulons les principes que nous avons adopté pour réaliser notre objectif.

Le principe fondamental que nous avons suivi pour réaliser la construction de facettes 3D a été introduit dans la Section 2.5.1. Il s'agit d'éviter dans la mesure du possible, tout calcul direct sur les segments 3D. D'une part, pour éviter de travailler sur des données entachées d'une forte erreur de localisation, et d'autre part pour essayer de dégager en 2D le plus d'informations relatives aux propriétés 3D des facettes à reconstruire. Ces informations nous permettent par la suite de mieux contraindre notre processus de construction 3D. Pour ce faire, nous avons alors décidé d'utiliser une méthode (cf Section 2.5.2) qui utilise conjointement l'information région et l'information contour pour regrouper les segments 2D en paquets de segments supposés être les projections de segments 3D qui sont coplanaires.

La conception du constructeur de facettes s'est alors décomposée en quatre étapes :

- *la première étape* consiste à construire les facettes 2D : hypothèse de projection de facettes 3D dans les images. Cette reconstruction utilise deux informations duales : les régions et les segments. Nous avons ainsi décrit un protocole d'utilisation de ces deux informations. De plus afin de parer aux imperfections de la segmentation en régions nous avons proposé une approche *multiseuil* permettant de générer différentes cartes régions d'une même image avec des seuils différents.

- la *deuxième étape* revient à la phase d'appariement des facettes 2D des triplets d'images stéréoscopique traitées.
- la *troisième étape* met en œuvre une technique fondée sur des principes de la géométrie projective qui permet dans un premier temps de vérifier en 2D, l'hypothèse de planarité sous jacente à la structure de facettes 2D. Cela nous donne un moyen de contrôler la validité des hypothèses que nous émettons. Dans un deuxième temps, cette technique est utilisée pour retrouver les classes de triplets de facettes 2D, correspondant à des facettes 3D coplanaires. D'autres applications (concernant la recherche de propriétés telles que l'opacité des facettes) qui n'effectuent que des calculs en 2D, sont également expliquées.
- la *quatrième phase* permet de reconstruire en 3D les facettes en tirant profit de toutes les informations relatives aux propriétés 3D extraites en 2D.

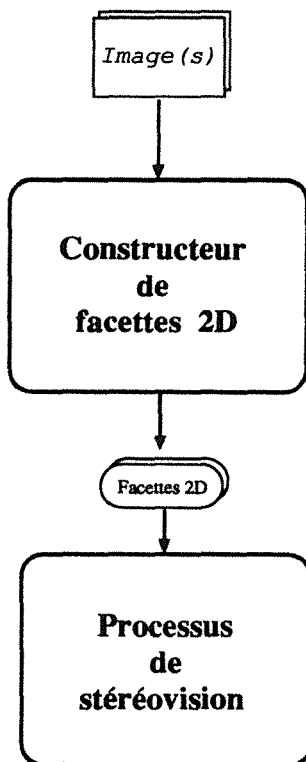
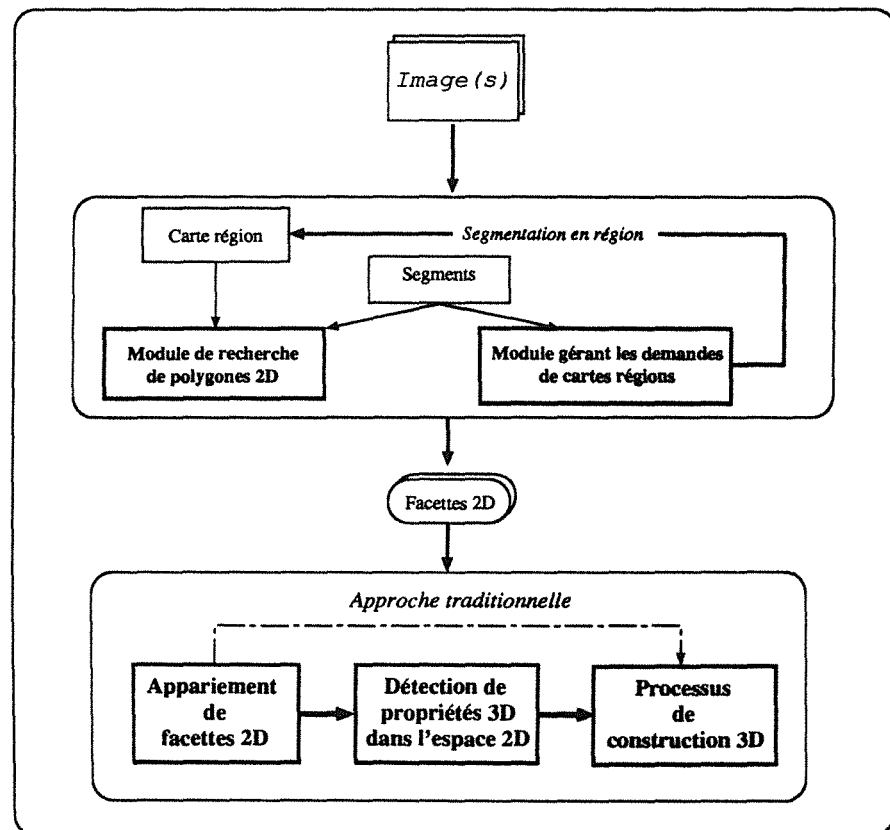
Architecture globale
du constructeur de facettesArchitecture détaillée
du constructeur de facettes

Figure 2.5. Architecture du constructeur de facettes 3D.

Quant à l'architecture du système dédié à la construction de facettes 3D, elle se décompose en deux parties (Figure 2.5) :

- (1) la première partie se charge de générer des hypothèses 2D : *le constructeur de facettes 2D*. La présentation détaillée du constructeur de facettes 2D fait l'objet du prochain chapitre.
- (2) la seconde partie constitue le module de stéréovision des facettes. Après la mise en correspondance des facettes 2D extraites d'un triplet d'images données nous passons à l'étape de validation des hypothèses de planarité. Ce module de validation, que nous introduisons dans le processus de stéréovision, représente la plus grande originalité du système que nous proposons. La phase de reconstruction 3D des facettes est activée en fin de traitement. Sa spécificité provient de l'utilisation des contraintes 3D faites par le module de validation des propriétés 3D dans l'espace 2D.

3

La construction de facettes 2D

Ce chapitre est dédié à la description du module de construction de facettes 2D dont les principes ont été introduits dans le chapitre précédent (Section 2.5.2). L'architecture du constructeur de facettes 2D comporte deux modules (Figure 3.1). Le premier, intitulé *module de construction de polygones 2D*, effectue un cycle complet de construction de facettes 2D, en présence d'une carte de régions donnée. Le deuxième module, intitulé *contrôleur de demande des cartes régions*, se charge de contrôler l'alimentation du premier module en cartes régions.

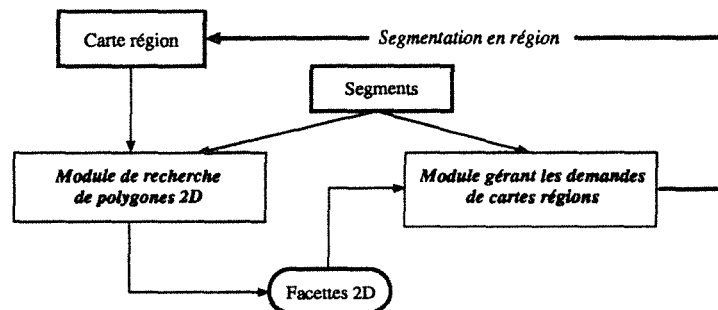


Figure 3.1. Architecture globale du constructeur de facettes 2D.

Avant de procéder à la présentation détaillée des différents modules constituant ce constructeur, nous débutons dans la Section 3.1 par une présentation de définitions qui nous sont utiles. Par la suite, nous décrivons les heuristiques utilisées ainsi que l'algorithme de construction de facettes 2D qui en découle. Mais auparavant la Section 3.2 présente la phase d'initialisation de cet algorithme alors que les sections 3.3 et 3.4 présentent les autres étapes de l'algorithme. La Section 3.5 présente des résultats expérimentaux. Enfin, dans les sections 3.6 et 3.7 nous abordons une discussion d'une éventuelle extension du constructeur de facettes 2D pour intégrer de nouvelles informations tels les points de fuite ou les résultats d'un détecteur de coins.

3.1 Quelques définitions

Nous définissons ici les différentes notions que nous allons utiliser ultérieurement. Nous y explicitons en particulier la relation établie entre les régions et les segments d'une part et les régions et les facettes d'autre part.

3.1.1 Définition d'une facette 2D

Nous tenons, avant de donner la définition d'une facette 2D, à rappeler celle d'une facette 3D. Nous désignons par facette 3D une surface plane d'un objet de la scène observée. En supposant que nous évoluons dans un monde ne comportant pas de surfaces transparentes ni de surfaces texturées, une facette 3D peut-être décrite comme étant un polygone 3D plan opaque de surface homogène.

Définition 1 Une facette 2D, supposée être la projection d'une facette 3D, est définie comme étant un polygone 2D entourant une zone d'intensité homogène. A toute facette est attachée la notion de côté interne et côté externe pour désigner son intérieur et extérieur :

$$f \rightarrow (Reg-Int(f), Reg-Ext(f))$$

En pratique, une facette 2D f est construite en recherchant une suite fermée de segments 2D connexes formant ainsi un polygone 2D. L'homogénéité de l'intensité à l'intérieur de la facette f est confirmée si la segmentation en régions fournit une unique région associée à l'intérieur de cette facette.

Ainsi, pour une carte région donnée, nous définissons les deux terminologies suivantes : *facettes finales* et *facettes incertaines*. Nous désignons par facettes finales, les facettes 2D qui, par rapport à la carte région courante, sont des polygones 2D qui entourent une même et unique région de cette carte. Le terme de facette incertaine désigne toute suite ouverte de segments 2D connexes ou toute suite fermée de segments connexes qui n'entoure pas une région unique. Remarquons que les facettes incertaines d'une première carte région pourront devenir des facettes finales d'une deuxième carte région et vice-versa.

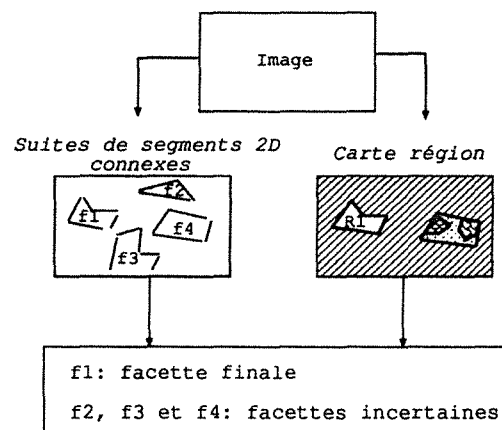


Figure 3.2. Facettes finales et facettes incertaines.

La Figure 3.2 illustre un exemple où il existe une facette finale et plusieurs facettes incertaines par rapport à une carte région donnée. La facette $f1$ est dite finale car utilisant l'information région disponible, elle répond exactement à la définition d'une facette 2D. Quant aux facettes $\{f2, f3, f4\}$ ce sont toutes, pour des raisons différentes, des facettes incertaines :

1. $f2$ est un polygone 2D n'entourant aucune région.

2. f_3 est une suite ouverte de segments 2D.
3. f_4 est un polygone 2D qui entoure plusieurs régions.

Par la suite, nous notons $f = (s_1, \dots, s_n)$ toute facette 2D f , finale ou incertaine. Chaque segment s_i désigne un segment appartenant au bord de f . Cette suite de segments 2D est ordonnée suivant la connexité des segments entre eux.

3.1.2 A propos des segments

Un segment 2D est obtenu par l'approximation polygonale effectuée sur les points de contours trouvés dans l'image. Pour nos expérimentations, nous avons utilisé le détecteur de Deriche intégré dans le logiciel Visilog. Nous avons orienté tous les segments de l'image. Pour ceci nous fixons un sens d'orientation et en déduisons pour chaque segment le point d'origine et le point d'extrémité permettant de parcourir le segment dans le sens *origine* \rightarrow *extrémité*. Ceci permet de parler du côté gauche et de côté droit d'un segment et de lui associer, en présence de l'information région, les notions de régions à gauche et de régions à droite :

$$s \rightarrow (R_g^s, R_d^s)$$

De plus, à tout segment s faisant partie d'une facette f , nous définissons relativement à f les notions de $Reg-Int^f(s)$ et $Reg-Ext^f(s)$ par :

$Reg-Int^f(s_i) = R_g^{s_i}$ (resp. $Reg-Int^f(s_i) = R_d^{s_i}$) si, par rapport au segment s_i , l'intérieur de la facette est à sa gauche (resp. à sa droite). La même définition s'applique pour définir $Reg-Ext^f(s_i)$ en fonction de $R_g^{s_i}$ et $R_d^{s_i}$.

Définition 2 Nous désignons par zone support d'un segment s , l'ensemble des pixels situés à une distance inférieure à un seuil donné ϵ de s (ϵ étant de l'ordre de cinq pixels).

C'est relativement à cette zone support que les relations entre les segments et les régions sont établies.

3.1.3 La correspondance région-segment

Définition 3 La correspondance segment \leftrightarrow région se calcule par la superposition de la carte des segments 2D sur la carte région courante.

Nous associons à chaque segment s , préalablement orienté, les deux ensembles de régions R_g^s et R_d^s situées respectivement à sa gauche et à sa droite. Cette détermination se fait en étudiant les régions qui existent dans la zone support du segment s (cf. Figure 3.3).

$$s \rightarrow (R_g^s, R_d^s) \text{ où } \begin{cases} R_d^s : \text{Ensemble des étiquettes des régions à droite du segment} \\ R_g^s : \text{Ensemble des étiquettes des régions à gauche du segment} \end{cases}$$

Définition 4 Un segment s est dit inscrit dans une région R , si $R \in R_g^s \cap R_d^s$. Si $R_g^s \cap R_d^s = \emptyset$ nous parlons de segment pertinent.

Il est évident que s peut être inscrit dans plus d'une région : $R_g^s \cap R_d^s = \{R_i, i = 1..n\}$.

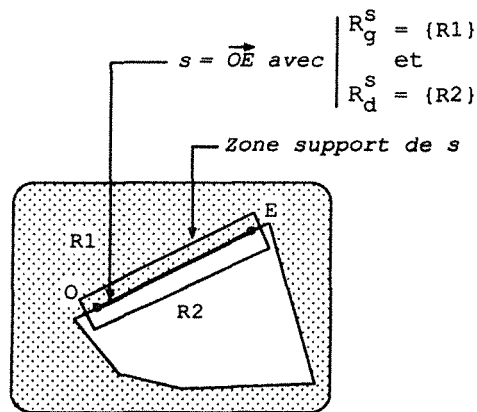


Figure 3.3. Correspondance région-segment.

3.1.4 La correspondance région-facette

A partir de la correspondance entre régions et segments, nous définissons la correspondance entre régions et facettes. A toute facette, nous associons deux ensembles d'étiquettes des régions correspondant respectivement aux régions situées à l'intérieur et celles situées à l'extérieur de la facette.

Soit $f = (s_1, \dots, s_n)$ une facette formée par n segments. L'information $Reg-Int(f)$ et $Reg-Ext(f)$ est calculée comme suit :

$$f \rightarrow (Reg-Int(f), Reg-Ext(f)) \text{ où } \begin{cases} Reg-Int(f) = \cup_{i=1..n} Reg-Int^f(s_i) \\ Reg-Ext(f) = \cup_{i=1..n} Reg-Ext^f(s_i) \end{cases}$$

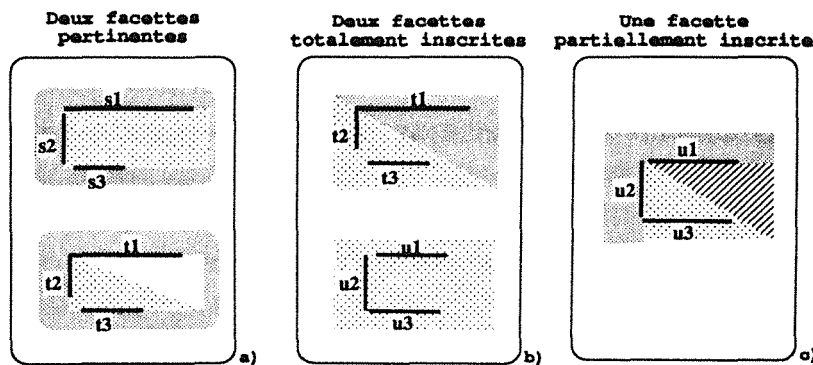


Figure 3.4. Facettes pertinentes et facettes incertaines.

Définition 5 Une facette f est pertinente ou inscrite suivant l'information région qui lui est associée :

1. Si $(Reg-Ext(f) \cap Reg-Int(f) = \emptyset)$ alors la facette f est dite facette pertinente (Figure 3.4.a).

2. Pour toute région R telle que $R \in \text{Reg-Int}(f) \cap \text{Reg-Ext}(f)$, la facette f est dite inscrite dans la région R . Nous distinguons deux types de facettes inscrites :

- (a) une facette f est dite totalement inscrite si $\text{Reg-Ext}(f) = \text{Reg-Int}(f)$ (Figure 3.4.b).
- (b) une facette f est dite partiellement inscrite si $\text{Reg-Ext}(f) \cap \text{Reg-Int}(f) \neq \emptyset$ et si il existe une région $R \in \text{Reg-Int}(f) \cup \text{Reg-Ext}(f)$ telle que f soit non inscrite dans R (Figure 3.4.c).

3.2 Module d'initialisation

Cette étape d'initialisation permet de construire les premières facettes incertaines du système. Ces facettes sont construites à partir de la carte des segments sans utilisation d'aucune information région. Nous allons, avant d'expliquer cette phase d'initialisation, présenter le prétraitement relatif aux segments.

3.2.1 Prétraitement sur les segments

Lors de la construction de facettes 2D, nous recherchons souvent les segments voisins d'un segment donné. Pour accélérer cette recherche, nous utilisons la technique des baquets [Knu75] [Aya90]. Cette technique de tri réalise le partitionnement de l'image en fenêtres carrées appelées baquets. Chaque baquet est ensuite relié à l'ensemble des segments 2D qui le traversent. Inversement, chaque segment s est également associé à l'ensemble des baquets qu'il traverse. La recherche des voisins d'un segment s en une extrémité donnée de s se fait parmi l'ensemble des segments attachés aux baquets associés à l'extrémité traitée.

Initialement les champs de l'information région associée à tout segment sont tous affectés d'une même valeur R représentant une région fictive. Intuitivement, ceci revient à dire que virtuellement nous avons une carte région où il n'y a qu'une seule région R . Ainsi, initialement, tout segment est dit inscrit et par voie de conséquence toute facette du système est également dite inscrite.

3.2.2 Construction des premières facettes incertaines

Considérant que l'information segment est une information fiable, nous estimons que les jonctions de type jonction-L entre segments, quand elles existent, permettent de définir des indices pertinents pour amorcer la recherche d'un polygone 2D. Pour ceci nous allons rechercher les chaînes de segments connexes et convexes. La restriction aux chaînes convexes est une heuristique que nous avons adoptée pour éviter de construire des chaînes provenant de surfaces non planes. Mais cette restriction n'est faite que lors de l'initialisation : en effet si par la suite les informations régions et segments corroborent l'existence d'une facette 2D non convexe, cette dernière est construite et gardée dans le système.

La recherche des jonction-L entre segments est réalisée suivant la méthode décrite par Skordas dans [Sko88]. Cette méthode consiste à ouvrir une fenêtre de taille $w * w$

où w est de l'ordre de quelques pixels autour des points extrémités Ext_s de chaque segment s puis à déterminer la liste des segments dont au moins une extrémité est située dans cette fenêtre. Cette recherche est facilitée par le prétraitement effectué sur les segments (les baquets). Afin de procéder au choix du segment candidat s_{cand} , qui formera avec s une jonction-L, Skordas propose de mettre en œuvre les quatre critères suivants (les notations sont indiqués dans la Figure 3.5) :

1. $\theta_1 \leq \theta_{(s,s_{cand})} \leq \pi - \theta_1$. θ_1 est un angle de l'ordre de quelque degrés et permet de supprimer les segments colinéaires au segment s .
2. ($d_1 \leq w/2$) et ($d_2 \leq w/2$).
3. il ne doit pas y avoir de segment contenu dans une fenêtre autour de $[I, Ext_s]$ et dont l'orientation relative par rapport à celle du segment I, Ext_s soit inférieure à θ_1 .
4. le critère précédent, appliqué cette fois à une fenêtre ouverte autour du segment $I, Ext_{s_{cand}}$.

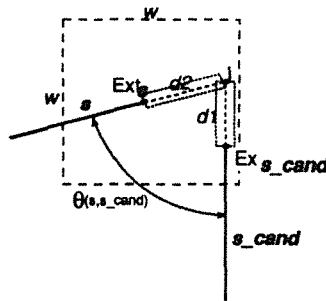


Figure 3.5. Construction d'une jonction-L ([Sko88]).

Chaque jonction-L est considérée comme définissant une facette incertaine. L'intérieur de chacune de ces facettes est défini comme étant le côté du secteur entrant de l'angle défini par la jonction-L. Afin de déterminer les portions de chaînes convexes, nous procédons à la fusion des facettes f_i et f_j , qui se partagent un même segment s et qui vérifient :

$$Reg-Int^{f_i}(s) = Reg-Int^{f_j}(s)$$

En d'autres termes, l'intérieur de la facette f_i par rapport au segment s est situé du même côté que celui de la facette f_j par rapport à s (Figure 3.6).

Ainsi, nous avons initialisé l'ensemble des facettes incertaines du système. L'ensemble des facettes finales est initialisé à vide.

3.3 Module de construction de polygones 2D

L'approche adoptée consiste à utiliser différentes cartes régions (issues de la segmentation d'une même image avec des seuils différents) et une carte segment unique, pour chercher les facettes 2D. A l'étape i , le constructeur de facettes dispose :

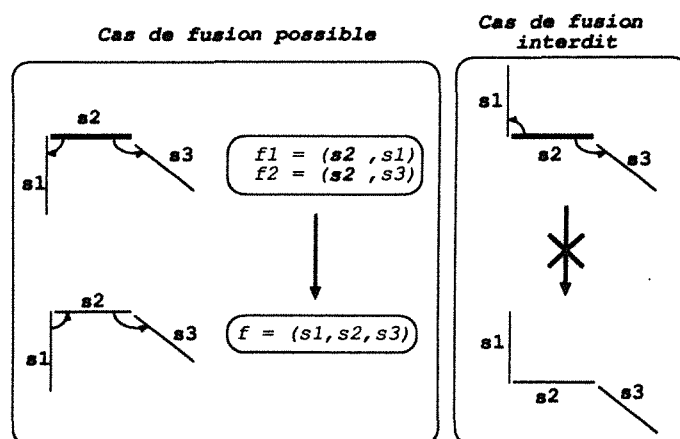


Figure 3.6. Fusion des premières facettes incertaines du système.

1. d'une carte région donnée, calculée suivant ses besoins. Le choix de la carte utilisée fait l'objet de la section suivante,
2. d'un ensemble de facettes finales qui ne sont donc plus à traiter,
3. d'un ensemble de facettes incertaines provenant de l'étape $i - 1$. Initialement cet ensemble regroupe les facettes créées à partir des jonction-L (cf. Section 3.2.2),
4. de l'ensemble des segments. A chaque segment est associé l'information indiquant si il est inscrit dans la carte région de l'étape précédente.

La vocation première du module de construction de polygones 2D est de mettre en œuvre un protocole d'utilisation conjointe de l'information région, contenue dans la carte région courante, et de l'information segment 2D pour engendrer des facettes finales et lever l'ambiguïté attachées aux facettes incertaines de l'étape $i - 1$.

L'utilisation d'une carte région se décompose en deux phases explicitées dans la suite [CM91]. La première phase utilise la nouvelle carte région pour tenter de lever l'ambiguïté relative aux facettes incertaines du système à l'étape courante, afin d'en faire des facettes finales. La deuxième phase du cycle utilise la carte région courante, pour continuer le processus de construction de facettes mais cette fois en l'initialisant à partir l'information région et épuise ainsi toute l'information contenue dans cette carte.

3.3.1 Fonction de recherche d'un segment candidat

Définissons en premier lieu une fonction qui à un segment s donné et une région R , recherche dans le voisinage de l'extrémité E donnée de s , un segment s_{cand} tel que :

1. s_{cand} soit un segment pertinent bordant R ,
2. il n'existe aucun autre segment pertinent qui borde R et qui se situe entre le segment s et s_{cand} .

La recherche de s_{cand} s'effectue de la manière suivante. On délimite autour de l'extrémité E de s une fenêtre de recherche carrée de taille $\delta * \delta$. La valeur δ est deux à trois fois plus importante que le seuil de tolérance w utilisé pour la recherche des jonction-L puisque d'une part w s'est avéré trop petit pour trouver une jonction-L et d'autre part nous ne travaillons plus à l'aveugle : l'information région va nous guider pour la recherche du segment candidat. A cette zone de recherche nous associons l'ensemble des baquets qu'elle intersecte. Chaque liste de segments attachée à un de ces baquets est triée afin de ne garder que les segments s' tels que $R \in (R_g^{s'} \cup R_d^{s'})$ et $R \notin (R_g^{s'} \cap R_d^{s'})$. Nous formons ainsi l'ensemble $Cand$ qui réunit les segments candidats potentiels. Parmi tous les segments s' de cet ensemble le segment s_{cand} est celui qui minimise la quantité :

$$Min(Dist(Orig(s'), E), Dist(Ext(s'), E))$$

Si l'ensemble $Cand = \emptyset$ alors la fonction retourne échec.

Par la suite, nous ferons appel à cette fonction par $Rech_cand(s, E, R)$ où s désigne le segment pour lequel nous recherchons un segment s_{cand} du côté de l'extrémité E et qui borde la région R .

3.3.2 Phase-1 : étude des facettes incertaines

Nous commençons par classer les facettes incertaines en trois classes en fonction de l'information région courante (Figure 3.7) :

- La classe $\mathcal{F}1$ composée de facettes pertinentes ayant chacune une unique région interne.
- La classe $\mathcal{F}2$ composée de facettes entourant chacune plusieurs régions. Cette définition n'exclut pas les facettes partiellement inscrites.
- La classe $\mathcal{F}3$ composée des facettes totalement inscrites.

Les facettes composant la classe $\mathcal{F}3$ sont des facettes pour lesquelles la carte région courante ne présente aucun intérêt puisque la région interne et la région externe associées à chacune de ces facettes relativement à cette carte, sont les mêmes. Pour chacune de ces facettes, il s'agit de trouver la carte région qui leur corresponde et fasse ressortir nettement leurs régions interne et externe. Les facettes de cette classe restent donc des facettes incertaines à la fin de ce cycle.

Les deux autres classes $\mathcal{F}1$ et $\mathcal{F}2$ définissent l'ensemble des facettes à traiter par la carte région du cycle courant. A chacune de ces deux classes nous associons une action qui lui est spécifique (Figure 3.7) : *décomposition* pour la classe $\mathcal{F}2$ et *fermeture* pour la classe $\mathcal{F}1$. Toute nouvelle facette créée, à la suite de l'application d'une de ces actions, est classée suivant l'information région qui lui est attachée dans une des trois classes définies plus haut. Nous présentons ci-après chacune de ces deux actions.

A La décomposition

Cette opération est relative aux facettes de la classe $\mathcal{F}2$. Pour toute facette appartenant à cette classe, le problème vient du fait que plusieurs régions internes lui sont

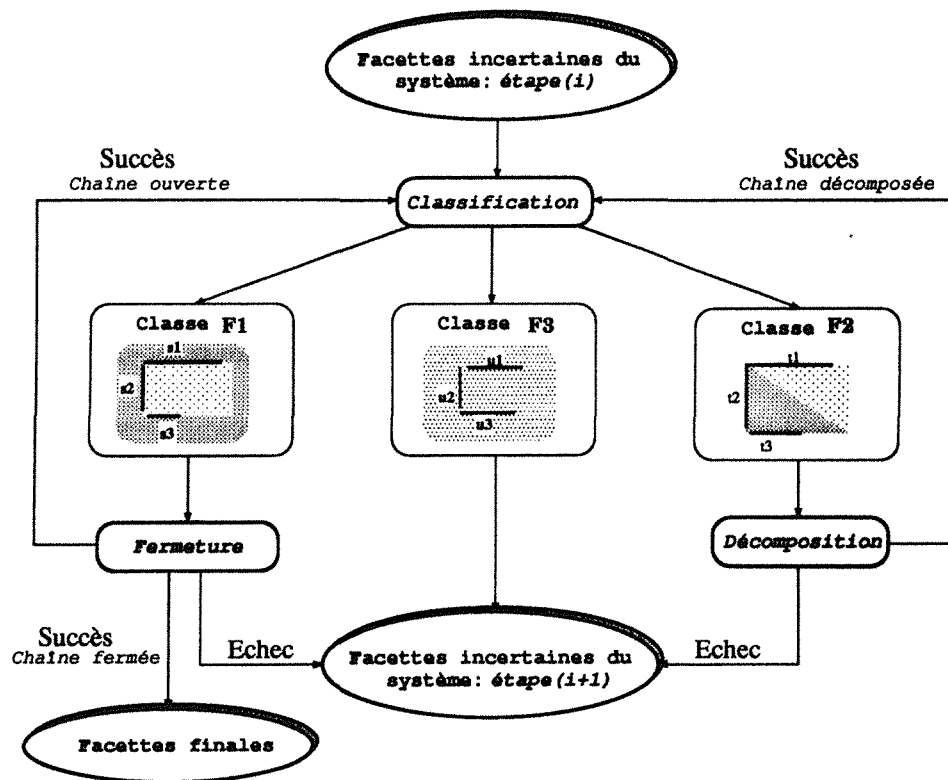


Figure 3.7. Cycle de déroulement de la phase-1 à l'étape i .

associées. La décomposition a pour but d'essayer de diviser chacune de ces facettes en plusieurs facettes dont au moins une soit de classe $\mathcal{F}1$.

Nous distinguons deux types de décomposition possible. L'une concerne la décomposition d'une jonction en deux et l'autre concerne le partage d'un segment en deux. Détaillons ces deux actions.

A-1 Décomposition d'une jonction

Cette opération concerne une facette f de la classe $\mathcal{F}2$ qui comporte dans la liste des jonctions qui la compose, une jonction (s_i, s_j) telle que la région interne bordée par s_i soit différente de celle bordée par s_j (Figure 3.8) :

$$Reg-Int^f(s_i) \cap Reg-Int^f(s_j) = \emptyset$$

Il s'agit alors de rechercher dans l'espace délimité par la jonction (s_i, s_j) , un segment s de la carte des segments qui permet la décomposition de la jonction ambiguë en deux jonctions, (s_i, s) et (s, s_j) . Quand cette décomposition est validée par l'existence d'un segment s , la facette incertaine initiale f est retirée du système et est remplacée par les deux facettes suivantes :

$$f = (s_1, \dots, s_i, s_j, \dots, s_n) \rightarrow \begin{cases} f_1 = (s_1, \dots, s_i, s) \\ f_2 = (s, s_j, \dots, s_n) \end{cases}$$

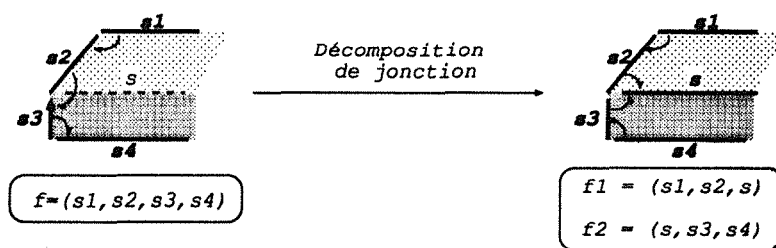


Figure 3.8. Décomposition d'une jonction.

Ces deux nouvelles facettes sont initialement étiquetées comme incertaines et sont classées dans une des trois classes $\mathcal{F}1$, $\mathcal{F}2$ et $\mathcal{F}3$ suivant leurs caractéristiques par rapport à l'information région courante. Signalons que si la facette f à décomposer est une chaîne fermée de segments, la facette f n'est retirée du système que si les facettes filles (f_1, f_2) sont des chaînes fermées de segments connexes. Autrement dit, nous ne permettons pas de remplacer une facette fermée par des facettes incertaines.

Soit E_i l'extrémité mise en jeu dans la jonction (s_i, s_j) , la recherche du segment s se fait alors avec l'appel de la fonction $Rech_cand(s_i, E_i, Reg-Int^f(s_i))$. Dans le cas où cette fonction n'arrive pas à trouver un segment s permettant le partage de la facette initiale, nous déclarons qu'il y a échec de décomposition pour la facette traitée.

A-2 Décomposition d'un segment

Cette opération est plus délicate. Elle concerne les facettes f de $\mathcal{F}2$ qui comportent une jonction (s_i, s_j) telle que l'un des deux segments, supposons s_j , borde deux régions différentes :

$$\begin{cases} Reg-Int^f(s_i) \cap Reg-Int^f(s_j) = R_1 \\ Reg-Int^f(s_j) = \{R_1, R_2\} \end{cases}$$

Une telle facette f borde deux régions par l'intermédiaire du seul segment s_j . Nous partageons alors ce segment s_j en deux segments s_j^1 et s_j^2 sous la condition suivante :

$$\begin{cases} s_j = s_j^1 \cup s_j^2 \\ Reg-Int^f(s_j^1) = R_1 \\ Reg-Int^f(s_j^2) = R_2 \end{cases}$$

Chacun de ces deux segments est dit *segment inféré* (Figure 3.9). Signalons que comme les cartes régions sont utilisées dans le sens des résolutions croissantes nous sommes assurés de ne jamais être confrontés au phénomène inverse : demande de fusion de segments inférés.

La recherche du point de coupure se fait de la manière suivante. Nous procédons à un partage dichotomique du segment. A chaque étape nous sélectionnons la partie du

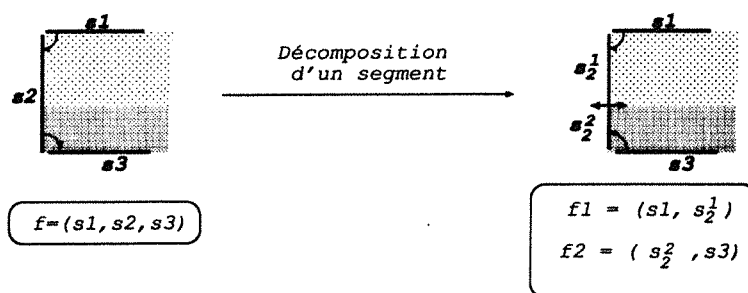


Figure 3.9. *Décomposition d'un segment.*

segment dont la zone support contient les deux régions $R1$ et $R2$. Quand la taille de la partie du segment devient de l'ordre de deux pixels nous en prenons un comme point de coupure du segment.

Remarque 1 *La décomposition d'un segment est une opération dont le résultat n'est conservé que si elle aboutit à une facette finale.*

La facette incertaine initiale f est remplacée temporairement dans le système par les deux facettes f_1 et f_2 :

$$f = (s_1, \dots, s_i, s_j, \dots, s_n) \rightarrow \begin{cases} f_1 = (s_1, \dots, s_i, s_j^1) \\ f_2 = (s_j^2, \dots, s_n) \text{ si } s_n \neq s_j \text{ sinon } f_2 = \emptyset \end{cases}$$

La facette incertaine f initiale n'est pas définitivement retirée du système. En effet, si à la fin de la phase-1 aucune des facettes (f_1, f_2) générées par l'intermédiaire des segments inférés ne donne lieu à une facette finale, nous retirons les segments inférés ainsi que les facettes qui leur sont relatives et nous replaçons dans le système la facette incertaine f .

La décomposition d'un segment est une opération plus délicate que la décomposition d'une jonction puisque l'on remet en doute le chaînage effectué par les outils de détection de contours que nous supposons plus stables que les outils de segmentation en régions. Il est alors raisonnable de ne pas retenir du système les décompositions de segments qui n'ont mené à rien dans le cycle courant.

B- La fermeture de facettes

Cette opération concerne les facettes appartenant à la classe \mathcal{F}_1 . Elle est l'unique opération qui peut créer des facettes finales. La fermeture de facettes consiste à fermer les suites ouvertes de segments 2D, en recherchant les segments manquants. Cette recherche est effectuée en se basant sur l'information région interne associée à la facette traitée.

Soit $f = (s_1, \dots, s_i)$ une facette à fermer. La recherche d'un segment s_{cand} s'effectue dans le voisinage de l'extrémité libre du segment s_i qui borde $Reg-Int(f)$ par l'intermédiaire de la fonction $Rech_cand(s_i, Ext-libre(s_i), Reg-Int(f))$. Si un tel segment existe

alors la facette f est remplacée dans le système par la facette $f' = f \cup s_{cand}$ (Figure 3.10.a). Si le segment s_{cand} appartient à une autre facette de même région interne que la facette f nous procédons alors à l'opération de *fusion de facettes* qui consiste à fusionner les deux chaînes de segments en une seule (Figure 3.10.b).

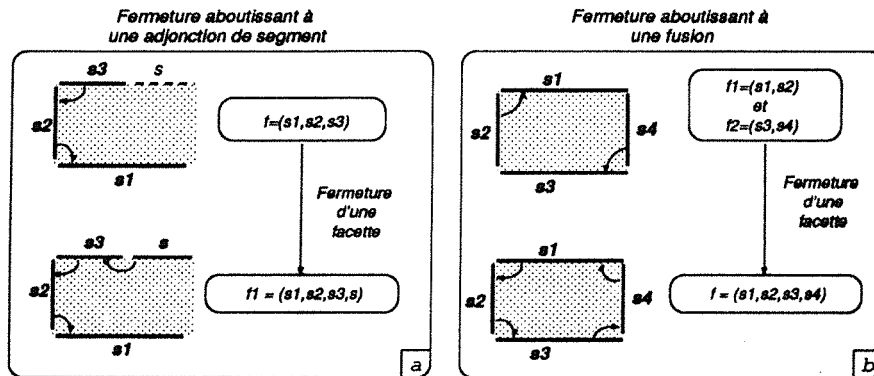


Figure 3.10. Fermeture d'une facette.

Remarquons que l'opération de fermeture peut engendrer une facette de type $\mathcal{F}2$ si le segment s_{cand} ajouté borde une autre région en plus de la région interne de la facette. Une opération de décomposition de type partage de segment peut être alors nécessaire.

Si aucun segment s_{cand} n'a été trouvé alors la facette f est gardée comme incertaine pour cette étape. Remarquons que les chaînes fermées de segments connexes engendrées par l'opération de fermeture sont des facettes à ne plus utiliser avec la carte région courante.

Une classification est alors nécessaire pour déterminer parmi ces facettes celles qui sont finales et celles qui restent incertaines et qui feront éventuellement, l'objet d'un nouveau traitement avec une nouvelle carte région.

Nous considérons comme facette finale toute suite fermée de segments connexes de type $\mathcal{F}1$ ne contenant pas de segments inscrits dans sa région interne. Cette dernière restriction tient compte du fait que chaque segment inscrit dans une région interne d'une facette, peut ultérieurement intervenir dans la décomposition de cette facette, en présence d'une carte région plus fine rendant ces segments pertinents. Toute facette ne répondant pas à ce critère est déclarée comme incertaine.

3.3.3 Phase-2 : recherche des facettes à partir des régions

Après avoir utilisé la carte région courante pour traiter les facettes incertaines du système, nous nous proposons d'utiliser l'information région pour rechercher de nouvelles jonction-L dans l'ensemble des segments 2D n'appartenant pas à des facettes (Figure 3.11). Ces nouvelles hypothèses de facettes incertaines de classe $\mathcal{F}1$ viennent réalimenter la chaîne de traitement de la *phase-1*.

L'objectif de cette phase est de parer aux imperfections possibles de l'étape d'initialisation du processus de construction de facettes (Section 3.2.2) en utilisant l'information région. En effet, l'ensemble initial des facettes est généré en se fondant sur la

recherche de jonctions entre segments. Cette recherche, basée sur un critère de proximité des extrémités de segments entre eux peut, à cause d'une mauvaise segmentation ou d'un mauvais choix du seuil de tolérance de proximité, ignorer des jonction-L importantes.

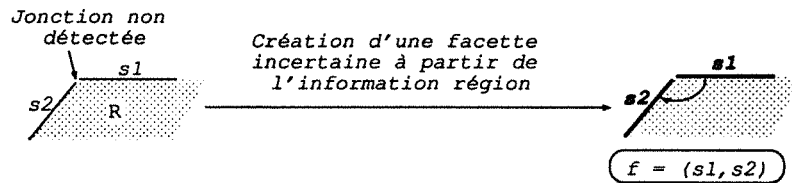


Figure 3.11. Jonction créée à la Phase-2.

Ainsi la *phase-2* a pour objectif d'extraire de nouvelles informations pertinentes pour la construction de facettes 2D, et ce à partir de la carte région courante. Notons d'abord que cette phase ne concerne ni toutes les régions ni tous les segments. En effet, afin d'éviter des redondances de traitement il faut tenir compte des régions et des segments utilisés à la *phase-1*. Ainsi, avant de procéder à la recherche de nouvelles facettes, nous classifions les régions et les segments de la manière suivante :

- toutes les régions de la carte courante ne sont pas pertinentes pour la construction de nouvelles facettes finales. Ainsi nous créons un ensemble de régions dites stériles noté $Reg_{stérile}$ qui regroupe les régions à ne pas prendre en considération. Cet ensemble est initialisé par toutes les régions internes des facettes traitées dans la *phase-1*. En effet, toute l'information portée par ces régions a déjà servi pour établir des facettes finales avec ou sans succès.

$$Reg_{stérile} = \{ \text{Des régions internes des facettes traitées dans la } phase-1 \}$$

- les segments sont également triés. Seuls les segments dits *libres*, ne faisant partie d'aucune facette du système et qui étaient inscrits à l'étape précédente, sont explorés dans cette phase pour la recherche de nouvelles facettes 2D. Cette restriction portée sur le type de segment à traiter répond à une optique d'efficacité. En effet, la restriction aux segments libres permet d'éviter de revisiter une facette déjà traitée en *phase-1*. Quant à la condition d'inscription dans une région à l'étape $i - 1$, elle provient de la remarque suivante. Soit s un segment libre et déjà pertinent à l'étape $i - 1$. Ce segment s a donc été traité à la *phase-2* de l'étape $i - 1$ et apparemment sans succès puisqu'il est resté libre. Cet insuccès provient soit du fait que le segment s ne fait pas partie d'une facette 2D, soit du fait que l'ancienne segmentation en régions n'était pas assez fine pour détecter la facette f auquel appartient le segment s . Cependant comme nous savons que le segment s est pertinent il ne peut être la cause directe de l'insuccès de la fermeture de la facette recherchée. Cet insuccès provient vraisemblablement d'un autre segment s' à qui la carte régions de l'étape $i - 1$ ne convenait pas. Ainsi, notre restriction permet de ne tenir compte que des segments qui peuvent devenir pertinents à l'étape courante, comme le segment s' , et donc porteur d'une information nouvelle pouvant permettre de construire de nouvelles facettes 2D.

Présentons la gestion de la création de nouvelles facettes incertaines à partir des régions sous forme d'une fonction.

Fonction *Création_facette*(*carte_région*, *segments*)
Pour tout *Segment s libre, inscrit à l'étape i - 1*
et pertinent à l'étape courante
Faire
Si *s borde une région R unique avec $R \notin \text{Reg}_{\text{stérile}}$* Alors
 /** Recherche de s_{cand} formant
 avec s une jonction-L bordant R **/
 $s_{\text{cand}} = \text{Rech_cand}(s, \text{origine}(s), R)$
Si *Exist(s_{cand})* Alors
 Poser $f = (s, s_{\text{cand}})$
 Procéder à la fermeture de la facette f
Si *Echec de la fermeture* Alors *rejeter la facette.*
Sinon *Ajouter cette nouvelle facette au système*
 $\text{Reg}_{\text{stérile}} = \text{Reg}_{\text{stérile}} \cup R$
Fin Fonction *Création_facette*

Signalons aussi que tout comme pour l'opération de décomposition de segments, les facettes générées par cette phase sont conservées dans le système que si elles se présentent sous forme d'une suite fermée de segments 2D décrivant une facette 2D de type $\mathcal{F}1$. Ceci est fait dans le but de ne pas encombrer le système de facettes incertaines qui proviennent d'hypothèse faite par la *phase-2*.

3.4 Le module de contrôle de la demande des cartes régions

Rappelons que l'objectif de la segmentation en régions est de partitionner topologiquement l'image en zone vérifiant les mêmes critères donnés d'homogénéité. L'idée sous-jacente à ce partitionnement est de faire correspondre à chaque surfaces des objets de la scène une région unique. Cependant, la segmentation en régions est loin de répondre à cet idéal car c'est est une opération délicate puisque les différentes surfaces des objets d'une scène ne possèdent pas les même critères d'homogénéité. Ainsi, il est quasi-impossible de segmenter une image en régions pertinentes (où à chaque surface d'un objet correspond une région unique) avec les même critères globaux ; les phénomènes tels la sur-segmentation ou encore la fusion apparaissent à chaque fois.

Disposant d'un outil de segmentation en régions donné et afin de pallier aux inconvénients de cet outil, nous nous proposons d'utiliser un ensemble de cartes régions relatives à une même image et obtenu après application de l'outil segmentation avec différentes valeurs de paramètres. Cet ensemble de cartes régions nous permet en fait, de profiter des différentes informations régions pertinentes que délivre chacune d'entre elles et ainsi de mettre en évidence, en plusieurs segmentations, la totalité des surfaces

des objets de l'images. Dans notre cadre d'application cet ensemble de cartes régions est généré par une application itérative de l'outil de segmentation en régions avec différentes valeurs de paramètres calculées par le constructeur de facettes 2D suivant ses besoins. L'utilisation de ces cartes se fait en passant des cartes comportant peu de détail -petit nombre de régions- à celles qui contiennent plus de détails -nombre grandissant de régions-.

Avant de passer à la présentation de la gestion de la génération des cartes régions, nous commençons par présenter l'outil de segmentation en régions que nous avons utilisé dans la version actuelle du constructeur de facettes 2D. Puis, nous introduisons l'heuristique que nous avons établi pour rechercher les seuils de segmentation adéquats avec lesquels nous relançons la segmentation en régions. Comme notre approche se décompose en deux étapes, nous réservons à chacune d'entre elles une section explicatives.

3.4.1 Présentation de l'outil de segmentation en régions

Les outils de segmentation en régions sont généralement des outils qui utilisent plusieurs étapes [WD88] telles que la croissance des régions, l'élimination de petites régions etc., ce qui fait accroître le nombre de paramètres à gérer. L'utilisation de ces outils est d'autant plus délicate que le nombre de paramètres est important. Dans notre cas, nous disposons d'un outil de segmentation en régions écrit par Fairfield [Fai90] facile à utiliser car ne dépendant que d'un seul paramètre.

L'approche adoptée par Fairfield est de renforcer le contraste de l'image à traiter avant de passer à l'étape de segmentation en régions proprement dite. Ainsi étant donné une image \mathcal{I} , il applique sur cette image (Figure 3.12.a) un algorithme de *renforcement de contraste*¹ qui engendre une image dite *clarifiée* \mathcal{I}' (Figure 3.12.b). Cet algorithme a pour effet de renforcer les contrastes entre les points de discontinuité et son auteur le compare aux algorithmes dit de diffusion anisotropique [PM89]. Perona et Malik ont présenté la technique de diffusion anisotropique comme étant une méthode de suppression des détails trop fins sans toutefois affecter la localisation des points ni occasionner la disparition des points à forts contraste.

La segmentation en régions est alors effectuée sur l'image clarifiée \mathcal{I}' et devient un simple algorithme de coloriage. Pour une valeur donnée t du paramètre de l'outil, tout couple (x, y) de pixels voisins, au sens des 8-voisins, est considéré comme faisant partie de la même région si

$$|I'(x) - I'(y)| \leq t.$$

Soulignons que lors d'une utilisation itérative décroissante pour la valeur du paramètre de cet outil, nous sommes certains de ne jamais refusionner des régions qui étaient distinctes aux étapes précédentes. En effet, en prenant deux valeurs t_1 et t_2 , pour le paramètre de l'outil avec $t_1 > t_2$, deux pixels x et y appartenant à une même région à l'étape t_2 sont *a fortiori* dans une même région à l'étape t_1 . Cette remarque indique qu'une utilisation itérative de l'outil de segmentation avec des valeurs de paramètres décroissants, permet d'obtenir des cartes régions qui ne font qu'affiner l'information

1. *Toboggan contrast enhancement.*

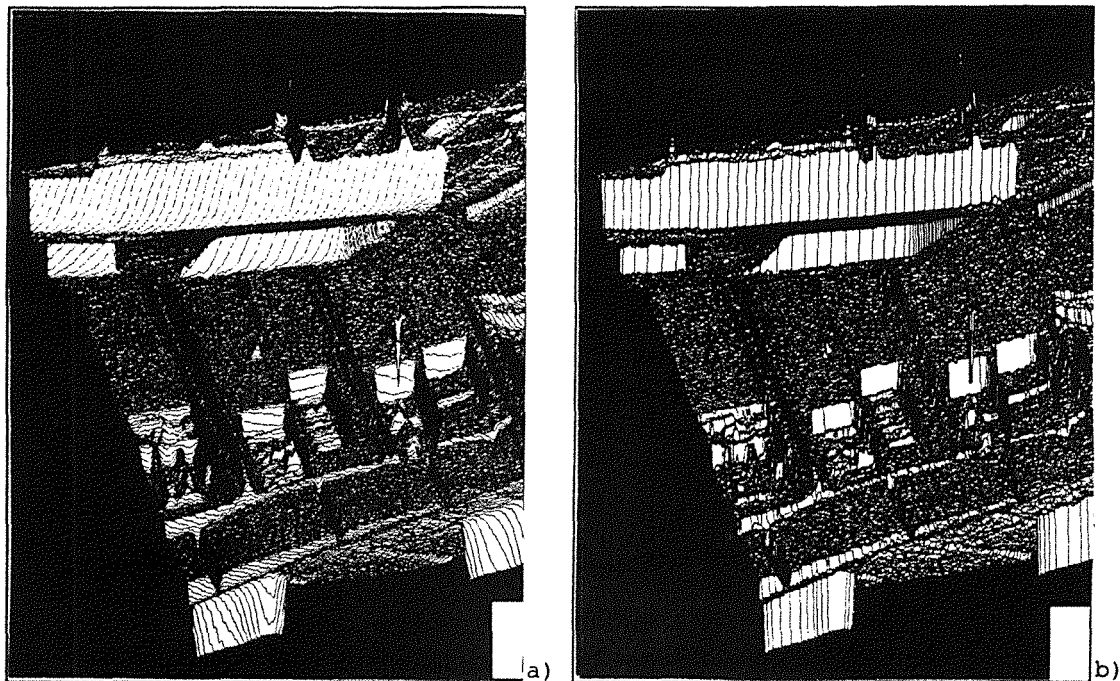


Figure 3.12. Effet de l'algorithme de renforcement d'image. a) Nappe de l'image initiale. b) Nappe de l'image clarifiée.

régions: pas de phénomène de fusion à l'étape i de deux régions distinctes de l'étape $i - 1$, ce qui mettrait en péril notre algorithme de construction de facettes 2D.

3.4.2 Heuristique adoptée pour la gestion des demandes de cartes régions

Comme nous l'avons souligné dans la section précédente, le seul paramètre à gérer pour la segmentation en régions est la valeur maximale acceptée pour le contraste que doivent présenter deux pixels voisins faisant partie d'une même région.

Le calcul des différentes valeurs de seuil segmentation permettant d'engendrer les différentes cartes, répond au besoin du protocole utilisé pour la construction de facettes 2D à partir de l'information région et segment. En effet, la création d'une facette finale f n'est possible que s'il existe une carte région par rapport à laquelle tous les segments composant f sont pertinents et bordent une même région. De ce fait, les segments inscrits dans une carte région sont des segments sans intérêt pour la construction de facettes. Notre but va consister à engendrer un ensemble de cartes régions permettant d'associer à chaque segment s au moins une carte région qui fasse ressortir nettement sa région gauche et sa région droite.

Une segmentation en régions est dite *pertinente* pour un segment s si en l'appliquant dans la zone support du segment s , ce dernier se trouve associé à sa gauche et à sa droite à une région unique. Notre idée est de commencer par associer à chaque segment,

l'intervalle des valeurs possibles à affecter au paramètre de l'outil de segmentation qui évite, dans la zone support de ce segment, les phénomènes de sur-segmentation ou de fusion. Puis, partant de ces différents intervalles, nous calculons les seuils adéquats pour la segmentation en régions permettant d'obtenir l'ensemble de cartes régions utiles à la construction de facettes 2D.

Pour rechercher les intervalles de valeurs relatives à chaque segment assurant une segmentation pertinente autour de ce segment, nous devons tenir compte des trois critères suivants :

critère-1 *tenir compte de la spécificité de l'outil de segmentation en régions utilisé* : la valeur du paramètre représente la différence d'intensité tolérée entre deux pixels voisins pour permettre leur regroupement en une seule région.

critère-2 *pas de fusion* : l'apparition d'une région unique de part et d'autre du segment doit être évitée. Il s'agit alors de rechercher la valeur maximale C_{max}^s , possible pour le paramètre de segmentation évitant le phénomène de fusion.

critère-3 *pas de sur-segmentation* : l'apparition de plusieurs petites régions non significatives de part et d'autre du segment s doit être évitée. Il s'agit, dans ce cas, de rechercher la plus petite valeur possible, C_{min}^s , à affecter au paramètre de l'outil de segmentation évitant ce phénomène.

C'est en tenant compte de ces contraintes que les seuils de segmentation vont être établis [Cha92]. La Figure 3.13 illustre les résultats obtenus avec l'application de la segmentation avec différentes valeurs de seuils à la zone support d'un segment donné. Pour une valeur supérieure à C_{max}^s l'effet de fusion est visible; le segment est complètement inscrit dans une même région. Pour un seuil pertinent, la segmentation en régions fait apparaître des contours régions séparant les régions gauche et droite du segment. Enfin, une valeur de seuil plus petite que la valeur C_{min}^s , fait apparaître une multitude de petites régions non significatives autour du segment s .

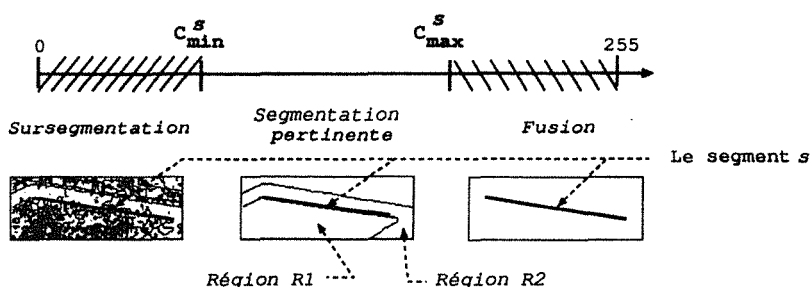


Figure 3.13. Intervalle des valeurs pertinentes contrôlant la segmentation en régions de part et d'autre du segment.

Pour déterminer ces valeurs C_{min}^s et C_{max}^s associées à chaque segment s , nous utilisons une technique qui s'inspire fortement des techniques de seuillages. L'idée de base que nous avançons est qu'à chaque région doit correspondre une *plage d'intensités* qui regroupent les intensités des différents pixels composant la région. Notre objectif

consiste à rechercher, pour chaque segment s , les plages d'intensités relatives à ses régions gauches et droites, pour permettre par un simple calcul de distance entre les plages d'intensités de déterminer les seuils pertinents de segmentations pour ce segment s . Pour déterminer les plages d'intensités relatives aux régions de la zone support d'un segment, nous étudions les deux histogrammes d'intensité construits autour de ce segment s : l'un concernant le côté gauche du segment et le deuxième le côté droit.

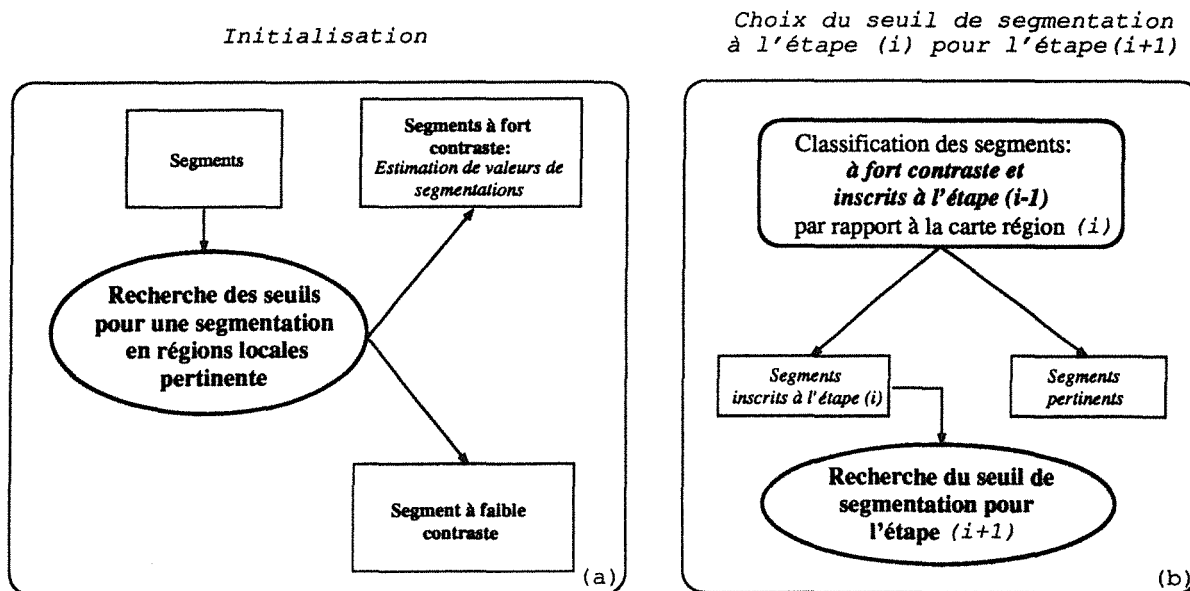


Figure 3.14. Principe du contrôle de la génération des cartes régions.

Après avoir déterminé les valeurs de contrastes pertinents de chaque segment nous pouvons rechercher les seuils adéquats pour la segmentation en régions. En résumé, la gestion de la demande de cartes régions se décompose en deux étapes :

- **étape-1** ou *étape de calcul des contrastes pertinents relatif à chaque segment* (Figure 3.14.a). Cette étape correspond à une étape d'initialisation. Elle consiste à associer à chaque segment 2D s , l'intervalle de contrastes permettant d'obtenir une segmentation locale pertinente pour s c'est-à-dire provoquant l'apparition de régions significatives de part et d'autre de s . La détermination de cet intervalle de valeurs se fait par étude de l'histogramme des valeurs d'intensité lumineuse autour de chaque segment qui nous indique la borne inférieure assurant la non division des régions bordant ce segment et la borne supérieure assurant la non fusion de ces régions. Pour certains segments, dont le contraste est faible, cette recherche de seuils peut aboutir à un échec.
- **étape-2** ou *étape de calcul des seuils de segmentation* (Figure 3.14.b). Notre objectif est de créer des cartes régions différentes permettant à chaque fois de faire décroître le nombre de segments inscrits au profit du nombre de segments porteur d'une information région pertinente. Pour ce faire, les différents intervalles de valeurs associés aux segments inscrits à l'étape précédente sont utilisés pour déterminer le seuil de segmentation à choisir pour l'étape courante du traitement. Le processus est ainsi itéré jusqu'à ce qu'aucun segment ne soit inscrit.

3.4.3 Etape-1 : Calculs des contrastes pertinents d'un segment

A- Contraintes pour la recherche des contrastes pertinents

Nous allons formaliser les trois critères précédemment cités afin de pouvoir les utiliser pour la recherche des seuils pertinents relatifs à chaque segment.

• **Critère-1 La spécificité.** La première condition, relative à la spécificité de l'outil de segmentation, est très importante car elle conditionne la méthode de recherche des valeurs pertinentes pour la segmentation en régions relative à un segment. Cette spécificité implique que, pour trouver les paramètres d'une segmentation pertinente autour d'un segment s , il faut étudier le contraste de chaque pixel de la zone support avec ses 8-voisins. Ensuite, à partir des valeurs obtenues, il faut estimer les seuils de regroupement qui fusionneraient ou qui sur-segmenteraient les régions pertinentes de la zone support du segment s . Mais cette étude, fondée sur la répartition spatiale de l'intensité, est trop locale et coûteuse.

Nous allons donc effectuer une étude plus globale. En tenant compte du fait que l'algorithme de Fairfield effectue une étape de *clarification de l'image* avant de passer à sa segmentation, nous allons étudier la distribution de l'intensité lumineuse dans la zone support du segment relativement à l'image clarifiée. Notons ici que relativement à l'image clarifiée, la zone support d'un segment s présente une distribution d'intensité lumineuse qui permet souvent de bien séparer les intensités des pixels composant les régions gauche et droite du segment (Figure 3.15).

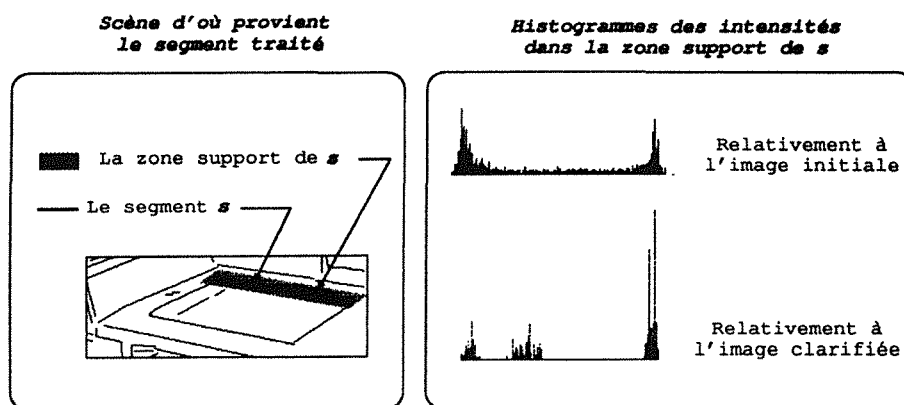


Figure 3.15. Distributions des intensités dans la zone support d'un segment relativement à l'image initiale et à l'image clarifiée.

L'idée est alors d'étudier la distribution de l'intensité lumineuse autour d'un segment, et de déterminer la plus petite et la plus grande valeur à affecter au paramètre de l'outil pour éviter la sur-segmentation et la fusion dans la zone support du segment. Pour ceci, la zone support de chaque segment est partagée en deux parties relatives respectivement à la zone gauche et à la zone droite du segment s . Pour chacune de ces parties nous construisons l'histogramme des intensités correspondantes, notés G et D .

Définition 6 Nous définissons la notion de plage d'intensité P_R relative à une région R comme étant l'intervalle défini par la plus petite et la plus haute valeur d'intensité attachées aux pixels composant cette région.

$$P_R = [\min\{I(x) \in R\}, \max\{I(x) \in R\}] \text{ où } I(x) \text{ désigne l'intensité du pixel } x$$

Nous associons à ces plages d'intensité la notion de distance ensembliste $Dist$ définie comme suit : soient $P_1 = [I_1, I_2]$ et $P_2 = [I_3, I_4]$ deux plages d'intensité :

$$Dist(P_1, P_2) = \begin{cases} 0 & \text{si } P_1 \cap P_2 \neq \emptyset \\ \min(|I_1 - I_3|, |I_2 - I_4|) & \text{sinon} \end{cases}$$

Chacun des histogrammes \mathcal{G} et \mathcal{D} est initialement décrit par les plages d'intensité P_i – composantes connexes – qui le constitue : $P_i = [I_1^i, I_2^i]$ de \mathcal{G} sont les plus grands intervalles tels que $\forall I \in P_i, \mathcal{G}(I) \neq 0$.

Nous définissons la taille d'une plage $P = [I_1, I_2]$ par rapport à un histogramme H comme étant la somme des valeurs relatives à cette plage :

$$Taille(P) = \sum_{I=I_1}^{I_2} H(I)$$

Posons P_i la plage i de l'histogramme considéré. Nous définissons alors les quantités suivantes (la définition symétrique s'applique au côté droit) :

1. $Taille_{Reg}^{totale}$ désigne la superficie de la zone support du segment qui est constante pour un segment donné.
2. $Taille_{Reg}^{gauche}$ désigne le nombre de pixels formant l'ensemble des régions gauches (resp. droites) du segment. Elle est définie comme suit :

$$Taille_{Reg}^{gauche} = \sum_{P_i \in \mathcal{G}} Taille(P_i)$$

Cette valeur dépend directement des valeurs de l'histogramme et varie en fonction des modifications subies par ce dernier.

3. $Taille_{R^g}$ désigne la superficie de la plus grande région gauche noté R^g de la zone support du segment et

$$Taille_{R^g} = \max_{P_i \in \mathcal{G}} Taille(P_i)$$

A cause du décalage entre la position du segment et de les limites entre les régions gauche et droite du segment, la décomposition initiale de l'histogramme \mathcal{G} (respectivement \mathcal{D}) en plages d'intensité correspond rarement aux régions à gauche (respectivement à droite) du segment. En effet, et comme le montre la Figure 3.16, qui traite du même segment que celui de la Figure 3.15, certaines plages affectées à l'histogramme \mathcal{G} devraient de toutes évidences faire partie en fait de l'histogramme \mathcal{D} (ou vice-versa).

C'est pour ceci qu'il faut procéder à des transformations de la distribution de ces deux histogrammes afin d'obtenir des informations cohérentes. Pour ce faire, nous allons introduire une transformation qui va nous permettre de modifier les histogrammes de

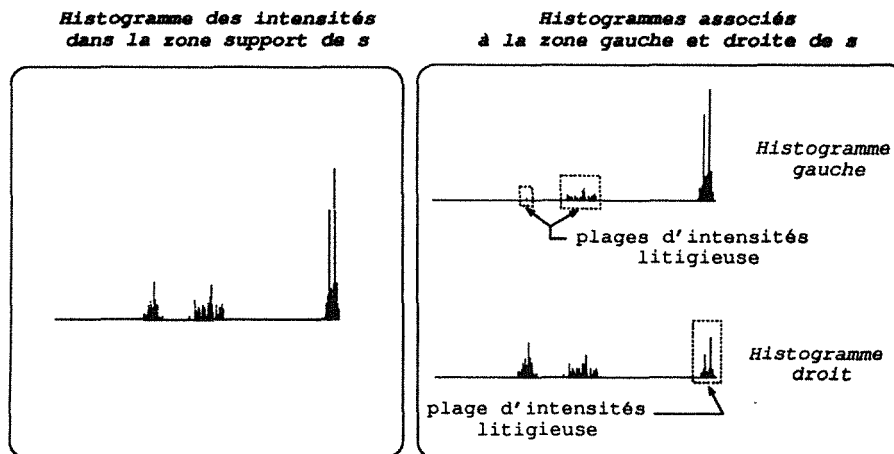


Figure 3.16. Distributions des plages initiales des plages des histogrammes \mathcal{G} et \mathcal{D} .

manière à pouvoir associer, d'une manière automatique, à chaque région droite et gauche d'un segment s leurs différentes plages d'intensité relatives.

• **Critère-2 Pas de fusion.** Ce critère stipule qu'il faut des régions différentes de part et d'autre du segment. Afin de tenir compte d'un décalage possible entre les contours des régions et les segments correspondants, nous considérons que la segmentation en régions est significative si la taille des régions dites gauches (respectivement droites) vérifie la contrainte suivante :

$$\begin{cases} Taille_{Reg}^{droite} \leq 75\% (Taille_{Reg}^{totale}) \\ Taille_{Reg}^{gauche} \leq 75\% (Taille_{Reg}^{totale}) \end{cases} \quad (C1)$$

Ce choix de la valeurs 75% est expérimental. Il tient compte en fait de la largeur de notre zone support du segment qui est de 5 pixels ainsi que du décalage possible des extrémités du segments de l'ordre de 1 pixel en moyenne.

• **Critère-3 Pas de sursegmentation.** Ce critère demande qu'il y ait de part et d'autre d'un segment donné, une région dominante en taille, notée respectivement R^d et R^g . Ainsi, nous introduisons une deuxième contrainte :

$$\begin{cases} Taille_{R^d} \geq 75\% (Taille_{Reg}^{droite}) \\ Taille_{R^g} \geq 75\% (Taille_{Reg}^{gauche}) \end{cases} \quad (C2)$$

B- Algorithme de recherche des seuils pertinents

Définissons à présent la fonction avec laquelle nous procédons à la transformation des deux histogrammes initiaux, \mathcal{G} et \mathcal{D} , afin de déterminer les plages d'intensité des régions gauche et droite du segment étudié.

Définition 7 Nous définissons la fonction Fus qui, prenant une valeur δ et deux plages d'intensité P_1 et P_2 , construit deux plages d'intensité P'_1 et P'_2 définies comme suit :

$$Fus(P_1, P_2, \delta) \longrightarrow (P'_1, P'_2) = \begin{cases} (\emptyset, P_1 \cup P_2) & \text{si } \begin{cases} Dist(P_1, P_2) \leq \delta \\ Taille(P_1) < Taille(P_2) \end{cases} \\ (P_1 \cup P_2, \emptyset) & \text{si } \begin{cases} Dist(P_1, P_2) \leq \delta \\ Taille(P_2) \leq Taille(P_1) \end{cases} \\ (P_1, P_2) & \text{si } Dist(P_1, P_2) > \delta \end{cases}$$

En d'autre terme, quand les deux plages d'intensité traitées proviennent du même histogramme, et qu'elle sont à une distance inférieure à δ l'une de l'autre alors ces deux plages sont réunies en une seule (Figure 3.17.a). En revanche, si chacune de ces plages provient d'un histogramme différent, la fonction Fus a pour effet de transférer la plus petite en taille vers l'autre histogramme en la réunissant à la plage provenant de ce dernier (Figure 3.17.b).

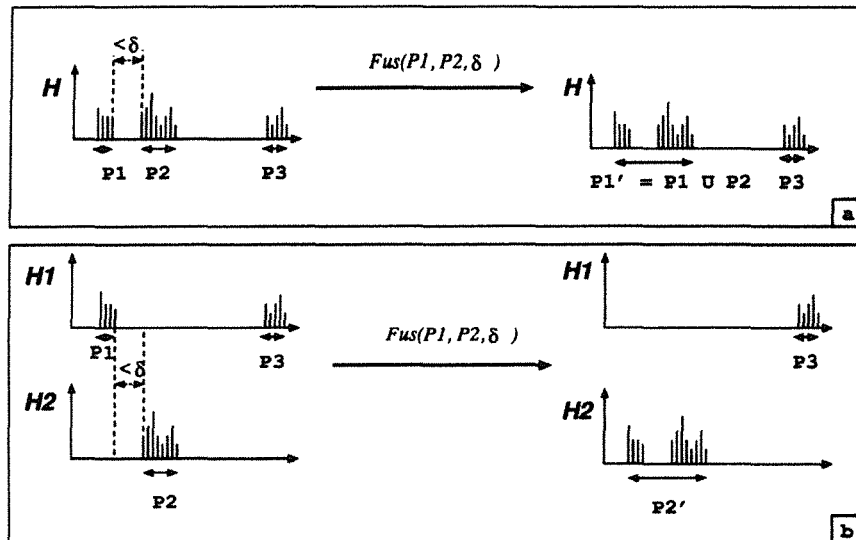


Figure 3.17. Illustration de l'effet de la fonction Fus .

La transformation des histogrammes \mathcal{G} et \mathcal{D} se fait en appliquant la fonction Fus avec le paramètre δ . L'application de cette fonction se fait dans l'ordre suivant :

1. initialement, les plages d'un même histogramme sont fusionnées suivant la distance δ ,
2. puis les couples de plages, provenant respectivement de \mathcal{G} et \mathcal{D} , sont fusionnés.

Après chaque application de la fonction Fus , la répartition des plages entre les histogrammes change, provoquant des modifications des tailles des plages et par conséquent de la distribution dans les histogrammes.

L'algorithme de recherche des contrastes significatifs pour un segment se décompose en deux étapes. Détaillons ces deux étapes :

1- *Prétraitement* : les histogrammes \mathcal{G} et \mathcal{D} relatifs à un segment donné peuvent avoir des plages d'intensité qui s'intersectent, c'est-à-dire qu'il existe une plage d'intensité P_i^G de \mathcal{G} et une autre P_j^D de \mathcal{D} telles que (cf. 3.16) :

$$Dist(P_i^G, P_j^D) = 0$$

Pour supprimer ces intersections et donc lever les ambiguïtés de seuillage, nous appliquons une première fois la fonction $Fus(P_i^G, P_j^D, 0)$. Cette opération a pour effet de réunir en une seule plage les plages qui s'intersectent et de les classer suivant le critère de taille dans l'un des deux histogrammes. Remarquons que, si après cette opération, nous avons

$$Taille_{Reg}^{droite} = 0 \text{ ou } Taille_{Reg}^{gauche} = 0$$

alors nous sommes en présence d'un cas d'échec. En effet, ceci signifie que la phase initiale de séparation des plages a échoué; les plages des deux histogrammes sont rassemblée en une seule. Ce cas se produit quand le contraste entre les deux côtés du segment n'est pas assez prononcé pour apparaître sur la distribution des histogrammes. La détection des plages d'intensité concernant les régions gauche et droite du segment est alors impossible. Un tel segment est dit *ambigu* pour la détection des paramètres de l'outil de segmentation en régions.

2- Recherche des seuils de sur-segmentation et de fusion

- le seuil de sursegmentation correspond à la plus petite valeur δ avec laquelle nous pouvons appliquer la fonction Fus sans mettre en défaut l'une des deux contraintes (C1) ou (C2) ,
- le seuil de fusion correspond à la plus grande valeur δ pour le paramètre distance de la fonction Fus , qui après son application sur les deux histogrammes \mathcal{G} et \mathcal{D} permet de vérifier les contraintes (C1) et (C2).

S'il n'existe aucune valeur δ permettant après application de la fonction Fus de vérifier les conditions (C1) et (C2), nous sommes de nouveau en présence d'un cas d'échec.

L'algorithme qui découle de cette spécification est donné en annexe (cf. Annexe A).

C- Amélioration de l'algorithme de recherche des seuils

A l'expérimentation l'algorithme de détection des contrastes pertinents s'avèrent souvent non fructificatif pour les grands segments: car l'algorithme est incapable de séparer les plages d'intensité relatifs aux régions gauche et droite de part et d'autre du segment. Il est alors essentiel de se demander comment améliorer l'efficacité de

l'algorithme précédent. Pour cela, revenons sur la spécificité de l'outil de segmentation utilisé. En fait, seul le contraste entre un pixel et ses 8-voisins a une influence. Or, en étudiant le comportement de l'intensité des pixels composant la zone support du segment, nous avons perdu la notion de répartition spatiale et nous avons permis la prise en compte de différence de contraste entre des pixels physiquement éloignés entraînant la détection de faux contrastes. Aussi, afin de tenir compte de cette remarque, nous procédons à un partitionnement des segments en morceaux de longueur inférieure ou égale à une constante l . Cette longueur est dans la pratique d'une quinzaine de pixels. Puis, pour chaque partie de ce segment, nous recherchons les seuils pertinents relatifs à la segmentation locale. Afin de propager l'information entre les différentes parties composant le segment, un recouvrement de l'ordre de quelques pixels est maintenu entre elles. La Figure 3.18 illustre le partage d'un segment en deux composantes donnant chacune un intervalle de valeurs pertinentes pour la segmentation en régions.

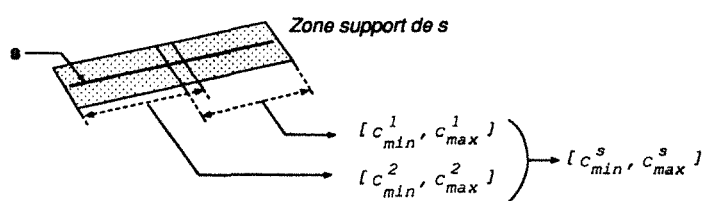


Figure 3.18. Décomposition d'un segment pour la recherche de seuils pertinents.

La détermination de l'intervalle des valeurs pertinentes relatives à un segment est faite à partir des intervalles relatifs aux différentes parties selon lesquelles il a été décomposé. Soit s un segment décomposé en n parties notées s_1, \dots, s_n . Après le calcul de l'intervalle pertinent $[C_{min}^{s_i}, C_{max}^{s_i}]$ pour chaque partie s_i et si aucun cas d'échec n'a été produit nous déterminons l'intervalle correspondant au segment s entier comme suit :

$$\begin{cases} C_{min}^s = \text{Max}_{i=1}^n (C_{min}^{s_i}) \\ C_{max}^s = \text{Min}_{i=1}^n (C_{max}^{s_i}) \end{cases}$$

Ayant ainsi calculé les extrémités de l'intervalle des valeurs pertinentes nous vérifions que $C_{min}^s \leq C_{max}^s$ si cette inégalité n'est pas vérifiée, le segment est déclaré ambigu.

3.4.4 Etape-2 : Calculs des seuils pertinents pour la segmentation

Le module de calculs des seuils pertinents pour la segmentation permet de réactiver à chaque étape l'outil de segmentation avec une nouvelle valeur de paramètre calculée de manière à générer à chaque fois une carte région utile à la construction de facettes 2D. La philosophie de ce module va consister à respecter les deux contraintes suivantes :

1. éviter les conflits occasionnés par une utilisation non ordonnée des cartes régions,
2. engendrer un ensemble de cartes régions permettant d'associer à chaque segment s au moins une carte région par rapport à laquelle le segment s est dit être non inscrit.

En réponse à la première contrainte, nous allons relancer l'outil de segmentation en régions avec des valeurs de seuil qui décroissent strictement.

Pour répondre à la deuxième contrainte, nous nous proposons de calculer la valeur du paramètre de segmentation en utilisant les intervalles de valeurs pertinentes associées à chaque segment non ambigu. Le principe de base utilisé est de rechercher parmi les segments inscrits du système, celui dont la zone support est la plus rapidement sur-segmentée. Ceci se traduit par la recherche d'un segment inscrit s qui a la plus grande valeur pour C_{min}^s . La fonction de choix de la valeur du paramètre de segmentation s'exprime très simplement en fonction des contrastes locaux de ce segment. La valeur du paramètre permet d'obtenir une carte région par rapport à laquelle le segment s est pertinent. Nous nous restreignons aux segments inscrits car ce sont ces segments qui en se transformant en segments pertinents, deviennent porteur d'une information nouvelle et peuvent alors intervenir dans l'élaboration de nouvelles facettes 2D.

Présentons à présent la fonction que nous avons choisi pour calculer à chaque étape la valeur du paramètre de l'outil de segmentation² :

$$t = \frac{C_{min}^{s_cand} + \text{Min}(t_{ancien}, C_{max}^{s_cand})}{2}$$

où t_{ancien} est la valeur précédente du paramètre de l'outil de segmentation en régions et s_cand vérifie les quatre conditions suivantes :

1. s_cand est un segment inscrit dans une région³ de la carte région précédente,
2. $C_{min}^{s_cand} < t_{ancien}$,
3. $C_{min}^{s_cand}$ est la plus haute valeur parmi tous les C_{min}^s des segments inscrits du système,
4. $C_{max}^{s_cand}$ est la plus petite valeur parmi tous les C_{max}^s des segments inscrits s qui sont tels que $C_{min}^s = C_{min}^{s_cand}$.

Les deux premières conditions permettent d'assurer la terminaison et la convergence du système de construction de facettes 2D alors que les deux dernières assurent l'efficacité du choix du segment dont la zone support délimite une surface qui, en fonction des résultats de la recherche du contraste, s'est avérée être la plus sensible à la sur-segmentation.

Initialement, tous les segments sont initialisés comme étant inscrits et $t_{ancien} = \infty$. Après chaque création d'une carte région nous affectons à t_{ancien} la valeur courante de t .

3.5 Résultats expérimentaux

Nous venons de présenter en détail dans les sections précédentes, l'approche adoptée pour extraire les facettes 2D d'une image. Nous illustrons dans cette section des

-
2. Nous avons pris soin de prendre la valeur milieu de l'intervalle des seuils de contraste pour éviter les extrémités qui sont de nature plus sujettes à des erreurs de détections que les valeurs intermédiaire.
 3. Ce segment ne fait donc pas partie d'une facette 2D finale.

résultats d'exécution du module d'extraction de facettes 2D. L'organisation de cette section permet de montrer les résultats des différentes étapes de l'exécution pour l'extraction des facettes d'une image. Puis nous présentons d'autres résultats finaux de construction de facettes sur un triplet d'images stéréoscopique⁴.

A- Visualisation des résultats des différentes étapes de la construction de facettes 2D

La Figure 3.19 montre l'image en niveau de gris traitée et l'ensemble de segments initial utilisé.

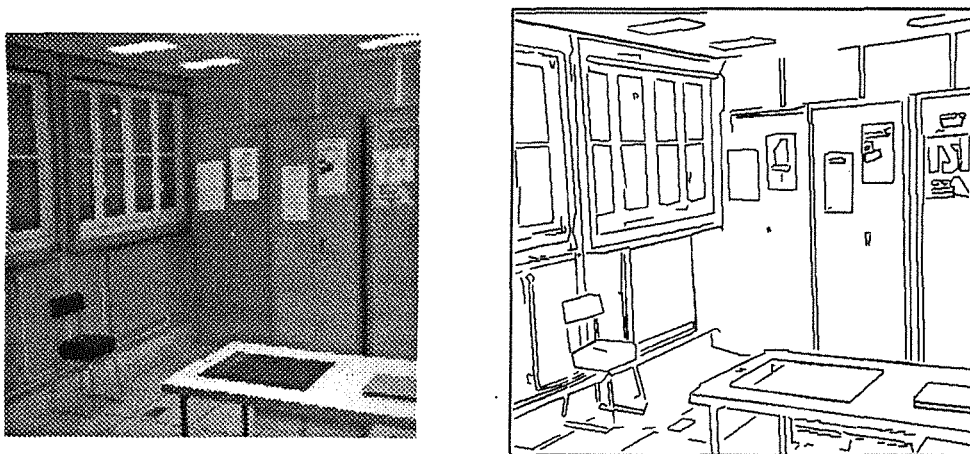


Figure 3.19. *Image initiale et l'ensemble de segments extraits.*

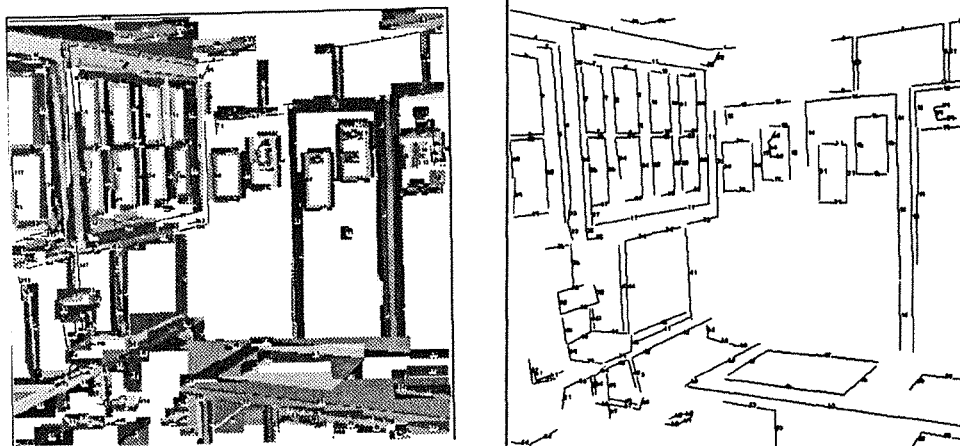
Par la suite, le premier couple d'images de la Figure 3.20 illustre le résultat de l'étape d'initialisation qui :

1. d'une part associe à chaque segment un intervalle des valeurs de contrastes pertinents. Pour visualiser le résultat de cette étape de recherche de contraste nous avons présenté le résultat de la segmentation en régions réalisée autour de chaque segment non ambiguë avec la valeur moyenne de l'intervalle des contrastes qui lui est associé,
2. et d'autre part délivre le premier ensemble de facettes 2D incertaines du système.

Les couples d'images restant de la Figure 3.20 ainsi que ceux de la Figure 3.21 représente progressivement les résultats de chaque étape d'exécution. Nous avons reproduit ici les différentes cartes régions utilisées et l'ensemble des facettes traitées avec succès avec chacune de ces cartes.

⁴ Remarquons que d'autres résultats sont visibles dans les autres chapitres puisque la partie stéréovision du constructeur est alimentée à partir des résultats du constructeur 2D.

Résultat de l'initialisation



Processus de construction de facettes 2D

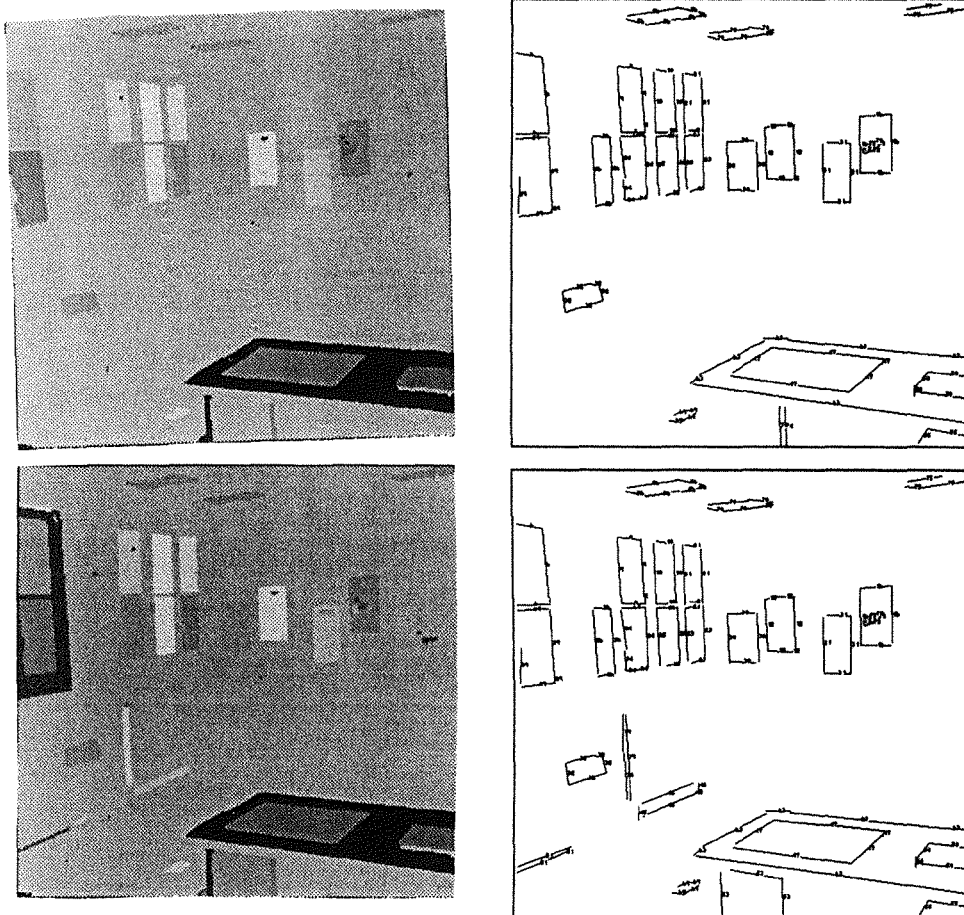


Figure 3.20. *Résultats des différentes étapes de construction de facettes 2D.*

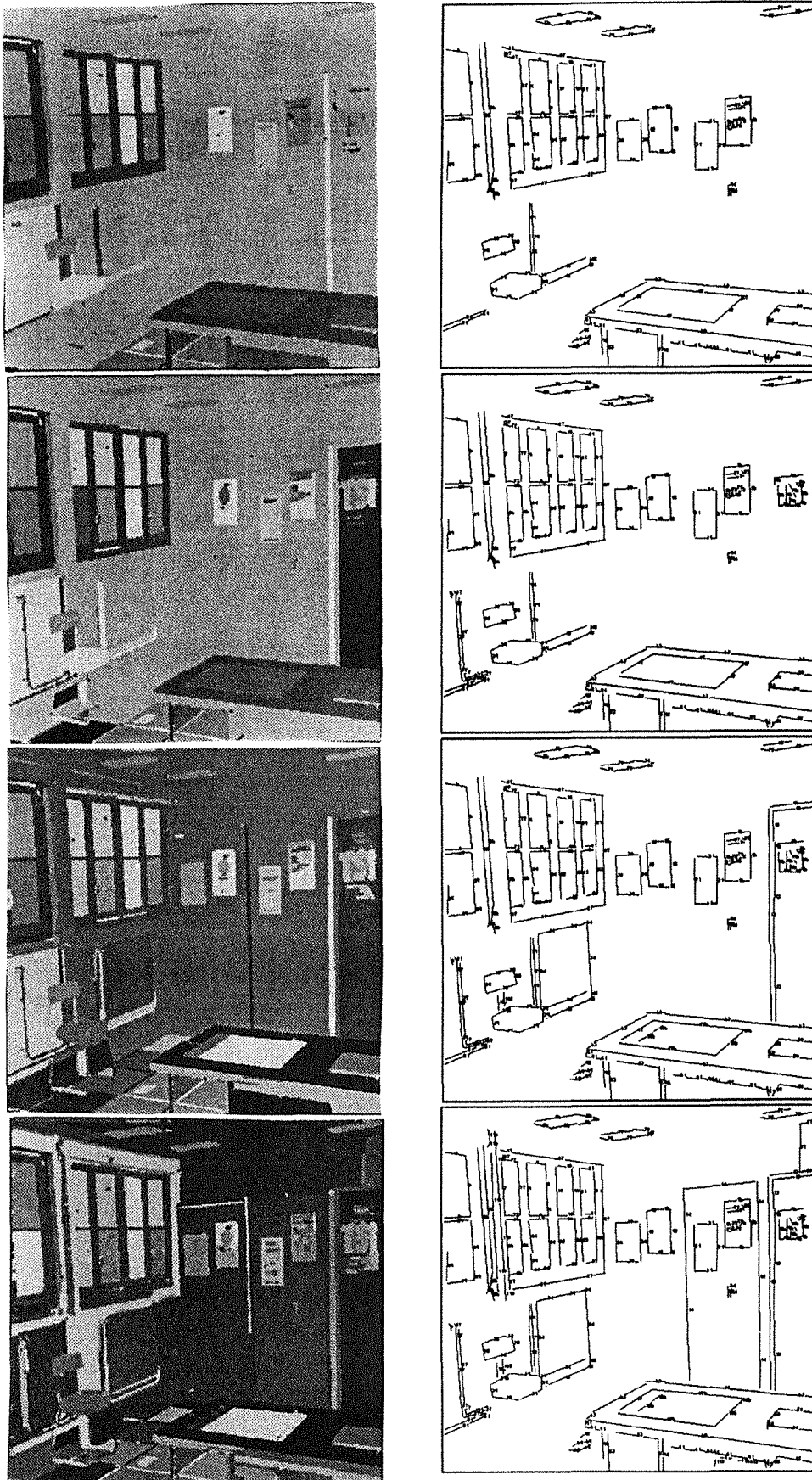


Figure 3.21. Résultats (suite).

Facettes 2D relatives à un triplet d'images

Les trois figures suivantes représentent chacune l'image en niveau de gris utilisée, l'ensemble des segments initial qui en est extrait et l'ensemble de facettes 2D construit pour cet image.

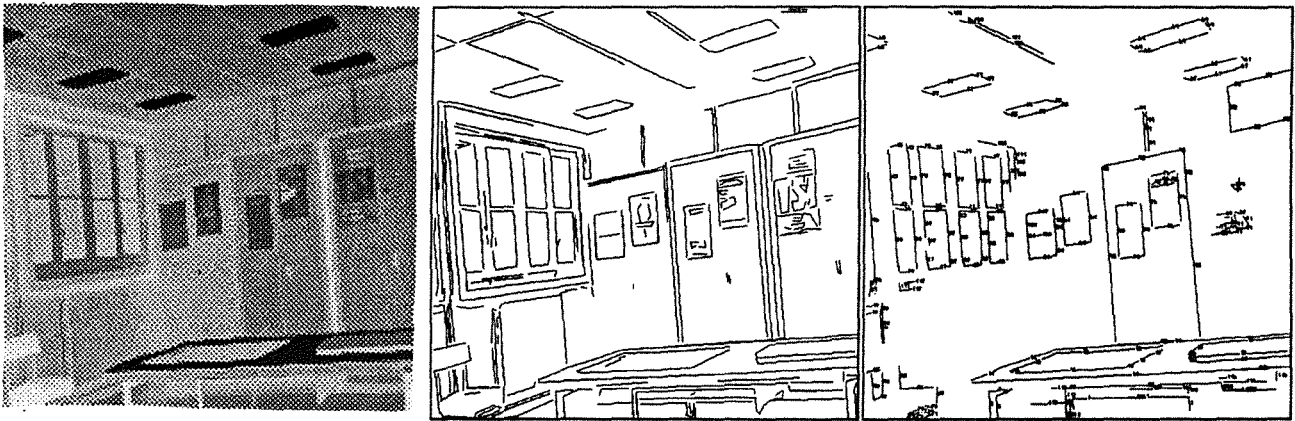


Figure 3.22. Facettes 2D de l'image de la caméra 1.

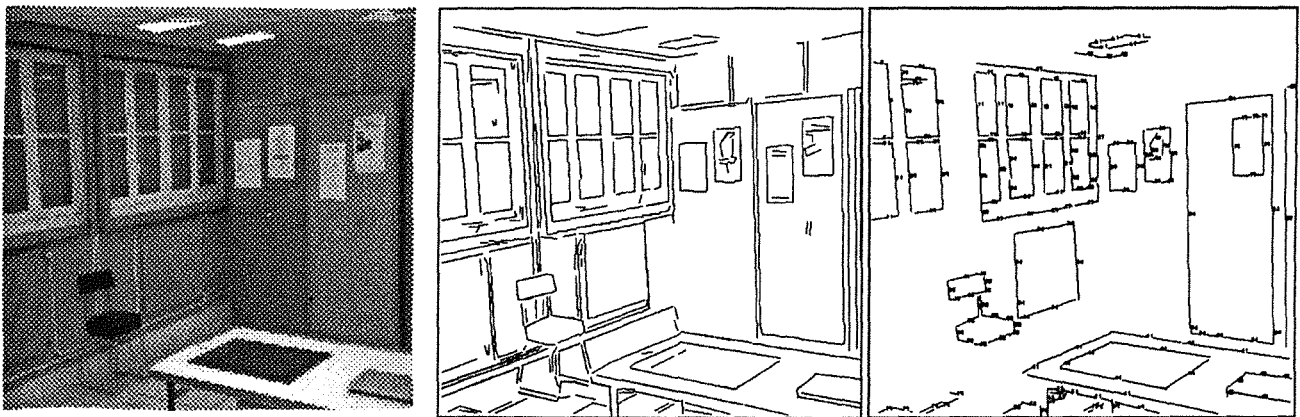


Figure 3.23. Facettes 2D de l'image de la caméra 2.

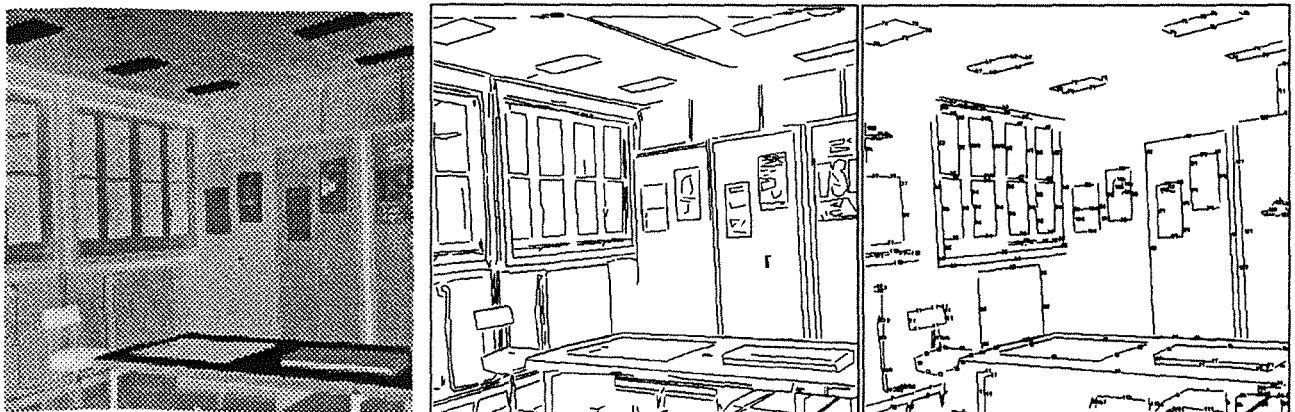


Figure 3.24. Facettes 2D de l'image de la caméra 3.

3.6 Discussion

Dans ce chapitre nous avons présenté la philosophie du constructeur de facettes 2D qui consiste à utiliser, de manière combinée, des sources d'information multiples. Nous nous sommes essentiellement focalisés sur la gestion des deux informations duales que sont les régions et les segments et qui ont permis l'implantation d'une première version du constructeur de facettes 2D. Cependant nous avons également étudié la prise en compte de résultats de recherche des points de fuite et de ceux d'un détecteur de coins. Cette version étendue (Figure 3.25) du constructeur de facettes 2D n'est pas encore implantée. En effet, ayant dans notre cas jugé que le constructeur de facettes actuel donne des résultats satisfaisant, nous avons choisi de développer la partie stéréovision de notre système plutôt que d'améliorer la version actuelle du constructeur de facettes 2D. Signalons également que la partie détection de coins est en cours d'élaboration dans notre équipe, et que l'introduction de ses résultats est de ce fait encore impossible.

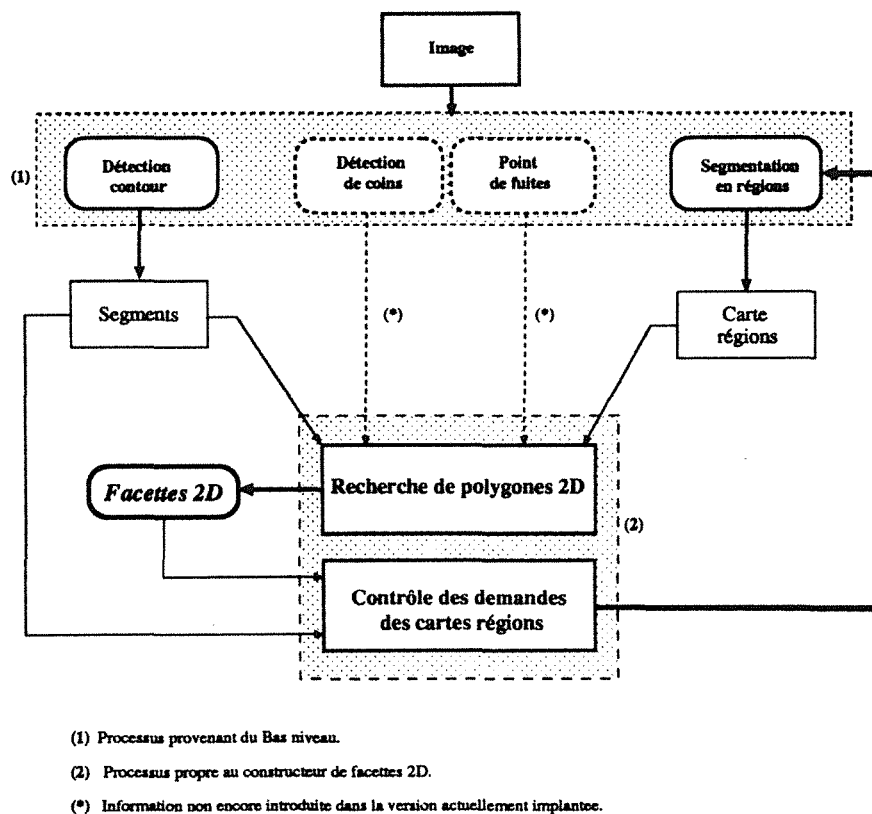


Figure 3.25. Architecture étendue du constructeur de facettes 2D.

3.6.1 Introduction de l'information point de fuite

La prise d'images est modélisée par une projection perspective. Cette transformation, étudiée plus en détail dans le Chapitre 5, a la caractéristique de transformer un faisceau de droites 3D parallèles en un faisceau de droites 2D passant par un point p_{∞} . Ce point p_{∞} définit ce que l'on appelle un *point de fuite* (Figure 3.26). Quan [QM89]

et Lutton [Lut90] ont tous deux développés des algorithmes de détection de points de fuite dans les images en recherchant des points d'accumulation relatifs aux intersections de segments. A chaque point de fuite, ils associent la liste des segments qui y sont attachés et forment alors un groupe directionnel (Figure 3.27).

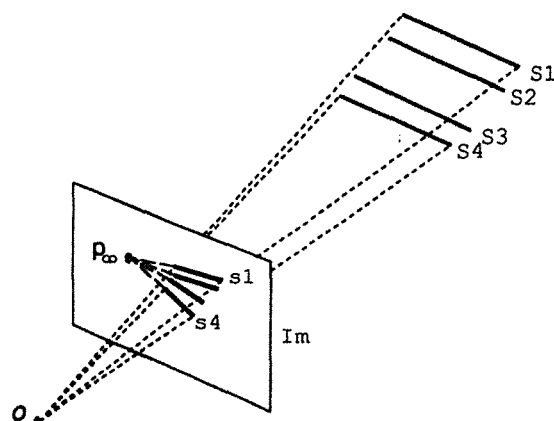


Figure 3.26. Le point de fuite pour un ensemble de segments 3D parallèles.

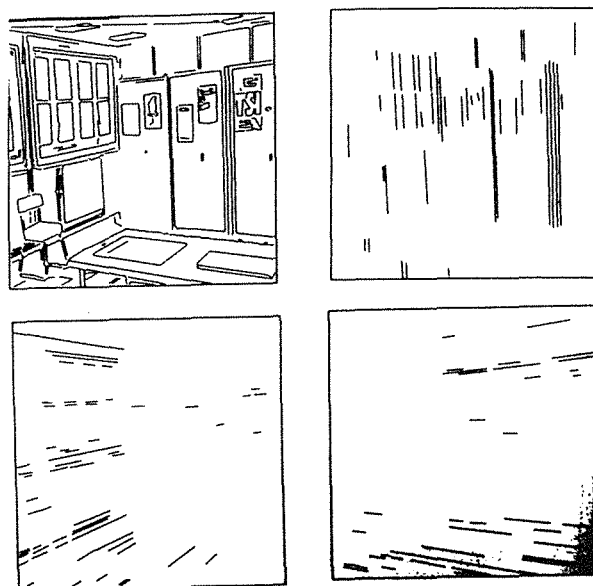


Figure 3.27. Partant de l'image des segments en haut à gauche, nous récupérons trois groupes directionnels représentés chacun dans une image [Qua89].

Tout comme Quan [Qua89], nous pouvons utiliser l'information point de fuite pour contrôler l'étape de construction des indices visuels, en rejetant certaines des facettes 2D ne correspondant pas à la projection d'une surface plane 3D.

Propriété 1 Soient trois segments 2D (s_1, s_2, s_3) dont nous connaissons les points de fuite respectifs (p_1, p_2, p_3) . Nous avons l'implication suivante (Figure 3.28) :

$$(s_1, s_2, s_3) \text{ coplanaires} \implies (p_1, p_2, p_3) \text{ alignés}$$

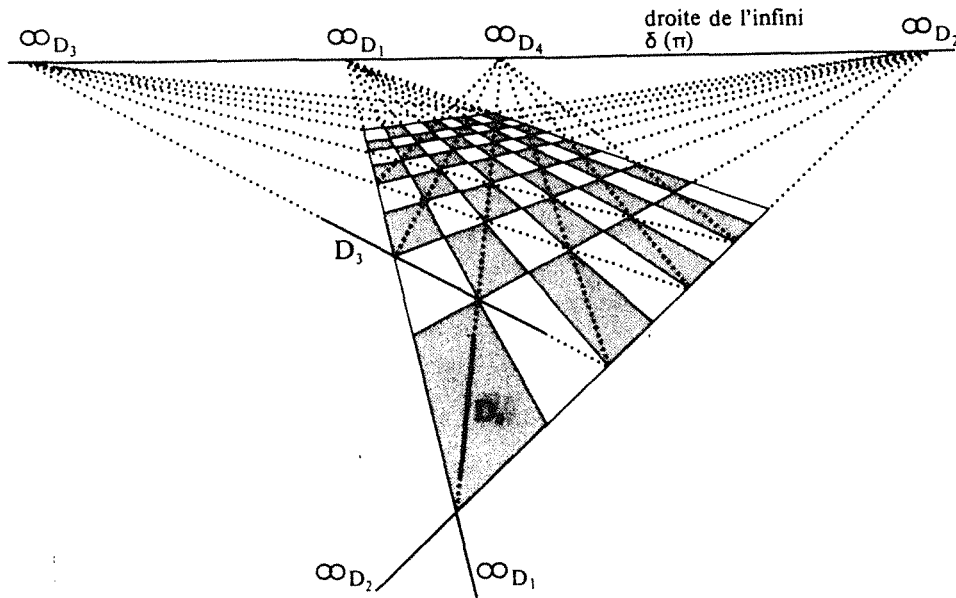


Figure 3.28. Points de fuite des droites coplanaires d'un échiquier vu en perspective ([leh88] page 116).

L'utilisation directe de la Propriété 1 dans le constructeur de facettes peut se faire en vérifiant que pour toute facette, l'ensemble des points de fuite associés aux segments qui la composent, appartiennent à une même droite qui n'est autre que la droite à l'infini de la facette étudiée. Ce test peut être introduit lors du processus de fermeture des facettes incertaines, en rejetant tout nouveau segment mettant en péril la planarité de la facette traitée.

Si l'utilisation des points de fuite semble intéressante car complètement monoculaire, il faut noter que c'est une information qui ne peut pas être calculée pour l'ensemble des segments d'une image et peut même être complètement inexistante dans certains cas. Par exemple, dans le cas où les objets sont proches de la caméra, la recherche de points de fuite est mise en défaut car la transformation perspective ayant un faible effet se rapproche plus d'une projection orthogonale. Remarquons par ailleurs que pour une utilisation efficace de cette information il est nécessaire de tenir compte des erreurs commises lors de la détection des points de fuite [Lut90]. Les segments dont les directions sont proches de la ligne d'horizon ont un point de fuite qui est très mal détecté pouvant ainsi faire basculer l'appartenance d'un segment d'un groupe directionnel à un autre.

Remarque 2 *Insistons enfin sur le fait que l'utilisation de cette information monoculaire même si elle est rattachée à tous les segments, ne peut en aucun cas confirmer la planarité des facettes 3D relatives aux facettes 2D traitées. Nous ne pouvons l'utiliser que pour éviter d'engendrer certaines fausses hypothèses de planarité (Figure 3.29).*

En effet, la Propriété 1 n'est qu'une implication. Il existe donc des cas où, bien que les points de fuites (p_1, p_2, p_3) de trois droites soient alignés, les droites ne sont pas coplanaires. Prenons deux plans parallèles Π et Π' qui définissent une même droite à

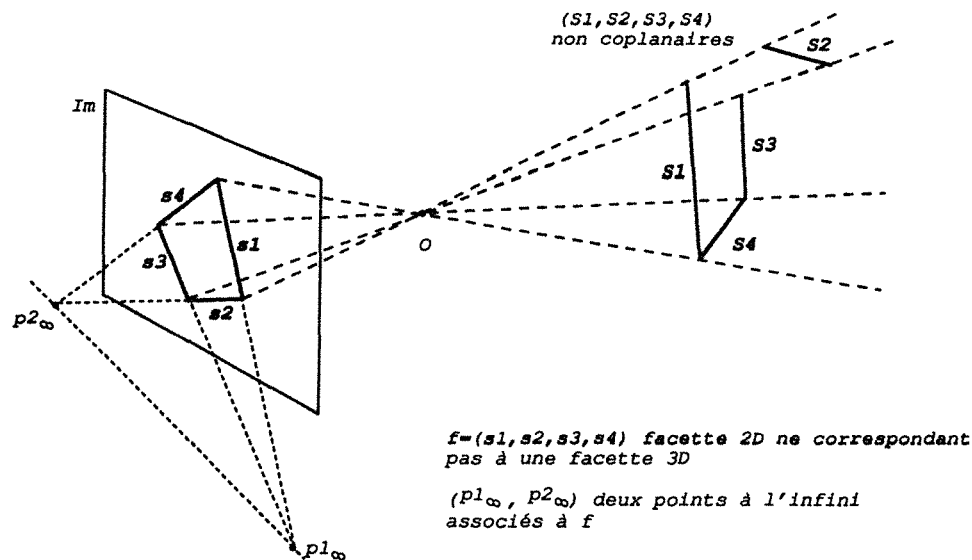


Figure 3.29. Cas d'une facette non plane non détectée par la Propriété 1.

l'infini Δ_∞ . Prenons deux droites $D1$ et $D2$ dans Π et une troisième $D3$ dans Π' . Par construction ces trois droites sont non coplanaires. Pourtant leur points de fuite sont alignés puisqu'ils appartiennent à Δ_∞ .

3.6.2 Introduction des résultats d'un détecteur de coins

L'étape d'initialisation du processus de construction de facettes est fondée sur la recherche des jonctions-L dans l'image. Cette recherche est faite pour l'instant en utilisant un algorithme inspiré des travaux de Skordas [Sko88]. Mais il est évident qu'une recherche au niveau des points de contours serait plus fiable à confirmer ou infirmer une hypothèse de jonction. La recherche de coins tels qu'elle est suggérée par [GD91] peut combler ce vide d'information. En effet, un détecteur de coin permettrait de déterminer des points possibles de jonctions. Ces coins forment alors une information très fiable pour initialiser la recherche de jonctions dans leur voisinages dont ils seront le point de connexion. Ceci nous éviterai de réaliser une recherche de jonctions-L à l'aveugle comme il est le cas pour l'instant dans le constructeur de facettes 2D. Ainsi les points de jonctions-L, calculés comme étant l'intersection des deux segments résultat de l'approximation de chaînes de points de contours, seraient remplacés par les coins qui sont mieux localisés. Cette remarque concernant l'amélioration de la localisation des points de jonctions est surtout intéressante pour la partie stéréovision du constructeur de facettes. En effet, comme nous le montrons dans les chapitres suivant nous préfererons utiliser l'information segment à celle des points sommets des facettes qui ont une localisation déféctueuse entrainant une reconstruction peu stable.

La mise en œuvre de cette amélioration est encore au stade de l'étude puisque le détecteur de coin est en cours de réalisation par S. Tabbone dans le cadre de sa thèse.

3.6.3 Autres travaux sur la reconstruction de facettes 2D

Plusieurs personnes se sont intéressées au problème de la structuration d'images dans le but de faire apparaître les propriétés structurelles de la scène observée. Dans une approche similaire à la notre, nous citons les travaux de Baron et Briot [BB91] qui pour rechercher les structures de la scène observée suggèrent de réaliser un regroupement des segments en paquet de segments supposé chacun provenir de la projection des limites d'une même surface d'objet. Ce regroupement est réalisé en utilisant les deux représentations duales de l'image : les contours et les régions. Pour ce faire, les segments entourant une même région d'une carte région donnée (résultat d'un outil de segmentation qui tient compte des lignes de contours) sont réunis ensemble pour former des groupements perceptuels pertinents correspondant dans notre terminologie aux facettes 2D. Notons que la ressemblance entre nos deux approches pour la construction des structures 3D de la scène s'arrête à ce niveau ; en effet, les auteurs expliquent que pour la partie stéréovision leur direction future consiste dans un premier temps à donner une description de l'image sous forme d'un graphe relationnel entre les segments utilisant pour ceci les relations inférées par les régions : adjacence, inclusion et contiguïté. Cette description leur permet de mettre en œuvre une mise en correspondance fondée sur les cliques maximales (travaux de Skordas [HS89]). Puis supposant être dans un monde polyédrique, ils suggèrent alors de rechercher les plans 3D en supposant la coplanarité des segments appartenant à un même regroupement pertinent et en rejetant tous ceux qui ne vérifient pas cette contrainte de coplanarité en 3D.

Un autre travail connexe au notre est celui réalisé par Zagrouba [ZK92]. Si ce travail présente le même objectif que le notre puisqu'il s'agit de rechercher les facettes composant les scènes observées par stéréovision⁵, l'approche adoptée est quant à elle totalement différente. En fait, bien que s'inscrivant dans le cadre d'une utilisation conjointe de l'information région et segment, l'approche utilisée s'apparente plus aux approches développées dans les travaux fondés sur la reconstruction de régions 3D, comme le travail de Wrobel [WD88], qu'au notre. En effet, l'idée proposée pour rechercher les facettes 2D – projection de surfaces d'objets 3D – est de procéder initialement à une mise en correspondance des régions 2D du couple d'images stéréoscopiques suivie d'une mise en correspondance des segments de contours qui entourent une paire de régions homologues. Chaque ensemble de segments homologues ainsi construit est alors considéré comme étant un couple de facettes 2D homologues. Ainsi, dans ce travail les facettes 2D, d'un couple stéréoscopique, sont construits par couple de facettes homologues. Ceci constitue une autre différence avec notre approche puisque pour notre part nous commençons par traiter individuellement chaque image du triplet stéréoscopique et ce n'est que par la suite que nous procédons à la mise en correspondance des facettes 2D comme il est expliqué dans le prochain chapitre.

Pour finir, nous reproduisons dans la Figure 3.30 un résultat montré dans cet article. Nous pouvons constater qu'aux facettes 2D extraites par ce système ils manquent souvent des segments. A notre avis, cette faiblesse n'est qu'une conséquence directe des problèmes de la mise en correspondance inhérents à la primitive région.

5. Dans un cadre de banc stéréoscopique binoculaire.

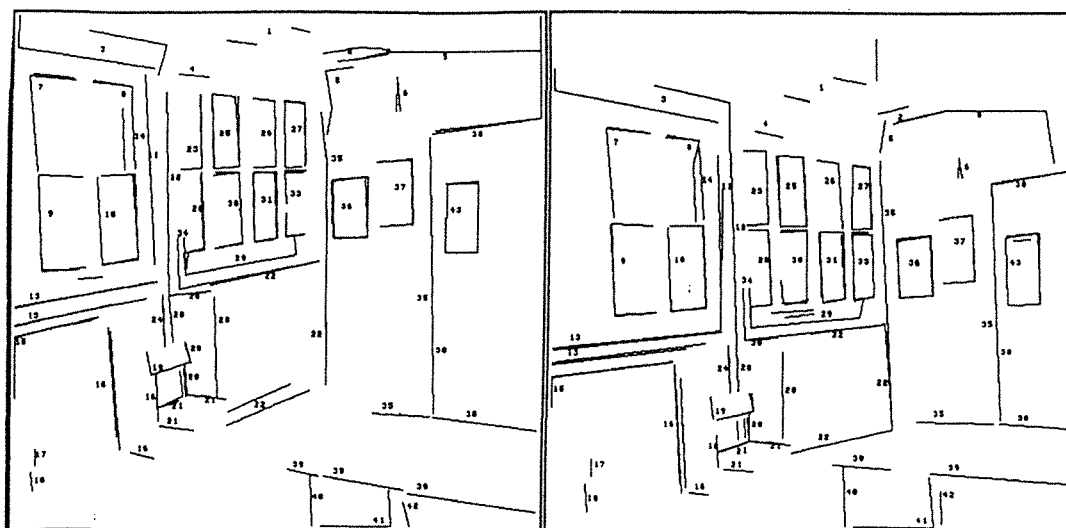


Figure 3.30. Résultats de construction de facettes 2D[ZK92]

3.7 Conclusion

Dans ce chapitre nous avons présenté en détail le constructeur de facettes 2D tel qu'il est actuellement implanté ainsi que les améliorations qui peuvent lui être apportées à court terme. L'avantage de ce constructeur vient de l'approche adoptée qui consiste à utiliser les résultats d'outils du bas niveau pour construire des primitives plus évoluées. De plus l'utilisation de différentes cartes régions a permis de parer aux imperfections de la segmentation en régions.

Une des critiques que l'on peut formuler à l'égard de notre constructeur de facettes 2D est la dépendance de ce système à l'outil de segmentation en régions utilisé puisqu'il nous est impossible de prendre un autre outil de segmentation en régions sans changer tout le module composant le contrôleur de la génération des cartes régions. Cependant l'idée reste valide et il s'agit de trouver le moyen de corrélérer les paramètres de l'outil utilisé avec le résultat escompté.

Par ailleurs, malgré les résultats satisfaisant du constructeur de facettes 2D, notons que la demande itérative des cartes régions, qui se traduit par des réactivations de l'outil de segmentation en régions, ralentit le constructeur de facettes 2D. En fait deux choix sont possibles :

1. soit préparer à l'aveuglette une série de cartes régions et les injecter, dès le départ, dans le constructeur de facettes 2D,
2. soit, préparer les cartes régions au fur et à mesure en réponse à une demande exprimée par le constructeur de facettes. Cette demande se manifeste par la réactivation de l'outil de segmentation avec une valeur du paramètre calculée de manière à permettre la construction de nouvelles facettes finales.

Nous avons choisi pour notre implantation de procéder suivant la dernière méthode. Ceci s'est traduit par l'utilisation d'une heuristique permettant de demander unique-

ment les cartes jugées pertinentes pour la suite du traitement.

Enfin, seule la segmentation en régions est utilisée de manière itérative augmentant ainsi l'efficacité de son utilisation. Ne serait-il pas aussi profitable de songer à une utilisation de type multiéchelle pour les segments permettant une redemande de segmentation contours pour affirmer ou infirmer une hypothèse de décomposition suggérée par l'information région ? La réponse à cette question est délicate et il s'agit en fait d'établir de nouvelles règles d'utilisation de l'information région et segment en se gardant de trop se fier à l'information région.

4

Mise en correspondance des facettes 2D

Nous venons de présenter au chapitre précédent, notre approche pour extraire les facettes 2D des images. L'utilisation de triplets d'images stéréoscopiques de la scène permet de reconstruire ces primitives 2D en trois dimension.

La première tâche du reconstituteur de facettes 3D consiste à rechercher dans le champ visuel commun aux 3 caméras, les triplets de facettes 2D homologues, c'est-à-dire les triplets de facettes provenant de la projection d'une même surface 3D de la scène. Cette phase d'association de primitives 2D, connue sous le nom de "mise en correspondance" ou "appariement", permet ensuite de procéder à la reconstruction des primitives de facettes 3D.

Ce chapitre, consacré à la mise en correspondance de facettes 2D, présente tout d'abord une étude bibliographique des recherches existants dans le domaine afin de situer notre travail (Section 4.1). Nous décrivons ensuite dans la section 4.2, l'approche que nous avons choisie et concluons en présentant quelques résultats.

4.1 La mise en correspondance en stéréovision

La mise en correspondance recherche dans un couple ou un triplet d'images stéréoscopiques les primitives image qui résultent de la projection d'une même primitive 3D de la scène. Cette tâche, aisée pour l'humain, représente l'un des problèmes des plus importants et des plus difficiles de la vision artificielle. Plus particulièrement, dans le cadre de la stéréovision, le phénomène d'occultation ou encore des structures répétitives dans une image, mènent souvent à une confusion lors de l'appariement de primitives.

Une littérature abondante est consacrée à la mise en correspondance en stéréovision. Mentionnons, entre autres, l'article de Barnard et Fischler [BF82] qui présente l'état de l'art en 1982. Dans cet article, les auteurs décrivent les différents travaux, des années avant 1982, concernant la mise en correspondance dans le cadre de la stéréovision binoculaire. Aucune trace de mise en correspondance en stéréovision trinoculaire n'est décrite dans cet article car ce n'est qu'à partir de l'année 1986 que cette technique est appliquée. Ainsi, Yachida *et al.* [YKK86] sont parmi les premiers à avoir introduit l'utilisation d'une troisième caméra en stéréovision comme outil essentiel de vérification des appariements effectués. Plusieurs équipes ont alors développé des méthodes de mises en correspondance trinoculaire. Une étude bibliographique comprenant, cette

fois, les méthodes binoculaires et trinoculaires est présentée dans l'article de Dhond et Aggrawal dans [DA89]. Cet article présente en fait l'état de l'art en 1989. Signalons que deux ans plus tard, ces deux auteurs présentèrent une évaluation des méthodes de stéréovision trinoculaire par rapport aux méthodes binoculaire où ils conclurent que l'utilisation de trois caméras permettait de réaliser une mise en correspondance plus fiable sans pour autant effectuer un nombre d'opérations beaucoup plus important [DA91]. Notons cependant que le champ de vision commun à deux caméras, plus étendu que celui de trois caméras, permet de mettre un plus grand nombre de primitives en correspondance.

Afin d'introduire notre module d'appariement des facettes 2D rappelons brièvement les principes de la mise en correspondance en stéréovision.

Un processus de mise en correspondance stéréoscopique, repose essentiellement sur trois points :

1. *la nature des primitives à mettre en correspondance.* Ce choix dépend d'une part de la nature des informations 3D que nous souhaitons reconstruire -points, segments ...- et d'autre part des primitives 2D qui peuvent être extraites des images.
2. *les contraintes permettant de construire et de sélectionner les bons appariements.* Le choix de ces contraintes est important et dépend du type de la scène observée et du banc stéréoscopique utilisé : binoculaire ou trinoculaire.
3. *la technique de mise en correspondance.* Le choix de la technique dépend fortement des choix réalisés pour les deux points précédents.

4.1.1 Les primitives images à apparier

Différentes primitives peuvent être choisie pour réaliser l'appariement. Nous pouvons les classer globalement en deux catégories suivant l'information intrinsèque qui leur est associée :

- *les primitives locales.* Dans cette catégorie Moravec à introduit les points d'intérêt [Mor77]. Ces points de l'image représentent localement une forte variance de luminance dans les deux directions. La pauvreté de l'information portée par cette primitive et la difficulté qui en résulte pour la mise en correspondance ont contraint les chercheurs à adopter des points plus significatifs que ceux de Moravec : les points de contours [Gri81] [YKK86]. Pour pallier au caractère encore local de l'information portée par les points de contours, certains chercheurs comme Meygret [MTB91] et Zhao [ZSC91], proposent de réaliser une mise en correspondance de chaînes de points de contours. Cependant, l'utilisation des points de contours n'a pas résolu la complexité de la mise en correspondance et l'intérêt s'est porté alors vers des primitives moins locales.
- *les primitives évoluées.* Ayant constaté la difficulté de la mise en correspondance occasionnée par des primitives trop locales comme les points, et traitant des images de scènes d'intérieur où la structure "segments 2D" est aisée à extraire, plusieurs chercheurs se sont intéressés à la mise en correspondance des segments de contours [AF85] [AL87] et [LL86].

Comme d'une part le nombre de segments dans une image est inférieur au nombre de points, et que d'autre part les attributs géométriques utilisables sont plus riches et plus discriminants que ceux attachés aux points, des algorithmes plus performants ont pu voir le jour. Horaud et Skordas [HS89] proposent pour leur part de réaliser une description de l'image sous forme d'un graphe qui tient compte des regroupements tels les jonctions entre segments. La mise en correspondance des segments dans ce cas, comporte une information plus générale permettant d'améliorer le nombre d'appariement de segments. Néanmoins, des problèmes de confusions persistent surtout dans le cadre binoculaire.

Un autre type de primitives porteur d'une information intrinsèque plus riche est également utilisés : les régions [WD88, Vin91]. Néanmoins, cette primitive séduisante au premier abord, pose des problèmes directement liés à son extraction. Comme nous l'avons souligné auparavant (cf. §2.5.2) la segmentation en régions est une opération instable associant souvent à une région réelle 3D, des régions 2D d'aspects différents d'une image à une autre. Ce qui entraîne une mise en correspondance délicate à mettre en œuvre.

4.1.2 Les contraintes en stéréovision

La plus importante difficulté de la mise en correspondance vient du fait qu'une primitive donnée dans une image peut être, à première vue, associée à plusieurs primitives de la seconde image¹. Se pose alors le problème du choix du bon candidat. Afin de lever les ambiguïtés et de permettre de dégager un ensemble d'appariement corrects, un certain nombre de contraintes ont été proposées.

Contrainte de la géométrie épipolaire

La géométrie du système de prise de vue stéréoscopique impose aux primitives homologues de respecter la contrainte épipolaire (cf. Section 2.3). Dans le cadre de la stéréovision binoculaire cette contrainte permet de réduire l'espace bidimensionnel de recherche des appariements potentiels, à un espace unidimensionnel : celui relatif aux deux droites épipolaires homologues. Cette contrainte constitue la plus forte contrainte géométrique utilisée lors de la mise en correspondance dans les systèmes stéréoscopique calibrés.

Dans le cadre de la stéréovision trinoculaire, la contrainte épipolaire déduite directement de la contrainte épipolaire binoculaire, est encore plus forte. La troisième image est généralement utilisée pour vérifier une hypothèse d'appariement entre les deux premières. En effet, soit (p_1, p_2) un couple de points candidats provenant des images Im_1 et Im_2 . Soient E_{13} et E_{23} les droites épipolaires dans l'image Im_3 associées respectivement au point p_1 et au point p_2 (Figure 4.1). Nous avons la propriété suivante :

Si le point d'intersection de E_{13} et E_{23} coïncide avec un point $p_3 \in Im_3$ alors le triplet (p_1, p_2, p_3) est un de points triplet d'homologues.

1. Comme le dit Bonnin [Bon91] "...rien ne ressemble plus à un segment d'une image qu'un autre segment d'une autre image.."

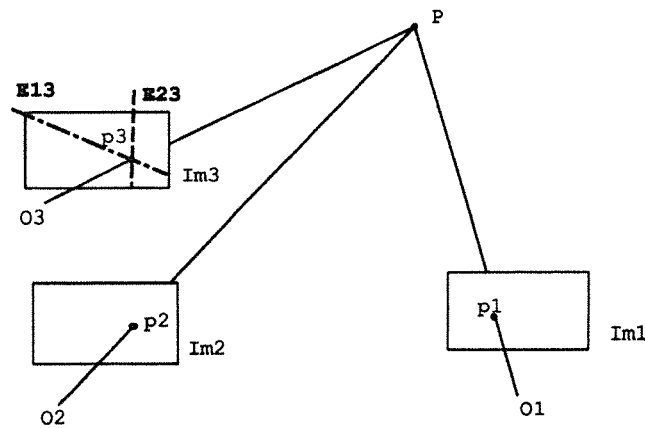


Figure 4.1. *Contrainte épipolaire dans un système trinoculaire. Le point p_3 est à l'intersection des droites épipolaires E_{13} et E_{23} calculées respectivement à partir des points p_1 et p_2 .*

Hypothèses de Marr-Poggio

Dans leur article [MP76], Marr et Poggio ont introduit des contraintes qui sont couramment utilisées, sous différentes formes, par tous ceux qui ont abordé la mise en correspondance en stéréovision. Ces hypothèses sont les suivantes :

- *l'unicité.* Cette hypothèse spécifie qu'à chaque point d'une image correspond au plus un point dans la deuxième image. Elle est généralement prise en compte par tous les algorithmes de mise en correspondance. Cependant, cette contrainte d'unicité est mise en défaut par les alignements accidentels ou par l'existence d'objets transparents dans la scène.
- *la continuité.* Pour la définir, introduisons tout d'abord la notion de disparité. Dans un banc stéréoscopique rectifié², la disparité d'un couple de points homologues est définie comme étant la différence entre les abscisses images de ces points. Intuitivement la disparité quantifie le paralaxe entre deux images et est fonction de la distance du point 3D au banc stéréoscopique. Partant du fait que les scènes observées sont composées de surfaces régulières, Marr et Poggio ont proposé de vérifier la continuité par morceaux de la fonction distance entre des couples de primitives appariés voisins. Cette contrainte fut traduite en exigeant la continuité par morceaux de la fonction disparité entre des couples de primitives appariés voisins.

Contraintes diverses

Par la suite, d'autres contraintes ont été proposées :

2. Lignes épipolaires horizontales.

- *la ressemblance d'attributs* relative soit à la photométrie comme la luminance, soit à la géométrie locale autour des primitives à appairer. Cette contrainte, bien qu'attractive, est généralement insuffisante pour lever les ambiguïtés.
- *la contrainte d'ordre* spécifie que l'ordre de présentation des primitives entre deux épipolaires homologues est préservé. Ceci n'est pas toujours vrai, il suffit d'observer que dans certain cas l'ordre des pattes d'une chaise n'est pas le même d'une image à l'autre.
- *la contrainte de planarité* particulière aux travaux de Takahashi et Tomita [TT88]. Après avoir mis en correspondance des segments 2D, Takahashi et Tomita proposent de vérifier que les triplets de segments supposés provenir des contours d'une unique surface plane vérifie une contrainte de planarité. Cette contrainte n'est autre que la contrainte de planarité utilisée par Bonnin que nous avons présenté dans le Chapitre 2. L'utilisation unique de cette contrainte globale comme contrainte d'appariement³ ne permet pas d'exploiter toute la puissance de cette dernière.

4.1.3 Les techniques de mise en correspondance

Différentes techniques de la mise en correspondance ont été utilisées :

- *la corrélation* est l'une des premières techniques de mise en correspondance qui a été utilisée. Dans le cadre de son travail, Moravec [Mor77] recherche la meilleure corrélation entre des fenêtres centrées sur des points caractéristiques, les points de Moravec. C'est une méthode qui coûte très cher en temps et qui n'est pas très efficace car très sensible au manque de texture et à la présence de structures répétitives.
- *la mise en correspondance hiérarchique*. L'idée de base de cette méthode est d'utiliser deux pyramides d'images de niveaux de résolutions différents -une pyramide pour chaque image stéréoscopique- et de les mettre en correspondance. La première mise en correspondance se fait au niveau le plus bas, celui donc qui contient un nombre d'information restreint mais "sûres". Cette mise en correspondance initiale permet par la suite de résoudre les conflits d'appariements à une échelle plus fine. Une propagation graduelle de la mise en correspondance des niveaux les plus bas aux niveaux les plus hauts est réalisée avec un affinement des contraintes mises en jeu. Cette méthode a été utilisée par Moravec [Mor77] en combinaison avec une méthode de corrélation permettant de réduire le temps de calcul de cette dernière. Gagalowicz *et al.* [Gag89, RG91] et [Vin91] utilisent également cette technique pour mettre en correspondance les régions qui généralement se présentent sous des formes différentes d'une image à l'autre.
- *la programmation dynamique*. Dans ce cas, la mise en correspondance est vue comme étant un problème de recherche d'un chemin optimal dans un graphe

3. A l'encontre de Bonnin [Bon91], Takahashi et Tomita ne l'utilisent plus lors de la reconstruction 3D.

où les nœuds représentent tous les couples d'appariements potentiels et dont la résolution se fait par des techniques de la programmation dynamique.

Cette méthode a été initialement utilisée par Baker et Binford [BB81] pour appairier les points de contours de deux images. Leur démarche consiste à effectuer une analyse ligne épipolaire par ligne épipolaire. La mise en correspondance d'une paire de lignes maximise un critère de ressemblance globale sous l'hypothèse de la préservation de l'ordre des primitives. Dans le cadre d'une mise en correspondance de régions fondée sur un algorithme de relaxation, Wrobel propose d'intégrer un algorithme programmation dynamique appliqué aux points de contours, pour imposer la contrainte de continuité des lignes contours, afin de mieux guider l'appariement des régions [MW84, WD88].

- *l'isomorphisme de sous graphe* utilisé par Horaud et Skordas [HS89] pour la mise en correspondance de segments. Ces auteurs construisent, à l'aide des segments, une description de l'image sous la forme d'un graphe tenant compte de certains regroupements caractéristiques tels les jonctions. La mise en correspondance se fait en utilisant des algorithmes de recherche d'isomorphisme de sous-graphe et donne de bons résultats.
- *les algorithmes de relaxation*. Le principe fondamental de ces techniques consistent à affecter un poids à une association entre deux primitives, puis à faire intervenir l'influence du voisinage pour faire évoluer ce poids : réduction en cas de conflit ou renforcement en cas de cohérence.

Marr et Poggio [MP76] ont présenté l'un des premiers algorithmes de relaxation en stéréovision donnant des résultats impressionnants dans le cas de stéréogrammes aléatoires. Ainsi, les primitives sont des points distribués aléatoirement sur des surfaces en trois dimensions relativement continues. Grimson [Gri81] a implanté une nouvelle version de cet algorithme en utilisant des points de contours et en introduisant l'utilisation d'une approche multirésolution.

- *les algorithmes fondés sur la prédiction-vérification des hypothèses (PVH)*. Pour diminuer les ambiguïtés dans les appariements, Ayache et Faverjon [AF85] ainsi que Robert de St Vincent [Vin86] proposent d'utiliser une méthode de propagation et vérification récursive d'hypothèses. Initialement ils associent à chaque segment les segments situés de son voisinage⁴. Une propagation récursive des hypothèses d'appariement sur les segments voisins est réalisée jusqu'à ce qu'il n'y ait plus de voisins à appairier. Par la suite, l'hypothèse initiale n'est retenue que si elle a permis de générer un nombre suffisamment élevé d'appariements consistants.
- *la stéréovision trinoculaire*. Yachida *et al.* [YKK86] ont présenté un des premiers système trinoculaire pour la mise en correspondance de points de contours avec pour unique contrainte la contrainte épipolaire trinoculaire. Un algorithme fort simple en résulte mais la pauvreté de l'information portée par le point et les erreurs de localisation de ces derniers a encouragé les chercheurs à utiliser des primitives plus riches comme le segment de contours. Ainsi, Ayache et Lustman

4. Ceci est fait au moyen de l'utilisation de la technique des baquets, approche que nous avons repris pour notre implantation.

[AL87] proposent un algorithme trinoculaire pour les segments qui en plus des contraintes de compatibilité d'attributs introduit la contrainte épipolaire trinoculaire. Cet algorithme, dont nous nous sommes inspirés, réalise un appariement assez fiable, et fournit de nombreux triplets de segments permettant une reconstruction 3D plus riche.

4.2 Appariement de facettes 2D

Nous présentons dans cette partie la mise en correspondance trinoculaire des facettes 2D. Rappelons qu'une facette 2D est définie par une chaîne de segments 2D; ainsi l'appariement de facettes se traduit par une mise en correspondance de chaînes de segments 2D. De ce fait, la mise en correspondance des facettes 2D délivre non seulement des triplets de facettes mais également des triplets de segments. L'algorithme d'appariement que nous avons développé se compose de deux étapes :

1. *une mise en correspondance globale* qui permet de former les triplets de facettes homologues,
2. *une mise en correspondance locale* qui apparie les segments 2D provenant d'un triplet de facettes 2D homologues en utilisant une méthode similaire de celle développée par Ayache et Lustman [AL87].

Les contraintes mises en jeu pour cet appariement sont décrites dans la section 4.2.1, l'algorithme est détaillé dans la section 4.2.2.

L'implantation de la partie mise en correspondance des facettes 2D nous a été facilitée par le travail effectué par J.C Di Martino dans le cadre de son DEA [Mar91]. Le sujet traité dans ce travail, consistait à implanter la mise en correspondance locale. Par la suite, nous avons souvent réutilisé des procédures développées par Di Martino, qui, par conception, sont compatibles avec notre structure de données, pour réaliser la partie mise en correspondance de facettes 2D du module de stéréovision du constructeur de facettes 3D.

4.2.1 Les contraintes de mise en correspondance de facettes

Pour réaliser l'appariement nous mettons en jeu quatre critères relatifs aux segments 2D appartenant à une facette :

- (a) *la contrainte d'épipolarité trinoculaire* inhérente au banc stéréoscopique utilisée dans le cadre de notre application.
- (b) *la contrainte de direction* [AB80] [Sko88] correspondant à la disparité en orientation relative à l'intervalle d'angles $[\theta_{min}, \theta_{max}]$ permis pour un segment candidat à un appariement (Figure 4.2).

Reprenons l'explication introduite par Ayache [Aya90] permettant de bien saisir la signification de cette contrainte. Nous savons qu'un point p_j homologue d'un point p_i se situe obligatoirement sur l'épipolaire $E(p_i)$ associée à p_i dans l'image Im_j . En utilisant les données du calibrage, il est possible de décrire le déplacement $p_i(P)$ sur la droite épipolaire $E(p_i)$ en fonction du déplacement du point 3D P

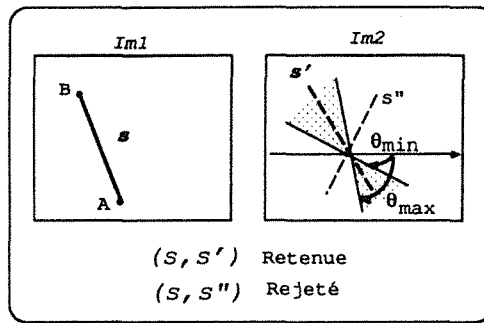


Figure 4.2. *Contrainte de direction.*

sur la ligne de point de vue décrite par O_i, p_i . Autrement dit, quand la distance de déplacement de P est comprise entre $[\rho_{min}, \rho_{max}]$, ce qui est le cas dans le cadre d'une scène d'intérieur où les points 3D se déplacent entre $[0m, 10m]$ par exemple, alors p_j se déplace sur l'épipoilaire dans un intervalle bien délimité: $[d_{min}(\rho_{min}), d_{max}(\rho_{max})]$. Ainsi, à tout segment $s_i = [a_j, b_j]$, il est possible de déterminer pour chacune de ses extrémités l'intervalle de déplacement de son homologue dans l'image Im_j . Ce calcul permet en fait d'associer à tout segment s_i de l'image Im_i un intervalle un intervalle $[\theta_{min}, \theta_{max}]$ de directions permises pour son homologue potentiel.

- (c) *la contrainte de positionnement* [DWD92] relative à la notion de facette 2D. Tous les segments des trois images sont orientés de la même manière (cf. §3.1.2); ceci permet de définir la "droite" et la "gauche" d'un segment. Nous pouvons donc, pour chaque segment faisant partie d'une facette, déterminer si cette dernière se trouve à sa droite ou à sa gauche. La contrainte que nous imposons à un couple de segments homologues faisant chacun partie d'une facette 2D donnée, et d'avoir cette facette située du même coté (cf. Figure 4.3).

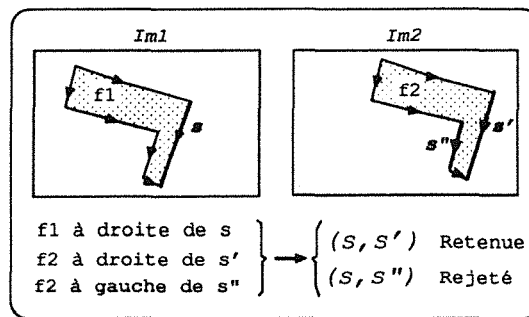


Figure 4.3. *Contrainte de positionnement.*

- (d) *la contrainte d'unicité*. Appliquée à la primitive point, cette contrainte impose à tout point de Im_1 de n'appartenir qu'à un unique triplet d'homologue⁵. Cette

5. Cette contrainte suppose qu'il n'existe pas de points 3D de la scène alignés avec le centre objet optique d'un caméra.

contrainte ne peut pas être appliquée dans ces termes, aux primitives segments. En effet, remarquons que les segments peuvent présenter une segmentation différentes d'une image à l'autre. Ainsi, un segment s_1 dont la projection sur l'image Im_j est décomposée en deux segments, doit être apparié à ces deux segments. En fait, la contrainte d'unicité pour les segments va se traduire comme suit [Aya90] : le point milieu d'un segment ne peut admettre, dans chacun des deux autres plans images, qu'un unique homologue.

Nous introduisons également un critère global pour la mise en correspondance de facettes qui est le suivant :

- (e) *la contrainte de l'ordre d'enchaînement.* Soit $Appar : f_1 \rightarrow f_2$ la fonction d'appariement qui associe la facette f_2 à f_1 . Pour chaque segment de la facette f_1 la fonction $Appar$ définit le segment homologue de la facette f_2 . Cette application $Appar$ doit être soit une fonction croissante en prenant comme ordre sur les segments l'ordre d'enchaînement des segments de la facette.

4.2.2 L'algorithme de mise en correspondance

Après la définition des différentes contraintes utilisées, présentons le processus de mise en correspondance des facettes provenant des trois images du système trinoculaire (Im_1, Im_2, Im_3). Soient $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ les ensembles de facettes respectifs des trois images stéréoscopiques. L'algorithme se compose de deux étapes explicitées ci-après.

A- La mise en correspondance globale

La mise en correspondance globale consiste à associer à une facette 2D f_1 d'une image, Im_1 par exemple, deux facettes provenant respectivement de Im_2 et Im_3 qui sont supposées être ses homologues. Pour ce faire, nous sélectionnons le plus grand segment s_1 non apparié de f_1 et nous recherchons les homologues de s_1 dans Im_2 et Im_3 . Notons p le point milieu de s_1 et $E_{12}(p)$ et $E_{13}(p)$ les épipolaires correspondant au point p dans les plans images respectifs Im_2 et Im_3 . La recherche des homologues de s_1 se fait parmi les segments 2D $s_2 \in Im_2$ (respectivement $s_3 \in Im_3$) appartenant à des facettes de \mathcal{F}_2 (respectivement de \mathcal{F}_3) respectant les contraintes suivantes :

- (i) s_2 (respectivement s_3) intersecte l'épipolaire $E_{12}(p)$ dans Im_2 (respectivement $E_{13}(p)$ dans Im_3). Cette contrainte n'est autre que la contrainte d'épipolarité binoculaire,
- (ii) s_2 (respectivement s_3) vérifie les contraintes (b) et (c).

Cette recherche permet en un premier temps de sélectionner des facettes 2D dans chaque image Im_2 et Im_3 qui puissent être des homologues potentiels de la facette f . Dans l'exemple illustré par la Figure 4.4, nous associons à f d'une part les facettes (f_1, f_2) de \mathcal{F}_2 et d'autre part les facettes (fa, fb) de \mathcal{F}_3 . Notons qu'à chacune de ces facettes candidates, une hypothèse de mise en correspondance binoculaire du segment s initial a été réalisée, délivrant quatre couples d'appariements candidats :

$$\left\{ \begin{array}{l} \text{Relativement à } Im_2 : (f, f_1) \leftrightarrow (s, s_1) \text{ et } (f, f_2) \leftrightarrow (s, s_2) \\ \text{Relativement à } Im_3 : (f, fa) \leftrightarrow (s, sa) \text{ et } (f, fb) \leftrightarrow (s, sb) \end{array} \right.$$

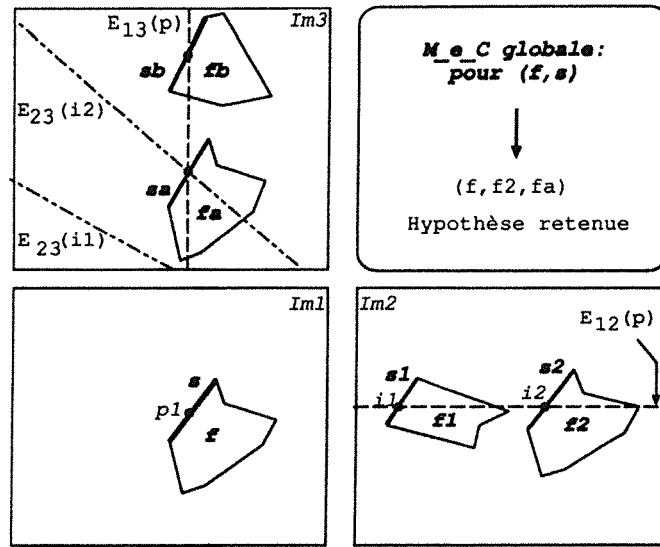


Figure 4.4. Résultat de la mise en correspondance globale.

La mise en œuvre de la contrainte d'épipolaire trinoculaire (a) permet d'élaguer parmi tous ces couples de segments ceux qui ne vérifient pas la contrainte d'épipolarité entre Im_2 et Im_3 . Les triplets restants constituent un ensemble noté *Trip_Fac_Cand*. Dans l'exemple illustré par la Figure 4.4, ceci permet de garder uniquement l'hypothèse du triplet (f, f_2, f_a) . Après cette étape de mise en correspondance globale, il s'agit d'activer la mise en correspondance locale sur chacun des triplets de facettes appartenant à l'ensemble *Trip_Fac_Cand* permettant de conserver le meilleur appariement.

Fonction $M_C_Globale(Im_1, Im_2, Im_3)$

$Trip_Fac_Cand = \emptyset$

Pour chaque facette non appariée $f \in \mathcal{F}_1$ Faire

- Prendre $s \in f$: s plus grand segment de f
- Prendre p , le point milieu de s
- Construire $E_{12}(p)$ et $E_{13}(p)$, les épipolaires correspondant au point p
- Construire le sous ensemble de facettes $F' \subseteq \mathcal{F}_2$ (resp. $F'' \subseteq \mathcal{F}_3$) :

Chaque facette $f_i \in F'$ a un segment s'_i telle que

$$E_{12}(p) \cap s'_i \neq \emptyset$$

Contraintes (b) et (c) vérifiées

/** Former l'ensemble des triplets retenus **/

- Pour chaque $f_i \in F'$ Faire

Si Trouver $f_j \in F''$ tel que (s, s'_i, s''_j) vérifie (a)

Alors $Trip_Fac_Cand = Trip_Fac_Cand \cup \{(f, f_i, f_j)\}$

Fin Pour

- $M_C_Locale(Trip_Fac_Cand)$

Fin Pour

Fin Fonction *M_C_Globale*

Remarquons que dans certains des cas particuliers, une facette 2D de l'image Im_1 peut correspondre à plus d'une facette 2D homologue de l'image Im_2 comme illustré dans la Figure 4.5. La phase de mise en correspondance globale de f appliquée entre (Im_1, Im_2, Im_3) ne retient que le triplet (f, f_1, fa) . Il est évident que le triplet (f, f_2, fa) est également à retenir. Pour ce faire, la fonction est relancée trois fois, partant à chaque fois d'une nouvelle image de base ce qui permet de visiter toutes les facettes des trois images :

$$\begin{cases} M_C_Globale(Im_1, Im_2, Im_3) \\ M_C_Globale(Im_2, Im_3, Im_1) \\ M_C_Globale(Im_3, Im_1, Im_2) \end{cases}$$

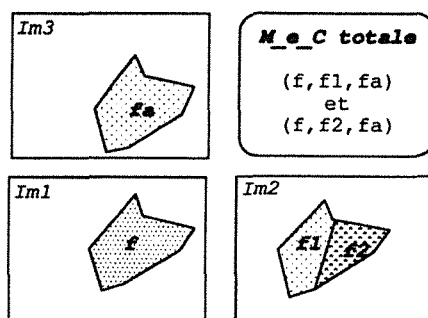


Figure 4.5. Appariement d'une facette f de Im_1 qui se présente sous forme de deux facettes dans l'image Im_2 .

B- La mise en correspondance locale

La procédure précédente permet pour une facette donnée de construire les hypothèses de triplets de facettes homologues qui lui sont relatifs. L'étape suivante consiste à effectuer l'appariement local permettant de mettre en correspondance les segments des triplets de facettes homologues. Ainsi, partant d'un triplet de facettes 2D homologues de $TF = (f_1, f_2, f_3) \in Trip_Fac_Cand$, l'étape de mise en correspondance locale délivre leur ensemble de triplets de segments 2D homologues qui en découle :

$$(f_1, f_2, f_3) \rightarrow \{(s_1^1, s_1^2, s_1^3), \dots, (s_n^1, s_n^2, s_n^3)\} \text{ où } \begin{cases} f_i = (s_1^i, \dots, s_p^i), p \geq n \\ \text{et} \\ (s_1^i, \dots, s_n^i) \subset (s_1^i, \dots, s_p^i) \end{cases}$$

Ainsi, pour chaque triplet de *Trip_Fac_Cand*, une recherche classique de mise en correspondance des segments, mettant en œuvre les contraintes (a) (b) (c) (d) composant chacune des facettes, est réalisée.

A la fin de cette recherche, seuls les triplets répondant à la contrainte d'enchaînement (e) et ayant aboutit à une mise en correspondance d'au moins deux segments de la facette initiale, sont gardés. Nous retenons le triplet de facettes de l'ensemble *Trip_Fac_Cand* qui a permis le nombre maximal de mises en correspondance.

4.3 Résultats expérimentaux

Comme cette partie présente peu d'originalité, nous nous contenterons de montrer quelques résultats puisque implicitement les résultats du chapitre 6, montre en fait et les résultats du constructeur de facettes et celui du module de mise en correspondance.

La Figure 4.6 montre le triplet d'images des segments extraits de l'image de la mire. En fait, la phase de construction de facettes 2D a extrait de chacune des images des facettes 2D. Les segments portant des numéros sont des segments faisant parties de facette 2D. Nous remarquons que tous les segments ne porte pas de numéros indiquant que lors de l'appariement de facettes ces segments ne sont pas étudiés. Les ensembles de facettes 2D extrait chacun d'une image forment la donnée en entrée de notre logiciel de construction de facettes 2D.

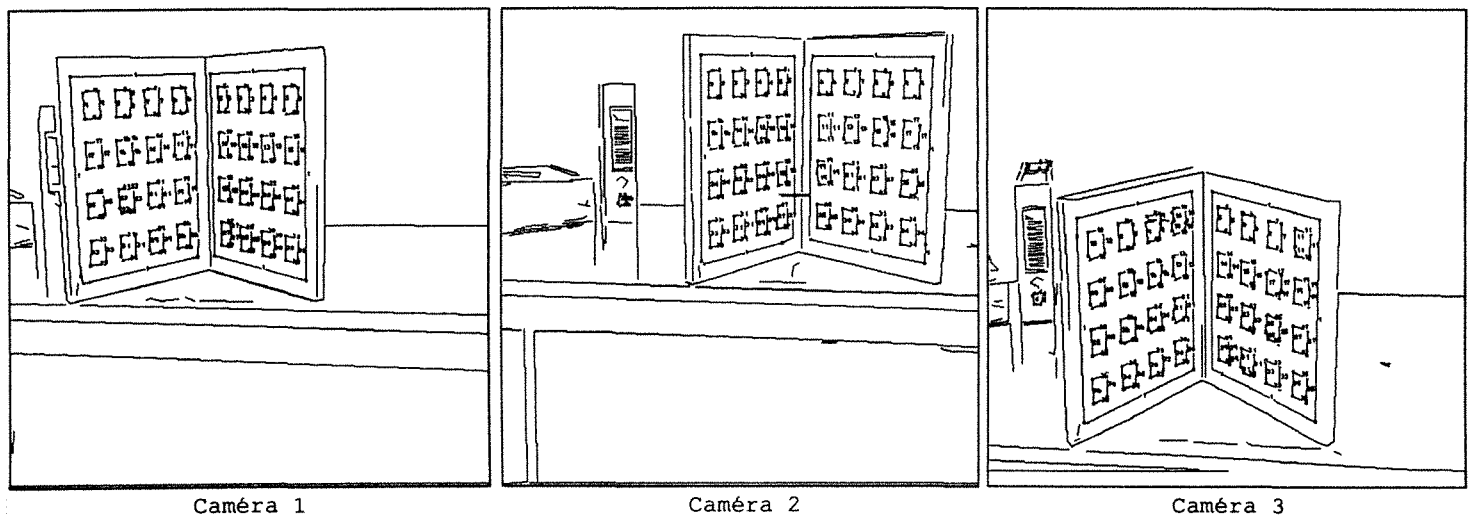


Figure 4.6. Représentation du triplet d'images de segments. Seuls les segments portant un numéro font partie d'une facette.

La Figure 4.7 représente l'ensemble de facettes appariés. Les triplet de facettes homologue sont composés des facettes 2D provenant de chacune des images est portant le même numéro.

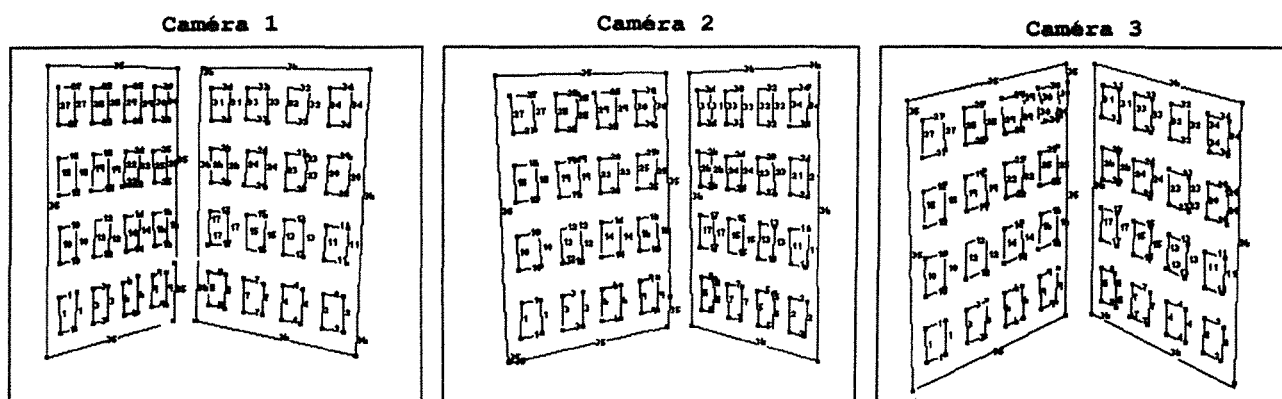


Figure 4.7. Résultat d'appariement de facette 2D.

Par la suite nous montrons un autre résultats de mise en correspondance. Cet exemple nous permet de soulever une remarque importante: la validité de l'hypothèse de planarité des facettes 2D n'est pas toujours vérifiée même en dehors de tout phénomène d'occultation. En effet, la Figure 4.8 montre le résultat du constructeur de

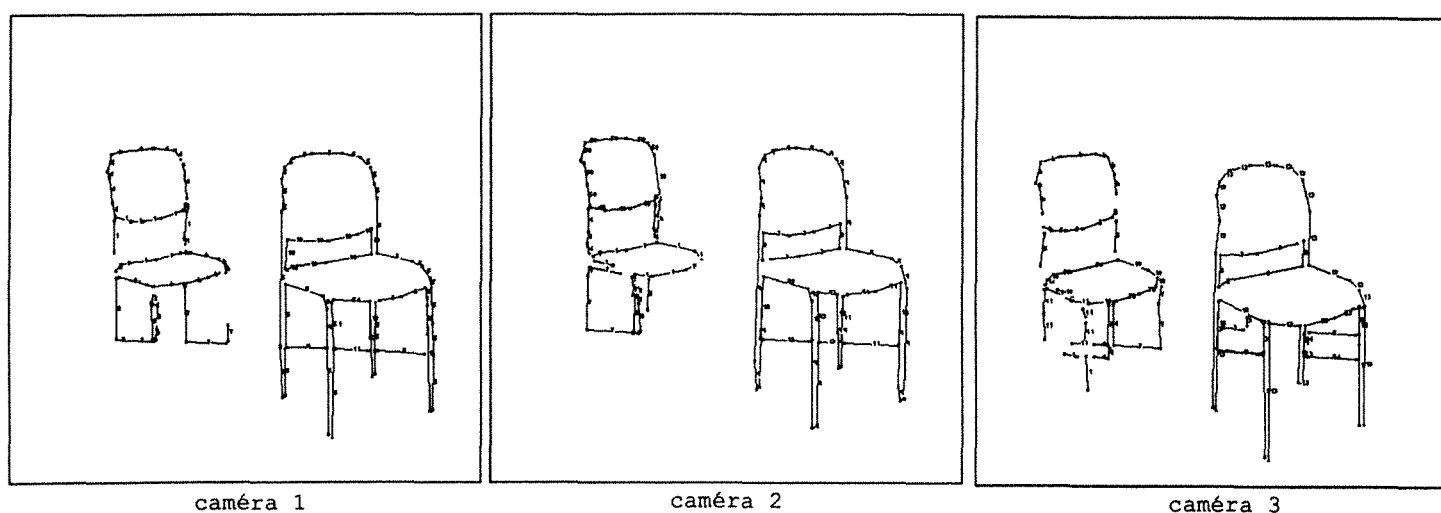


Figure 4.8. Les images des triplets de facettes.

facette sur un triplet d'image d'une scène composée de deux chaises. Nous pouvons alors remarquer que, dans chacune des images, le constructeur de facettes regroupe toutes les parties de la chaise de droite pour en faire une unique facette 2D. Si la mise

en correspondance de ces facettes 2D ne pose aucun problème (Figure 4.9) il est vrai aussi qu'aucune information indiquant la non validité de la planarité n'a pu être trouvée. Cette constatation nous a encouragé à chercher une technique qui permet après avoir réalisé la mise en correspondance de détecter la non planarité, évidente pour l'humain, de certains triplet de facettes 2D. Les deux chapitres suivant apporte notre contribution dans la résolution de ce problème.

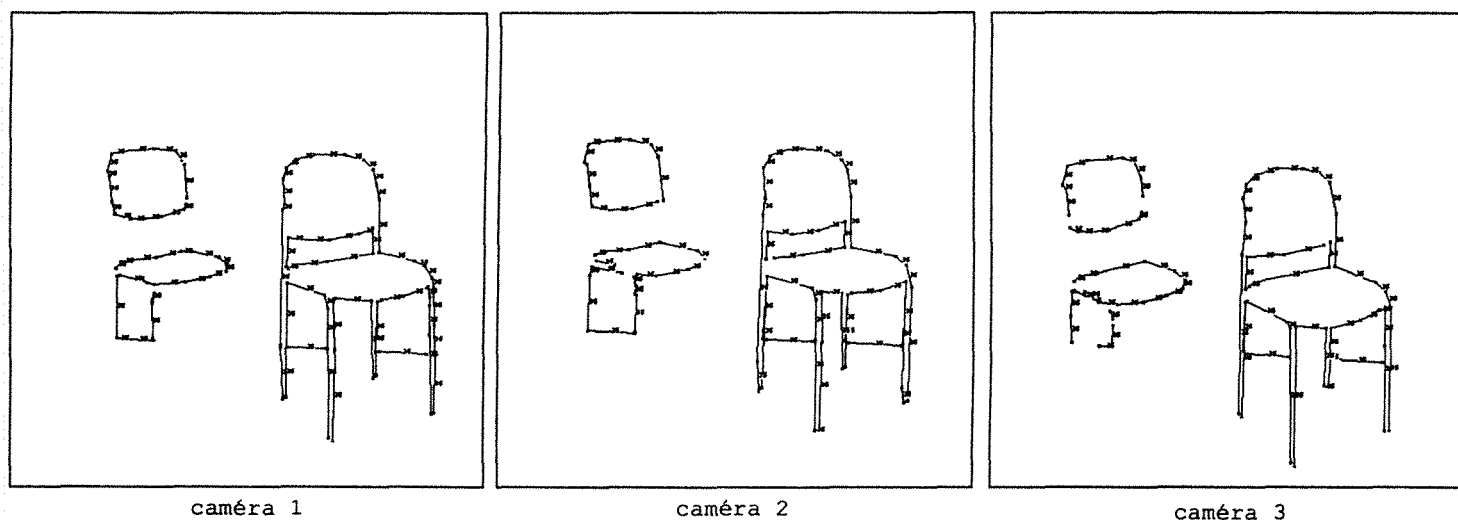


Figure 4.9. Les triplets de facettes homologues.

4.4 Discussion

L'appariement de facettes revient essentiellement à un appariement trinoculaire de segments 2D sous des contraintes inhérentes à la structure de facettes 2D. Ainsi, l'algorithme qui en découle permet de sélectionner plus rapidement les paquets de segments à appairer avec le module de mise en correspondance locale ce qui renforce la fiabilité des appariements des segments faisant partie d'une facette. Cependant, il est à remarquer que le nombre d'appariements de segments a diminué puisque tous les segments d'une image n'appartiennent pas forcément à une facette.

Soulignons enfin que l'algorithme d'appariement de facette se complète très bien par un algorithme d'appariement de segments 2D dans un système trinoculaire. En effet, l'optique générale du niveau intermédiaire de MAGIC est de délivrer au haut niveau des facettes 3D et des segments 3D. Il serait alors judicieux, après avoir réalisé la mise en correspondances des facettes 2D, d'effectuer la mise en correspondance des segments 2D de l'image qui n'ont pas été traités. Dans ce cas l'architecture du système global de mise en correspondance résulte d'un couplage entre le module de mise en correspondance de facettes 2D de notre système et celui des segments 2D (cf. Figure 4.10). L'ensemble des segments 2D serait décomposé en deux parties disjointes: les segments faisant partie d'au moins une facette 2D et les segments dits "libres" qui

n'appartiennent à aucune facette. Dès lors, ayant filtré l'ensemble initial de segments de ceux qui ont été appariés par le processus d'appariement de facettes, nous pouvons lancer le processus d'appariement de segments 2D sur le restant des segments.

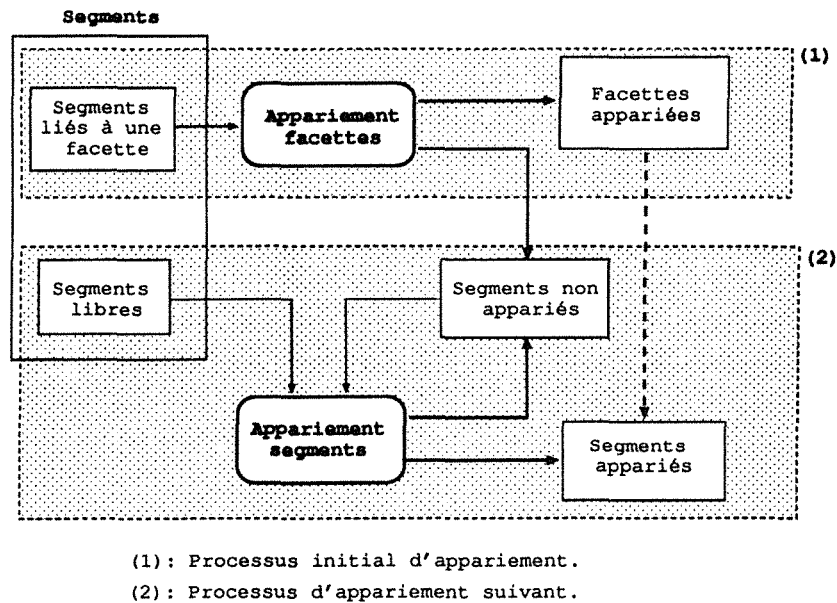


Figure 4.10. *Processus d'appariement de segments 2D avec la notion de facettes 2D.*

5

Géométrie projective et test de coplanarité

Les facettes 2D sont supposées provenir de la projection perspective de surfaces 3D planes (Chapitre 3 et [CM91]). Il est évident que cette affirmation n'est pas toujours vraie : des phénomènes tels l'occultation ou la mauvaise segmentation, peuvent entraîner la construction de facettes 2D ne correspondant à aucune surface plane 3D. Notre objectif est d'écartier ces fausses facettes avant de passer à la phase de reconstruction 3D. A cet effet, nous proposons un outil permettant de vérifier l'hypothèse de planarité des facettes 3D, à partir des n-uplets de facettes 2D appariées; nous considérons donc que l'étape d'appariement des primitives est déjà traitée. La vérification de la planarité se fait en utilisant des principes de la géométrie projective et a essentiellement été inspirée par les travaux de Mohr [Moh92], de Quan [QM91] et de Morin [MMG91] sur l'application de la géométrie aux problèmes de la vision artificielle.

Nous consacrons ce chapitre uniquement à l'étude théorique du problème de la vérification de la coplanarité d'un ensemble de primitives 3D (points, segments ou droites) à partir de leurs images 2D appariées. La mise en oeuvre de ces contraintes dans le constructeur de facettes 3D est explicitée dans le chapitre suivant.

Ce chapitre s'articule autour de neuf sections. Dans la Section 5.1 nous présentons quelques utilisations pratiques des résultats de la géométrie projective en vision artificielle. Les trois sections suivantes, § 5.2, § 5.3 et § 5.4, sont réservées à introduire un ensemble de définitions plus spécifiques à notre objectif. Puis, nous passons à la présentation des différents résultats concernant l'étude de la coplanarité. Après avoir introduit dans la Section 5.5 la contrainte de coplanarité d'une droite et d'un plan, nous énonçons dans la Section 5.6, la contrainte relative à la coplanarité d'un ensemble de triplets de points homologues. La Section 5.7 est réservée à la présentation d'une contrainte de coplanarité de couples de droites, ou de segments, homologues. Dans l'avant dernière section de ce chapitre, nous présentons deux autres approches, décrites dans la littérature, et qui permettent l'étude de la coplanarité de points détectée en 2D. Une synthèse clot ce chapitre.

5.1 Utilisation des invariants projectifs en vision

5.1.1 Pourquoi utiliser les invariants projectifs ?

Revenons sur le modèle sténopé qui est souvent adopté pour représenter la géométrie d'une caméra. Ce modèle permet d'assimiler la prise d'image à une transformation perspective parfaite dont nous avons présenté la formulation matricielle dans la Section 2.1. Les paramètres de cette matrice sont déterminés à l'étape de calibrage et la phase de reconstruction 3D par triangulation utilise explicitement ces données. Cependant, l'étude d'une image peut être menée soit en utilisant des principes de reconstruction qui repose sur les invariants de la géométrie projective soit en utilisant les données issues de la phase de calibrage. Dans la Figure 5.1 nous illustrons les deux cheminements possibles :

1. dans le premier cas, la phase de calibrage délivre les paramètres de la transformation perspective utilisée. La reconstruction se fait alors par triangulation,
2. dans le second cas, les paramètres des caméras sont des inconnues du problème. Ici aussi, deux approches sont possibles :
 - (i) d'une part, nous avons l'approche développée par Sparr [SN90, Spa92] qui partant de l'hypothèse d'une projection perspective parfaite propose de retrouver la position relative des points de la scène en utilisant le concept de descripteur affine d'une configuration de points. Ce descripteur contient en fait l'information affine de la position relative des points entre eux. L'étude des propriétés de ce descripteur sous la projection perspective, permet de dégager un théorème général qui lie le descripteur d'une configuration donnée de points à celui de son image. Ceci permet par la suite de réaliser une reconstruction relative. Cette approche est bien adaptée aux scènes composées de parallélogramme. Remarquons que sous l'unique hypothèse de projection perspective parfaite il est possible de dégager en monoculaire l'information : point de fuite. Cette information permet de former des paquets de segments 2D, où chaque paquet représentent l'image d'une classe de segments 3D tous parallèles entre eux. La recherche des points de fuite ne nécessite aucune connaissance a priori sur l'image, si ce n'est que nous supposons que l'effet de perspective est visible dans l'image. Cette recherche fait l'objet de divers travaux. Certains chercheurs utilisent les points de fuite pour réaliser des appariements entre un modèle et une image [Qua89], d'autres s'en servent pour localiser les points de prise de vue dans le cas d'images aériennes [Lut90].
 - (ii) d'autre part, nous retrouvons les approches qui supposent avoir en plus du modèle sténopé la donnée *a priori* d'un ensemble de points appariés dits *points de références*. Dans ce cas, utilisant les invariants projectifs ces approches arrivent à extraire des informations pertinentes de l'image étudiée. C'est dans ce cadre d'étude que nous plaçons la méthode que nous présentons dans ce chapitre.

Revenons au cas (ii). L'utilisation des invariants projectifs pour résoudre les problèmes posés par l'étude des images en n'ayant aucune connaissance des paramètres des caméras, a été abordée par différentes personnes au cours de ces vingt dernières années. Elle aboutit généralement à des reconstructions relatives aux points de référence.

Duda et Hart montrent, dans le chapitre intitulé "*Projectives Invariant*" de leur livre [DH73], qu'une reconstruction des images des centres optiques d'un système binoculaire est possible, en ayant uniquement la connaissance de quatre couples de points homologues coplanaires. Cette approche ouvre des perspectives sur la reconstruction fondée sur l'utilisation d'invariants projectifs et sur la reconstruction relative. Plus récemment, différents travaux [MMIQ91, Moh92, Fau92] concernant l'utilisation de la géométrie projective et plus précisément de l'application du principe de conservation du birapport, décrivent des techniques qui permettent d'extraire différentes informations 3D des images 2D : ainsi, ils expliquent comment retrouver l'image d'un point de l'espace, déterminer la géométrie épipolaire ou encore localiser les centres optiques des caméras, uniquement à partir de la donnée d'un certain nombre de points appariés. Ces travaux présentent une nouvelle approche de la problématique de la vision.

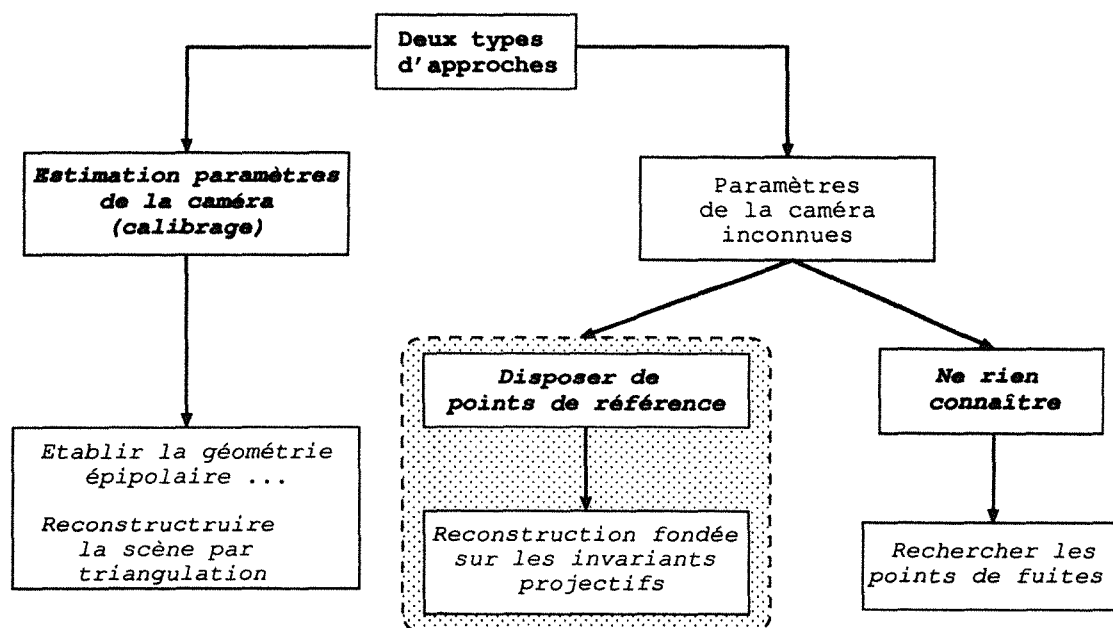


Figure 5.1. Comment aborder l'étude des images 2D ?

5.1.2 Et dans notre cas ?

Dans notre cadre d'application, nous disposons de toutes les données de calibrage. Il semble alors nécessaire d'expliquer la démarche qui nous a conduit à utiliser les invariants projectifs. Nous avons exprimé notre intention de rechercher dans l'espace 2D, le plus d'informations possibles concernant des propriétés 3D : planarité et coplanarité de facettes, afin de mieux contraindre la construction des facettes 3D (Section 2.5.1). La réponse à notre ambition consiste à trouver des propriétés projectives qui permettent de décrire la planarité 3D d'une facette à partir de la donnée de ses images.

Dans l'espace 3D euclidien, la planarité d'un ensemble de points 3D P_{3d} peut être caractérisée par l'étude de l'intersection de toutes les droites passant par deux points de P_{3d} avec un plan Π donné. Ce plan Π représentera par la suite *un plan de référence*. En effet, un ensemble de points 3D P_{3d} est dit être composé de points coplanaires si et seulement si nous sommes dans l'un de ces deux cas :

1. tous les points de P_{3d} appartiennent à Π ,
2. les points d'intersections des droites passant par deux points de P_{3d} avec le plan Π sont tous alignés.

Cette approche pour caractériser la coplanarité d'ensemble de points 3D, peut être transcrite en une propriété projective utilisant les images des points composant P_{3d} et la donnée d'images de 4 points qui définissent une base projective de Π . Introduisons à présent les différentes définitions et propriétés projectives utiles pour notre problème.

5.2 Rappels sur la géométrie projective

Ayant assimilé le mécanisme de prise de vue d'une image au mécanisme d'une projection perspective parfaite¹, nous nous trouvons alors dans le cadre de la géométrie projective. Utilisant cet outil mathématique fondamental, il devient possible de caractériser les transformations subies par les propriétés 3D attachées aux primitives, tel le parallélisme entre droites, lors d'une projection perspective. Ceci permet ensuite de déterminer le type d'information 3D pouvant être extrait projectivement d'une ou de plusieurs images 2D.

La géométrie du projectif prend ses sources dans la géométrisation de la perspective. Ceci se matérialise par l'adjonction des points à l'infini² à un espace affine permettant ainsi de manipuler ces points particuliers. Leur introduction à donner naissance à de nouveaux axiomes formalisant ainsi la géométrie projective.

La littérature propose de nombreux ouvrages concernant la géométrie projective ; le livre [Bko91] permet au lecteur de connaître les origines de cette géométrie et sur les changements qu'elle a occasioné dans le raisonnement mathématique. Certains articles comme [MA90] ou [Fau92] par exemple, illustrent l'application directe des résultats de la géométrie projective à la vision. Les ouvrages [Cox74] [Ber77] et [Tis83] nous ont fournis les définitions que nous avons utilisés et que nous présentons ci-après.

5.2.1 Espace projectif, coordonnées homogènes et homographies

1. Nous avons adopté le modèle de sténopé pour la caméra.
 2. Chaque point à l'infini est associé à un faisceau de droites parallèles. Il représente le point d'intersection de ce faisceau en projectif.

Définition 8 Soit V un espace vectoriel de dimension $n + 1$ sur le corps des réels \mathcal{R} . L'espace projectif $P(V)$ est défini comme l'espace quotient de $V - \{0\}$ par la relation d'équivalence :

$$\text{Si } X \text{ et } Y \in V, X \sim Y \iff \exists(\lambda \in \mathcal{R}, \lambda \neq 0), \text{ tel que } X = \lambda Y.$$

Ainsi X et Y représentent le même point p de l'espace projectif. Le $(n+1)$ -uplet $X = (x_1, \dots, x_{n+1})$, défini à un facteur près, est appelé les coordonnées homogènes du point p . Les points de coordonnées homogènes $(x_1, \dots, x_n, 0)$ définissent les points à l'infini.

Lorsque V est de dimension finie $(n + 1)$, la dimension de l'espace projectif $P(V)$ est alors de n . Tout espace projectif de dimension 2 est appelé *plan projectif*. Un espace projectif à une dimension est appelé *droite projective*.

Définition 9 Toute application linéaire en les coordonnées homogènes est une *homographie*.

De ce fait, la projection centrale (projection perspective) est une homographie. Remarquons que cette transformation ne conserve pas les invariants habituels de l'espace euclidien tels la distance entre deux points, l'angle entre deux vecteurs ou encore le parallélisme entre deux droites.

5.2.2 L'invariant de base de la géométrie projective: le birapport

Comme nous venons de le souligner, les invariants euclidiens conservés par le groupe des isométries ne le sont plus par les homographies. En fait, un nouveau et unique invariant de base existe pour ces applications: il s'agit du *birapport*.

Définition 10 Etant donnés 4 points (A, B, C, D) d'une droite projective, le birapport de ces points, noté $[A, B, C, D]$, est défini comme suit :

$$[A, B, C, D] = \frac{\overline{CA}}{\overline{CB}} * \frac{\overline{DB}}{\overline{DA}} \text{ où } \overline{XY} \text{ est une mesure algébrique de } XY$$

L'ordre des points a une importance dans le calcul du birapport. En réalité, étant donné quatre points, il existe six birapports différents suivant l'ordonnement de ces points. Mais ces birapports ne sont pas indépendants et il est facile de vérifier que :

$$\begin{cases} [A, B, C, D] = [A, B, D, C]^{-1} = [C, D, A, B] \\ [A, B, C, D] + [A, C, B, D] = 1 \end{cases}$$

Théorème 1 Le birapport est conservé (donc est un invariant) par toute homographie, en particulier par la projection perspective (qui, pour nous représente la transformation caractérise la prise de vue d'une image par une caméra).

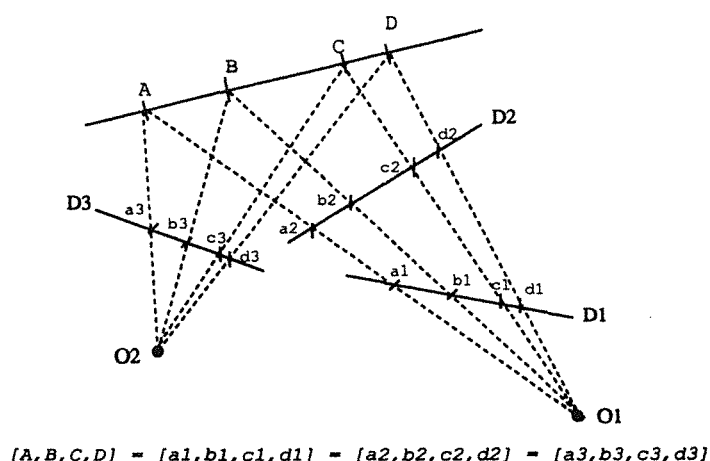


Figure 5.2. Conservation du birapport.

la Figure 5.2 illustre la conservation du birapport de quatre points (A, B, C, D) alignés, par la projection à partir du point O_1 sur une droite D_1 et D_2 . Cette quantité est également conservée par la projection de centre O_2 sur la droite D_3 .

Définition 11 Le birapport d'un faisceau de 4 droites (l_1, l_2, l_3, l_4) concourantes en O , est défini comme étant le birapport $[A, B, C, D]$ des points d'intersections du faisceau de droites avec une droite quelconque ne passant pas par le point O (Figure 5.3).

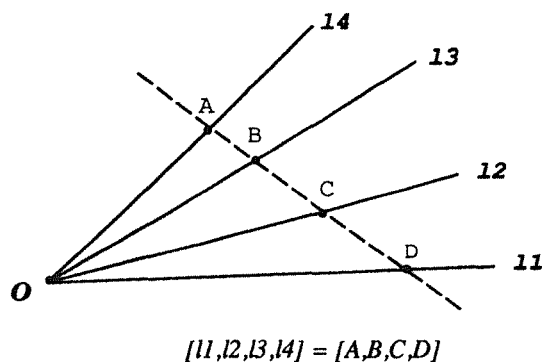


Figure 5.3. Birapport d'un faisceau de droites concourantes.

Le birapport d'un faisceau de droites est indépendant du choix de la droite qui coupe ce faisceau. Ce birapport s'exprime également au moyen des sinus des angles entre les droites du faisceau :

$$k = \frac{\sin(l_4 l_2)}{\sin(l_3 l_2)} * \frac{\sin(l_3 l_1)}{\sin(l_4 l_1)}$$

Définition 12 Le birapport d'un faisceau de 4 plans est défini comme étant le birapport des quatre droites d'intersections de ces plans avec un cinquième plan quelconque n'appartenant pas au faisceau.

5.2.3 Base projective

Une base d'un espace vectoriel V de dimension n est représentée par n vecteurs, alors qu'un espace affine de dimension n admet une base à $n + 1$ points. La base d'un espace projectif de dimension n est définie par $(n + 2)$ points.

En effet, rappelons que les coordonnées homogènes d'un point p de l'espace projectif $P(V)$ sont définies à un facteur près dans une base vectorielle \mathcal{E} de V . Ces coordonnées sont celles des points P dans \mathcal{E} qui correspondent à p . La donnée de $(n + 1)$ points $p(e_0), \dots, p(e_n)$ libres dans l'espace projectif $P(V)$ permet de reconstruire une infinité de bases vectorielles de V . Ces bases sont toutes définies par les mêmes vecteurs à un facteur de proportionnalité près (la matrice de passage entre ces bases est une matrice diagonale). Afin de fixer une base unique permettant de définir les coordonnées homogènes des points de $P(V)$, il faut définir une contrainte sur ces $(n + 1)$ points entraînant cette unicité. Cette contrainte se traduit par l'ajout d'un nouveau point $p(e_{n+1})$ aux $n + 1$ premiers.

Définition 13 Une base d'un espace projectif $P(V)$ de dimension n est formée de $(n + 2)$ points tels qu'aucun sous-ensemble de $(n + 1)$ points n'appartienne à un hyperplan.

De ce fait, soit (e_0, \dots, e_n) une base de \mathcal{R}^{n+1} alors $(p(e_0), \dots, p(e_n), p(e_0 + \dots + e_n))$ un repère projectif de \mathcal{P}^n . Dans le cas d'un plan projectif \mathcal{P} , tout quadruplet de points (A, B, C, D) , non trois à trois colinéaires, définit une base projective.

5.2.4 Coordonnées projectives

Soit un espace projectif muni d'un repère projectif, il est intéressant de localiser les points de cet espace par des quantités calculées directement dans l'espace projectif lui-même. Le concept de coordonnées projectives permet de localiser tout point d'un espace projectif. Ces coordonnées projectives s'expriment en fonction du birapport.

A- Cas d'une droite

Soient 3 points (A, B, C) définissant un repère projectif d'une droite Δ . A tout point P de cette droite, nous associons $\lambda_P = [A, B, C, P]$ le birapport de ces points. Cette valeur détermine de manière unique la position du point P sur la droite Δ . La valeur λ_P représente la coordonnée projective de P dans la base (A, B, C) .

B- Cas d'un plan

Soient 4 points (A, B, C, D) formant une base projective \mathcal{B} d'un plan Π . Nous reprenons ici la définition de coordonnées projectives telle qu'elle est donnée dans [DH73]. Les coordonnées projectives d'un point P du plan Π dans \mathcal{B} sont définies par le triplet (k_1, k_2, k_3) des trois birapports des faisceaux de droites non indépendants suivants :

$$\begin{cases} k_1 = [AB, AC, AD, AP] \\ k_2 = [BA, BC, BD, BP] \\ k_3 = [CA, CB, CD, CP] \end{cases}$$

La notation XY représente la droite passant par le point X et le point Y . Il est facile de montrer que dans le cas particulier où le point P appartient à une droite passant par deux points de la base, le birapport k_1 prend une des valeurs particulières suivantes :

$$\begin{cases} k_1 = \infty & \text{si } P \text{ appartient à la droite passant par } (A, B) \\ k_1 = 1 & \text{si } P \text{ appartient à la droite passant par } (A, D) \\ k_1 = 0 & \text{si } P \text{ appartient à la droite passant par } (A, C) \end{cases}$$

Un raisonnement analogue peut être fait pour les deux birapports k_2 et k_3 .

Tout comme pour le cas d'un point, la donnée d'une valeur k permet de définir une droite unique l relativement à la base $\mathcal{B} = (A, B, C, D)$, telle que pour tout point $P \in l$, nous ayons $k = [AB, AC, AD, AP]$ (Figure 5.4).

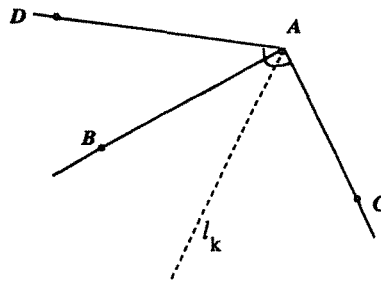


Figure 5.4. Droite l_k correspondant à une valeur de birapport k donné.

Ainsi deux birapports soigneusement³ choisis parmi les coordonnées projectives d'un point P , suffisent à définir ce point de l'espace projectif comme étant l'intersection des deux droites relatives chacune à un des birapports considérés (Figure 5.5). Par exemple, nous pouvons choisir les birapports k_1 et k_2 , si k_1 ou k_2 différents de ∞ .

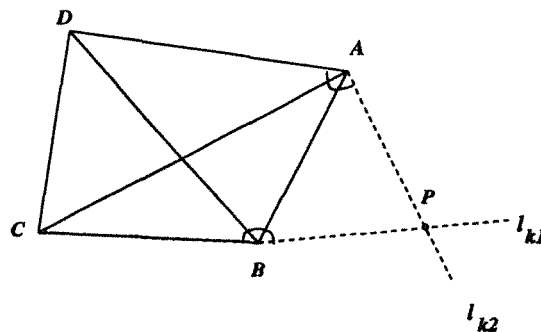


Figure 5.5. Coordonnées projectives dans un plan.

Le cas particulier où $k_1 = k_2 = \infty$, est un cas de dégénérescence. En effet, ces deux birapports définissent la même droite, à savoir la droite AB . Pour résoudre le problème, il suffit de prendre le couple (k_1, k_3) qui est alors un couple de birapports non dégénéré.

3. Eviter les cas limites.

Remarque 3 Pour simplifier l'écriture, nous désignons les points d'un espace projectif par deux de ses coordonnées projectives qui ne forment pas un cas dégénéré et par abus de notation, nous posons $P = (k_1, k_2)_B$.

Les coordonnées projectives sont invariantes par toute transformation perspective. La démonstration faite dans le livre de Duda et Hart [DH73] repose sur la définition et l'invariance du birapport d'un faisceau de plans.

5.3 Système de bases projectives en correspondance

Nous assimilons un plan image à un espace projectif. Si pour la recherche de contraintes de coplanarité des primitives nous ne faisons aucun appel à la connaissance des paramètres de calibrage, nous demandons en revanche à disposer d'images de points de références de configuration bien déterminée. La contrainte de coplanarité d'un ensemble de points 2D appariés, ou encore de droites 2D appariées, est établie via ces points de référence et plus précisément en utilisant la conservation de l'unique invariant de base de la géométrie projective : *le birapport*.

Définition 14 Soient n plans images distincts. Nous désignons par système de bases projectives en correspondance, un n -uplet de bases projectives dont chacune est inscrite dans un plan image, et provenant toutes de la projection d'un même quadruplet \mathcal{B} , de points 3D coplanaires. \mathcal{B} est appelée la base projective initiale du système.

Dans la Figure 5.6, le couple $((a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2))$ représente un système de bases projectives en correspondance issues de la base initiale $(A, B, C, D)^4$.

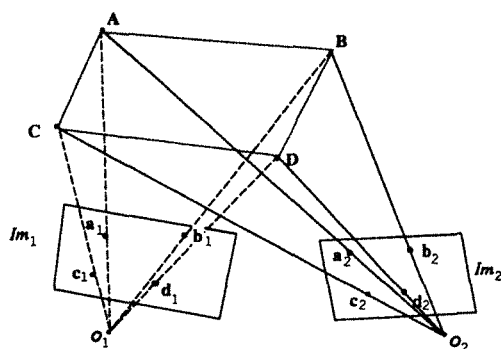


Figure 5.6. Couple de bases projectives en correspondance.

4. Nous désignons par des lettres minuscules les primitives dans les plans images alors que les lettres majuscules désignent des primitives 3D. Les plans de l'espace sont désignés par la lettre Π suivie d'un indice si nécessaire.

5.3.1 Notion de plan de référence

La suite de notre travail repose sur l'utilisation des systèmes de bases projectives en correspondance. Toutefois, dans un esprit de clarté et afin de pouvoir présenter intuitivement les résultats que nous allons énoncer, il est plus facile d'introduire la notion de *plan de référence*, sous-jacente à la notion de système de bases en correspondance. La notion de système de bases projectives en correspondances repose sur l'existence dans l'espace de quatre points coplanaires composant la base projective initiale du système (cf. Définition 14).

Définition 15 *Tout plan Π support d'une base projective initiale est appelé plan de référence.*

Il est évident qu'un plan Π , muni d'une base projective, n'est un plan de référence, que s'il ne contient aucun des centres optiques du banc utilisé.

Pour faciliter la compréhension, nous omettons de parler des systèmes de bases en correspondance et parlons de préférence de plans de références qu'ils induisent. Insistons sur le fait, que l'équation du plan de référence ainsi que les coordonnées 3D des points de la base initiale peuvent être complètement inconnues. La donnée d'un système de bases projectives en correspondance suppose uniquement de disposer d'images appariés de quatre points coplanaires dans l'espace.

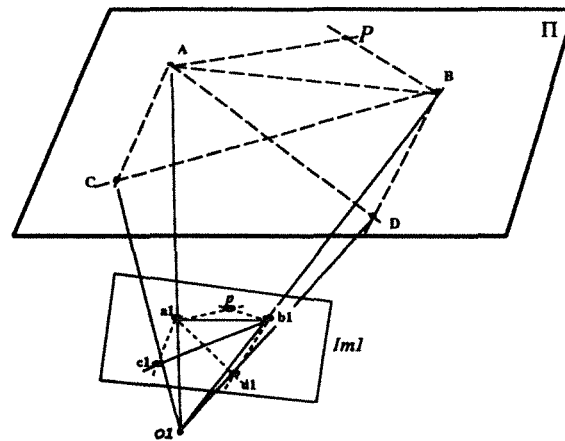
La propriété de base que nous décrivons ci-après est une conséquence directe de la propriété de conservation des coordonnées projectives des points.

Propriété 2 *Désignons par $(\mathcal{B}_1, \dots, \mathcal{B}_n)$ un système de bases projectives en correspondance. Soit Π le plan de référence muni de la base projective initiale \mathcal{B} correspondant à ce système, alors : tout point $P = (k_1, k_2)_{\mathcal{B}}$ de Π sur le plan image Im_i se projette en un point image p_i qui est défini par le même couple de coordonnées projectives dans la base projective \mathcal{B}_i , $p_i = (k_1, k_2)_{\mathcal{B}_i}$.*

La Figure 5.7 illustre le procédé de reconstruction de l'image du point P de Π dans le plan Im_1 en utilisant uniquement les coordonnées projectives de ce dernier. Ce résultat est directement obtenu par l'application de la Propriété 2.

5.3.2 La transformation $\mathcal{T}_{i \rightarrow j}^{\Pi}$

L'utilisation de la Propriété 2 permet pour tout point p_i du plan image Im_i , de retrouver son homologue $p_{i \rightarrow j}$ dans le plan image Im_j tel que le couple de points $(p_i, p_{i \rightarrow j})$ soit l'image d'un unique point $P(\Pi)$ appartenant au plan de référence Π . Le point $p_{i \rightarrow j}$ est défini par les mêmes coordonnées projectives dans \mathcal{B}_j que p_i dans \mathcal{B}_i . La construction de cet homologue se fait donc via le système de base projective en correspondance induit par le plan de référence Π . Ainsi, pour construire le point $p_{i \rightarrow j}$, des calculs exclusivement 2D ont remplacés les calculs 3D qui consistaient à déterminer le point d'intersection de la droite passant par le point principal de la caméra i et le point p_i avec le plan de référence, puis de projeter ce point sur le plan image Im_j .



$P = (k1, k2)_B$ et $p = (k1, k2)_{B1}$ avec $p = \text{Image}(P)$ sur ImI

Figure 5.7. Reconstruction de l'image d'un point P appartenant à un plan de référence.

Nous allons formaliser cette technique de reconstruction d'homologues particuliers, en introduisant l'application $\mathcal{T}_{i \rightarrow j}^\Pi$, définie comme suit :

$$\begin{aligned} \mathcal{T}_{i \rightarrow j}^\Pi : (Im_i, \mathcal{B}_i) &\longrightarrow (Im_j, \mathcal{B}_j) \\ p_i = (k_1, k_2, k_3)_{\mathcal{B}_i} &\longrightarrow p_{i \rightarrow j} = (k_1, k_2, k_3)_{\mathcal{B}_j} \end{aligned}$$

Nous notons également par $\mathcal{T}_{i \rightarrow j}^\Pi$, l'application duale, qui à une droite d_i du plan image Im_i fait correspondre la droite $d_{i \rightarrow j}$ dans le plan image Im_j .

Remarque 4 Il est facile de vérifier que l'application $\mathcal{T}_{i \rightarrow j}^\Pi$ a la propriété suivante :

$$\mathcal{T}_{i \rightarrow j}^\Pi \circ \mathcal{T}_{j \rightarrow k}^\Pi = \mathcal{T}_{i \rightarrow k}^\Pi$$

Dans la suite, et quand aucune confusion n'est possible nous omettons de rappeler le plan de référence par rapport auquel l'application est définie et nous la notons plus simplement $\mathcal{T}_{i \rightarrow j}$.

Afin d'éviter toute confusion, il importe de faire apparaître qu'en réalité nous associons à toute primitive $prim_i$, de type point ou droite de Im_i , deux primitives "homologues" de Im_j . En effet, étant donné un plan de référence Π , nous associons à $prim_i$ le couple de primitives de même type: $(prim_j, prim_{i \rightarrow j})$ où :

1. $prim_j$ correspond au "vrai" homologue de p_i : établi par l'étape d'appariement,
2. $prim_{i \rightarrow j}$ l'homologue de p_i construit par application $\mathcal{T}_{i \rightarrow j}^\Pi$ à p_i .

Nous sommes alors en droit de se demander d'une part quelle type de relation peut exister entre ces deux homologues, et d'autre part si la coïncidence de ces deux primitives en une seule a une signification particulière ou non.

Nous accordons à ces deux questions une place particulière. En effet, la réponse qui leur est faite, nous permet par la suite de justifier la nécessité de l'utilisation d'un banc stéréoscopique trinoculaire (Section 5.4.2).

A- Cas du point

Il est facile de voir que dans le cas d'un point p_i (différent de l'épipoles), les deux points p_j et $p_{i \rightarrow j}$ appartiennent à la même droite épipolaire dans Im_j associée à p_i . Ceci est visible dans la Figure 5.8.

Il devient alors évident que quand nous avons une coïncidence entre les deux points p_j et $p_{i \rightarrow j}$, ceci revient à dire que le point P et $P(\Pi)$ sont confondus. Et nous énonçons la propriété suivante :

Propriété 3

$(p_j = T_{i \rightarrow j}(p_i)) \Leftrightarrow$ le couple de vrai homologues (p_i, p_j) est issu de la projection d'un point 3D appartenant au plan Π

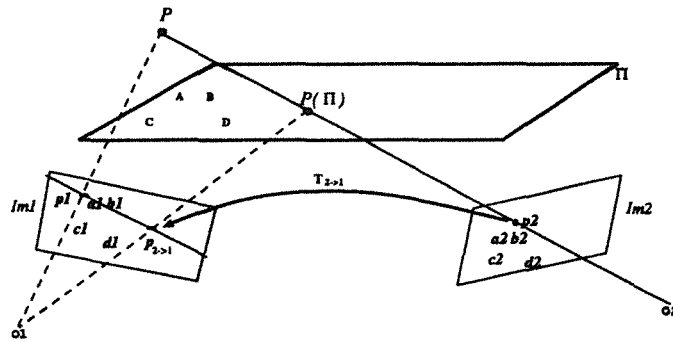


Figure 5.8. $T_{2 \rightarrow 1}^\Pi$ appliquée à un point.

B- Cas d'une droite

Dans le cas d'une droite d_i , la droite $d_{i \rightarrow j}$ correspond en fait à la projection de la droite $D(\Pi)$: la droite d'intersection du plan Π et du plan d'interprétation associé à d_i (Figure 5.9).

Déterminons à présent la signification de la coïncidence entre d_j et $d_{i \rightarrow j}$. A l'encontre du cas du point, l'égalité entre d_j et $d_{i \rightarrow j}$ ne permet pas de caractériser les droites 2D provenant de la projection d'une droite 3D appartenant au plan Π . En effet, dans le cas d'une droite nous énonçons la propriété suivante :

Propriété 4

$(d_j = T_{i \rightarrow j}^\Pi(d_i)) \Leftrightarrow \begin{cases} d_j \text{ et } d_i \text{ forment un couple de droites épipolaires homologues} \\ \text{ou bien} \\ D = D(\Pi) \end{cases}$

Notons finalement que dans le cas où d_j et d_i forment un couple de droites épipolaires homologues⁵, alors l'égalité $(d_j = T_{i \rightarrow j}^\Pi(d_i))$ est indépendante du choix du plan de référence Π .

5. Autrement dit la droite D dont ils sont les images n'appartient pas à un plan épipolaire du banc binoculaire Im_i et Im_j utilisé.

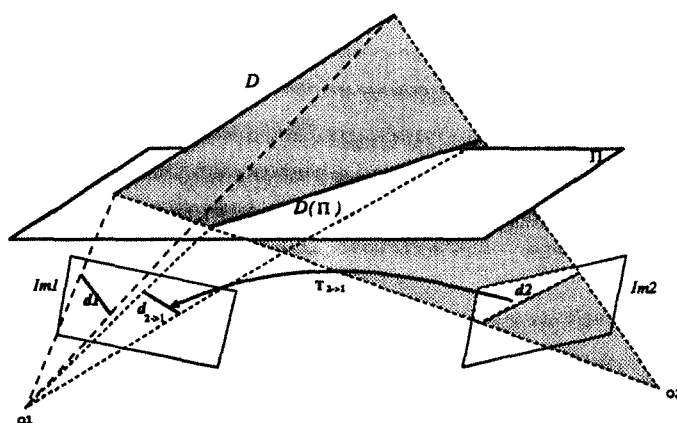


Figure 5.9. Représentation graphique de $T_{2 \rightarrow 1}^{\Pi}$ appliquée à une droite.

5.4 Trace d'une droite 3D sur un plan de référence

Dans cette section, nous reprenons un résultat établi dans l'article de Quan [QM91] sous le nom de *piercing point*. Nous reformulons ce résultat en parlant de trace d'une droite sur un plan Π . Cette notion de trace de droite nous servira ultérieurement à définir la contrainte de coplanarité d'un ensemble de points ou d'un ensemble de droites.

5.4.1 Définition de la trace d'une droite

Définition 16 La trace d'une droite 3D $L, T_{(D,i)}^{\Pi}$, dans le plan image Im_i par rapport au plan de référence Π est l'image de l'intersection de la droite avec le plan Π dans le plan Im_i .

Nous étendons cette notion de trace d'une droite à celle d'un segment 3D en associant cette dernière à la trace de la droite support du segment sur le plan de référence considéré.

Quand le plan image est parfaitement connu et qu'aucune confusion n'est possible, nous écrivons au lieu de $T_{(D,i)}^{\Pi}$ plus simplement T_L^{Π} omettant de rappeler le numéro du plan image d'où provient la trace.

5.4.2 Construction de la trace d'une droite

Dans cette section, nous présentons le moyen de déterminer la trace d'une droite sur un plan de référence Π dans le plan image Im_i , en ne connaissant de cette droite que ses images appariées et en n'effectuant aucun calcul dans l'espace 3D. En fait, tout revient à construire directement cette trace dans le plan image Im_i . Cette construction utilise le principe énoncé par Quan [QM91].

A- Cas d'une droite n'appartenant pas à un plan épipolaire

Soit L une droite n'appartenant pas à un plan épipolaire du banc stéréoscopique binoculaire utilisé. La trace T_L^Π de la droite L sur un plan Π dans le plan image Im_1 s'écrit alors comme suit :

$$T_L^\Pi = l_1 \cap T_{2 \rightarrow 1}(l_2) \quad (5.1)$$

Démonstration

Le problème est de retrouver l'image de l'ensemble des points d'intersection de la droite L avec un plan de référence Π donné relativement au plan image Im_i . Nous supposons que nous disposons des projections (l_1, l_2) de L sur les deux plans images ainsi que d'un système de bases en correspondance issues de Π noté (B_1, B_2) . Nous désignons par B la base initiale de Π .

Déterminer l'image de l'intersection de la droite L et du plan de référence Π revient à déterminer l'image de l'intersection des trois plans distincts⁶ suivants : Π , le plan d'interprétation de l_1 et le plan d'interprétation de l_2 . Ainsi, les points d'intersection recherchés sont les points d'intersection des deux droites de Π suivantes :

$$(L_1 = \Pi \cap Plan(O_1, l_1)) \text{ et } (L_2 = \Pi \cap Plan(O_2, l_2)) \quad (5.2)$$

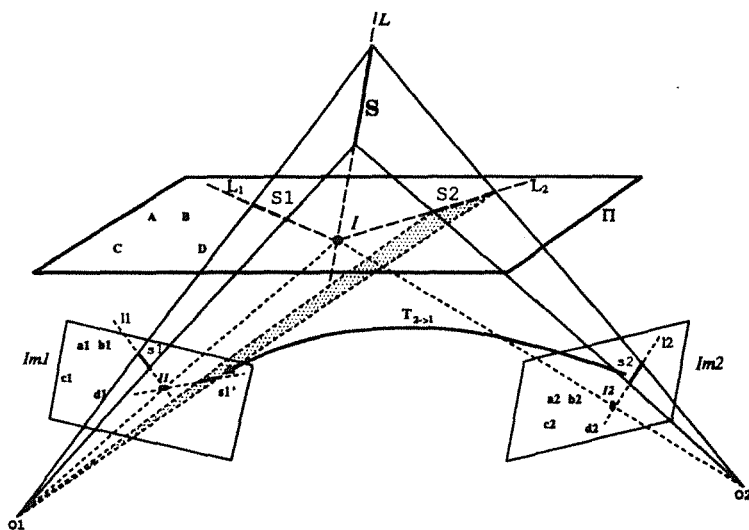


Figure 5.10. La détermination de la trace d'un segment S et de la droite L .

Le problème initial de la recherche de l'intersection d'une droite et d'un plan se ramène à la recherche de l'intersection de deux droites définies dans le plan de référence Π .

La trace de L sur Π relativement à Im_i est alors défini par l'intersection de l'image respective de L_1 et de L_2 dans le plan image Im_i . Il suffit donc de construire les images respectives des droites L_1 et L_2 dans Im_i . Prenons le cas de la trace relative au plan image Im_1 : l'image de la droite L_1 sur Im_1 est la droite l_1 . La droite L_2

6. Ces plans sont distincts car L n'est pas dans un plan épipolaire.

a les mêmes coordonnées projectives dans \mathcal{B} que l_2 dans \mathcal{B}_2 , cette caractérisation se déduit directement de la Propriété 2. Ainsi, comme $L_2 \in \Pi$, l'image de cette dernière dans Im_1 a également les mêmes coordonnées dans \mathcal{B}_1 ce qui résout le problème de la reconstruction de l'image de la droite L_2 dans Im_1 .

L'application $\mathcal{T}_{2 \rightarrow 1}$ appliquée à la droite l_2 permet donc de retrouver l'image de la droite dans le plan image Im_1 (Figure 5.10).

La résolution du problème initial de la reconstruction de la trace de L associé au couple (l_1, l_2) sur Π relativement au plan image Im_1 , revient donc à la recherche de l'intersection de droites particulières parfaitement définies dans le plan image Im_1 et qui sont les droites l_1 et $\mathcal{T}_{2 \rightarrow 1}(l_2)$. \diamond

B- Cas d'une droite appartenant à un plan épipolaire

Nous savons grâce à la Propriété 4, que dans le cas où la droite L appartient à un plan épipolaire du banc stéréoscopique Im_1 et Im_2 , ses images (l_1, l_2) sont tel que $\mathcal{T}_{2 \rightarrow 1}^\Pi(l_2) = l_1$. Ainsi, à partir de deux plans images aucune information relative à la trace ne peut être déduite de la manière décrite précédemment.

Cette dernière remarque met en évidence les limites d'un banc stéréoscopique binoculaire pour le calcul 2D de la trace. L'utilité d'un banc stéréoscopique trinoculaire, dont les centres optiques ne sont pas alignés devient donc indiscutable. Utilisant un système trinoculaire, nous savons alors que pour toute droite L , il existe au moins un couple de caméras par rapport aux quels L n'est pas située dans un plan épipolaire. Remarquons par ailleurs, que dans le cas où la droite est quelconque (droite contenu dans aucun plans épipolaires) l'information délivrée par la construction de la trace relativement à chacun des couples de caméras est alors une information redondante.

C- Résultat sur la construction de la trace d'une droite

Propriété 5 Soit L une droite dont on connaît le triplet d'images (l_1, l_2, l_3) provenant d'un banc stéréoscopique trinoculaire dont les caméras sont non alignées. La trace de L sur un plan Π dans le plan image Im_1 est alors définie par :

$$T_L^\Pi = l_1 \cap \mathcal{T}_{2 \rightarrow 1}(l_2) \cap \mathcal{T}_{3 \rightarrow 1}(l_3) \quad (5.3)$$

Remarque 5 En utilisant les propriétés relatives à l'application $\mathcal{T}_{i \rightarrow j}$ énoncées dans la Remarque 4, nous pouvons, à partir de la trace d'une droite L sur Π relativement au plan image Im_i , construire directement la trace de L sur Π relativement au plan Im_j en appliquant la relation suivante :

$$T_{(L,j)}^\Pi = \mathcal{T}_{i \rightarrow j}^\Pi(T_{(L,i)}^\Pi) \quad (5.4)$$

Ainsi, nous choisissons de caractériser la trace par ses coordonnées dans la base projective correspondante, laquelle provient de la base initiale rattachée à Π . Le plan image par rapport auquel nous effectuons la construction de la trace est choisi arbitraire et n'influe nullement sur le résultat. C'est pour cette raison que nous proposons d'effectuer la recherche de la trace exclusivement dans un même plan image : le plan

Im_1 .

Nous étendons les résultats de la trace d'une droite à celle d'un segment, et nous énonçons la définition suivante où S est un segment ayant pour triplet d'images (s_1, s_2, s_3) ses images respectives.

Définition 17 Nous appelons trace T_S^Π du segment S sur un plan Π dans le plan image Im_1 , la trace de la droite support de S sur Π . Ainsi, T_S^Π est composée par l'ensemble des points d'intersection de $l_1, T_{2 \rightarrow 1}(l_2)$ et $T_{3 \rightarrow 1}(l_3)$ où l_1, l_2 et l_3 sont respectivement les droites support des segments s_1, s_2 et s_3 .

5.5 Contrainte de coplanarité d'une droite et d'un plan de référence

La trace d'une droite sur un plan de référence donné permet de classer cette droite par rapport à ce plan. En effet, il est possible au vu de la trace de savoir si la droite est coplanaire ou non à ce plan. Énonçons ci-après la contrainte caractérisant la coplanarité d'une droite et d'un plan de référence; cette contrainte se déduit de la Propriété 4.

Contrainte 1 La coplanarité de la droite L à un plan de référence Π donné est donnée par la relation suivante :

$$L \in \Pi \iff (l_1 = T_{2 \rightarrow 1}(l_2) = T_{3 \rightarrow 1}(l_3))$$

Relevons le fait que dans le cas d'un segment S non coplanaire au plan Π ($S \notin \Pi$) il est possible de déterminer l'extrémité du segment la plus proche du plan de référence en utilisant la trace de ce segment sur un plan de référence Π . Ceci permet de déterminer en 2D, l'extrémité du segment par laquelle celui-ci ferait sa jonction avec le plan.

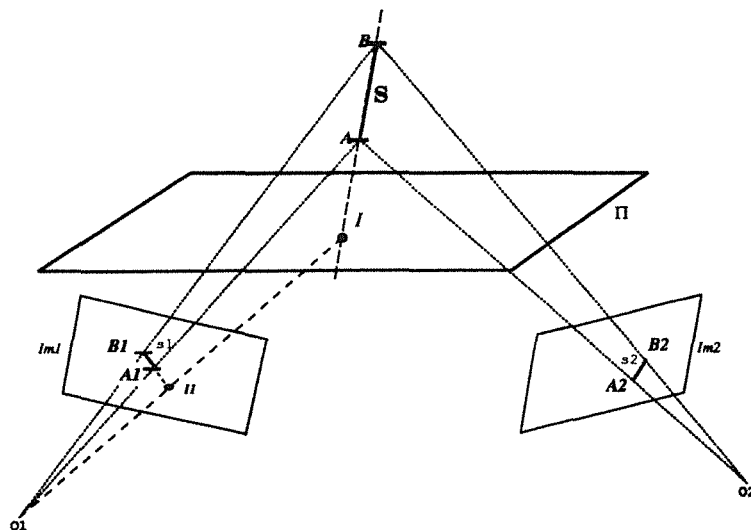


Figure 5.11. Détection de l'extrémité la plus proche de S à Π en utilisant la trace.

Propriété 6 Soit $(s_1 = [A_1, B_1], s_2 = [A_2, B_2], s_3 = [A_3, B_3])$ un triplet de segments homologues provenant d'un segment 3D S , intersectant le plan Π . Nous désignons le point de trace de ce segment sur Π par $i_1 = s_1 \cap \mathcal{T}_{2 \rightarrow 1}(s_2) \cap \mathcal{T}_{3 \rightarrow 1}(s_3)$. Si $i_1 \notin s_1$ alors l'extrémité la plus proche du plan Π est celle dont la distance d au point de trace est la plus proche : $d = \text{Min}(\text{dist}(A_i, I_i), \text{dist}(B_i, I_i))$.

Notons que si $i_1 \in s_1$ alors le segment s_1 traverse le plan de référence. Cette remarque est reprise ultérieurement dans la Propriété 11.

5.6 Caractérisation de la coplanarité de points 3D

Soit Pt_{3d} un ensemble de points 3D. Supposons que nous ne connaissons que les images de ces points sur les plans images ainsi que leur appariement. Désignons par Pt_{2d} l'ensemble des triplets de points appariés. Notre but est de caractériser la coplanarité de Pt_{3d} en travaillant sur Pt_{2d} [Cha92]. A cet effet, nous introduisons la notion de trace d'un ensemble de points qui nous permet de caractériser la coplanarité de cet ensemble. Tout comme pour la trace d'un segment, nous allons montrer qu'il est possible de déterminer la trace d'un ensemble de points 3D à partir de l'ensemble des images appariées de ces points.

5.6.1 Trace d'un ensemble de points 3D

Définition 18 La trace d'un ensemble Pt_{3d} de n points 3D sur un plan Π est l'ensemble des traces de toutes les droites passant par deux points de l'ensemble Pt_{3d} sur Π . Cette trace est notée Pt_{3d} .

La construction de la trace d'un ensemble de points découle de celle de la construction de droites présentée dans la Section 5.4. En effet, soient :

- (1) $Pt_{3d} = \{P_1, \dots, P_n\}$ et $Pt_{2d} = \{(p_1^1, p_1^2, p_1^3), \dots, (p_n^1, p_n^2, p_n^3)\}$ où (p_j^1, p_j^2, p_j^3) désigne le triplet de points 2D homologues correspondant aux images du point 3D P_j sur les plans images Im_1, Im_2 et Im_3 .
- (2) $L_{(u,v)}$ désigne la droite qui passe par les points P_u et P_v de Pt_{3d}
- (3) $l_{(u,v)}^i$ désigne la droite passant par les deux points (p_u^i, p_v^i) . Remarquons que $l_{(u,v)}^i$ est l'image sur Im_i de la droite $L_{(u,v)}$.

Nous avons alors :

$$T_{Pt_{3d}}^\Pi = \bigcup_{1,1}^{n,n} T_{L_{(u,v)}}^\Pi = \bigcup_{u,v=1}^{u,v=n} \{ \mathcal{T}_{2 \rightarrow 1}^\Pi(l_{(u,v)}^2) \cap \mathcal{T}_{3 \rightarrow 1}^\Pi(l_{(u,v)}^3) \cap l_{(u,v)}^1 \} \quad (5.5)$$

5.6.2 Contrainte de coplanarité de points 3D

Contrainte 2 Les notations utilisées sont les mêmes que celles introduites ci-avant. L'ensemble Pt_{3d} est composé de points coplanaires si et seulement si nous sommes dans l'un des cas suivants :

- Pour tout couple (u, v) $T_{L(u,v)}^\Pi = l_{(u,v)} \iff Pt_{3d}$ est composé de points coplanaires qui sont contenus dans le plan Π .
- Tous les points composant $T_{Pt_{3d}}^\Pi$ sont alignés sur une droite δ qui est l'image de la droite constituant l'intersection du plan support de Pt_{3d} et du plan Π (Figure 5.12).

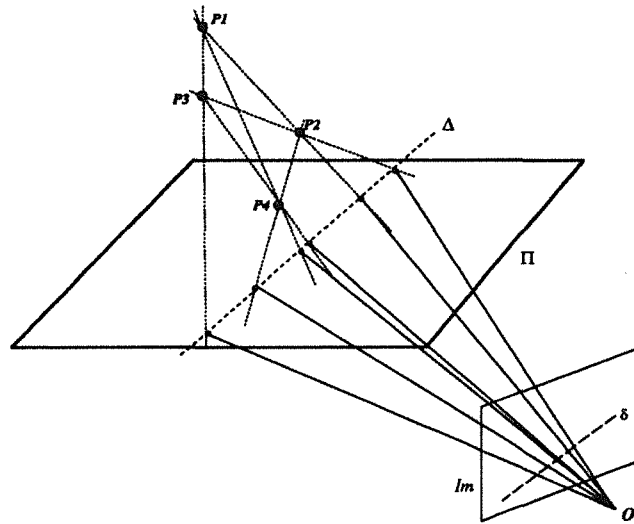


Figure 5.12. Trace d'un ensemble de 4 points coplanaires.

Démonstration

(\implies) Il est évident que si tous les points de Pt_{3d} sont coplanaires et s'ils appartiennent au plan Π alors la trace de chaque droite passant par un couple de points de Pt_{3d} est égale à son image. Si les points sont coplanaires sans être dans le plan Π alors l'intersection du plan support de ces points et du plan Π est une droite Δ dont l'image δ regroupe tous les points de la trace.

(\impliedby) Supposons que les P_i soient non coplanaires et démontrons que les points de la trace ne peuvent pas être alignés. Soient P_1, P_2 et P_3 trois points de l'ensemble Pt_{3d} non alignés; ils définissent un plan noté $\Pi_{Pt_{3d}}$. Désignons par Δ la droite d'intersection entre Π et $\Pi_{Pt_{3d}}$ et δ son image. Il est évident que nous avons $T_{(P_1, P_2, P_3)}^\Pi \subset \delta$. Supposons, sans perte de généralité que $P_1 \notin \Delta$. La question à résoudre est la suivante :

Pouvons-nous avoir un point $P_i \in Pt_{3d}$ et $P_i \notin \Pi_{Pt_{3d}}$ tel que les points de la trace définie par les quatre points (P_1, P_2, P_3, P_i) soient alignée ?

Pour répondre à cette question nous devons résoudre le système d'équations suivant :

$$\begin{cases} T_{(P_1, P_2, P_3, P_i)}^\Pi \subset \delta \implies L_{(1,i)} \cap \Pi \in \Delta \\ P_i \notin \Pi_{Pt_{3d}} \implies L_{(1,i)} \notin \Pi_{Pt_{3d}} \end{cases}$$

Ce système n'admet de solution que si $P_1 \in \Delta$, ce qui est contradictoire avec notre hypothèse de départ. \diamond

Tout ensemble de points ayant une trace ne correspondant pas à l'un des deux cas énoncés par la Contrainte 2, est donc composé de points non coplanaires. Insi connaissant la trace d'un ensemble de points 3D nous pouvons déterminer si ces derniers sont coplanaires ou non. De plus, une fois la coplanarité d'un ensemble Pt_{3d} de points est vérifiée, la race de ce dernier permet de distinguer le cas où Pt_{3d} est coplanaire au plan Π de celui où le plan support de Pt_{3d} est sécant avec Π en donnant l'image de cette intersection.

Remarque 6 Dans le cas où l'on connaît les paramètres 3D des plans de référence utilisés, la donnée des deux traces d'un ensemble de points coplanaires sur deux plans de références Π et Π' distincts permet de définir les paramètres 3D du plan support du polygone. C'est cette approche qui sera utilisée pour la reconstruction des facettes 3D (Chapitre 6).

5.6.3 Un algorithme de recherche de coplanarité un ensemble de points

Nous nous intéressons ici de plus près à la contrainte de coplanarité et cherchons à savoir si pour caractériser un ensemble de points 3D, nous avons réellement besoin de la trace de toutes les droites passant par au moins deux points de cet ensemble. La réponse est négative comme nous le montrons ci-après. En suivant la démonstration de la contrainte de planarité, nous constatons qu'il suffit de sélectionner un point P_i qui n'appartienne pas au plan de référence, et d'étudier la trace du faisceau de droites issues de ce point et passant par un autre point de l'ensemble considéré. Nous appelons ce point P_i point pivot. Un point pivot P_i d'images (p_i^1, p_i^2, p_i^3) vérifie la condition suivante :

$$P_i \text{ est un point pivot} \iff T_{2 \rightarrow 1}^\Pi(p_i^2) \neq p_i^1 \quad (5.6)$$

De cette remarque découle un algorithme simple de recherche de coplanarité d'un ensemble de points 2D à partir de leur trace.

La fonction que nous décrivons ci-après prend en entrée un ensemble de triplet de points homologues Pt_{2d} et rend un booléen suivant la validité ou pas de l'hypothèse relative à la coplanarité des points 3D correspondant au triplets de Pt_{2d} . Nous choisissons de construire la trace dans le plan image Im_1 . Rappelons que $l_{(i,j)}^l$ $l = 1 \dots 3$ désigne la droite du plan image Im_l passant par le couple de points (p_i^l, p_j^l) . Le triplet (p_i^1, p_i^2, p_i^3) désigne un triplet de points homologues.

Fonction *Rech_coplanaire_ens_point*($Pt_{2d} = \{(p_1^1, p_1^2, p_1^3), \dots, (p_n^1, p_n^2, p_n^3)\}$)
 /** Initialisation : Recherche d'un point pivot **/
 $Coplan_{\Pi} = \text{Vrai}; i = 1;$
Tant que ($Coplan_{\Pi}$ et ($i \leq n$))Faire
 Si $T_{2 \rightarrow 1}^{\Pi}(p_i^2) \neq p_i^1$ Alors /* On a un point pivot */
 $Coplan_{\Pi} = \text{Faux}; (pt_{pivot}^1, pt_{pivot}^2, pt_{pivot}^3) = (p_i^1, p_i^2, p_i^3)$
 Sinon /* On passe au suivant */
 $i = i + 1;$
Fin tant que

 /** Recherche de la coplanarité **/
Si $Coplan_{\Pi}$ Alors *La Coplanarité Validée*; /*1^{ère} partie de la Contrainte 2 */
Sinon
 $\delta = \emptyset;$
 Pour tout $(p_i^1, p_i^2) \neq (pt_{pivot}^1, pt_{pivot}^2)$ de P Faire
 Calculer $I_{(pivot,i)} = T_{2 \rightarrow 1}^{\Pi}(l_{(pivot,i)}^2) \cap T_{3 \rightarrow 1}^{\Pi}(l_{(pivot,i)}^3) \cap l_{(pivot,i)}^1$
 Si *Non* ($\text{Aligne}(I_{(pivot,i)} \cup \delta)$) Alors /** échec **/
 Coplanarité mise en défaut; Arrêt;
 Fin Pour
 Rendre coplanarité validée /* 2^{ème} partie de la Contrainte 2 */
Fin Fonction

Remarque 7 *Dans le cas où la coplanarité d'un ensemble de points est établie et si ces points n'appartiennent pas tous au plan de référence considéré, nous confondons, par abus de langage, la trace avec la droite qui la contient.*

5.6.4 Comment tester la coplanarité de deux ensembles de points ?

Le problème traité dans cette partie concerne la recherche des opérations à effectuer pour caractériser la coplanarité de deux ensembles de points Pt_{3d1}, Pt_{3d2} , à partir de la trace de chacun d'eux sur un même plan Π qui leur est non coplanaire. Remarquons en premier lieu, qu'il n'est pas possible à partir de l'égalité de leur trace $T_{Pt_{3d1}}^{\Pi}$ et $T_{Pt_{3d2}}^{\Pi}$ de déduire la coplanarité de ces ensembles entre eux. L'égalité $T_{Pt_{3d1}}^{\Pi} = T_{Pt_{3d2}}^{\Pi}$ est insuffisante pour prouver la coplanarité de deux ensembles de points, en effet si Pt_{3d1} et Pt_{3d2} ont des plans supports faisant partie d'un faisceau de plans passant par une droite Δ du plan Π , alors la trace de chacun de ces ensembles de points est l'image de Δ sur le plan image considéré pour calculer la trace (cf. Figure 5.13). Ainsi, pour déterminer la coplanarité de deux ensembles de points dont la trace relative à chacun d'eux par rapport au même plan de référence est donnée, nous procédons comme suit :

- si les traces indiquent que Pt_{3d1} et Π sont coplanaires d'une part et que Pt_{3d2} et Π sont coplanaires d'autre part alors Pt_{3d1} et Pt_{3d2} sont évidemment coplanaires.
- si $T_{Pt_{3d1}}^{\Pi} \neq T_{Pt_{3d2}}^{\Pi}$, alors Pt_{3d1} et Pt_{3d2} sont non coplanaires.
- si $T_{Pt_{3d1}}^{\Pi} = T_{Pt_{3d2}}^{\Pi} = \delta$, alors les ensembles Pt_{3d1} et Pt_{3d2} sont coplanaires si et seulement si la trace associée à une droite passant par un couple de points pivots provenant respectivement de Pt_{3d1} et Pt_{3d2} appartient à δ .

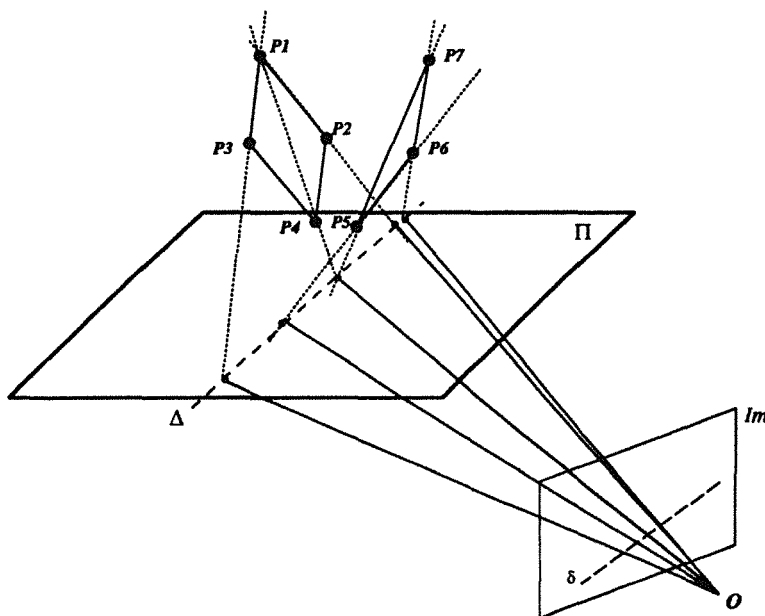


Figure 5.13. Exemple de non coplanarité de deux ensembles de points définissant une même droite de trace.

5.7 A propos d'une contrainte de coplanarité d'un ensemble de droites

L'étude de la contrainte de coplanarité d'un ensemble de points nous a conduit naturellement à l'étendre à un ensemble de droites. Soit $D_{3d} = \{D_1, \dots, D_n\}$ un ensemble de droites 3D et soit $D_{2d} = \{(d_1^1, d_1^2, d_1^3), \dots, (d_n^1, d_n^2, d_n^3)\}$ l'ensemble des triplets de droites homologues correspondants à D_{3d} .

Définition 19 La trace d'un ensemble de droites 3D D_{3d} est l'union des traces des droites composant D_{3d} .

Au contraire du cas des points, la contrainte de coplanarité de droites ne peut être confirmée par l'alignement des points de trace par rapport à un plan de référence. En effet, prenons le cas où D_{3d} est composé de droites appartenant chacune à un plan du

faisceau de plans passant par une droite Δ du plan Π (Figure 5.14), alors la trace $T_{D_{3d}}^{\Pi}$ est contenu dans l'image de Δ dans le plan image où nous effectuons la recherche de la trace. Les points composant cette trace sont alignés bien que par construction D_{3d} soit composé de droites non coplanaires. Remarquons que cette configuration est la seule qui permette d'infirmer l'implication suivante :

Si les points de $T_{D_{3d}}^{\Pi}$ sont alignés alors les droites de D_{3d} sont coplanaires.

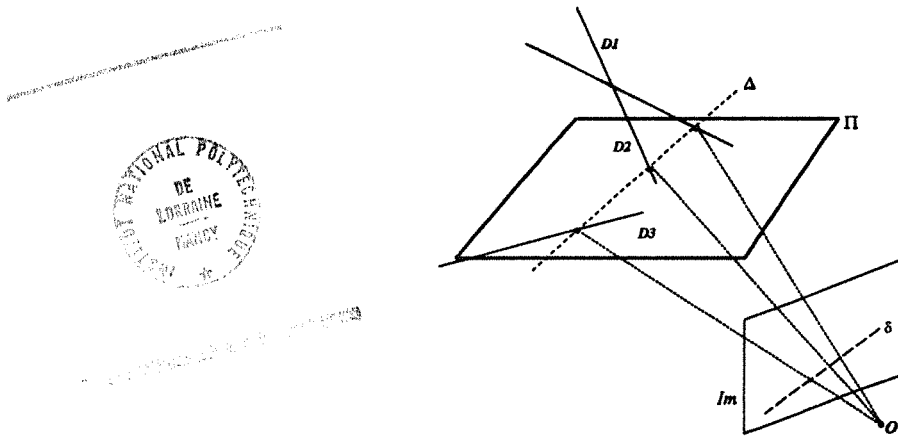


Figure 5.14. Exemple de non coplanarité d'un ensemble de droites ayant des points de trace alignés.

Remarque 8 De l'étude de la trace $T_{D_{3d}}^{\Pi}$ d'un ensemble de droite D_{3d} , nous pouvons seulement en déduire que :

- l'ensemble D_{3d} est composé de droites toutes coplanaires au plan Π si et seulement si la trace de chacune de ces droites est égale à l'image d'elle-même,
- si les points composant $T_{D_{3d}}^{\Pi}$ ne sont pas alignés alors les droites ne sont pas coplanaires.

La Remarque 8 est très importante. En effet, si nous disposons d'un ensemble de segments homologues et voulons éviter de travailler sur les extrémités de ces segments à cause de l'erreur de localisation dont ils sont entachés, nous sommes tenté de vérifier la coplanarité de ces segments en utilisant la trace relative à leur droites supports. Cependant, puisqu'il n'y a pas de contrainte aussi forte que celle sur les points, il faut, pour l'instant du moins, dans le cas où la trace attachée aux droites supports est composée de points alignés, revenir à des points homologues appartenant à chacun des segments et appliquer la contrainte de coplanarité de points.

5.7.1 Et si on prenait plus d'un plan de référence?

Nous venons de montrer que la non coplanarité des droites de D_{3d} , possédant une trace alignée, provient du seul cas où ces dernières font partie d'un faisceau de plans

contenant le plan de référence Π considéré. Une idée pour lever cette ambiguïté est de prendre un nouveau plan de référence Π' , ne faisant pas partie de ce faisceau de plans, et sur lequel nous recherchons également la trace de ces droites [Cha93]. Afin de garantir cette condition pour tous les ensembles de droites traités il suffit en fait de disposer de trois plans de références, Π , Π' et Π'' qui ne définissent pas un faisceau de plans. Nous énonçons alors la contrainte de coplanarité de droites comme suit :

Contrainte 3 Soient Π , Π' et Π'' trois plans de référence ne décrivant pas un faisceau de plans. D_{3d} est dit composé de droites coplanaires si et seulement si nous sommes dans l'un de ces deux cas :

1. $T_{D_{3d}}^{\Pi_i} = \cup_{i=1}^n d_i^1$ où Π_i est l'un des trois plans de référence donnés Π , Π' ou Π'' .
2. Il existe trois droites δ , δ' et δ'' telles que $T_{D_{3d}}^{\Pi} \subset \delta$, $T_{D_{3d}}^{\Pi'} \subset \delta'$ et $T_{D_{3d}}^{\Pi''} \subset \delta''$.

Démonstration

La première partie de la contrainte étant évidente, nous démontrons seulement la seconde partie. Il est aisé d'admettre que si D_{3d} est composé de droites coplanaires alors la trace de D_{3d} sur chacun des plans génère une droite. Prouvons donc l'implication suivante :

$$\exists(\delta, \delta', \delta'') / \left\{ \begin{array}{l} T_{D_{3d}}^{\Pi} \subset \delta \\ T_{D_{3d}}^{\Pi'} \subset \delta' \\ T_{D_{3d}}^{\Pi''} \subset \delta'' \end{array} \right\} \implies D_{3d} \text{ est composée de droites coplanaires}$$

Remarquons que, comme Π , Π' et Π'' ne forment pas un faisceau de plans, alors au moins l'une des deux assertions suivantes est vraie : $\delta \neq \delta'$ ou $\delta \neq \delta''$. En effet, désignons par $\delta_{\Pi}^{\Pi'}$ (resp. $\delta_{\Pi}^{\Pi''}$) l'image de la droite d'intersection des plans de références Π et Π' (resp. Π et Π''). Si $\delta = \delta'$ alors forcément $\delta = \delta_{\Pi}^{\Pi'}$ et comme $\delta_{\Pi}^{\Pi'} \neq \delta_{\Pi}^{\Pi''}$ (car Π , Π' et Π'' ne forment pas un faisceau de plans) il est alors impossible d'avoir $\delta = \delta''$.

• Premier cas ($\delta \neq \delta'$)

$T_{D_{3d}}^{\Pi} \subset \delta$ signifie que les droites de D_{3d} appartiennent à des plans faisant partie d'un faisceau de plans passant par une droite Δ de Π ayant pour image δ . D'autre part $T_{D_{3d}}^{\Pi'} \subset \delta'$ signifie que D_{3d} fait partie d'un faisceau de plans passant par une droite Δ' de Π' ayant pour image δ' . Ainsi Δ et Δ' sont deux droites distinctes de l'espace 3D par lesquelles passent un unique plan 3D. Ce plan contient toutes les droites de D_{3d} .

• Deuxième cas ($\delta = \delta'$)

Le fait que $\delta = \delta'$ implique nécessairement $\delta \neq \delta''$. Ainsi, avec un raisonnement analogue au précédent, et en remplaçant Π' par Π'' , nous démontrons l'implication. \diamond

Remarque 9 La démonstration précédente permet de constater, que pour tout ensemble de triplets de droites homologues, seules deux traces sur deux plans de références

suffisent à démontrer sa planarité. Il suffit de vérifier que ces deux traces ne sont pas toutes deux confondues avec la droite d'intersection des deux plans de référence considérés. Nous associons donc pour chaque triplet les deux traces suivantes (δ_1, δ_2) où :

$$\begin{cases} \text{Si } \delta \neq \delta' \text{ Alors } (\delta_1, \delta_2) = (\delta, \delta') \\ \text{Si } \delta = \delta' \text{ Alors } (\delta_1, \delta_2) = (\delta, \delta'') \end{cases}$$

Reprenons le cas particulier illustré précédemment par la Figure 5.14. L'utilisation d'un second plan de référence Π' nous permet de conclure que les trois droites ne sont pas coplanaires car les points composant leur traces sur le nouveau plan de référence n'engendrent pas une droite. Cette remarque est illustrée dans la Figure 5.15.

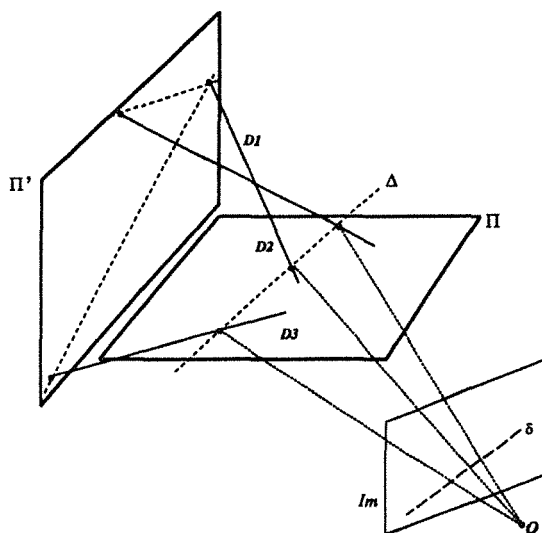


Figure 5.15. La non coplanarité d'un ensemble de droites est détectée en utilisant deux plans de références Π et Π' .

5.7.2 Un algorithme de recherche de coplanarité d'un ensemble de droites

Pour étudier la coplanarité de droites, nous venons de montrer que nous avons implicitement besoin de trois plans de référence que nous notons Π , Π' et Π'' . Nous choisissons de construire la trace d'un ensemble de droite dans le plan image Im_1 . Les différentes propriétés montrées ci-avant permettent de construire un algorithme de recherche de coplanarité d'un ensemble de droites.

Nous décrivons ci-après la fonction de recherche de coplanarité. Cette fonction prend en entrée un ensemble de triplets de droites homologues D_{2d} et rend un booléen⁷ suivant la validité ou pas de l'hypothèse de coplanarité emise sur les droites 3D correspondantes aux triplets de D_{2d} . La notation (d_i^1, d_i^2, d_i^3) désigne un triplet de droites homologues.

7. Remarquons qu'une légère modification de cette fonction permet en cas de coplanarité de rendre l'équation de la droite contenant les points de traces

Fonction *Rech_coplanaire_ens_dte*($D_{2d} = \{(d_1^1, d_1^2, d_1^3), \dots, (d_n^1, d_n^2, d_n^3)\}$)
 /** Initialisation **/
 $Cop_{\Pi} = \text{Vrai}; Cop_{\Pi'} = \text{Vrai}; Cop_{\Pi''} = \text{Vrai};$
 $i = 1;$
 $P_{\Pi} = \emptyset; P_{\Pi'} = \emptyset; P_{\Pi''} = \emptyset;$ /** Ensembles des points de trace **/
 Tant que (Cop_{Π} ou $Cop_{\Pi'}$ or $Cop_{\Pi''}$ et ($i \leq n$) **Faire**
 $P_{\Pi} = P_{\Pi} \cup (T_{3 \rightarrow 1}^{\Pi}(d_i^3) \cap P_{\Pi} \cup (T_{2 \rightarrow 1}^{\Pi}(d_i^2) \cap d_i^1);$
 $P_{\Pi'} = P_{\Pi'} \cup (T_{3 \rightarrow 1}^{\Pi'}(d_i^3) \cap T_{2 \rightarrow 1}^{\Pi'}(d_i^2) \cap d_i^1);$
 $P_{\Pi''} = P_{\Pi''} \cup (T_{3 \rightarrow 1}^{\Pi''}(d_i^3) \cap T_{2 \rightarrow 1}^{\Pi''}(d_i^2) \cap d_i^1);$
 Si $d_i^1 \notin P_{\Pi}$ **Alors** $Cop_{\Pi} = \text{Faux};$ /** S_i n'est pas coplanaire à Π **/
 Si $d_i^1 \notin P_{\Pi'}$ **Alors** $Cop_{\Pi'} = \text{Faux};$
 Si $d_i^1 \notin P_{\Pi''}$ **Alors** $Cop_{\Pi''} = \text{Faux};$
 $i = i + 1;$
Fin Faire
 /** Validation de la coplanarité **/
 Si (Cop_{Π} ou $Cop_{\Pi'}$ ou $Cop_{\Pi''}$)
Alors Retourner(*Coplanarité Validée*); /** 1^{ère} partie de la Contrainte 3 **/
Sinon
 Si (Non Aligné(P_{Π})) ou (Non Aligné ($P_{\Pi'}$)) ou (Non Aligné ($P_{\Pi''}$))
Alors Retourner(*Coplanarité non validée*); /** Cas d'échec **/
Sinon Retourner(*Coplanarité Validée*); /** 2^{ème} partie de la Contrainte 3 **/
Fin Fonction

5.7.3 Une méthode de construction des plans de références

Pour établir la contrainte de coplanarité de droites nous avons besoin de la donnée de trois plans de référence n'appartenant pas à un même faisceau de plans. Dans ce qui suit, nous répondons aux deux questions suivantes :

1. comment vérifier à partir de la donnée de trois systèmes de coordonnées projectives en correspondance, que les plans de références correspondants ne forment pas un faisceau de plans ?
2. quel est le nombre minimum d'images de points de référence dont nous avons réellement besoin pour décrire trois systèmes de coordonnées projectives en correspondance ?

Répondre à la première question revient à vérifier que les droites d'intersection des plans de références ne sont pas confondues, c'est-à-dire de vérifier que les images de ces droites, dans un des plans images par exemple le plan Im_1 , ne sont pas confondues. Reprenant les notations introduites dans la Section 5.7.1, il suffit de vérifier que $\delta_{\Pi'}^{\Pi'} \neq \delta_{\Pi''}^{\Pi''}$. Or, nous remarquons que $\delta_{\Pi'}^{\Pi'}$ (resp. $\delta_{\Pi''}^{\Pi''}$) n'est que la droite engendrée par la trace des points formant le système de bases en correspondance provenant de Π sur le plan de référence Π' (resp. de Π sur Π'') ou réciproquement. Ainsi la vérification est immédiate.

Répondons à présent à la seconde question. La nécessité de disposer de trois systèmes de coordonnées projectives en correspondance sous-entend qu'il nous faut connaître les images appariées de 12 points de référence. Ces points doivent se scinder en trois

groupes de quatre points coplanaires qui, de plus, ne doivent pas définir des plans faisant partie d'un même faisceau de plans. Nous allons montrer que nous n'avons besoin que de la donnée des images appariées de 6 points de référence (A, B, C, D, E, F) pour définir trois plans de références, et donc trois bases projectives en correspondance, qui ne forment pas un faisceau de plans. Pour ceci, il faut que les 6 points de référence soient tels que quatre d'entre eux soient coplanaires (supposons (A, B, C, D)) et définissent donc le plan de référence Π et que les deux autres points (E, F) n'appartiennent pas à Π . Nous pouvons dès lors procéder à la construction des deux autres plans de référence.

1- Construction du plan de référence Π'

Soit le triplet de points (A, E, F) . Ce triplet définit un plan 3D que nous considérons comme le plan de référence Π' . Afin d'avoir un système de bases projectives en correspondance provenant d'une base initiale de Π' , il nous faut rajouter un quatrième point de ce plan. Nous allons le construire en utilisant la notion de trace.

La trace des trois points (A, E, F) sur Π définit une droite passant par a_1 l'image connue de A sur le plan image Im_1 . Cette droite n'est autre que la droite intersection des deux plans de référence notée $\delta_{\Pi}^{\Pi'}$. Prenons alors un point p_1 quelconque de la droite $\delta_{\Pi}^{\Pi'}$ tel que $p_1 \neq a_1$ et associons lui les points p_2 et p_3 définis comme suit : $p_2 = \mathcal{T}_{1 \rightarrow 2}^{\Pi}(p_1)$ et $p_3 = \mathcal{T}_{1 \rightarrow 3}^{\Pi}(p_1)$. Le triplet de points (p_1, p_2, p_3) est alors, par construction, l'image d'un point 3D P appartenant au plan Π' . Nous avons ainsi construit un quatrième point qui s'ajoute aux trois premiers pour former une base projective de Π' : (A, E, F, P) . Nous avons par conséquent construit un système de bases projectives en correspondance attaché au plan Π' .

2- Construction du plan de référence Π''

Il est évident qu'une démarche analogue à celle décrite plus haut va nous permettre de construire le troisième système de bases projectives en correspondance qui nous est nécessaire. Ainsi (b_1 étant l'image de B dans Im_1) :

1. nous prenons Π'' le plan défini par les trois points (B, E, F) , si $b_1 \notin \delta_{\Pi}^{\Pi'}$.
2. nous prenons Π'' le plan défini par les trois points (C, E, F) , si $b_1 \in \delta_{\Pi}^{\Pi'}$. Comme (A, B, C, D) forme une base projective si $b_1 \in \delta_{\Pi}^{\Pi'}$, il est vrai que $c_1 \notin \delta_{\Pi}^{\Pi'}$.

Ainsi, à partir de la donnée de six points de référence, nous venons de montrer que nous pouvons construire trois systèmes de bases projectives en correspondance provenant de trois plans de référence Π , Π' et Π'' qui ne forment pas un faisceau de plans.

5.8 Autre approche pour détecter la coplanarité de points

Dans cette section nous présentons une autre approche qui s'inscrit dans un contexte proche du nôtre : développer une méthode permettant d'infirmer ou de confirmer la coplanarité d'un ensemble de points. Cette approche a été énoncée par Mohr [MQVB92].

5.8.1 Utilisation de la géométrie épipolaire

Nous présentons ici une méthode de recherche de coplanarité qui utilise explicitement les épipolaires. Cette méthode nous a été décrite par Mohr. Nous reproduisons dans ce qui suit la propriété ainsi que sa démonstration présentée dans [MQVB92] et [Fau92].

Contrainte 4 Soient A, B, C, D quatre points $\mathcal{3}D$, non trois à trois colinéaires, qui se projettent respectivement en a_1, b_1, c_1, d_1 et en a_2, b_2, c_2, d_2 sur les deux plans images Im_1 et Im_2 . Soit un couple de diagonales et son homologue dans les deux plans images (a_i, b_i et c_i, d_i par exemple). L'intersection des diagonales dont aucune n'est confondue avec une épipolaire donne, dans chaque plan image, un point d'intersection noté I_1 et I_2 .

(A, B, C, D) sont coplanaires $\iff (I_1, I_2)$ sont sur des épipolaires homologues

Démonstration

(\implies) Il est évident que si (A, B, C, D) soient coplanaires alors (I_1, I_2) sont des homologues et donc situés sur des épipolaires homologues.

(\impliedby) Supposons que (A, B, C, D) sont non coplanaires. Posons D' le point d'intersection de la ligne de vue de d_1 avec le plan défini par les trois points (A, B, C) . Ce point se projette dans le plan image Im_2 en un point d'_2 distinct de d_2 . Par construction d'_2 et d_2 appartiennent à la même épipolaire c'est-à-dire celle associée à d_1 . Posons I'_2 le point d'intersection des deux diagonales a_2, b_2 et c_2, d'_2 . Ce point appartient à Δ_2 l'épipolaire associée au point I_1 dans le plan image Im_2 . Le point I_2 peut-il appartenir à Δ_2 ? Cela ne peut être vrai que si Δ_2 est confondue avec une des diagonales a_2, b_2 et c_2, d_2 ce qui est impossible. \diamond

Afin d'éviter les cas limites où une des diagonales coïncide exactement avec une épipolaire il suffit de prendre un autre couple de diagonale.

5.8.2 Discussion

Remarquons que la méthode de Mohr (cf. Section 5.8.1) peut-être, au contraire de la notre, appliquée aux systèmes stéréoscopique binoculaire. En revanche, elle nécessite la connaissance des épipoles des deux caméras utilisées. Son implantation est facile (tous les calculs sont fait dans les plans images), et les quelques expérimentations que nous en avons faites ont prouvé l'efficacité de cette méthode.

Enfin pour terminer, signalons un avantage certain présenté par la méthode qui repose sur les traces, lors de la recherche des groupes de points coplanaires. En effet, la méthode décrite dans la Section 5.6.4 permet de décrire un moyen simple pour engendrer des hypothèses de coplanarité entre groupes de triplet de points: puisque seuls les ensembles de points présentant une même trace sur un même plan de référence sont des candidats potentiels pour vérifier une coplanarité. Ce critère permet de réduire l'espace de recherche des ensembles de triplets de points coplanaires ce que la méthode

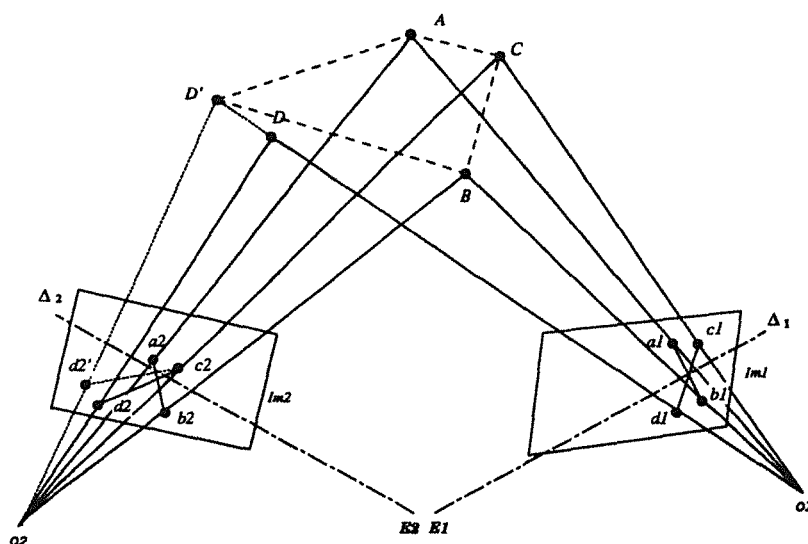


Figure 5.16. *Contrainte de coplanarité de points utilisant la géométrie épipolaire.*

des épipolaires ne permet pas. Cette recherche des ensemble de points (ou encore de droites) coplanaires fait l'objet d'une étude plus approfondie dans le prochain chapitre.

5.9 En résumé

Dans ce chapitre nous avons abordé le problème de l'étude de la coplanarité des primitives dans une approche qui s'intègre dans le cadre de recherche de conservation d'invariants projectifs. Cette approche a donc un champ d'application plus large que celui auquel nous la destinons : la construction de facettes 3D, où tous les paramètres de calibrage sont connus. En effet, nous avons uniquement supposé que la phase d'appariement des primitives, de type points, segments ou droites, a été faite au cours d'un traitement antérieur et nous n'avons fait aucune hypothèse sur la connaissance ou non des données des paramètres du calibrage du banc stéréoscopique utilisé. Cependant, nous avons exigé de disposer de trois caméras dont les centre optiques sont non alignés. Puis, nous avons supposé que nous connaissons des systèmes de bases projectives en correspondance.

La notion fondamentale que nous avons défini et utilisé pour établir nos différentes propriétés est la notion de trace d'une primitive sur un plan de référence. Le point fort de cette notion est incontestablement, que cette trace peut-être complètement déterminée à partir de calculs effectués dans un plan image quelconque, donc un espace 2D, et ce en utilisant la géométrie projective. Puis, utilisant cette dernière et en mettant en application des résultats de la géométrie projective nous avons proposé des contraintes permettant de déterminer la coplanarité d'un ensemble de couples de points homologues et d'un ensemble de couples de droites homologues. L'application directe de ces résultats pour la reconstruction 3D de facettes est présentée dans le chapitre suivant.

Nous avons également montré que si nous ne connaissons qu'un seul système de bases projectives en correspondance, donc un unique plan de référence, tout problème de coplanarité de primitives (droites ou segments) étudié à partir de la trace se ramène à un problème de coplanarité de points (Contrainte 2). Mais dès que nous avons trois systèmes de bases projectives en correspondance, donc trois plans de référence et plus précisément 6 points de références, tout problème de coplanarité peut être ramené à un problème de coplanarité de droites (Contrainte 3).

6

Utilisation du test de coplanarité

Ce chapitre décrit d'une part les différents processus de l'étape de prétraitement, spécifique à la partie stéréovision de notre système et dont le principe est présenté dans la section 6.1, et d'autre part l'étape de reconstruction 3D. Tout d'abord, comme la mise en application des résultats du chapitre précédent nécessite la donnée de systèmes de bases projectives en correspondance, nous réservons la Section 6.2 pour décrire la méthode utilisée pour la construction de ces derniers. Par ailleurs, sortant du cadre des données théoriques (non bruitées), la mise en application de ces techniques, fondées sur la géométrie projective, nécessite la prise en compte du bruit dans les images. La Section 6.3 traite de ce problème de l'erreur et présente les solutions que nous y apportons. Puis, nous passons dans les sections 6.4 et 6.5 à la description respectivement du test 2D de planarité d'une facette et du processus permettant de construire les groupes de triplets de facettes homologues images de facettes 3D coplanaires. La Section 6.6, termine l'étude des processus de prétraitement, en présentant la technique qui permet de détecter en 2D les facettes qui ne vérifient pas la propriété d'opacité. Nous passons ensuite, dans la Section 6.7, à la présentation du processus de reconstruction des facettes 3D. Ce processus qui repose sur l'information *trace* permet de réaliser une reconstruction sous contrainte de planarité et de coplanarité. La section 6.9 clot ce chapitre en discutant les résultats présentées dans la Section 6.8.

6.1 Processus composant l'étape de prétraitement

Nous avons supposé que les facettes 2D représentent la projection des surfaces planes de la scène. Notre objectif, à présent va consister à vérifier en 2D cette hypothèse, de manière à contraindre le processus de construction de facettes 3D à maintenir cette planarité. Le chapitre précédent décrit une méthode qui permet, à partir de la donnée de plans de références, de déterminer en 2D les propriétés vérifiées par des primitives 3D dont on ne connaît que les images appariées. Dans le constructeur de facettes, la mise en application de cette méthode, fondée sur la notion de trace, permet d'effectuer une *étape de prétraitement* sur les triplets de facettes homologues avant de reconstruire les facettes 3D. Les différents processus qui composent cette étape de prétraitement sont (Figure 6.1):

le processus de recherche de planarité : ce processus permet d'écarter de l'ensemble des triplets de facettes homologues les triplets qui ne vérifient pas l'hypothèse de planarité sous-jacente à la notion de facettes 2D,

le processus de recherche des classes de facettes coplanaires : toujours dans l'optique d'extraire en 2D le plus d'information possible sur les propriétés 3D relatives aux facettes à reconstruire, nous utilisons de nouveau l'information trace associée aux triplets de facettes 2D pour rechercher les classes de triplets de facettes homologues images de facettes 3D coplanaires,

le processus de recherche des facettes non opaques : en utilisant la notion de système de bases projectives en correspondance nous décrivons une technique qui permet de déterminer les facettes qui ne sont pas opaques, ce qui peut être intéressant pour déterminer les facettes virtuelles.

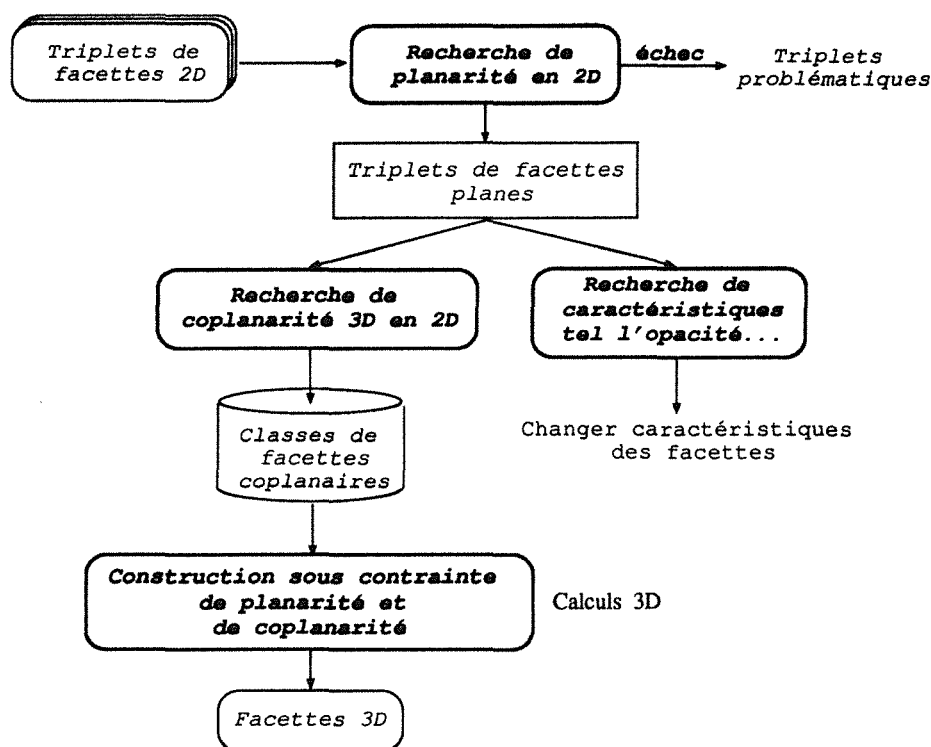


Figure 6.1. Prétraitement des triplets de facettes homologues et construction des facettes 3D

6.2 Construction des systèmes de bases projectives

Rappelons que les résultats présentés dans le chapitre précédent reposent sur la donnée *a priori* de trois systèmes de bases projectives en correspondance. Comme dans notre cadre d'application nous disposons des paramètres de calibrage, nous proposons une méthode très simple et efficace pour la construction de ces systèmes de bases projectives ; nous fixons arbitrairement dans l'espace trois plans Π_1 , Π_2 et Π_3 tel que :

- (i) chaque plan Π_i répond à la définition d'un plan de référence (Définition 15),
- (ii) Π_1, Π_2 et Π_3 ne définissent pas un faisceau de plans.

Puis, nous fixons dans chacun de ces trois plans, quatre points définissant une base projective initiale. Utilisant alors les matrices de calibration nous associons à chaque base projective initiale, le système de bases projectives en correspondance qu'elle induit. D'autre part, sachant que plus les points d'une base projective sont écartés meilleure est la précision obtenue sur le calcul des coordonnées projectives [MQVB92], nous prenons soin de bien choisir les points virtuels de manière à obtenir des bases projectives composées de points écartés.

Remarque 10 *Cette méthode de construction des bases projectives en correspondance nous permet d'émettre l'hypothèse, que dans notre cadre d'application, nous utilisons des bases projectives "parfaites". Ceci justifie la relation univoque que nous établissons entre les erreurs trouvées sur la trace et les erreurs inhérentes aux primitives 2D dans les images réelles.*

En fait, la mise en application des résultats du chapitre précédent ainsi que ceux de ce chapitre, nécessite également la donnée des images des droites d'intersections $\delta_{\Pi_i}^{\Pi_j}$ des plans de références entre eux (cf. Section 6.5) et de la droite à l'infini de chacun de ces plans relativement au plan image Im_1 (cf. Section 6.3). Nous désignons par $\mathcal{D}_{\infty}^{\Pi_1}$, $\mathcal{D}_{\infty}^{\Pi_2}$ et $\mathcal{D}_{\infty}^{\Pi_3}$ ces droites à l'infini qui correspondent respectivement aux plans de référence Π_1 , Π_2 et Π_3 .

Connaissant parfaitement les plans de références en 3D, nous déterminons la droite $\Delta_{\Pi_i}^{\Pi_j}$ intersection des plans Π_i et Π_j . La projection de cette dernière dans le plan image, dans lequel nous effectuons les calculs, est la droite¹ $\delta_{\Pi_i}^{\Pi_j}$.

Quant à la construction² de la droite à l'infini de chacun des plans de référence, elle se fait en utilisant la propriété suivante (Figure 6.2) :

Propriété 7 *La droite à l'infini D_{∞}^{Π} d'un plan Π dans le plan image Im est égale à la droite d'intersection du plan Im et du plan Π_O : plan parallèle au plan Π et passant par le centre optique O .*

Rappelons, enfin, un résultat, qui nous est utile pour la suite, relatif au point de fuite de toute droite appartenant à un plan Π :

Propriété 8 *Le point de fuite p_{∞} d'une droite δ image d'une droite $\Delta \in \Pi$, est définie comme étant le point d'intersection de δ et de la droite D_{∞}^{Π} droite à l'infini du plan Π dans l'image Im : $p_{\infty} = \delta \cap D_{\infty}^{\Pi}$ (Figure 6.2).*

1. Notons que $\delta_{\Pi_i}^{\Pi_j}$ peut théoriquement être reconstruite en n'effectuant que des calculs 2D. Pour ce faire, il suffit de remarquer que la droite $\delta_{\Pi_i}^{\Pi_j}$ n'est autre que la trace du plan Π_i sur le plan Π_j (ou vice-versa). Cette droite est donc construite en considérant la trace des 4 points coplanaires provenant de la base \mathcal{B}^{Π_i} rattachée au plan Π_i relativement au plan de référence Π_j .

2. Les équations nécessaires à cette construction sont fournies en annexe.

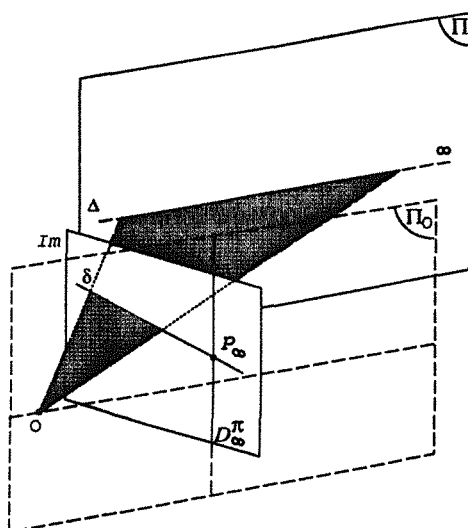


Figure 6.2. La construction : de la droite à l'infini D_∞^Π du plan Π et du point de fuite p_∞ de la droite Δ appartenant à Π .

6.3 A propos des erreurs sur les traces

Les primitives 2D extraites des images réelles sont souvent entachées d'erreurs. Ces erreurs, dues au bruit, se répercutent sur le résultat de tout calcul qui utilise ces primitives. Dans notre cas, nous avons pu constater expérimentalement que l'erreur de localisation³ des droites 2D d'un triplet de droites homologues (l_1, l_2, l_3) ne permet plus de vérifier la convergence théorique des trois droites l_1 , $\mathcal{T}_{2 \rightarrow 1}^\Pi(l_2)$ et $\mathcal{T}_{3 \rightarrow 1}^\Pi(l_3)$. Or comme nous l'avons montré la trace T_L^Π de la droite L sur un plan de référence Π , qui est par définition l'image du point d'intersection de L avec le plan Π , se calcule comme suit (§ 5.4.2) :

$$T_L^\Pi = l_1 \cap \mathcal{T}_{2 \rightarrow 1}^\Pi(l_2) \cap \mathcal{T}_{3 \rightarrow 1}^\Pi(l_3) \quad (6.1)$$

La non convergence constatée expérimentalement, met en défaut ce processus de calcul théorique de la trace de la droite L sur le plan de référence Π , puisque en pratique nous obtenons dans la majorité des cas :

$$T_L^\Pi = \emptyset \quad (6.2)$$

Afin de mieux comprendre l'impact de l'erreur entachées aux primitives 2D sur le calcul de la trace et par conséquent de mieux l'interpréter nous allons visualiser le problème en 3D.

6.3.1 Origine et interprétation de l'erreur

Soit S un segment ayant pour droite support la droite L et admettant pour images 2D le triplet (s_1, s_2, s_3) . Soit (l_1, l_2, l_3) le triplet de droites support de (s_1, s_2, s_3) . Etu-

3. D'après la Remarque 10 seule l'erreur sur les primitives 2D est en cause.

dions tout d'abord, l'intersection binaire suivante⁴ :

$$p_{(i,j)} = \mathcal{T}_{i \rightarrow 1}(l_i) \cap \mathcal{T}_{j \rightarrow 1}(l_j) \text{ avec } i \neq j \text{ et } i = 1..3, j = 1..3 \quad (6.3)$$

L'équation 6.3 détermine en fait l'image de l'intersection de la droite support du segment $S_{(i,j)}$: segment résultant de la triangulation à partir du couple de segments 2D (s_i, s_j) , et du plan Π . C'est là précisément que réside l'explication de la non convergence des droites l_1 , $\mathcal{T}_{2 \rightarrow 1}(l_2)$ et $\mathcal{T}_{3 \rightarrow 1}(l_3)$. En effet, comme l'ont constaté tous ceux qui ont travaillé dans le cadre de la stéréovision et de la reconstruction de segments 3D, en pratique les trois plans d'interprétations issus d'un triplet de segments homologues ne forment pas un faisceau de plans (Figure 6.3.a). L'erreur de localisation sur les primitives 2D, entraîne lors de la triangulation la reconstruction de trois segments 3D $(S_{(1,2)}, S_{(1,3)}, S_{(2,3)})$ approximant le segment réel S .

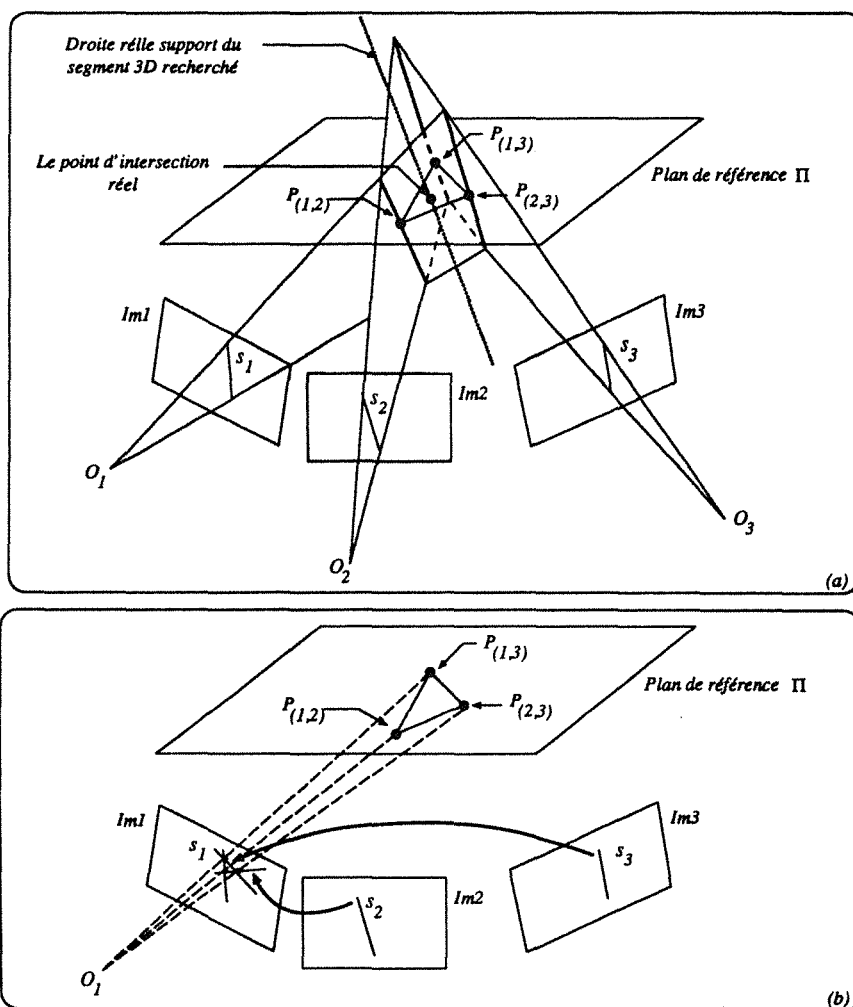


Figure 6.3. Erreur dans la construction des traces. (a): les erreurs de reconstruction représentées en 3D. (b): leur conséquences sur la recherche des traces.

4. l'application $\mathcal{T}_{1 \rightarrow 1}$ est l'application identité.

En supposant qu'il n'y ait pas de biais systématique sur les mesures, nous pouvons alors admettre que le segment S se situe à l'intérieur du volume V délimité par les droites support des trois segments reconstruits (S_1, S_2, S_3) . Le point d'intersection de la droite support de S avec un plan de référence Π se situe alors à l'intérieur de la surface correspondant à l'intersection du plan Π avec le volume V . Par conséquent la trace T_S^Π recherchée se situe dans l'image $Triang_{(s_1, s_2, s_3)}$ de cette surface. Nous disons que $Triang_{(s_1, s_2, s_3)}$ représente la *zone de présence de la trace* recherchée. Il est facile de constater que $Triang_{(s_1, s_2, s_3)}$ est en fait délimitée par les trois traces suivantes (Figure 6.3.b) :

$$\begin{cases} p_{(1,3)} = l_1 \cap \mathcal{T}_{3 \rightarrow 1}(l_3) \\ p_{(1,2)} = l_1 \cap \mathcal{T}_{2 \rightarrow 1}(l_2) \\ p_{(2,3)} = \mathcal{T}_{3 \rightarrow 1}(l_3) \cap \mathcal{T}_{2 \rightarrow 1}(l_2) \end{cases} \quad (6.4)$$

Cas d'un segment appartenant à un plan épipolaire

Examinons maintenant l'effet de l'erreur sur les triplets particuliers de segments (ou de droites) homologues qui contiennent un couple de segments (supposons (s_1, s_2)) dont l'orientation est proche de l'orientation des épipolaires. Une telle orientation indique que nous sommes proche du cas caractéristique, mentionné dans la Propriété 4, où la droite 3D appartient à un plan épipolaire. Cette propriété indique qu'aucune information pertinente sur la trace ne peut être extraite de ce couple de segments.

Cependant en raison des erreurs et lors de l'expérimentation, les droites l_1 et $\mathcal{T}_{2 \rightarrow 1}(l_2)$ ne sont pas confondues et ont un point d'intersection qui a la "sémantique" d'un point de trace. Or, ce point issu de $l_1 \cap \mathcal{T}_{2 \rightarrow 1}(l_2)$ est par essence erroné. Aussi, pour éviter la prise en compte de ces points de trace éronnés, nous choisissons d'écarter, pour tout triplet de segments homologues contenant un couple de segments (s_i, s_j) dont l'orientation est proche des épipolaires, le point issu de trace issu de ces deux segments pour ne garder que les deux autres points de trace.

6.3.2 La prise en compte des erreurs

Tout l'intérêt de la trace réside dans la possibilité de vérifier la coplanarité d'un ensemble de primitives 3D à partir de la donnée de leur triplets d'images. Cette vérification revient à prouver que l'ensemble des primitives homologues vérifie l'une des deux conditions suivantes (cf. Contrainte 3 et Contrainte 2) :

- (i) soit la coïncidence entre les homologues réels et ceux reconstruits par $\mathcal{T}_{2 \rightarrow 1}$ et $\mathcal{T}_{3 \rightarrow 1}$ au quel cas, les primitives 3D sont coplanaires au plan de référence utilisé,
- (ii) soit l'alignement des traces associées à cet ensemble de triplets de primitives 2D.

Nous allons à présent décrire la démarche que nous avons suivit dans chacun de ces cas, pour tenir compte de l'erreur.

Cas (i) : coplanarité de la facette avec le plan de référence

En raison de l'importance des erreurs dans les images, il n'est pas possible de vérifier la superposition exacte des droites l_1 , $\mathcal{T}_{2 \rightarrow 1}(l_2)$ et $\mathcal{T}_{3 \rightarrow 1}(l_3)$.

Afin de tenir compte de cette remarque, nous allons en pratique vérifier que les droites l_1 et $\mathcal{T}_{3 \rightarrow 1}(l_3)$, respectivement l_1 et $\mathcal{T}_{2 \rightarrow 1}(l_2)$, ont des directions qui sont dites α -confondues car elles définissent entre elles un angle plus petit qu'un seuil de tolérance α donné (α de l'ordre de 3 degré).

Cas (ii) : pas de coplanarité de la facette avec le plan de référence

Soit $S_{2d} = \{ts_1, \dots, ts_n\}$ un ensemble de n triplets de segments 2D homologues dont nous voulons étudier la coplanarité 3D. Dans § 6.3.1 nous avons vu qu'en raison de l'erreur nous étions amené à associer à chaque triplet de segments 2D ts_i non plus un point de trace mais une zone de présence $Triang_i$ relative à la trace du segment 3D sur le plan Π . De ce fait, pour vérifier l'hypothèse de coplanarité relative à S_{2d} , nous allons moduler la condition d'alignement des points de traces en cherchant à prouver l'existence d'une droite unique qui traverse les différentes zones de présence $Triang_i$, $i = 1..n$. Le problème revient alors à décrire une méthode de construction de cette droite.

Revenons sur la notion de zone de présence $Triang_i$ relative à la trace du segment 3D sur le plan Π . La zone de présence $Triang_i$ est en fait l'image de la surface $Surf_i$ intersection du plan Π avec le volume V_i associé au triplet ts_i . Si nous supposons que le segment réel S_i associé au triplet ts_i a une forte probabilité de se situer au centre de V_i , le point d'intersection de S_i avec Π a par conséquent une forte probabilité d'être au centre de gravité de cette surface $Surf_i$. Cette remarque, retranscrite en terme de trace, implique que la trace d'un segment est supposée être proche de l'image de l'isobarycentre de $Surf_i$ et qui est donc un point situé dans $Triang_i$. Il est évident qu'étant dans le cadre d'une projection perspective la mesure du barycentre qui est une mesure affine n'est pas conservée : l'image de l'isobarycentre de $Surf_i$ n'est pas égale à l'isobarycentre de $Triang_i$. Il faut alors construire l'image de l'isobarycentre dans Im_1 sans calculs 3D. Nous apportons, dans la Section 6.3.3, une solution à ce problème.

En résumé, notre démarche pour prouver la coplanarité d'un ensemble de segments consiste en un premier temps à rechercher au sens des moindres carrés la droite δ_{approx} qui approxime au mieux les images des isobarycentres des surfaces de présence. Puis, dans un deuxième temps et pour valider la coplanarité des primitives 3D nous devons vérifier que la droite δ_{approx} traverse bien chacune des zones de présence associée à la trace (Figure 6.4).

6.3.3 Construction de l'image d'un isobarycentre

Avant de présenter la reconstruction de l'image de l'isobarycentre d'un ensemble de trois points 3D à partir de leur points images, nous commençons par présenter le procédé de construction de l'image g_{ab} du point milieu G_{AB} de deux points A et B .

Image de l'isobarycentre de deux points

A partir des images a, b des points A, B et de la donnée du point de fuite de la droite passant par ces deux points il est possible de calculer l'image de l'isobarycentre

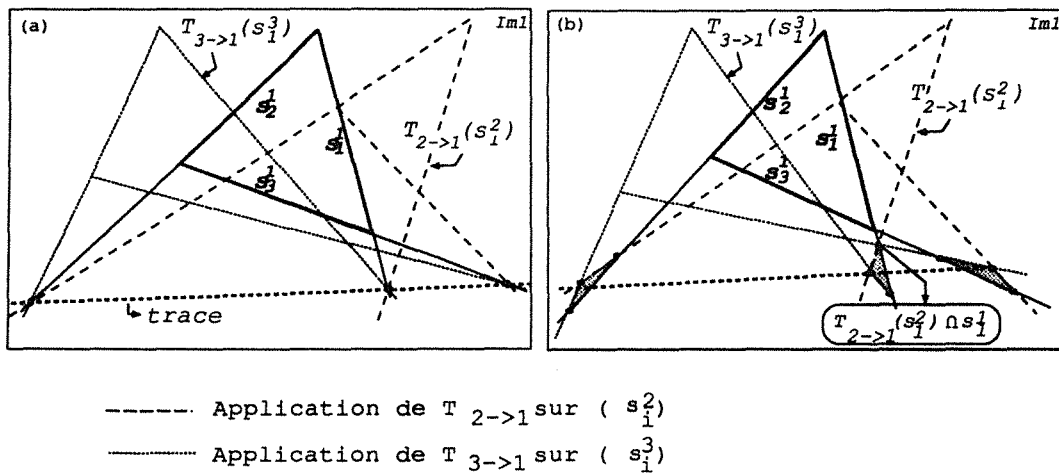


Figure 6.4. La droite de trace associée à un ensemble de segments coplanaires : (a) en théorie : elle passe par les points de traces, (b) en pratique : elle traverse chacune des zones de présence.

de A, B . Comme les mesures algébriques ne sont plus conservées, la reconstruction repose sur l'utilisation de l'unique invariant disponible : *le birapport*.

Rappelons ici une convention importante utilisée en géométrie projective :

$$\frac{\infty}{\infty} = 1 \tag{6.5}$$

Utilisant la Convention 6.5 la valeur du birapport des quatre points particuliers, A, B, G_{AB} et $\infty_{(AB)}$: le point à l'infini associé à la droite passant par A et B , est la suivante :

$$[A, G_{AB}, B, \infty_{(AB)}] = \frac{\overline{BA}}{\overline{BG_{AB}}} * \frac{\overline{\infty_{(AB)}G_{AB}}}{\overline{\infty_{(AB)}A}} = 2 \tag{6.6}$$

Par conservation du birapport, nous en déduisons alors, l'égalité suivante définie entre les quatres points images :

$$[a, g_{ab}, b, p_{\infty}] = \frac{\overline{ba}}{\overline{bg_{ab}}} * \frac{\overline{p_{\infty}g_{ab}}}{\overline{p_{\infty}a}} = 2 \tag{6.7}$$

où p_{∞} est le point de fuite de la droite passant par les points A et B : image du point à l'infini $\infty_{(AB)}$. Ce point p_{∞} est entièrement déterminé par la donnée de a et b et par la droite à l'infini du plan de référence considéré (cf. Propriété 8).

Comme les points a, b et p_{∞} sont trois points alignés qui forment une base projective de la droite δ l'équation 6.7 permet de définir un unique point g_{ab} qui est l'image de l'isobarycentre⁵ des points 3D A et B (Figure 6.5).

5. Les équations nécessaires à cette construction sont données en annexes.

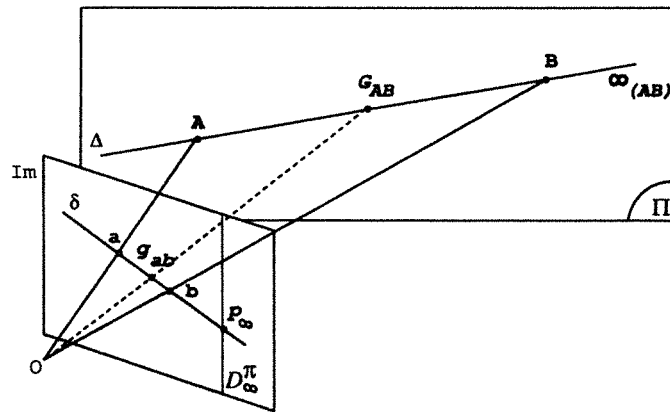


Figure 6.5. La construction de l'isobarycentre de deux points.

Image de l'isobarycentre de trois points

Pour construire l'isobarycentre de trois points nous allons suivre un raisonnement analogue à celui décrit plus haut. Soit (a, b, c) trois points images et g_{abc} l'image de leur isobarycentre en 3D à construire. Soit G_{AB} le point milieu de A et B . Le point 3D G_{ABC} isobarycentre de (A, B, C) est alors défini par l'équation suivante :

$$[G_{AB}, G_{ABC}, C, \infty_{(C, G_{AB})}] = 3/2 \quad (6.8)$$

ainsi nous en déduisons que :

$$[g_{ab}, g_{abc}, c, p_{\infty}^{g_{ab}c}] = 3/2 \quad \text{où} \quad (6.9)$$

1. g_{abc} est l'image du point G_{ABC} ,
2. g_{ab} l'image de G_{AB} du point milieu de A et B , est construit en utilisant l'équation 6.7,
3. $p_{\infty}^{g_{ab}c}$ est le point de fuite de la droite passant par les deux points : c et g_{ab} .

Ainsi, l'équation 6.8 permet de construire l'image de l'isobarycentre de chacune des zones de présence de la trace d'un triplet de primitives 2D.

6.4 Processus de vérification de la planarité

Introduisons en premier lieu la définition de la trace d'une facette. Une facette 2D peut être décrite de deux manières différentes : soit en fonction des segments qui la compose soit en fonction de ses sommets.

Définition 20 La trace $T_u^{\Pi_i}$, d'un triplet TF_u de facettes 2D sur un plan de référence Π , est la trace sur Π associée suivant la description adoptée pour TF_u , soit aux triplets de segments soit aux triplets de sommets.

Les résultats introduits dans le chapitre précédent nous offrent deux approches possibles pour effectuer la vérification de la planarité relative aux triplets de facettes 2D. En effet, nous pouvons étudier la planarité :

- (i) soit en utilisant la contrainte de coplanarité de points,
- (ii) soit en utilisant la contrainte de coplanarité de segments.

L'approche (i) sous entend l'utilisation de la description des facettes 2D au moyen de leur sommets obtenus par intersection des couples de segments adjacents composant une facette et sous entend également la mise en œuvre de l'une des deux méthodes introduites dans le chapitre précédent : soit celle fondée sur l'utilisation des épipoles (Section 5.8.1) ou bien celle qui utilise le principe de la trace induite par la Contrainte 2 (Section 5.6).

La critique essentielle que nous faisons à l'approche (i) est d'utiliser les points sommets pour décrire les facettes 2D. Or, en plus de la mauvaise localisation de ses points nous pouvons engendrer des faux triplets de sommets. En effet, comme le montre la Figure 6.6 les phénomènes d'occultation peuvent engendrer de faux sommets et donc de faux triplets de points.

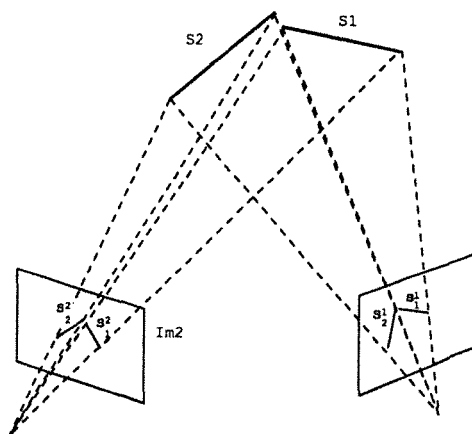


Figure 6.6. Un couple de points images issu de l'intersection des images de deux segments non coplanaires

De plus, l'utilisation des points ne permet pas d'étendre ce test aux facettes 2D problématiques : chaînes ouvertes de segments 2D. En effet, comme ces dernières ne sont pas des polygones 2D, elles ne peuvent être complètement décrites en terme de sommets. Les points extrémités des segments extrêmes de la chaîne, composant la facette, doivent alors faire partie de la description de la facette en terme de points caractéristiques. Or, les points extrémités d'un triplets de segments homologues ne sont pas forcément homologues. S'introduit alors le problème de recherche des vrais homologues d'un point extrémité d'un segment. Ceci compromet donc l'extension du système actuel à la prise en compte des facettes dites incertaines.

Pour ces raisons nous avons retenue pour l'implantation l'approche (ii) [Cha93]. Cette approche utilise la description des facettes au moyen de la suite de segments

2D qui la composent ce qui permet la mise en application de la Contrainte 3 relative à la coplanarité de droites. Ainsi, étant donné un triplet de facettes 2D homologues : $TF_u = (f_1, f_2, f_3)$, nous procédons à la construction de sa trace relativement à chacun des trois plans de référence Π_1, Π_2 et Π_3 . Ces traces sont calculées à partir des triplets de segments homologues relatifs à TF_u :

$$TF_u = (f_1, f_2, f_3) \rightarrow \{(s_1^1, s_1^2, s_1^3), \dots, (s_n^1, s_n^2, s_n^3)\} \text{ où } s_j^i \in f_i, j = 1..n \text{ et } i = 1..3$$

6.4.1 Test de planarité

Le processus de vérification de la planarité des facettes à reconstruire repose sur les explications fournies dans le Section 6.3. Nous allons expliquer le processus de construction de traces pour le cas de non coplanarité de la facette et des plans de références. Relativement à chaque plan de référence Π_i , nous déterminons pour chaque triplet de segment (s_j^1, s_j^2, s_j^3) sa zone de présence $Triang_{(s_j^1, s_j^2, s_j^3)}^{\Pi_i}$ de la trace sur le plan Π que nous notons pour des raisons de simplicité $\nabla_j^{\Pi_i}$ et l'image de l'isobarycentre $ISO_j^{\Pi_i}$ relatif à cette dernière. Ainsi, nous avons associé à chaque triplet l'information trace relative à chacun des plans de référence $\Pi_i, i = 1..3$, comme suit :

$$(f_1, f_2, f_3) \rightarrow \{(s_1^1, s_1^2, s_1^3), \dots, (s_n^1, s_n^2, s_n^3)\} \rightarrow \begin{cases} \{(\nabla_1^{\Pi_1}, ISO_1^{\Pi_1}), \dots, (\nabla_n^{\Pi_1}, ISO_n^{\Pi_1})\} \\ \{(\nabla_1^{\Pi_2}, ISO_1^{\Pi_2}), \dots, (\nabla_n^{\Pi_2}, ISO_n^{\Pi_2})\} \\ \{(\nabla_1^{\Pi_3}, ISO_1^{\Pi_3}), \dots, (\nabla_n^{\Pi_3}, ISO_n^{\Pi_3})\} \end{cases}$$

Pour chaque ensemble $ISO^{\Pi_i} = \{ISO_j^{\Pi_i} : (x_j^{\Pi_i}, y_j^{\Pi_i}), j = 1..n\}$ nous allons calculer les paramètres $(a_{\Pi_i}^*, b_{\Pi_i}^*, c_{\Pi_i}^*)$ de la droite T^{Π_i} approchant au mieux les points de ISO^{Π_i} . Il s'agit donc de chercher $(a_{\Pi_i}^*, b_{\Pi_i}^*, c_{\Pi_i}^*)$ qui minimise au sens des moindres carrés la relation suivante :

$$\sum_{i=1}^n (a_{\Pi_1}^* x_i^{\Pi_1} + b_{\Pi_1}^* y_i^{\Pi_1} + c_{\Pi_1}^*)^2 \quad (6.10)$$

Nous vérifions ensuite que chaque droite T^{Π_i} ainsi calculée traverse chaque zone de présence $\nabla_j^{\Pi_i}$ associée à l'ensemble ISO^{Π_i} . Les triplets qui ne vérifient pas cette dernière condition sont décrétés non plane et considérées comme problématiques alors que les autres triplets sont considérés provenir d'une facettes 3D planes. Ainsi pour les triplets de facettes TF_u considérés comme plans, nous avons calculé trois droites $T_u^{\Pi_1}, T_u^{\Pi_2}$ et $T_u^{\Pi_3}$ qui approximent aux mieux leur trace sur chacun des plans de références Π_i .

6.4.2 Traces caractéristiques

Pour terminer notons qu'après application du test de planarité, nous pouvons répondre, pour chaque triplet de facettes homologues prouvé plan, aux questions suivantes :

1. ce triplet est-il l'image d'une facette coplanaire à un des plans de référence Π_i ?

2. ce triplet appartient-il à un des faisceaux de plans définis par chacun des couples de plans de référence (Π_i, Π_j) ?

Les réponses faites à ces questions ont une grande importance pour la recherche des classes coplanaires comme il est montré dans la prochaine section. Pour l'instant, ces réponses vont nous permettre de rappeler la notion de traces caractéristiques présentée dans le chapitre précédent (Remarque 9). Il est évident qu'une fois la planarité validée, l'information trace relativement au trois plans de références est redondante. En effet :

1. l'information de coplanarité d'une facette à un des plans de référence Π_i , obtenue par application du test de planarité, rend redondante l'information trace sur les deux autres plans,
2. dans le cas où le triplet de facettes homologues n'est pas l'image d'une facette coplanaire à un des plans de référence, deux traces seulement, dites caractéristiques, parmi les trois calculées $T_u^{\Pi_1}$, $T_u^{\Pi_2}$ et $T_u^{\Pi_3}$ suffisent à caractériser ce triplet. Nous reprenons alors la convention, introduite par la Remarque 9, et nous définissons pour chaque triplet un couple de traces caractéristiques (δ_1, δ_2) où :

$$\begin{cases} \text{Si } T_u^{\Pi_1} \neq T_u^{\Pi_2} \text{ Alors } (\delta_1, \delta_2) = (T_u^{\Pi_1}, T_u^{\Pi_2}) & \text{(a)} \\ \text{Si } T_u^{\Pi_1} = T_u^{\Pi_2} \text{ Alors } (\delta_1, \delta_2) = (T_u^{\Pi_1}, T_u^{\Pi_3}) & \text{(b)} \end{cases}$$

Remarquons que la formule (a) concerne en fait toutes les facettes qui ne font pas partie du faisceau de plans décrit par les plans de référence (Π_1, Π_2) alors que la formule (b) concerne uniquement les facettes faisant partie de ce faisceau de plans.

6.5 Processus de recherche des facettes coplanaires

L'application du test de planarité permet de partager l'ensemble initial de triplets de facettes 2D homologues en deux :

1. d'une part, l'ensemble \mathcal{F}_{plan} qui regroupe les triplets de facettes 2D vérifiant la planarité 3D,
2. d'autre part, l'ensemble \mathcal{F}_{prob} qui regroupe les triplets de facettes 2D ambiguës (non planes) et qui sont pour l'instant rejetée du système.

Nous nous proposons de partitionner l'ensemble des triplets de facettes de l'ensemble \mathcal{F}_{plan} , en classes de triplets images de facettes 3D coplanaires. Cette recherche permet de contraindre plus fortement la phase de reconstruction 3D en exigeant la recherche d'un unique plan support pour une classe de facettes coplanaires. Pour réaliser ce partitionnement, une méthode simple mais combinatoire, consiste à prendre deux à deux les triplets de facettes et à prouver que leur réunion produit un ensemble de triplets de segments homologues qui vérifie le test de coplanarité. L'approche que nous proposons est nettement moins combinatoire et permet d'inférer les facettes coplanaires entre elles au moyen de l'étude de leur traces déjà calculées [CB93].

Examinons, dans un premier temps, l'ensemble \mathcal{F}_{plan} . Cet ensemble peut être décomposé en cinq sous ensembles disjoints comme suit(cf. Section 6.4.2):

$$\mathcal{F}_{plan} = \mathcal{F}_{coplan}^{\Pi_1} \cup \mathcal{F}_{coplan}^{\Pi_2} \cup \mathcal{F}_{coplan}^{\Pi_3} \cup \mathcal{F}_{\Pi_2}^{\Pi_1} \cup \mathcal{F}_{\Pi_3}^{\Pi_1}, \text{ où} \quad (6.11)$$

- (a) $\mathcal{F}_{coplan}^{\Pi_i}$ $i = 1..3$: représente l'ensemble des triplets de facettes 2D qui vérifient la première partie de la Contrainte 3 et qui proviennent donc de la projection de facettes 3D coplanaires au plan de référence Π_i .
- (b) $\mathcal{F}_{\Pi_2}^{\Pi_1}$: représente les triplets de facettes 2D qui ne sont pas des projections de facettes 3D planes coplanaires à un des plans de référence choisis (cf. la Contrainte 3) et qui de plus ne font pas partie du faisceau de plans défini par les deux plans de références Π_1 et Π_2 . Le couple de traces caractéristiques de chacun des triplets de facettes est défini relativement aux plans de référence (Π_1, Π_2) .
- (c) $\mathcal{F}_{\Pi_3}^{\Pi_1}$: les triplets de facettes 2D qui ne sont pas des projection de facettes 3D planes coplanaires à un des plans de référence choisis mais qui font partie du faisceau de plans décrit par les deux plans de références Π_1 et Π_2 . Le couple de traces caractéristiques de chacun des triplets de facettes est défini relativement aux plans de référence (Π_1, Π_3) .

Ici, deux remarques s'imposent. La première porte sur le fait que tout triplet pris dans un de ces sous ensembles ne peut vérifier une coplanarité 3D avec un autre triplet d'un autre sous ensemble. La seconde porte sur le fait que chaque sous ensemble noté $\mathcal{F}_{coplan}^{\Pi_i}$ forment une classe d'équivalence de triplets provenant de la projection de facettes 3D coplanaires. Le problème revient alors à partitionner des deux sous-ensembles restants $\mathcal{F}_{\Pi_2}^{\Pi_1}$ et $\mathcal{F}_{\Pi_3}^{\Pi_1}$ en classes de triplets coplanaires.

A- Décomposition de $\mathcal{F}_{\Pi_j}^{\Pi_1}$

Soit TF_u un triplet de $\mathcal{F}_{\Pi_j}^{\Pi_1}$ $j = 2, 3$ tel que $T_u^{\Pi_1}$ et $T_u^{\Pi_j}$ soient les droites de traces caractéristiques associées à TF_u . Ce couple de traces caractéristiques relatifs aux deux plans de référence Π_1 et Π_j n'est pas indépendant. En effet, soit $\delta_{\Pi_1}^{\Pi_j}$ l'image de la droite intersection des deux plans de référence Π_1 et Π_j sur le plan image Im_1 , alors il est aisé de voir que la contrainte suivante lie les deux traces de TF_u avec $\delta_{\Pi_1}^{\Pi_j}$:

$$T_u^{\Pi_1} \cap T_u^{\Pi_j} \in \delta_{\Pi_1}^{\Pi_j} \quad (6.12)$$

Il est évident aussi qu'à cause de l'erreur cette contrainte inhérente au concept de trace ne va pas être exactement vérifiée en pratique. En effet, expérimentalement nous vérifions que les points d'intersection de $T_u^{\Pi_1} \cap \delta_{\Pi_1}^{\Pi_j}$ et $T_u^{\Pi_j} \cap \delta_{\Pi_1}^{\Pi_j}$ sont en fait situés à une distance ε l'un de l'autre. Comme cette caractéristique à la notion de trace nous est nécessaire pour établir les classes de coplanarité, nous allons procéder au réajustement du couple de traces caractéristiques associé à chaque triplet de facettes de façon à ce qu'il vérifie la Contrainte 6.12. Ce réajustement va revenir à poser un nouveau problème d'optimisation tenant compte de cette nouvelle contrainte.

Mais avant de présenter le système d'équations à résoudre soulignons qu'il existe deux représentation minimale pour une droite suivant son orientation. Cette représentation minimale permet de réduire le nombre de paramètres d'une droite à deux :

$$\begin{cases} \text{Cas des droites non horizontales : } x + b y + c = 0 \\ \text{Cas des droites non verticales : } a x + y + c = 0 \end{cases} \quad (6.13)$$

Ayant choisi la représentation adéquate pour chacune des droites à calculer nous pouvons poser le système d'équations. Dans la suite, nous le présentons uniquement dans le cas où le choix adéquat pour les deux droites est la première représentation du Système 6.13. Supposons que nous avons par ailleurs $\delta_{\Pi_1}^{\Pi_j} : x + \beta_1^j y + \gamma_1^j = 0$. Cela revient à poser le système de minimisation suivant et où il s'agit de retrouver les valeurs de $(b_{\Pi_1}^*, c_{\Pi_1}^*)$ et $(b_{\Pi_j}^*, c_{\Pi_j}^*)$:

$$\begin{cases} \text{Minimiser } \sum_{i=1}^n (x_i^{\pi_1} + b_{\pi_1}^* y_i^{\pi_1} + c_{\pi_1}^*)^2 \\ \text{Minimiser } \sum_{i=1}^n (x_i^{\pi_j} + b_{\pi_j}^* y_i^{\pi_j} + c_{\pi_j}^*)^2 \\ \text{Sous la contrainte : } \left(\frac{c_{\pi_j}^* b_{\pi_1}^* - c_{\pi_1}^* b_{\pi_j}^*}{b_{\pi_j}^* - b_{\pi_1}^*} \right) + \beta_1^j \left(\frac{c_{\pi_1}^* - c_{\pi_j}^*}{b_{\pi_j}^* - b_{\pi_1}^*} \right) + \gamma_1^j = 0 \end{cases} \quad (6.14)$$

La contrainte de ce système exprime la Relation 6.12. Cependant, ce n'est pas cette formulation directe du problème que nous allons utiliser pour retrouver la valeur de nos paramètres. En effet, nous allons changer la présentation du problème et ainsi aboutir à un problème de minimisation plus aisé à résoudre et admettant la même solution que le Système 6.14. Nous allons exprimer le fait que la droite $T_u^{\Pi_1}$ ainsi que la droite $T_u^{\Pi_j}$ passe par un même point, de coordonnées cartésiennes : (x_0, y_0) , qui appartient à la droite $\delta_{\Pi_1}^{\Pi_j}$. Les seuls paramètres sont alors $b_{\pi_1}^*$, $b_{\pi_j}^*$ et y_0 : nous allons donc minimiser la quantité A suivante :

$$A = \sum_{i=1}^n (x_i^{\pi_1} + b_{\pi_1}^* y_i^{\pi_1} + y_0(\beta_1^j - b_{\pi_1}^*) + \gamma_1^j)^2 + \sum_{l=1}^n (x_l^{\pi_j} + b_{\pi_j}^* y_l^{\pi_j} + y_0 * (\beta_1^j - b_{\pi_j}^*) + \gamma_1^j)^2$$

La résolution du problème de minimisation $Minimiser(A)$ revient à résoudre le système suivant :

$$\begin{cases} \delta(A)/\delta y_0 = 0 \\ \delta(A)/\delta b_{\pi_1}^* = 0 \\ \delta(A)/\delta b_{\pi_j}^* = 0 \end{cases} \quad (6.15)$$

Ce système engendre une équation du 6^{ième} degré en y_0 que nous résolvons en utilisant la formule de Laguerre⁶.

6. Pour réaliser les calculs nous avons utilisé l'outil Maple et pour l'implantation les fonctions du Numerical Recipies [PFTV88]

A présent, pour chaque triplet de TF_u la résolution de l'équation 6.15 permet d'une part de réajuster le couple de traces $T_u^{\Pi_1}$ et $T_u^{\Pi_j}$ et d'autre part de déterminer les coordonnées du point $P_u^{(\Pi_1, \Pi_j)} = T_u^{\Pi_1} \cap T_u^{\Pi_j}$:

$$P_u^{(\Pi_1, \Pi_j)} : \left(\frac{c_{\Pi_j}^* b_{\Pi_1}^* - c_{\Pi_1}^* b_{\Pi_j}^*}{b_{\Pi_j}^* - b_{\Pi_1}^*}, \frac{c_{\Pi_1}^* - c_{\Pi_j}^*}{b_{\Pi_j}^* - b_{\Pi_1}^*} \right) \quad (6.16)$$

Soient TF_u et TF_v deux triplets de $\mathcal{F}_{\Pi_j}^{\Pi_1}$ $j = 2, 3$ et soit $T_u^{\Pi_1}$, $T_u^{\Pi_j}$, $T_v^{\Pi_1}$ et $T_v^{\Pi_j}$ les droites de traces caractéristiques qui leur sont associés relativement à Π_1 et Π_j où ($j = 2, 3$). Nous énonçons le théorème suivant (Figure 6.7) :

Théorème 2 TF_u et TF_v de $\mathcal{F}_{\Pi_j}^{\Pi_1}$ $j = 2, 3$ sont la projection de deux facettes 3D coplanaires si et seulement si :

$$T_u^{\Pi_1} = T_v^{\Pi_1} \text{ et } T_u^{\Pi_j} = T_v^{\Pi_j}. \quad (6.17)$$

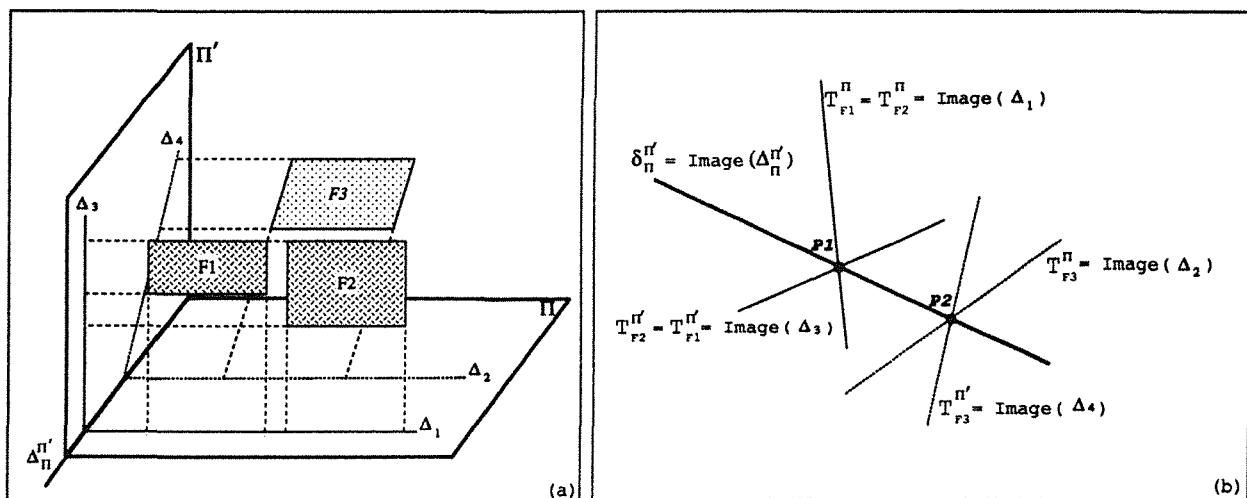


Figure 6.7. (a) La représentation 3D de trois facettes et de deux plans de références Π_1 et Π_2 . (b) La représentation de la trace de ces trois facettes sur Π_1 et Π_2 , f_1 et f_2 sont coplanaires : ils définissent le même couple de traces quant à f_3 elle définit un autre couple de traces.

La preuve de ce théorème est évidente car $\mathcal{F}_{\Pi_j}^{\Pi_1}$ représente l'ensemble des triplets de facettes 2D qui ne font pas partie du faisceau de plans défini par les deux plans de références Π_1 et Π_2 (cf. la Contrainte 3 et la Remarque 9). Une propriété évidente découle de ce théorème.

Propriété 9 Deux triplets facettes TF_u et TF_v sont la projection de deux facettes 3D coplanaires, alors ils doivent nécessairement vérifier :

$$P_u^{(\Pi_1, \Pi_j)} = P_v^{(\Pi_1, \Pi_j)} \quad (6.18)$$

C'est donc relativement à ce point d'intersection que les hypothèses de facettes coplanaires vont être générées. Cependant soulevons le fait que pour prouver la coplanarité des facettes 3D associées à TF_u et TF_v , la Condition 6.18 n'est qu'une condition nécessaire et non suffisante. En effet, prenons le cas particulier de facettes 3D appartenant à un faisceau de plan contenant le plan Π_1 (Figure 6.8.a). Alors par construction, la Condition 6.18 est vérifiée alors que les facettes 3D ne sont pas coplanaires (Figure 6.8.b). Il faut donc valider cette hypothèse de coplanarité.

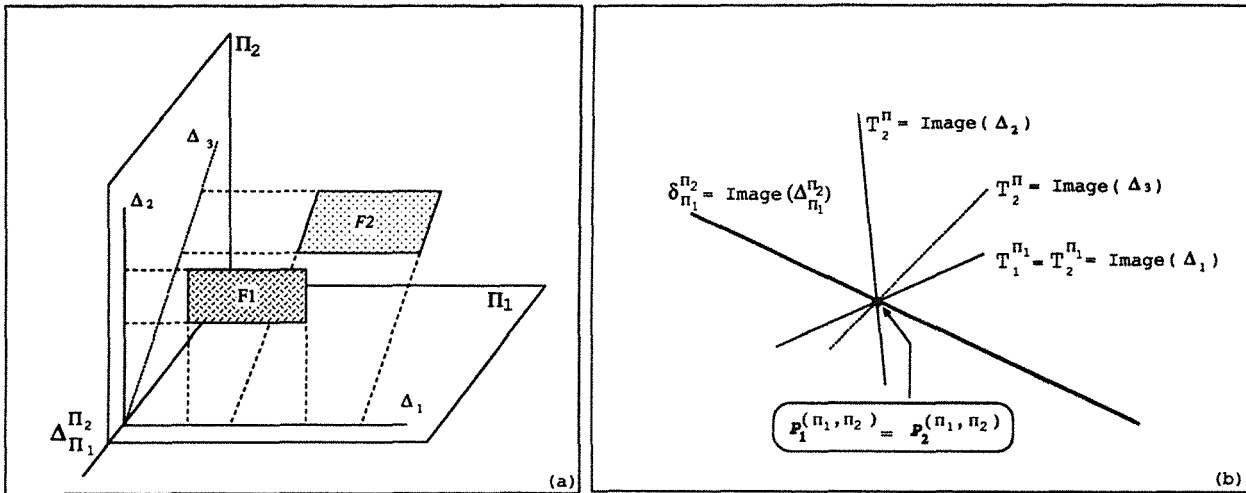


Figure 6.8. Cas de facettes 3D appartenant au même faisceau de plan que Π_1 .

B- Algorithme de recherche des classes coplanaires

Nous associons donc à chaque triplet de facettes $TF_u \in \mathcal{F}_{\Pi_j}^{\Pi_1}$, ($j = 2, 3$) les informations suivantes :

$$TF_u \longrightarrow (T_u^{\Pi_1}, T_u^{\Pi_j}, P_u^{(\Pi_1, \Pi_j)}, Cop_u),$$

où Cop_u représente l'ensemble des triplets de facettes 2D coplanaires à TF_u .

L'algorithme de recherche des facettes coplanaires met en application les résultats énoncés dans le paragraphe précédent. Ainsi, cet algorithme se décompose en deux parties : une première partie qui forment des hypothèses de coplanarité en se fondant sur la Propriété 9 et une seconde partie qui utilise le Théorème 2 pour vérifier la validité des hypothèses de coplanarité.

• Première étape. S'inspirant de la technique des baquets introduite dans la Section 3.2), la droite $\delta_{\Pi_1}^{\Pi_j}$ est décomposée en n intervalles se chevauchant. Chaque intervalle étant d'une longueur ε (Figure 6.9). A chaque intervalle I , nous associons l'ensemble des triplets TF_u dont le point $P_u^{(\Pi_1, \Pi_j)}$ appartient à I . Cette décomposition permet ainsi de créer des agglomérats de triplets de facettes associés à des points de $\delta_{\Pi_1}^{\Pi_j}$ qui sont dits être ε -voisin :

$$TF_u, TF_v \text{ font partie d'un même agglomérat ssi } \|P_u^{(\Pi_1, \Pi_j)} - P_v^{(\Pi_1, \Pi_j)}\| \leq \varepsilon.$$

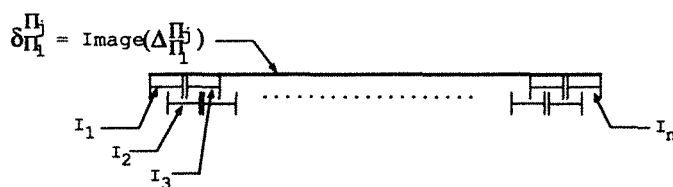


Figure 6.9. Décomposition de $\delta_{\Pi_1}^{\Pi_j}$.

Dans l'exemple illustré dans la Figure 6.7, l'application de cette première étape génère deux agglomérats: $\mathcal{A}_1 = \{TF_1, TF_2\}$ associé au point P_1 et $\mathcal{A}_2 = \{TF_3\}$ associé à P_2 .

• Seconde étape. Chaque agglomérat constitue un ensemble d'hypothèses de triplets de facettes susceptibles d'être coplanaires. Cette hypothèse sous jacente à chaque agglomérat est vérifiée par application de la Contrainte 2. En effet, en pratique et à cause du bruit, deux triplets TF_u et TF_v faisant partie d'un même agglomérat, sont supposés former un couple coplanaires si les directions de leur droites de trace relativement à un même plan de référence sont α -confondues: les droites de traces définissent un angle plus petit que α donné. Ainsi pour vérifier l'hypothèse de coplanarité entre TF_u et TF_v , nous réutilisons alors la Contrainte 3 pour vérifier la coplanarité de l'union des segments des deux facettes. Si la coplanarité est validée, une nouvelle trace est recalculée à partir de l'union de tous les segments des facettes dites coplanaires.

Initialement pour tout triplet TF_u nous posons $Cop_u = TF_u$. Pour chaque agglomérat \mathcal{A} nous procédons de la manière suivante:

Fonction Coplanarité(\mathcal{A})

Tantque $\mathcal{A} \neq \emptyset$

Prendre TF_u de \mathcal{A} ;

Pour tout TF_v de \mathcal{A} Faire

Si ($T_u^{\Pi_1}$ et $T_v^{\Pi_1}$ sont α -confondues) et ($T_u^{\Pi_j}$ et $T_v^{\Pi_j}$ sont α -confondues)

Alors /* Valider que l'union de leurs segments sont réellement coplanaires */

Si $Plan(Cop_u \cup Cop_v)$

Alors /* Unir les classes de TF_u et de TF_v */

$Cop_u = Cop_u \cup Cop_v$; $Cop_v = Cop_u$;

/* Recalculer la trace commune associées à Cop_u */

$T_u^{\Pi_1} = Trace(Cop_u)$ sur Π_1 ; $T_u^{\Pi_j} = Trace(Cop_u)$ sur Π_j ;

Pour tout TF_l de Cop_u Faire $T_l^{\Pi_1} = T_u^{\Pi_1}$; $T_l^{\Pi_j} = T_u^{\Pi_j}$; Fin Pour

Retirer TF_v de \mathcal{A} ;

Fin Pour

Fin Tantque

Fin Fonction

Une fois cette fonction appliquée à tous les agglomérats, nous obtenons les classes de triplets de facettes coplanaires. De plus, ayant pris soin de réajuster la trace en tenant compte de tous les segments supposés être coplanaires nous nous sommes assurés que l'information trace associée à chaque triplet contient l'information coplanarité.

6.6 D'autres propriétés exploitables en 2D

Toujours dans l'optique d'extraire le plus d'information possible sur les propriétés 3D en travaillant dans les plans images, nous avons développé des techniques permettant de retrouver des caractéristiques concernant les facettes 3D avant leur reconstruction 3D. Dans un premier temps nous énonçons les propriétés théoriquement puis nous introduisons leur application dans le cadre du constructeur de facettes 3D. A l'heure actuelle, ces applications n'ont pas encore été implantées dans la version actuelle du constructeur.

6.6.1 Position relative d'une primitive par rapport à un plan

Nous entendons par position relative d'une primitive par rapport à un plan de référence Π , le fait de déterminer si cette dernière est située à l'avant ou à l'arrière de ce plan, par rapport à un système de caméras donné. Il est évident que si nous voulons positionner une primitive par rapport à un plan Π il faut que ce plan présente la même face aux caméras qui composent le système stéréoscopique considéré. Ceci implique qu'il ne doit intersecter aucun des intervalle $[O_i, O_j]$ qui forme les bases du système stéréoscopique.

Comme pour les propriétés précédentes notre but est de ne pas étudier le problème en 3D, mais de le ramener à un calcul 2D. Dans la suite nous présentons trois propriétés de positionnement concernant respectivement les points, les segments et les facettes. Nous supposons que nous disposons d'un système de bases projectives en correspondance induit par la base projective associée au plan Π .

• **Position d'un point.** Soit (p_1, p_2) un couple de points images d'un même point 3D P . Nous cherchons à positionner P par rapport à Π . La propriété concernant la position relative d'un point par rapport à un plan Π s'énonce comme suit :

Propriété 10 Nous posons $p_{j \rightarrow i} = \mathcal{T}_{j \rightarrow i}^\Pi(p_j)$. Nous désignons par E_i l'épipoles de la caméra i . Trois cas sont possibles :

$$\left\{ \begin{array}{l} (p_i = p_{j \rightarrow i}) \iff \text{point 3D appartient au plan } \Pi \\ (p_{j \rightarrow i} \in [p_i, \infty[) \iff \text{point 3D situé en avant du plan } \Pi \\ (p_{j \rightarrow i} \in [E_i, p_i]) \iff \text{point 3D situé derrière le plan } \Pi \end{array} \right.$$

La Figure 6.10 illustre la Propriété 10. Les points M et N se projettent en un même point m sur le plan image 1. La position de $m2'$ et $n2'$ sur l'épipoles associé à m permet de déduire que N est au devant du plan de référence alors que M est à l'arrière de ce plan.

• **Position d'un segment.** La généralisation de ce résultat au cas d'un segment est immédiate. Soit (s_i, s_j, s_k) un triplet de segments appariés.

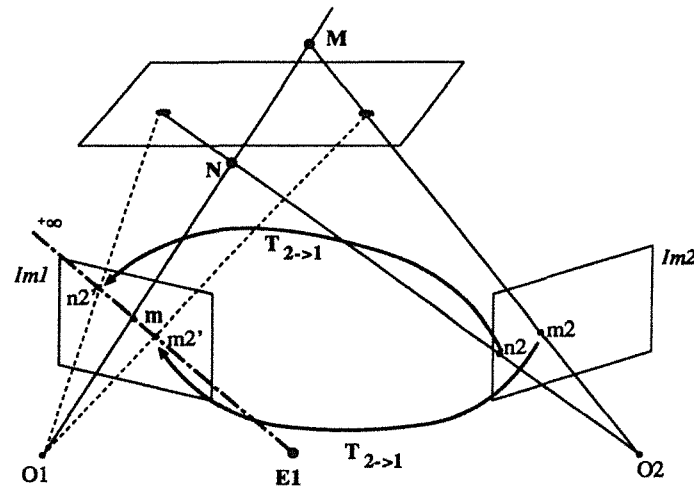


Figure 6.10. Caractérisation de la position d'un point par rapport à un plan.

Propriété 11 Nous posons $s_{j \rightarrow i} = T_{j \rightarrow i}(s_j)$ et $s_{k \rightarrow i} = T_{k \rightarrow i}(s_k)$. Alors trois cas sont possibles :

$$\begin{cases} (s_i = s_{j \rightarrow i}) \text{ et } (s_i = s_{k \rightarrow i}) \iff \text{segment appartient au plan } \Pi \\ (s_{j \rightarrow i} \cap s_i = \text{point}) \iff \text{le segment traverse le plan } \Pi \\ (s_{j \rightarrow i} \cap s_i = \emptyset) \iff \text{revient à étudier la position d'un point} \\ \quad \text{quelconque du segment par rapport au plan } \Pi \end{cases}$$

La Figure 6.11 illustre le cas de deux segments S et S' ayant la même droite support. L'étude du couple s_1, t_2 en prenant la droite épipolaire Δ_1 montre que le segment S' se situe à l'avant du plan alors que l'étude du couple s_1', t_2' en prenant la droite Δ_2 et en appliquant la Propriété 11 montre que le segment S est situé derrière le plan.

• **Position d'une facette.** Nous passons par la suite à la généralisation du résultat de positionnement au cas d'une facette prouvée plane. Soit $TF = (f_i, f_j, f_k)$ un triplet de facettes 2D homologues.

Propriété 12 $f_{j \rightarrow i} = T_{j \rightarrow i}(f_j)$ et $f_{k \rightarrow i} = T_{k \rightarrow i}(f_k)$ et nous désignons par T^Π la trace de TF par rapport au plan Π . Alors trois cas sont possibles :

$$\begin{cases} (f_i = f_{j \rightarrow i}) \text{ et } (f_i = f_{k \rightarrow i}) \iff \text{facette appartient au plan } \Pi \\ (T^\Pi \cap f_i \neq \emptyset) \iff \text{la facette intersecte le plan } \Pi \\ (T^\Pi \cap f_i = \emptyset) \iff \text{revient à étudier la position d'un point} \\ \quad \text{quelconque de la facette par rapport au plan } \Pi \end{cases}$$

6.6.2 Applications utiles de la propriété de positionnement

Nous énonçons à présent des applications directes du résultats théorique sur le positionnement d'une primitive par rapport à un plan de référence.

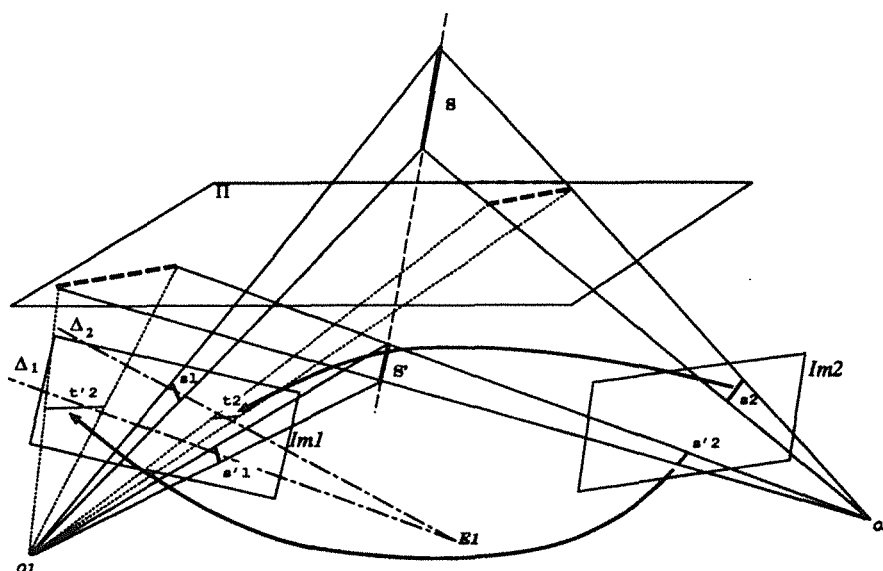


Figure 6.11. La présentation de la position d'un segment reconstruit par rapport à sa position relative au plan de référence

Nous supposons que nous disposons d'un triplet de facettes 2D homologues, $TF = (f_1, f_2, f_3)$ dont nous avons démontré la planarité. Toute l'astuce consiste à *prendre le plan support de la facette comme plan de référence*. Pour ceci il suffit de prendre quatre triplets de points homologues (généralement les sommets⁷) appartenant aux polygones 2D (f_1, f_2, f_3) , ces points forment alors un système de bases projectives en correspondance

- *Facette ne présentant pas la même face à deux caméras* : afin de savoir si le plan d'une facette F coupe le banc stéréoscopique, et de conclure que la facette ne présente pas la même face à deux caméras, il suffit de prendre un couple de points homologues (p_i, p_j) n'appartenant pas au plan de la facette Π_F (donc $T_{i \rightarrow j}^{\Pi_F}(p_i) \neq p_j$). Nous étudions alors en parallèle la position de ce couple de points par rapport au plan de la facette, dans les deux plans images en utilisant la Propriété 10. Si les conclusions sont contradictoires alors le plan de la facette coupe le banc stéréoscopique.
- *Opacité d'une facette* : une autre question peut se poser concernant l'opacité d'une facette. Supposons que dans chacune des surfaces délimitées par les polygones 2D f_1 et f_2 , deux facettes homologues, il existe un segment 2D, noté respectivement s_1 et s_2 . Supposons d'autre part que (s_1, s_2) forment un couple de segments homologues. Le problème que nous nous posons est de savoir si le segment 3D associé à ce couple de segments se situe à l'arrière ou au devant du plan défini

7. Comme il nous faut désigner 4 sommets pour former une base projective sur chaque facettes, pour les facettes représentées par un triangle nous utilisons la méthode décrite dans § 5.7.3 pour retrouver le quatrième point nécessaire.

par la facette sans en venir à la construction 3D ni de la facette ni du segment en question. Il suffit de prendre le plan de la facette pour plan de référence et d'étudier la position du segment en utilisant la Propriété 11.

- *Position d'une facette par rapport à une autre* : ayant deux couples de facettes appariées planes il est facile en prenant l'une d'entre elle comme plan de référence de positionner l'autre facette par rapport à la première en utilisant la Propriété 12.

6.7 Reconstruction de facettes 3D

Signalons tout d'abord que cette étape de reconstruction ne concerne que l'ensemble des triplets de facettes faisant partie de l'ensemble \mathcal{F}_{plan} .

La reconstruction de facettes repose sur le principe de triangulation et utilise le fait que la donnée de deux droites 3D distinctes d'un plan détermine parfaitement ce dernier. A ce stade du traitement, nous avons associé à chaque triplet TF_u de facettes 2D homologues planes un couple de traces caractéristiques sur deux plans de références connus. Remarquons par ailleurs que l'étape de prétraitement, qui précède l'étape finale de reconstruction 3D, a réajusté les traces de sorte que les droites de traces d'une même classe de facettes coplanaires soient exactement les mêmes. Ainsi, la reconstruction qui repose sur cette information traces va permettre de maintenir cette propriété de coplanarité en 3D. Présentons à présent le processus de construction de facettes 3D qui se décomposent en deux étapes :

étape de construction du plan support d'une facette : étant donné un triplet de facettes TF_u image d'une facette F_u , notre première tâche consiste à définir à partir de ses traces le plan support noté $\Pi(TF_u)$, de la facette F_u ,

étape de recherche des bords d'une facette : une fois le plan $\Pi(TF_u)$ déterminé, nous procédons à la recherche des limites de la facette F_u à partir des segments 2D qui délimitent les facettes 2D qui composent le triplet TF_u .

6.7.1 Etape de recherche du plan support d'une facette

Comme nous l'avons vu dans la section 6.5 l'ensemble \mathcal{F}_{plan} , dont nous voulons reconstruire les facettes 3D correspondantes, est décomposé en cinq sous-ensembles (Section 6.5) :

$$\mathcal{F}_{plan} = \mathcal{F}_{coplan}^{\Pi_1} \cup \mathcal{F}_{coplan}^{\Pi_2} \cup \mathcal{F}_{coplan}^{\Pi_3} \cup \mathcal{F}_{\Pi_1}^{\Pi_2} \cup \mathcal{F}_{\Pi_1}^{\Pi_3}$$

Le processus de construction du plan support d'une facette F_u dépend du sous-ensemble correspondant au d'images homologues TF_u associé à F_u .

A- Cas des facettes de $\mathcal{F}_{coplan}^{\Pi_i}$

Les triplets de facettes homologues appartenant au sous-ensemble $\mathcal{F}_{coplan}^{\Pi_i}$, $i = 1..3$ sont des triplets dont le correspondant 3D est prouvée être coplanaire au plan de référence Π_i aussi le plan $\Pi(TF_u)$ est complètement définie : $\Pi(TF_u) = \Pi_i$.

B- Cas des facettes de $\mathcal{F}_{\Pi_1}^{\Pi_i}$

A tout triplet de facettes TF_u faisant partie de l'ensemble $\mathcal{F}_{\Pi_1}^{\Pi_i}$ est associé le couple de traces suivant : $T_u^{\Pi_1}$ et $T_u^{\Pi_i}$. Chacune de ces traces définit une droite 2D dans le plan image. En calculant la droite intersection du plan d'interprétation de la droite $T_u^{\Pi_j}$ avec le plan de référence Π_j pour $j = 1, i$, nous obtenons deux droites 3D (Figure 6.12). Ces droites sont également les droites d'intersection du plan support de la facette avec les deux plans de références considérés. Ayant ainsi obtenu deux droites du plan support il est facile de déterminer les paramètres du plan support $\Pi(TF_u)$.

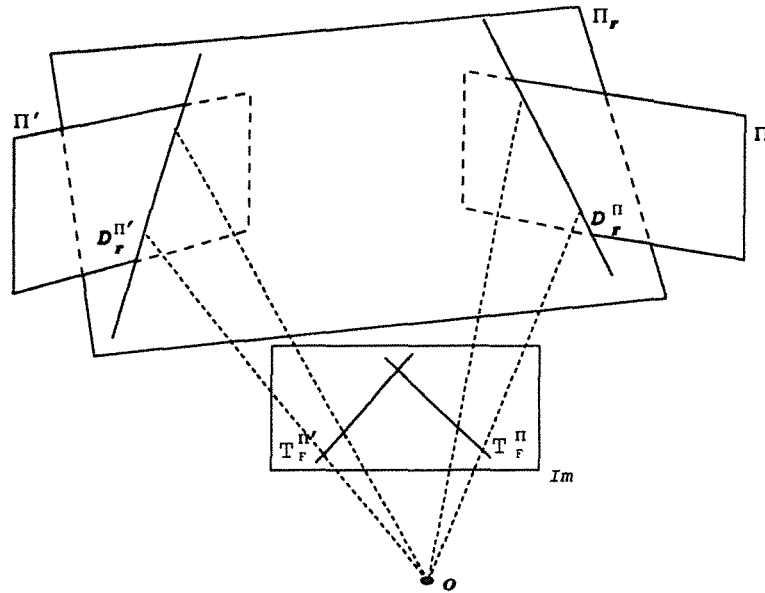


Figure 6.12. Construction de deux droites du plans support d'une facette F à partir de ses deux traces T_F^{Π} et $T_F^{\Pi'}$.

6.7.2 Etape de recherche des bords d'une facette

Après avoir déterminé pour chaque facette (et même de groupes de facettes en cas de coplanarité) l'équation de son plan support nous procédons à la recherche des segments qui la délimitent. La méthode proposée est simple. Elle consiste premièrement à définir par triangulation, sur le plan $\Pi(TF_u)$, le triplet de segments 3D (S_1, S_2, S_3) correspondant à chaque triplet de segments homologues (s_1, s_2, s_3) appartenant à la facette à construire. Par la suite et avec un calcul au moindre carré, nous déterminons pour chaque triplet (S_1, S_2, S_3) le segment 3D S correspondant. Ce segment S fait partie des segments limitant la facette 3D.

6.8 Résultats expérimentaux

Avant de passer à la présentation de quelques résultats, nous allons tout d'abord, dans la Section 6.8.1, parler d'un problème révélé par l'expérimentation et qui est celui

de la sensibilité de notre méthode aux erreurs de certains segments, dépendante de l'orientation du plan de référence choisi.

L'organisation des sections suivantes permet d'illustrer dans un premier temps le processus de construction des traces puis celui de la construction des facettes coplanaires. Dans ce dernier cas, un résultat sur un triplet d'images de synthèses permet de vérifier la conformité de notre module aux spécifications théoriques énoncées dans le chapitre courant ainsi que dans le chapitre précédent. Par la suite, nous donnons une application à des images réelles.

6.8.1 Trace de segments parallèles au plan de référence

En pratique, et malgré la prise en compte de l'erreur, nous avons pu constater, une certaine sensibilité des résultats de la trace, dépendante de l'orientation du plan de référence choisi. Il faut donc expliquer les raisons de cette sensibilité. L'étude expérimentale, que nous avons menée avec des images de synthèses bruitées, a fait apparaître clairement que la présence d'un segment parallèle au plan de référence, parmi l'ensemble des segments composant la facette traitée, pouvait perturber le résultat du test de planarité au point de le mettre en défaut. L'explication de ce phénomène est très simple.

D'un point de vue théorique, le calcul de la trace d'un segment parallèle au plan de référence ne diffère pas de celui d'un segment quelconque. Cependant, le point de trace d'un segment S , parallèle au plan de référence Π , est alors un point de la droite à l'infini D_{∞}^{Π} , du plan Π relativement au plan image Im_1 , et indique donc une direction. Aussi, lors de la construction de la trace d'une facette F comprenant un segment S parallèle au plan Π , le point de trace associé à S représente également le point de fuite de la droite de trace associée à la facette F . Ainsi donc, l'erreur de localisation des segments 2D images du segments S , se traduit, au niveau de la trace de la facette, par une erreur de localisation sur le point de fuite (et par conséquent sur la direction) de la droite de trace associée à la facette étudiée. Cette remarque explique l'importance de l'impact de l'erreur associée aux segments parallèles au plan de référence, sur l'information trace. Nous devons donc accorder à ces segments une place particulière.

Nous suggérons donc de traiter le cas des segments parallèles à part. Aussi, au lieu de la notion de zone de présence du point de trace, nous associons à un segment S parallèle au plan Π une notion de *zone de direction possible* qui est en fait : un intervalle sur la droite à l'infini du plan Π qui dépend de l'erreur associée aux images du segment S . Dans ces conditions, le test de planarité d'une facette par rapport à un plan de référence revient alors à vérifier l'une des deux conditions suivantes :

1. soit que tous les segments de la facette sont parallèles au plan de référence,
2. soit que d'une part tous les segments décrétés parallèles définissent des zones de directions qui s'intersectent⁸ : $[I]_{\infty}$ et que d'autre part la droite de trace de la facette (issue du calcul mené sur le reste des segments de la facette) admet un point de fuite situé dans $[I]_{\infty}$.

8. Il est évident que si une facette plane, non parallèle au plan de référence Π , admet des segments parallèles à Π alors ces segments sont parallèles entre eux.

Il reste alors à définir un moyen qui permettent de détecter les triplets de segments homologues images de segments parallèles (ou presque) à un plan de référence donné. Soit (s_1, s_2, s_3) la projection d'un segment 3D parallèle au plan de référence Π . Dans le cas parfait, la condition de parallélisme se traduit par la condition suivante :

$$l_1 \cap D_\infty^\Pi = \mathcal{T}_{2 \rightarrow 1}(l_2) \cap D_\infty^\Pi = \mathcal{T}_{3 \rightarrow 1}(l_3) \cap D_\infty^\Pi \quad (6.19)$$

Mais il est évident qu'en pratique et en raison de l'erreur sur les primitives 2D, cette triple égalité n'est plus vérifiée dans la majorité des cas. Aussi, pour détecter les cas de parallélisme, nous allons revenir à l'erreur sur les primitives 2D et nous allons émettre l'hypothèse (souvent admise) que l'erreur de localisation de chacune des extrémités d'un segment est de l'ordre d'un pixel. De ce fait, nous associons à l'image s_1 d'un segment S parallèle au plan de référence Π , un intervalle de directions, noté $[s_1]_\infty^\Pi$, sur la droite à l'infini D_∞^Π du plan Π . Cet intervalle représente les directions que peut présenter en fonction de l'erreur, le segment s_1 : l'image de S (Figure 6.13).

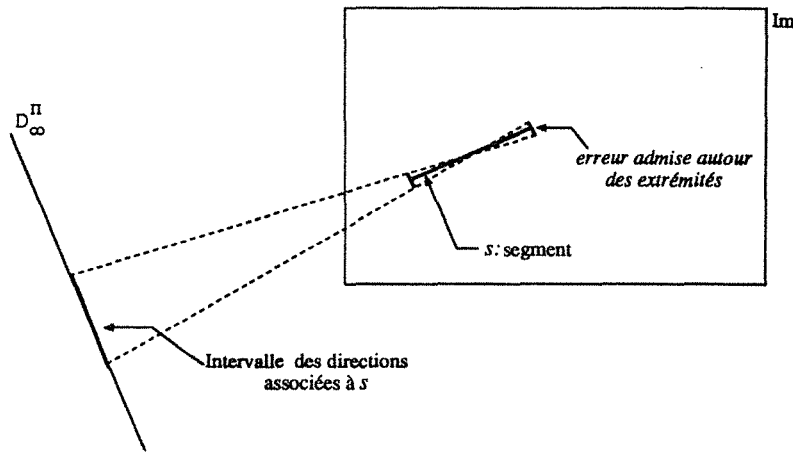


Figure 6.13. Erreur sur la direction de l'image d'un segment S , parallèle au plan Π .

De ce fait la condition théorique représentée par l'équation 6.19 est remplacée par une équation dérivée qui tient compte de l'hypothèse de l'erreur. Ainsi, soit (l_1, l_2, l_3) un triplet de droites homologues, pour que ce triplet soit considéré comme pouvant être l'image d'une droite parallèle (au sens de l'erreur admise) au plan de référence Π il faut qu'il vérifie l'équation suivante :

$$[l_1]_\infty^\Pi \cap \mathcal{T}_{2 \rightarrow 1}([l_2]_\infty^\Pi) \cap \mathcal{T}_{3 \rightarrow 1}([l_3]_\infty^\Pi) \neq \emptyset \quad (6.20)$$

6.8.2 Résultats du processus du calcul de la trace

Rappelons que la contrainte de planarité (Contrainte 3) se partage en deux parties :

- (i) la facette est coplanaire à un plan de référence : il s'agit de vérifier la coïncidence entre les segments homologues de la facette et les segments reconstruits par $\mathcal{T}_{2 \rightarrow 1}$ et $\mathcal{T}_{3 \rightarrow 1}$,
- (ii) la facette n'est pas coplanaire à un des plans de référence : il s'agit alors de vérifier l'alignement des traces associées au triplet de facettes 2D.

A- Application de la première partie de la Contrainte 3

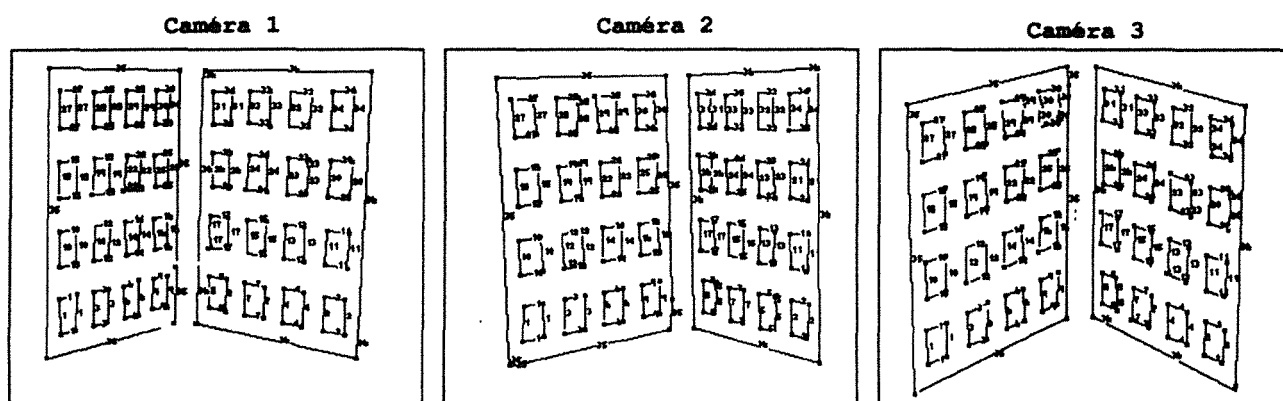


Figure 6.14. Triplets de facettes 2D homologues.

Dans la Figure 6.15, nous avons repris l'exemple des facettes de la mire (Figure 6.14) et nous avons choisis comme plan de référence le plan de l'un des pans de la mire. Les quatre points de la base sont les coins de la grande facette portant le label 35. Afin de ne pas charger l'image nous avons juste représenté l'application $\mathcal{T}_{2 \rightarrow 1}$ sur les segments composant les différents triplets de facettes.

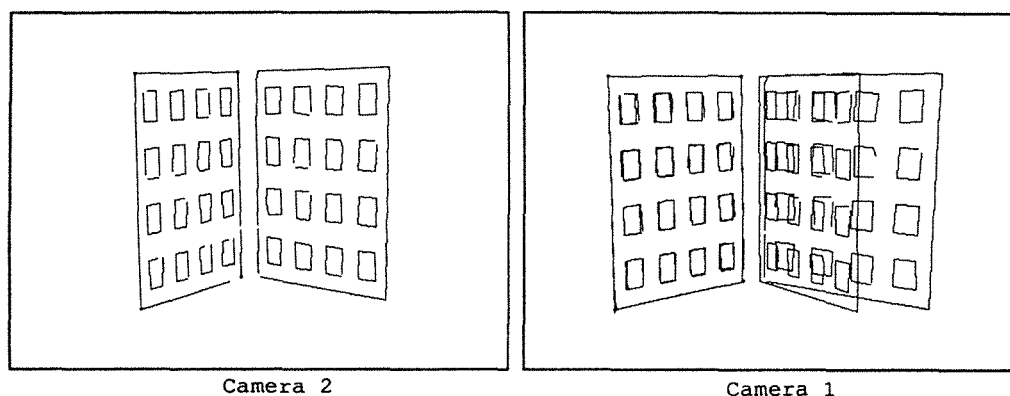


Figure 6.15. Trace de facettes coplanaires au plan de référence.

Nous remarquons que les facettes labellées $\{1,3,6,9,10,12,14,16,19,22, 25,27,28,29,30\}$ sont plans et coplanaires au plan de référence puisque tout triplet de segments (s_1^i, s_2^i, s_3^i) d'un de ces triplets de facettes vérifie que $\mathcal{T}_{2 \rightarrow 1}^\Pi(s_2^i) = s_1^i$ et $\mathcal{T}_{3 \rightarrow 1}^\Pi(s_3^i) = s_1^i$ et ainsi nous répondons à la première partie de la Contrainte 3.

Quant aux triplets de facettes labellés $\{31,33,32,34,26,24,23,21,17,15,13,11,8,7,4,2\}$, nous pouvons vérifier que pour leur couples de segments (s_1^i, s_2^i) nous avons $\mathcal{T}_{2 \rightarrow 1}^\Pi(s_2^i) \neq s_1^i$ nous en déduisons que leur planarité doit être étudiée en utilisant la deuxième partie de la contrainte 3.

B- Application de la deuxième partie de la Contrainte 3

Dans la Figure 6.16 nous avons choisis des plans de référence qui ne soient plus coplanaires au plan de la facette 35. Nous avons illustré le couple de droites qui approximent au mieux les points de traces issus de cette facette et qui valident l'hypothèse de planarité de cette facette.

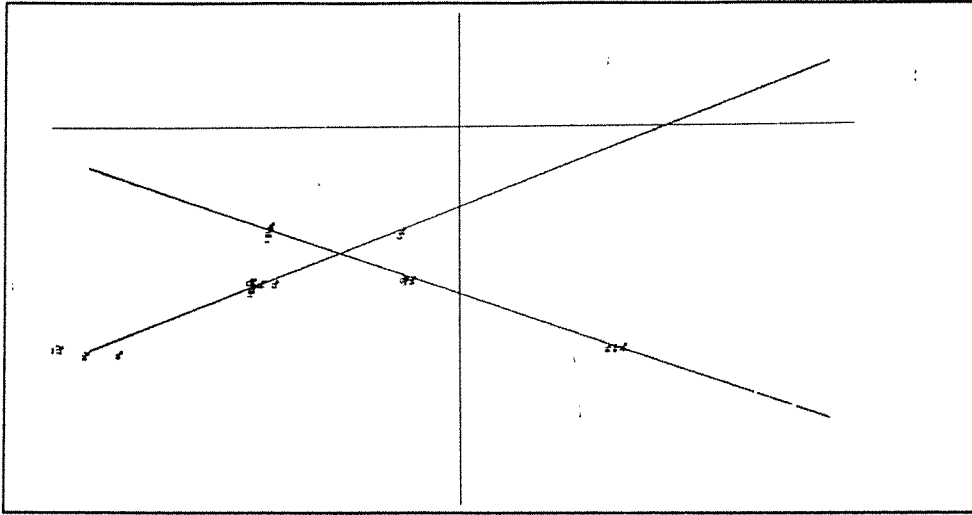


Figure 6.16. Couple de traces caractéristiques d'une facette plane.

C- Trace d'une facette non plane

La Figure 6.17.b illustre les points de trace non alignés d'une facette non plane (dont un couple d'image est donnée dans la Figure 6.17.a) sur un plan de référence quelconque.

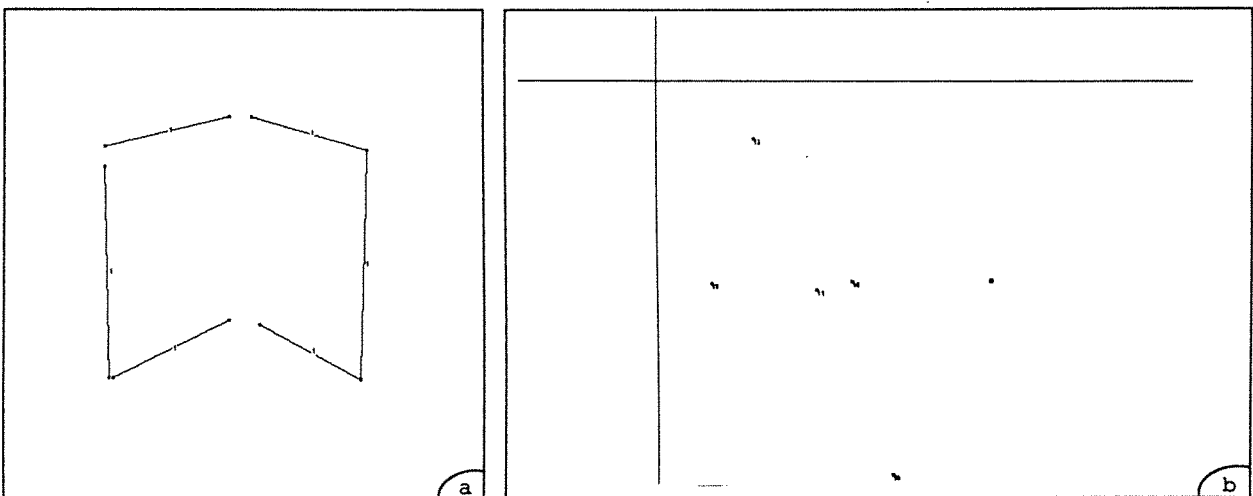


Figure 6.17. Points de traces d'une facette non plane.

6.8.3 Résultats de reconstruction 3D

A- Résultat sur une image de synthèse

Afin de vérifier la conformité de notre module de prétraitement aux résultats théoriques concernant la coplanarité et la reconstruction, nous avons testé notre module de stéréovision sur des images de synthèses : pas d'erreur de localisation sur les données. Les résultats expérimentaux coïncident avec les résultats théoriques puisque la reconstruction de la scène à partir des images de facettes 2D coïncident parfaitement avec le modèle initial.

La Figure 6.18 représente le modèle 3D de la scène utilisé, l'emplacement des centres optiques des caméras du banc stéréoscopique ainsi que les plans de références utilisés.

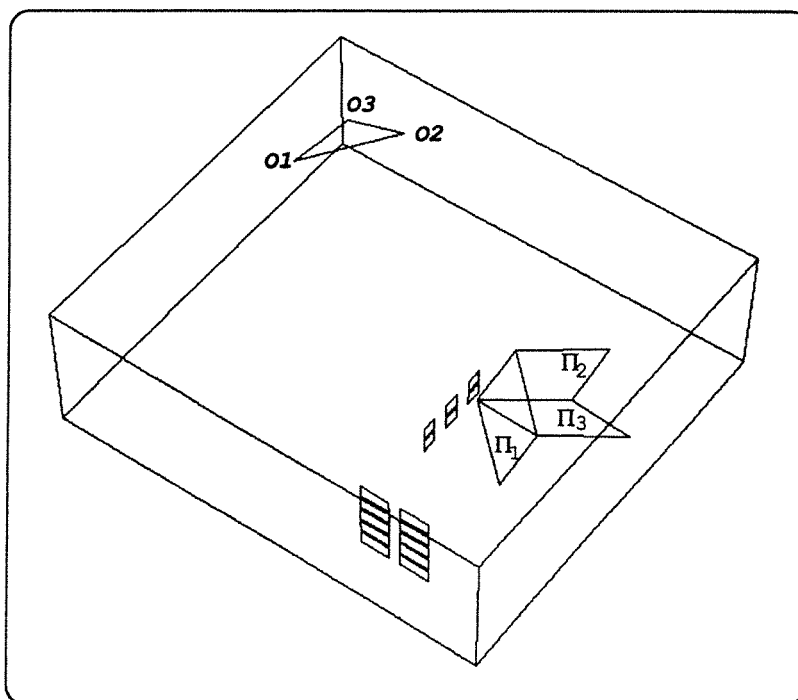


Figure 6.18. La scène de synthèse traitée.

La Figure 6.19 représente l'ensemble des facettes 2D délivrées en entrée au module de stéréovision. Une phase d'appariement est activée avant de procéder au calcul de la trace.

La Figure 6.20 montre l'ensemble des traces construites pour toutes les facettes de la scène relativement aux deux plans de références (Π_1, Π_2). Remarquons qu'étant dans le cadre parfait : pas d'erreur, nous ne voyons que quatre droites de traces représentant en fait deux couples de traces caractéristiques relative chacune à une classe de facettes coplanaires. Remarquons également la parfaite intersection de chacun des couples sur la droite d'intersection des deux plans de références $\delta_{\Pi_1}^{\Pi_2}$.

La Figure 6.21 représente le résultat de la dernière étape du constructeur de facettes 3D. Ces résultats, partant de données parfaites : la Figure 6.21 est la superposition du modèle initial avec les facettes reconstruites : aucun décalage n'est visible.

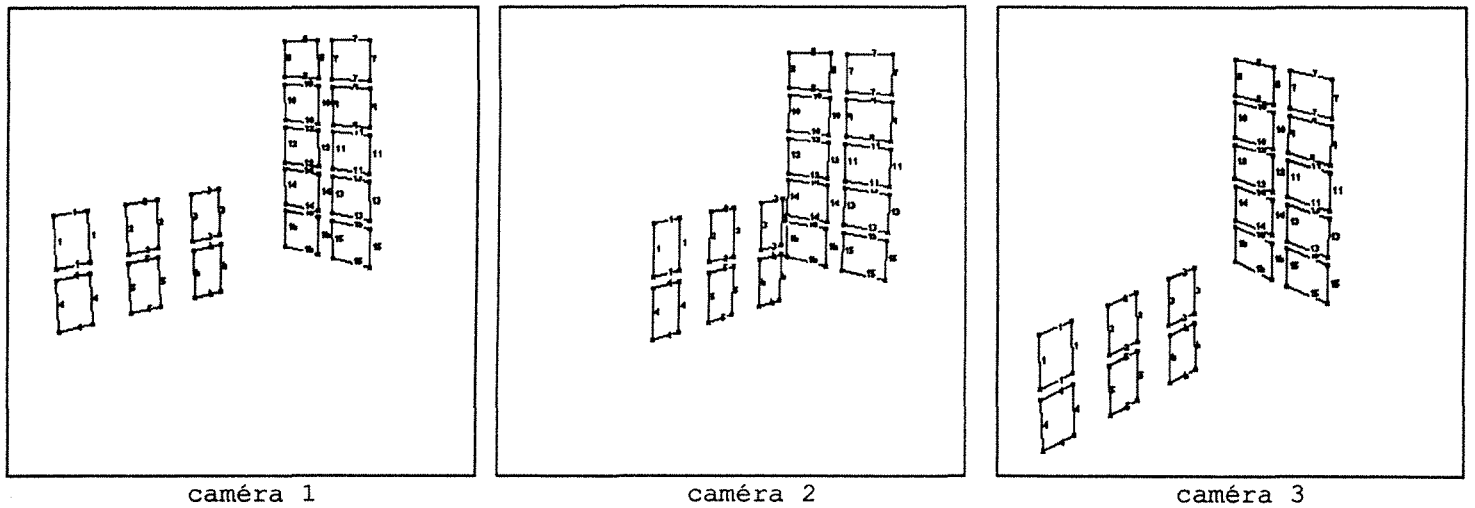


Figure 6.19. Les triplets d'images de synthèses.

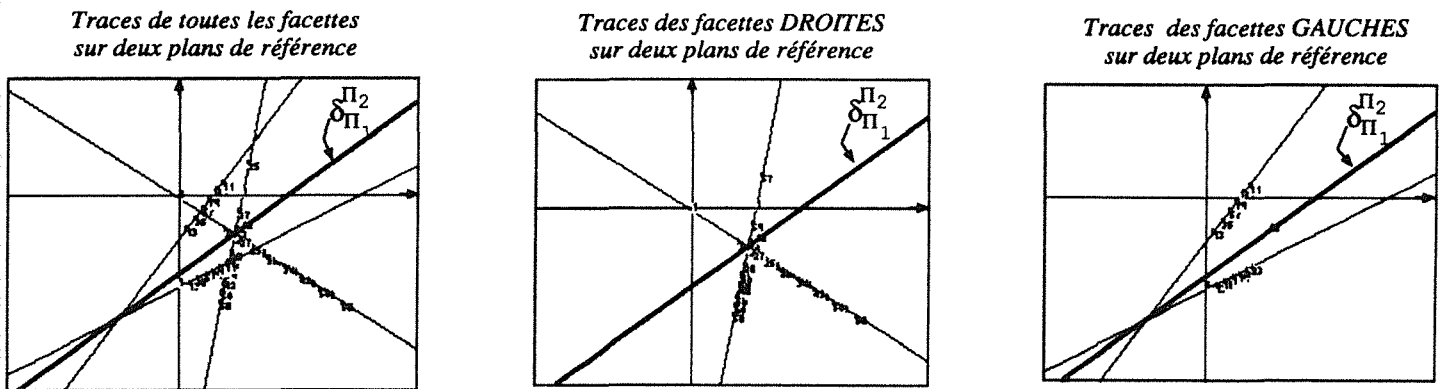


Figure 6.20. Trace calculée sur l'image de synthèse.

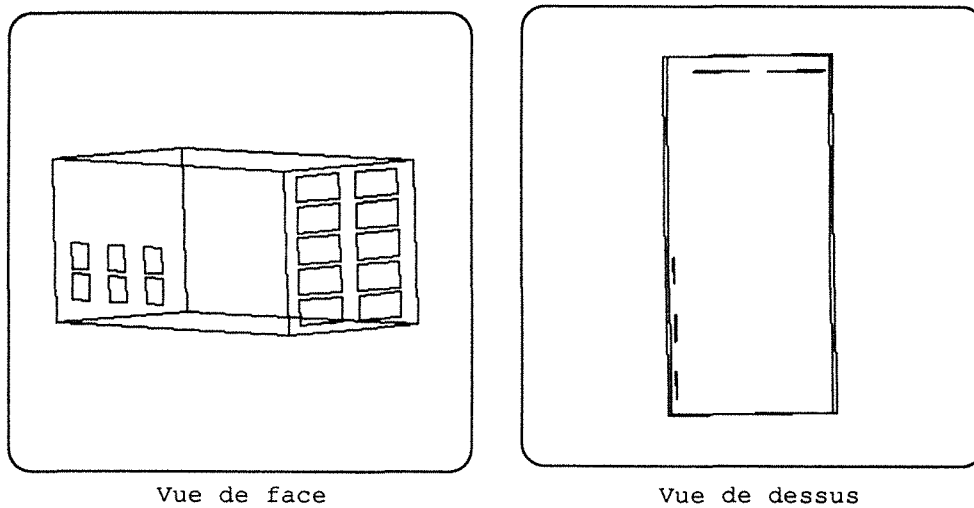


Figure 6.21. Résultats de la reconstruction des facettes de l'image de synthèse.

B- Résultats sur des images réelles

Nous passons à présent à la présentation des résultats de notre module appliqué à des triplets d'images réelles.

La Figure 6.22 représente l'ensemble de facettes 2D délivrée en entrée au module de stéréovision. Une phase d'appariement est activée avant de procéder au calcul de la trace.

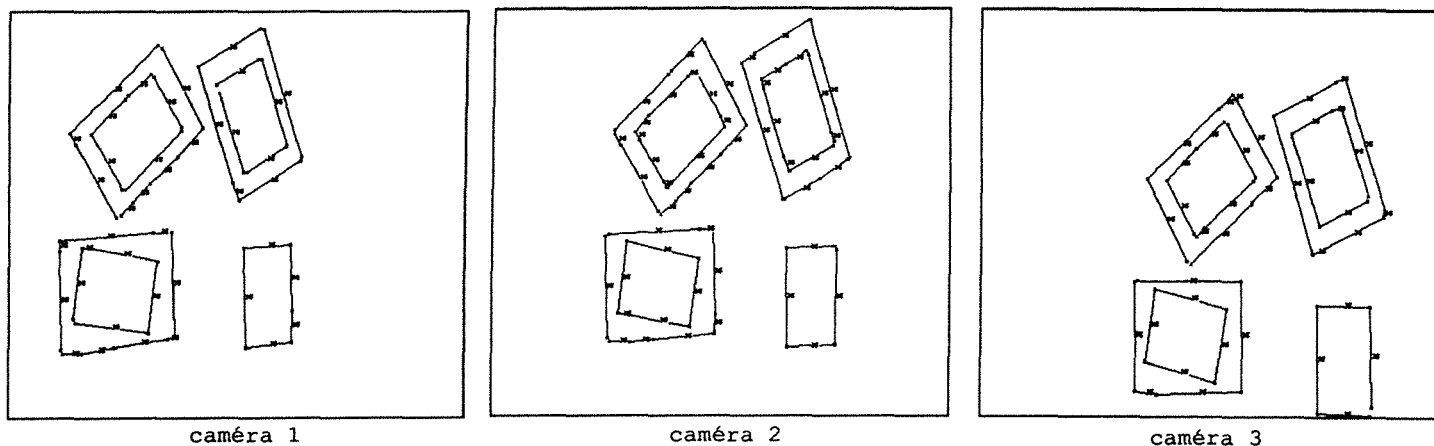


Figure 6.22. Triplet de facettes homologues : images réelles.

La Figure 6.23 représente l'ensemble des traces construites sur toutes les facettes de la scène relativement à deux plans de références caractéristiques. Remarquons que ces facettes sont en réalité toutes coplanaires et devraient donc présenter un couple de traces caractéristiques unique. Mais, à cause de l'erreur, cette unicité n'est pas vérifiée ; se dégage alors la notion de traces α -confondues.

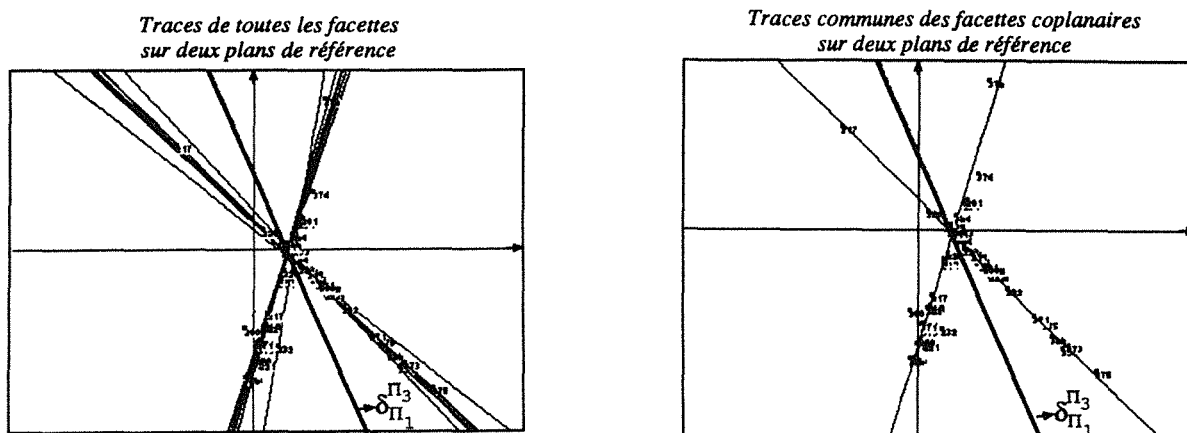


Figure 6.23. Traces des facettes de l'image réelle.

Après avoir construit la trace commune de cet ensemble de facettes, nous pouvons

procéder à la reconstruction 3D sous contrainte de coplanarité qui est comprise dans l'information trace associée à l'ensemble des facettes.

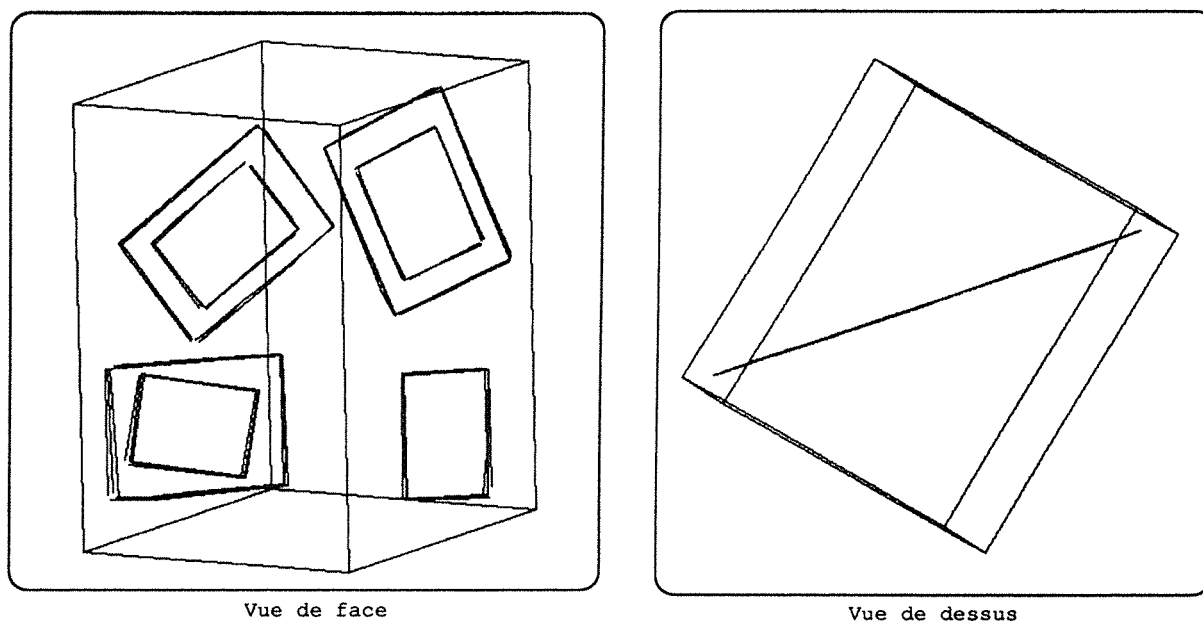


Figure 6.24. *Résultat de la reconstruction des facettes 3D.*

La Figure 6.24 représente le résultat de la dernière étape du constructeur de facettes 3D. Nous avons choisi ici de ne pas présenter le résultat final de la construction mais une étape intermédiaire qui permet de visualiser l'erreur que nous commettons sur la reconstruction. En effet, la Figure 6.24 visualise le plan support des facettes avec la rétroprojection des segments 2D de chaque triplet de segments homologues. Nous pouvons ainsi voir que chaque triplet décrit, sur le plan support de la facette des segments 3D assez proches. C'est le calcul au moindre carré autour de chacun de ces triplets de segments 3D qui permet de construire les limites des facettes.

Afin de visualiser notre apport au niveau de la reconstruction nous avons présenté dans le couple de figure Figure 6.25 et Figure 6.26 d'une part le résultat de l'appariement des segments qui délivre bien entendu un ensemble de triplet de segments plus nombreux que notre module et d'autre part les résultats de la reconstruction 3D des segments par l'algorithme de Ayache-lustman⁹. Nous pouvons voir que la coplanarité des segments n'est pas maintenue par cette reconstruction non contrainte. L'information trace permet quant à elle de détecter cette coplanarité en 2D et ainsi de la maintenir en utilisant une reconstruction sous contrainte.

9. Ces résultats nous ont été donnés avec les triplets des images par le laboratoire de Sophia Antipolis.

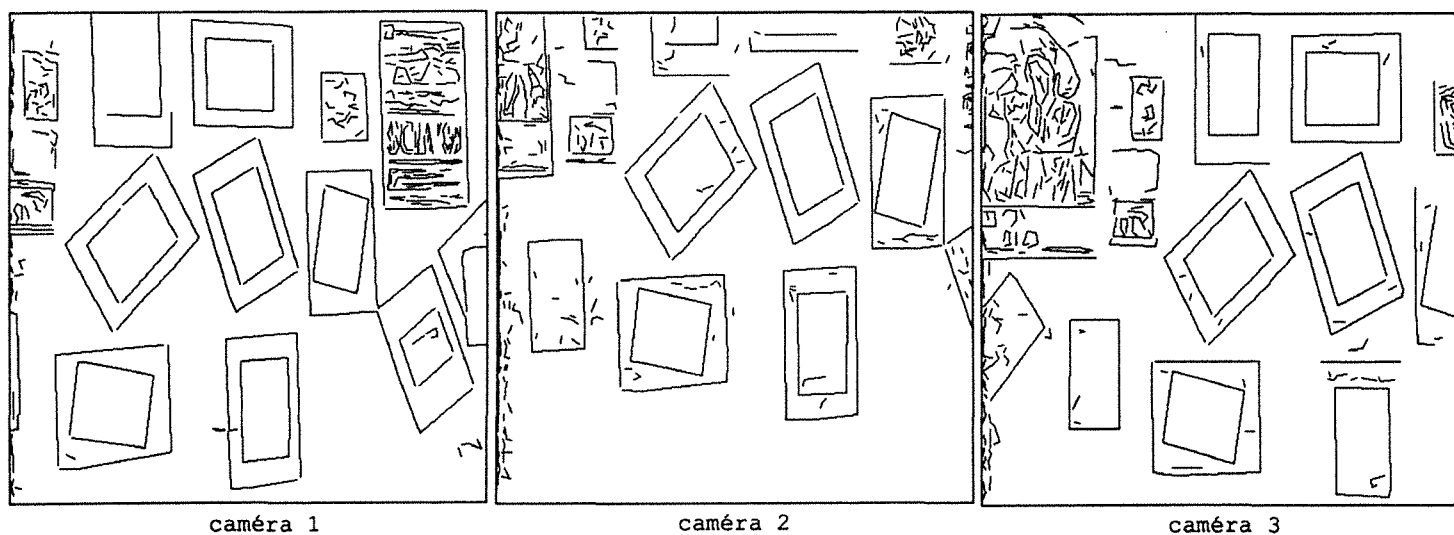


Figure 6.25. Triplet d'images des segments.

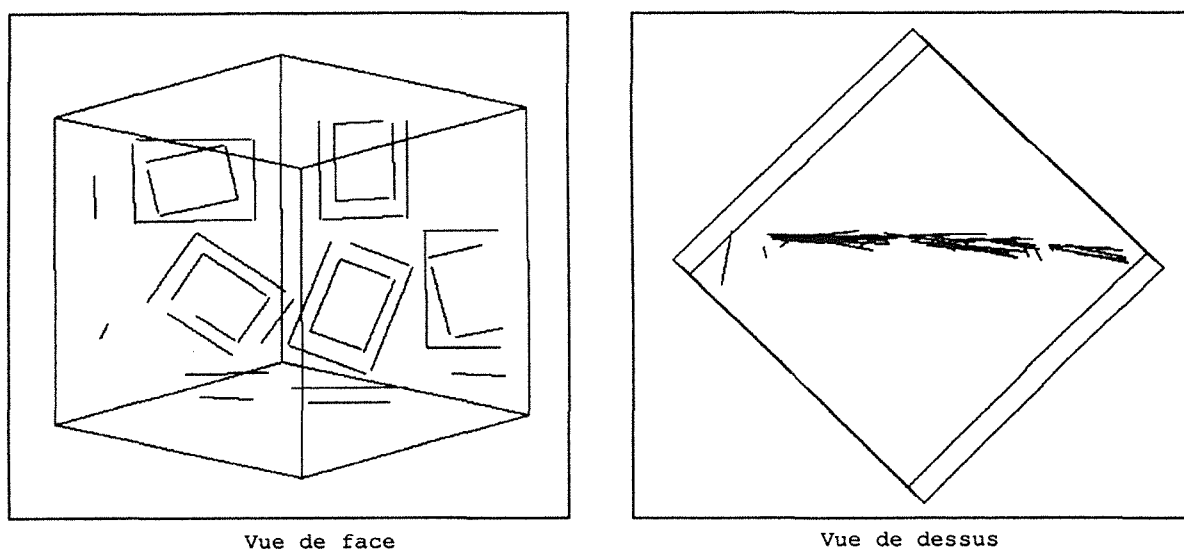


Figure 6.26. Résultat de la reconstruction des segments 3D.

6.9 Discussion

Dans ce chapitre nous avons présenté la dernière partie implantée dans notre système et qui repose essentiellement sur les résultats théoriques du chapitre précédent. Et il importe ici de relever la distance qui est apparue entre la théorie et sa mise en application. En effet, nous avons souligné dans le Chapitre 5 que la méthode, fondée sur les traces, peut être menée en dehors de toute connaissance *a priori* de données de calibrage, cependant nous avons vu en pratique, et ce à cause du bruit, que la connaissance de ces dernières nous est très utile puisqu'elle nous a permis :

1. d'établir un moyen souple pour construire les systèmes de bases projectives en correspondance nécessaires, qui nous permet d'émettre l'hypothèse de bases projectives parfaites,
2. de proposer, grâce à la connaissance de la droite à l'infini de ces plans de référence, une méthode de gestion de l'erreur. Remarquons ici que pour ce qui est des erreurs nous n'avons fait que transposer les problèmes des erreurs 3D de reconstruction par triangulation dans un espace 2D facilitant leur prise en compte.

Remarquons que le test de coplanarité tel que nous l'avons introduit a pour but de contraindre le constructeur à maintenir la coplanarité des facettes 3D. Mais, nous pouvons imaginer une autre méthode de vérification de la coplanarité destinée à servir comme outil de vérification rapide et non plus comme un prétraitement à la phase de reconstruction 3D. Cette méthode consiste à garder la première étape de l'algorithme, qui permet de formuler les hypothèses de coplanarité et à modifier la seconde de sorte à désigner un des triplets d'un agglomérat de facettes supposées coplanaires comme étant le plan de référence. Par la suite, la coplanarité des facettes se fait alors par le biais de la première partie de la contrainte 3. Ceci évite le calcul de la trace des triplets entre eux et permet de vérifier rapidement l'hypothèse de coplanarité.

Bien que les premiers résultats obtenus sur des images polyédriques répondent à notre objectif initial, il n'en reste pas moins que cette méthode est très sensible aux erreurs de localisation des primitives 2D et que cette sensibilité est d'autant plus importante que la direction d'un des segments étudiés est parallèle au plan de référence. Et à la réflexion, nous estimons que la donnée d'un seul "jeu" de trois plans de référence privilégie l'étude de certaines facettes par rapport à d'autres. Aussi nous paraît-il nécessaire de faire évoluer notre système en le munissant d'un "jeu" de plans de références plus important¹⁰ de sorte à pouvoir étudier chaque facette à l'aide de deux plans de références dont aucun des deux n'est une direction qui soit parallèle à un des segments de la facette. Le choix des plans adéquats se fait alors en utilisant le test décrit par l'équation de la Section 6.8.1 : Equation 6.20. Remarquons que dans cette variante de notre approche le test de coplanarité ne change pas du tout. En effet, la connaissance d'un couple de trace caractéristique d'une facette sur deux plans de références Π_1 et Π_2 permet de construire directement la trace de cette facette sur tout autre plan de référence Π dont on connaît la droite d'intersection avec les deux premiers plans, Π_1 et Π_2 utilisés.

Il faut encore ajouter que la méthode que nous avons développée, et qui repose essentiellement sur la contrainte de coplanarité des segments, est inadéquate pour les scènes presque polyédrique. En effet, comme les limites des objets comportent des courbures légères, les segments 2D résultant de l'approximation polygonale des chaînes de contours, résultat de la projection de ces limites, sont différents d'une image à l'autre. Cette différence de segmentation entraînent sur les triplets de segments homologues, des erreurs de plus grande amplitude. Ces erreurs perturbent alors le test de planarité. La solution pour traiter ces types de scènes se trouve en partie dans l'utilisation d'une méthode fondée sur la mise en correspondance de points puis de la contrainte de coplanarité de

10. Partitionner l'espace 3D avec plusieurs plans.

points. Cela dit cette solution ne résoud pas le problème de l'erreur mais évite l'étape d'approximation polygonale qui à notre avis ne convient pas du tout à ce type de scène.

Que pouvons nous dire maintenant à propos d'une technique de construction des segments 3D qui reposerait sur l'information trace? Cette reconstruction bien que possible, est à notre avis une démarche qui ne répond pas à notre objectif d'amélioration de la localisation des primitives 3D. En effet, la technique fondée sur les traces perd toute son efficacité dans le cas de segments individuels, puisque n'utilisant plus la contrainte de coplanarité nous ne disposons plus d'informations cumulatives qui permet de renforcer les informations des primitives les unes par rapport aux autres et diminuent ainsi l'effet de l'erreur.

7

Conclusion

Le travail présenté dans ce mémoire se situe dans le niveau intermédiaire du système de vision artificielle développé dans notre laboratoire et qui est fondé sur l'utilisation d'un banc stéréoscopique trinoculaire. Comme dans un premier temps, les scènes traitées sont supposées être essentiellement composées d'objets polyédriques, nous avons considéré que la description des objets peut être faite à partir d'assemblage des primitives tridimensionnelles : *facettes* et *segments*. Notre travail a alors eu pour objectif de développer un processus de stéréovision pour la reconstruction des facettes.

Après avoir présenté en détail l'approche que nous avons développée pour réaliser notre objectif, il convient à présent de faire le bilan de ce travail tant sur le plan des réalisations que sur le plan des perspectives. Nous avons préféré réserver à chacun de ces points une section. Ainsi, la section 7.1 présente le bilan général sur notre système, alors que la section 7.2 dresse une liste, non exhaustive, des perspectives que nous envisageons de mener dans la continuité de notre idée d'utiliser la géométrie projective pour dégager en 2D des propriétés 3D qui permettent de mieux contraindre la reconstruction. Une conclusion générale termine ce mémoire.

7.1 Bilan

Rappelons que le sens de notre recherche a consisté à retarder le plus tard possible la phase de reconstruction 3D afin d'en améliorer les résultats. Pour ce faire, nous avons dégagé une méthode qui consiste à réaliser dans un premier temps un regroupement des segments 2D d'une image en *facettes 2D*, puis à établir diverses propriétés 3D, telles la planarité ou la coplanarité sur les triplets de facettes 2D appariées. Ces propriétés établies en 2D, sont enfin utilisées pour contraindre le processus de construction des facettes à les maintenir en 3D.

Présentons dans un premier temps le bilan sur le constructeur de facettes 2D dont la philosophie est d'utiliser les informations éparses délivrées par les outils du bas niveau pour construire des primitives plus complètes. Dans cette optique, nous avons alors choisie d'établir et de mettre en œuvre un protocole d'utilisation d'informations théoriquement complémentaires : *les régions* et *les contours*. C'est ainsi que nous avons proposé d'une part un ensemble de règles décrivant un protocole d'utilisation d'une carte régions et d'une carte segments pour construire les facettes 2D d'une image ;

d'autre part nous avons défini une méthodologie d'utilisation des cartes régions qui consiste à utiliser itérativement un ensemble de cartes régions, dans une approche multiseuil, afin de parer aux imperfections des résultats des outils de segmentation en régions. Dans le cadre de la mise en application de cette méthodologie nous avons proposé une approche qui permet de gérer la redemande de cartes régions suivant les besoins du constructeur.

Toutefois, si le protocole que nous avons établi pour l'utilisation des segments et des régions 2D est indépendant des outils de segmentation utilisés pour les obtenir, il n'en est pas de même pour le processus de génération des différentes cartes régions. En effet, comme nous l'avons déjà constaté ce processus est *ad hoc* puisqu'il est fondée sur une heuristique intimement liée à l'algorithme de segmentation en régions utilisé. Cette dépendance rend impossible l'échange de l'outil de segmentation en régions par un autre sans modification du code du module de gestion de la redemande de cartes.

Dressons à présent le bilan de la partie stéréovision de notre système. Le point fort à retenir ce module réside dans la séparation que nous avons dressée entre la partie recherche en 2D des propriétés 3D et la phase de reconstruction.

La partie appariement ne présente aucune originalité si ce n'est d'avoir enrichi l'ensemble des contraintes généralement utilisés pour la mise en correspondance des segments 2D dans le cadre de la stéréovision par des contraintes intimement liés à la notion de facette 2D. Remarquons ici que si cette mise en correspondance permet de mieux lever les ambiguïtés l'ensemble total des segments mis en correspondance n'est pas aussi nombreux que celui donnés par les algorithmes de mises en correspondance de segments car tous les segments ne font pas forcément partie de facettes 2D. D'ailleurs comme nous l'avons suggéré à la fin du Chapitre 4, consacré à cette partie du constructeur, il serait intéressant de compléter ce processus de mise en correspondance par une mise en correspondance des segments libres du système.

Quand au module de vérification de propriétés 3D en 2D intégré dans notre système, il permet d'une part de vérifier en 2D la planarité des facettes 3D et d'autre part il permet de décrire un algorithme de recherche des classes de triplets de facettes : images de facettes coplanaires. En résumé, cet outil permet (théoriquement) de détecter à partir des triplets de polygones 2D appariés les informations suivantes :

1. déterminer l'ensemble des polygones 2D projections de polygones 3D plans,
2. trouver les classes de polygones 2D projections de facettes 3D coplanaires,
3. trouver les classes de polygones 2D projections de facettes 3D parallèle (cf. Section 7.2),
4. avoir une notion de position relative d'une facette par rapport à une autre.

Cependant, l'expérimentation sur des images bruitées a montré la sensibilité de ces techniques, fondées sur la géométrie projective, aux erreurs dans les images. En réalité, une amélioration de la localisation des points ou encore des segments est, à notre avis, la garantie la plus sûre pour une bonne utilisation de ces méthodes. Ceci implique, entre autre, qu'il est inutile pour l'instant d'utiliser notre technique (qui repose sur les segments 2D) pour traiter des scènes presque polyédriques (scènes d'intérieur comme

la scène des chaises avec des dossier légèrement arrondis) ; en effet, dans ce cas l'erreur sur la localisation des segments 2D est accentuée sur les contours légèrement arrondie et les résultats obtenus par la trace sont alors trop erronés pour être exploitables. La solution serait peut-être de ne pas passer à l'étape d'approximation polygonale et de travailler au niveau des chaînes de point de contours. Ceci implique des changement de fond dans le constructeur actuel qui repose sur l'information segment et non pas sur les chaînes de contours.

Pour terminer dressons le bilan de l'implantation. Notre travail a consisté à développer une chaîne complète de stéréovision. Cela a donné lieu à la réalisation de deux logiciels écrits en langage C et utilisant l'interface Sunview :

1. *le constructeur de facettes 2D* : se charge, à partir d'une image donnée et de son ensemble de segments 2D, de construire l'ensemble des facettes 2D suivant les critères qui ont été développés dans le protocole d'utilisation conjointe des régions et des segments. Le code du constructeur de facettes 2D intègre l'outil de segmentation en régions développé par Fairfield [Fai90]. Ce dernier, auquel nous avons apporté de légères modifications pour l'adapter à nos exigences d'implantation, est également écrit en C et nous a été fourni par le laboratoire de Sophia-Antipolis,
2. *la partie stéréovision de facettes* se compose de trois modules : le module d'appariement de facettes 2D, le module de vérification en 2D des propriétés 3D et le module de reconstruction de facettes 3D. Rappelons que la partie implantation du module d'appariement a été facilitée par le travail effectué par Di Martino [Mar91] dans le cadre de son DEA qui consistait de réaliser la mise en correspondance locale de facettes 2D. Quand au module utilisant la géométrie projective nous signalerons seulement que nous avons eu recours pour son implantation à l'utilisation du logiciel Maple pour établir les coefficients des équations de minimisation répondant à notre spécification et pour les résoudre aux fonctions, écrites en C, du *Numerical Recipes in C* [PFTV88].

7.2 Perspectives

Les perspectives que nous envisageons concernent toutes la partie stéréovision du constructeur de facettes et s'adressent particulièrement à la partie utilisation de la géométrie projective. En effet, de part son caractère passionnant cette voie de recherche nous intéresse beaucoup tant au niveau de son étude théorique qu'au niveau de l'étude de sa mise en application (gestion d'erreur).

La liste des problèmes abordés dans cette section n'est pas exhaustive mais contient les points qui nous semble les plus importants à traiter. Ainsi, dans un premier temps nous parlons du problème de la décomposition des triplets de facettes 2D non planes en parties planes. Puis, nous présentons un résultat théorique que nous avons établi s'intégrant parfaitement dans la logique du constructeur de facettes. Ce résultat permet théoriquement, tout comme pour la coplanarité, de donner en 2D les classes de facettes parallèles sans avoir à les construire. Mais, comme nous l'indiquons ensuite, la sensibilité de cette méthode à l'information point de fuite, a freiné son intégration

dans le constructeur. Enfin, nous proposons quelques améliorations dans le processus de reconstruction des facettes 3D.

• Décomposition des facettes non planes en partie plane

Nous avons souligné dans le Chapitre 6, que seuls les triplets de facettes homologues prouvées être l'image de facettes planes font l'objet d'une reconstruction 3D. Le fait de rejeter les autres triplets de facettes représente l'une des faiblesses de notre système des plus urgentes à corriger.

En effet, nous sommes en droit de nous interroger sur la robustesse de notre module de stéréovision dans le cadre de scènes fortement occultées. Dans de tels scènes les facettes 3D ne se projettent pas en *facettes 2D* vérifiant la contrainte de planarité. Généralement ces facettes 2D, même représentées par une chaîne de segments connexes comportent des segments provenant de l'objet occultant. Ces segments mettent en défaut la contrainte de planarité à vérifier par les facettes. Dans la version actuelle du constructeur de facettes 3D, de tels triplets sont rejetés du système. Il serait pourtant très avantageux de développer une méthode fondée sur la trace qui repère les segments occultants, les ôte de la structure facette 2D et ainsi récupère la partie visible de la facette à reconstruire¹. En réalité, le module de décomposition traite les deux problèmes suivants qui ont des "sémantique" différentes :

1. *élimination des segments occultants* : ce problème correspond à celui soulevé précédemment,
2. *décomposition d'une facette en partie plane* : sous certains angles de prise de vue, certaines arêtes ne sont pas détectées dans les images. De ce fait, deux surfaces connexes planes peuvent se trouver fusionnées et sont alors représentées par une seule facette 2D. Dans pareil cas, le triplet résultant ne vérifie pas la contrainte de planarité.

L'idée que nous proposons pour résoudre ce problème de décomposition consiste à établir une technique d'extraction, au moyen de la trace, des plus grands sous-ensembles de segments coplanaires entre eux de l'ensemble initial des segments composant la facette étudiée. Ces groupes de segments coplanaires peuvent alors faire l'objet d'une hypothèse de facettes 3D. Un gros travail de réflexion et de mise en application est encore à faire en ce sens.

• Test de parallélisme

Il peut sembler au premier abord paradoxal d'utiliser la géométrie projective pour retrouver des propriétés de parallélisme entre les facettes puisque cette dernière banalise le cas du parallélisme en le ramenant à un cas d'intersection normal difficile à caractériser sans la donnée explicite des points à l'infini. Mais dans notre cas, connaissant parfaitement la géométrie de notre banc stéréoscopique, nous pouvons associer à toute direction son point à l'infini. Ainsi la recherche des facettes parallèles entre elles, tout comme les facettes coplanaires, devient possible en 2D. Ce test dont nous avons

1. Remarquons qu'une fois la partie visible de la facette extraite, la vérification de l'hypothèse d'occultation peut être faite en utilisant la propriété de positionnement introduite dans la section 6.6.2.

étudié la partie théorique s'intègre parfaitement dans le module prétraitement intégrée dans la partie stéréovision de notre système.

Rappelons que le parallélisme entre les facettes peut revenir à vérifier que les plans de ces facettes définissent la même droite à l'infini par rapport au plan image étudié. Cette vérification en 2D se fait alors via la connaissance des droites à l'infini des plans de références utilisés. En effet, la détermination de la droite à l'infini d'une facette peut-être faite au moyen de sa trace. Soient Π et Π' deux plans de référence dont la droite d'intersection n'est pas parallèle au plan de la facette. L'intersection de la trace de la facette avec les droites à l'infini de chacun de ces plans détermine deux points distincts² appartenant à la droite à l'infini du plan de la facette et détermine ainsi la droite à l'infini de la facette (voir Figure 7.1.b).

Afin de couvrir tous les cas particuliers et d'établir la contrainte relative au test de parallélisme, nous devons faire appel à une nouvelle contrainte, concernant les plans de référence choisis. Ces derniers doivent vérifier les conditions suivantes :

1. Π, Π' et Π'' ne forment pas un faisceau de plans ce qui exclut également le cas particulier du parallélisme entre les trois plans,
2. Π'' n'est pas parallèle à la droite d'intersection des plans Π et Π' .

Contrainte 5 Désignons par TF_1 et TF_2 deux triplets de facettes appariés 2D et par F_1 et F_2 leur correspondants 3D. Notons par $T_{TF_i}^\Pi$, la trace définie par le triplet TF_i sur le plan Π . Alors F_1 et F_2 sont parallèles si et seulement si l'on est dans l'un de ces deux cas :

- le plan contenant l'une des deux facettes est confondu (resp. parallèle) avec l'un des plans de référence. Supposons sans perte de généralité que $F_1 \in \Pi$. Alors pour que F_2 définisse un plan parallèle à celui de TF_1 il faut et il suffit que $T_{TF_2}^\Pi = \mathcal{D}_\infty^\Pi$ ou que $F_2 \in \Pi$,
- les deux plans contenant les facettes intersectent chacun des plans de références. Alors TF_1 et TF_2 définissent deux facettes 3D parallèles si et seulement si (voir Figure 7.1) :

$$(T_{TF_1}^\Pi \cap T_{TF_2}^\Pi) \in \mathcal{D}_\infty^\Pi \quad \text{et} \quad (T_{TF_1}^{\Pi'} \cap T_{TF_2}^{\Pi'}) \in \mathcal{D}_\infty^{\Pi'} \quad \text{et} \quad (T_{TF_1}^{\Pi''} \cap T_{TF_2}^{\Pi''}) \in \mathcal{D}_\infty^{\Pi''} \quad (7.1)$$

Remarque 11 La droite définie par les points $P_1 = (T_{TF_1}^\Pi \cap T_{TF_2}^\Pi)$, $P_2 = (T_{TF_1}^{\Pi'} \cap T_{TF_2}^{\Pi'})$ et $P_3 = (T_{TF_1}^{\Pi''} \cap T_{TF_2}^{\Pi''})$ est la droite à l'infini des plans support des facettes correspondants à F_1 et F_2 . Elle est illustrée dans la Figure 7.1.b est désigné par $D_\infty^{F_1}$ et $D_\infty^{F_2}$.

2. C'est la restriction sur les plans de références dont la droite d'intersection ne doit pas être parallèle au plan de la facette qui permet d'assurer la construction de deux points distincts.

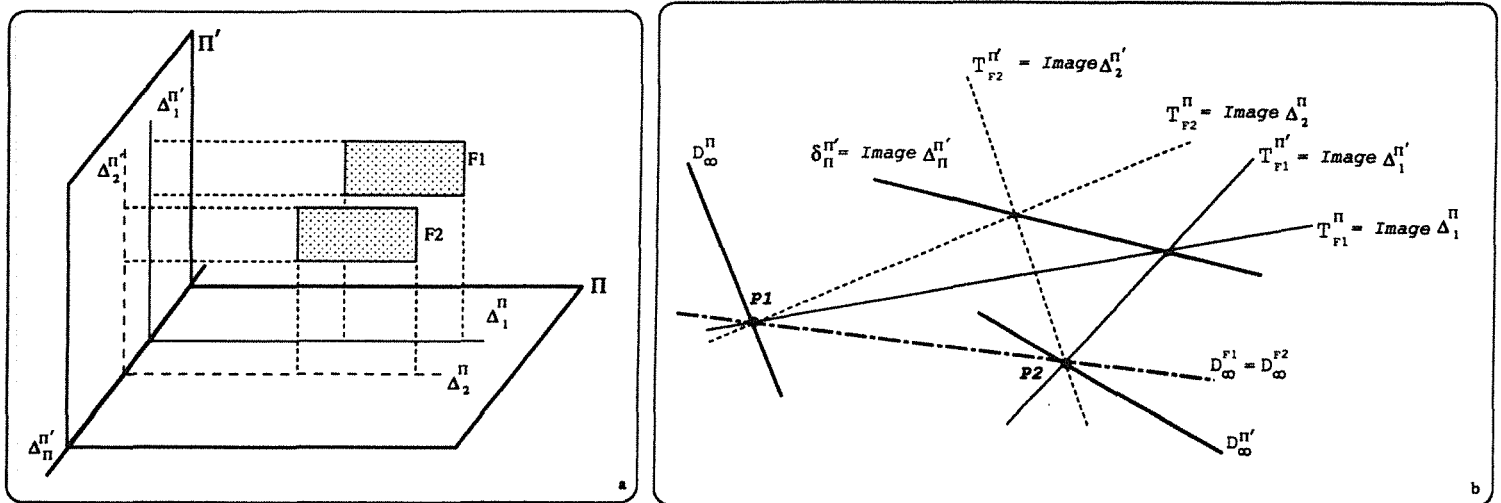


Figure 7.1. Parallélisme en 3D de facettes planes.

Dans le cas d'images réelles et en raison des erreurs de localisation, l'équation 7.1 n'est plus vérifiée. Si dans le cas de la coplanarité, la réponse au problème de la gestion d'erreur revenait à recalculer le couple de traces caractéristiques associés à l'ensemble des facettes coplanaires, puis de vérifier la planarité de l'ensemble, ceci est plus délicat dans le cas du parallélisme.

En fait, la solution la plus élégante revient d'abord à associer à la trace l'erreur que nous commettons dessus. Puis travaillant alors modulo cette erreur nous suggérons d'établir un test comme celui décrit dans la section 6.8.1 et qui permet de vérifier si les droites ont des directions qui peuvent être considérées comme parallèles. Ceci soulève donc un autre problème théorique important que nous n'avons pas traité dans ce mémoire : le problème de l'erreur relative à la construction de la trace. Ce problème, très intéressant, compléterait notre partie de mise en application des résultats établis dans le Chapitre 5 .

• Amélioration des résultats de reconstruction

Finalement, nous remarquons que si nos résultats de reconstruction semblent assez bons (pour les images polyédriques), aucune notion d'erreur 3D de localisation n'est délivrée avec ce résultat. Voulant avoir une idée sur l'erreur attachée à chaque plan de facette construit il semble alors intéressant de mettre en œuvre une technique de reconstruction qui repose sur le filtre de kalman [AF87, Tos87]. Cette technique permet³ d'incorporer l'information sur le bruit des mesures et donne un estimé de l'incertitude sur les paramètres calculés. Dans notre cas, ceci revient à écrire un système qui lie les paramètres du plan 3D recherché (prendre une représentation minimale) aux mesures dans les images : les segments 2D auxquels nous associons l'erreur souvent admise d'un pixel de décalage autour des points extrémités du segment.

3. L'utilisation de la technique du filtre de Kalman sous entend des hypothèses : système linéaire et bruit sur les mesures indépendants.

Une utilisation efficace du filtre de kalman demande généralement à disposer d'une bonne initialisation et à avoir un nombre assez important de mesures afin d'assurer la convergence de cette technique vers la meilleure solution. Remarquons alors que dans notre cadre d'application nous répondons à ces deux exigences puisque :

1. l'initialisation est faite par le plan reconstruit via les traces des facettes. Ce plan est souvent une très bonne approximation du plan réel,
2. l'utilisation de la contrainte de coplanarité ne fait qu'augmenter le nombre de segments intervenant dans la construction du plan support de toutes ces facettes.

Remarquons que l'adjonction de cette partie, qui ne pose aucun problème théorique puisqu'il a déjà été traité par [AF87], à notre constructeur peut se faire sans modifier le code déjà existant du constructeur.

7.3 Conclusion générale

La première partie du travail introduite dans ce mémoire tente de montrer l'intérêt des processus d'un niveau intermédiaire, qui ont pour principal objectif d'utiliser les différentes informations délivrées par le bas niveau en vue de structurer l'image pour faire apparaître les propriétés structurelles de la scène et ainsi en faciliter la compréhension. D'une façon générale, la mise en œuvre de tels processus nécessite d'abord de disposer de processus de bas niveau parfaitement adaptés au type de scène observée. La réalisation du constructeur de facettes 2D a pu être menée à bien grâce à l'existence d'outils du bas niveau produisant des résultats exploitables sur les images utilisées. Les résultats obtenus par le constructeur de facettes 2D permettent donc d'envisager l'intégration d'autres processus s'inscrivant dans une même logique.

La deuxième partie de ce travail s'inscrit dans l'esprit d'une étude d'une nouvelle approche pour la reconstruction sous contrainte. L'éclosion, au cours de ces dernières années, de divers travaux concernant des techniques qui reposent sur la géométrie projective m'ont permis de diriger le sens de mes recherches vers l'utilisation de cet outil mathématique attractif. Cependant la distance que l'on mesure entre les résultats théoriques et les résultats expérimentaux ne font que confirmer la difficulté de la mise en application des théories mathématiques aussi simples soient elles que celles utilisées dans ce mémoire. Et toute l'explication de la difficulté vient du passage aux données réelles bruitées. La robustesse et la stabilité de ces méthodes passe impérativement par une amélioration de la précision sur les primitives extraites des images ainsi que sur la phase de mise en correspondance.

Enfin, je tiens à dire que l'idée de l'utilisation des outils de la géométrie projective m'est venue après avoir assisté à la première réunion de l'équipe Movi, où les collègues du site de Grenoble nous ont initié à la géométrie projective. Je terminerai enfin en disant que pour ma part le plus agréable dans ce travail a été d'acquérir des connaissances sur un domaine dont j'ignorais tout et d'avoir grâce aux perspectives, dégagées de ce travail, encore des voies de recherche intéressantes à explorer.

Annexe A

Algorithme de recherche des contrastes pertinents pour un segment

Cet annexe présente l'algorithme qui découle de la présentation faite dans la section 3.4.3. Nous nous contenterons ici de rappeler uniquement les contraintes (C1) et (C2) pour faciliter la lecture de cet algorithme :

(C1) *Éviter la fusion* : il faut voir apparaître des régions différentes de part et d'autre du segment. Nous considérons que la segmentation en régions est significative si la taille des régions dites gauches (respectivement droites) vérifie la contrainte suivante :

$$\begin{cases} Taille_{Reg}^{droite} \leq 75\% (Taille_{Reg}^{totale}) \\ Taille_{Reg}^{gauche} \leq 75\% (Taille_{Reg}^{totale}) \end{cases} \quad (C1)$$

(C2) *Eviter la sursegmentation* : il faut voir apparaître de part et d'autre du segment une région dominante en taille, notée respectivement R^d et R^g . Ainsi, nous introduisons la deuxième contrainte :

$$\begin{cases} Taille_{R^d} \geq 75\% (Taille_{Reg}^{droite}) \\ Taille_{R^g} \geq 75\% (Taille_{Reg}^{gauche}) \end{cases} \quad (C2)$$

Fonction Recherche_seuil(s)

$C_{min}^s = \infty ; C_{max}^s = \infty ;$

/** Recherche de la valeur de sursegmentation: C_{min}^s **/

$\delta = 0 ;$ /** Cette valeur initiale permet d'effectuer la phase de prétraitement **/

$\delta_{ancien} = 0 ; Continue = Vrai ;$

Tant que Continue Faire

$Fus(\mathcal{G}, \mathcal{D}, \delta) ;$

```

Si Vérifier (C1, C2) Alors
    /** La valeur de sursegmentation est trouvée **/
     $C_{min}^s = \delta$ ; Continue = Faux;
Sinon
    /** Recherche de la prochaine distance à utiliser pour la
    fonction Fus, entraînant un changement certain dans ( $\mathcal{G}, \mathcal{D}$ ) **/
     $\delta_{ancien} = \delta$ ;
     $\delta = \delta + Min\_Distance\_plage(\mathcal{G}, \mathcal{D})$ ;
    Continue = ( $\delta_{ancien} == \delta$ ) /** Plus de modification possible **/
Fin Faire
    /** Recherche de la valeur de fusion :  $C_{max}^s$  **/
     $\delta_{ancien} = \delta$ ;
     $\delta = \delta + Min\_Distance\_plage(\mathcal{G}, \mathcal{D})$ ;
    Continue = ( $\delta_{ancien} == \delta$ );
Tant que Continue Faire
    Fus( $\mathcal{G}, \mathcal{D}, \delta$ ) ;
    Si Vérifier (C1, C2) Alors
         $\delta_{ancien} = \delta$ ;
         $\delta = \delta + Min\_Distance\_plage(\mathcal{G}, \mathcal{D})$ ;
        Continue = ( $\delta_{ancien} == \delta$ )
    Sinon
        /** La valeur de fusion est trouvée **/
         $C_{max}^s = \delta - 1$ ; Continue = Faux;
    Fin Faire
    Retourner ( $C_{min}^s, C_{max}^s$ );
Fin Fonction Recherche_seuil(s)

```

La fonction $Min_Distance_plage(\mathcal{G}, \mathcal{D})$ retourne une valeur correspondant à la plus petite distance possible entre deux plages provenant respectivement des histogrammes \mathcal{G} et \mathcal{D} . Il va de soi que cette plus petite distance occasionne des changements lors de l'application de la fonction $Fus(\mathcal{G}, \mathcal{D}, \delta)$.

Quand les deux histogrammes sont tels que l'un est composé d'une seule plage et l'autre ne comporte plus aucune plage, la fonction $Min_Distance_plage(\mathcal{G}, \mathcal{D})$ retourne alors 0. Aucune modification ne peut être apporté aux deux histogrammes puisque nous sommes dans le cas où nous avons une fusion totale des régions de la zone support du segment.

Annexe B

Calculs des coordonnées

B.1 Coordonnées d'un point sur une droite

Une base projective d'une droite Δ est définie par la donnée de trois points (A, B, C) . Tout point P de cette droite est alors complètement déterminé par la donnée de la valeur du birapport suivant : $k_p = [A, B, C, P]$.

Afin de retrouver les coordonnées cartésiennes de P , à partir de sa coordonnée projective il suffit d'écrire l'équation du birapport.

$$\text{Ainsi, en posant } P = (x_p, y_p) \text{ nous obtenons } \begin{cases} x_p = \frac{(1-k_p)x_a x_b + x_c(k_p x_a - x_b)}{x_c(k_p - 1) - k_p x_b + x_a} \\ y_p = \frac{(1-k_p)y_a y_b + y_c(k_p y_a - y_b)}{y_c(k_p - 1) - k_p y_b + y_a} \end{cases}$$

B.2 Coordonnées d'un point dans un plan

B.2.1 Coordonnées projectives à partir des coordonnées cartésiennes

Pour rechercher les coordonnées projective d'un point dans un plan, nous avons besoin de calculer des birapports de faisceau de droites. La recherche d'un birapport de quatre droites concourantes peut être réalisée en effectuant des calculs de déterminants [Moh92] en mettant en œuvre le théorème suivant :

Théorème 3 Soit l_1, l_2, l_3, l_4 un faisceau de droites passant par le point P_0 . Nous désignons par P_i un point de l_i distinct de P_0 . Nous désignons par Δ_{ijk} le déterminant de la matrice 3×3 où chaque ligne est composée par les coordonnées homogènes des points P_i, P_j, P_k :

$$\Delta_{ijk} = \det(P_i, P_j, P_k) = \begin{vmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{vmatrix}$$

$$\text{alors } [l_1, l_2, l_3, l_4] = [P_0 P_1, P_0 P_2, P_0 P_3, P_0 P_4] = \frac{\Delta_{013}}{\Delta_{014}} / \frac{\Delta_{023}}{\Delta_{024}}$$

Ainsi pour tout point P d'un plan donné nous pouvons lui associer ces coordonnées projective dans une base projective \mathcal{B} , en procédant au calcul de chacune par la méthode décrite par le Théorème 3.

B.2.2 Coordonnées cartésiennes à partir des coordonnées projectives d'un point

Inversement, ayant les coordonnées projective (k_1, k_2) d'un point P relativement à la base $\mathcal{B} = (A, B, C, D)$ ces derniers s'écrivent donc

$$\begin{cases} k_1 = \frac{\Delta_{ABD}}{\Delta_{ABP}} / \frac{\Delta_{ACD}}{\Delta_{ACP}} \\ k_2 = \frac{\Delta_{BAD}}{\Delta_{BAP}} / \frac{\Delta_{BCD}}{\Delta_{BCP}} \end{cases} \quad (\text{B.1})$$

la recherche des coordonnées cartésienne (x, y) de P , se fait en résolvant le système d'équations suivant qui se déduit du système précédent (B.1) :

$$\begin{cases} \Delta_{ABP} = K_1 * \Delta_{ACP} \\ \Delta_{BAP} = K_2 * \Delta_{BCP} \end{cases}$$

avec K_1 et K_2 deux constantes définies comme suit :

$$K_1 = 1/k_1 * \frac{\Delta_{ABD}}{\Delta_{ACD}} \text{ et } K_2 = 1/k_2 * \frac{\Delta_{BAD}}{\Delta_{BCD}}$$

Il est facile de démontrer que ce système est linéaire en x et y et nous avons la solution suivante :

$$\begin{cases} x = (x_a * K_1 + K_2 * (K_1 * x_c - x_b)) / (K_2 * K_1 - K_1 - K_2) \\ y = (y_a * K_2 + K_1 * (K_2 * y_c - y_b)) / (K_2 * K_1 - K_1 - K_2) \end{cases}$$

où (x_a, y_a) , (x_b, y_b) et (x_c, y_c) sont les coordonnées cartésiennes des points de la base A , B et C .

Annexe C

Construction de la droite à l'infini d'un plan

Notre objectif est de présenter les équations nécessaires pour la construction de la droite à l'infini d'un plan Π donné relativement à un plan image Im en utilisant la Propriété 7 énoncée dans le Chapitre 6. Pour se faire nous allons commencer par rappeler le modèle du sténopé.

Soit Rep_{monde} le repère absolu et $Rep_{caméra}$ le repère caméra d'origine le centre optique O , dont les axes x et y sont parallèles respectivement aux axes u et v du Rep_{image} et l'axe z est confondu avec l'axe optique (Figure C.1).

Les transformations entre repères s'écrivent comme suit :

- $Rep_{monde} \rightarrow Rep_{caméra}$: la transformation qui permet de passer du repère du Rep_{monde} au $Rep_{caméra}$ est une combinaison d'une rotation R et d'une translation T . Un point P de coordonnées (x_a, y_a, z_a) dans Rep_{monde} , aura les coordonnées (x_c, y_c, z_c) dans $Rep_{caméra}$ tel que :

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R \begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix} + T$$

La calibration permet de déterminer le vecteur de translation T et la matrice de rotation R .

- $Rep_{caméra} \rightarrow Rep_{image}$: la transformation dans ce cas est définie en utilisant les quatre paramètres suivants délivrés par la calibration :

1. u_0, v_0 : les coordonnées du centre optique de la caméra dans le repère image,
2. α_u, α_v : facteur d'échelle respectivement vertical et horizontal.

La relation entre les coordonnées du point $P (x_c, y_c, z_c)$ dans $Rep_{caméra}$ et celles de son image $p = (u_i, v_i)$ s'exprime comme suit :

$$\begin{cases} u_i = \alpha_u x_i + u_0 \\ v_i = \alpha_v y_i + v_0 \end{cases} \text{ avec } x_i = x_c/z_c \text{ et } y_i = y_c/z_c$$

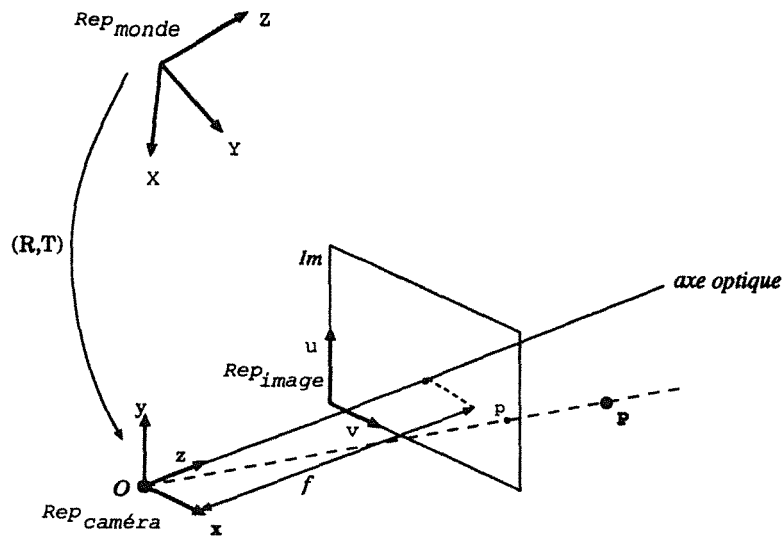


Figure C.1. Différents repères.

Soit alors un plan de référence Π d'équation $aX + bY + cZ + d = 0$ dans Rep_{monde} . D'après la Propriété 7, il s'agit de construire le plan Π_O : parallèle à Π et passant par le centre optique O de coordonnées $(x_o, y_o, z_o)_{Rep_{monde}}$. Dans le repère Rep_{monde} ce plan Π_O a pour équation :

$$\Pi_O : aX + bY + cZ - (ax_o + by_o + cz_o) = 0$$

Dans le repère $Rep_{caméra}$ l'équation du plan Π_O devient :

$$\Pi_O : \alpha x + \beta y + \gamma z = 0 \text{ avec } \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = R \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Il est facile alors de montrer que la droite à l'infini de Π_O : $D_{\infty}^{\Pi_O}$ intersection de Π_O et de Im est définie dans $Rep_{caméra}$ par l'équation suivante :

$$D_{\infty}^{\Pi_O} : \alpha \left(\frac{u_i - u_o}{\alpha_u} \right) + \beta \left(\frac{v_i - v_o}{\alpha_v} \right) + \gamma = 0 \quad (C.1)$$

Annexe D

Autres résultats du constructeur de facettes

Reconstruction de la mire.

Nous présentons dans la Figure D.1 le résultat de reconstruction des plans support des facettes coplanaires des images de la mire. Nous avons visualisé chacun de ces plans l'intersection de chaque plan d'interprétation issu d'un triplet de segments homologues composant les triplets de facettes étudiées des facettes avec le plan support adéquat. On peut voir que la construction n'est pas parfaite puisque certains triplets de segments n'engendrent pas une trace confondue sur le plan support de leur facette. La vue de dessus confirme l'existence d'une erreur lors de la reconstruction puisque nous pouvons voir que l'angle n'est pas parfait (angle droit) entre les plans de la mire.

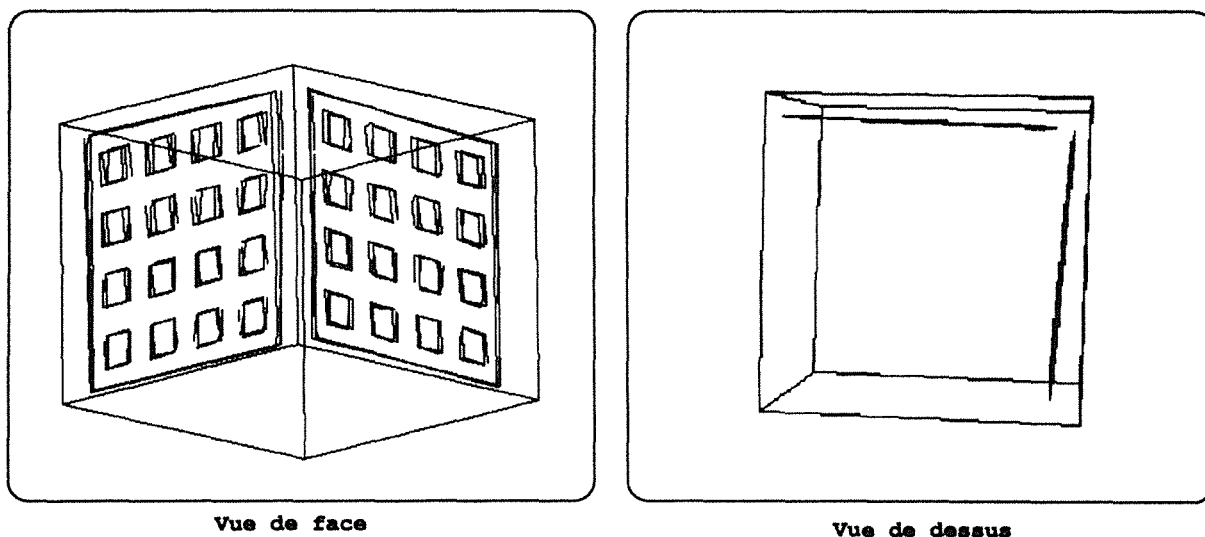


Figure D.1. Reconstruction des facettes de l'image de la mire.

Reconstruction à partir d'autres images de la séquence des posters.

Nous avons illustré dans la figure D.2 le résultat donné par l'étape précédente la recherche au moindre carré des limites des facettes. Le léger décalage entre les traces engendrés par un même triplet de point permet de constater que le plan support induit par la trace est un plan proche de la réalité.

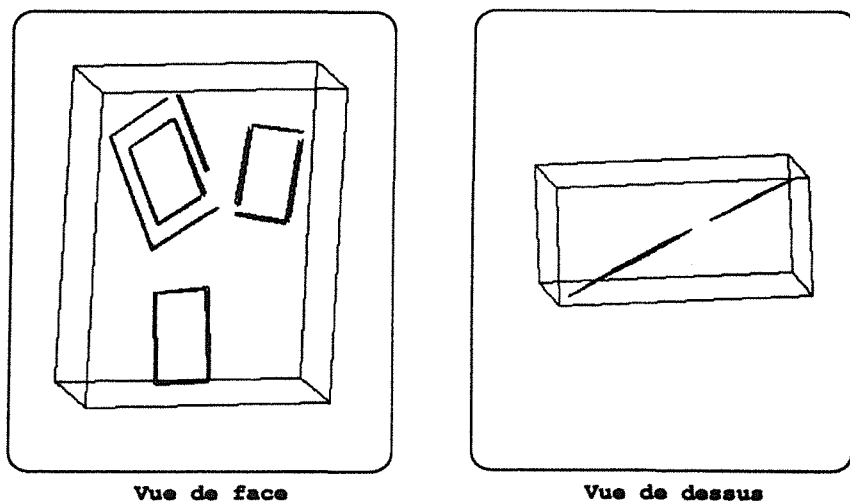


Figure D.2. *Reconstruction des facettes coplanaires.*

Bibliographie

- [AB80] Arnold (R.D.) et Binford (T.O.). – Geometric Constraints in Stereo Vision. *SPIE Image Processing for Missile Guidance*, vol. 238, 1980, pp. 281–292.
- [AF85] Ayache (N.) et Faverjon (B.). – A Fast Stereo Matcher Based on Prediction and Recursive Verification of Hypothesis. *In: Proceedings of the Third Workshop on Computer Vision: Representation and Control, Bellaire, USA*, pp. 73–82.
- [AF87] Ayache (N.) et Faugeras (O.). – Maintaining Representations of the Environment of a Mobile Robot. *In: ISSR*. ISSR, pp. 109–121. – Santa Cruz, 1987.
- [AL87] Ayache (N.) et Lustman (F.). – Fast and Reliable Passive Trinocular Stereovision. *In: Proceedings of the First International Conference on Computer Vision, London*, pp. 422–427.
- [Aya90] Ayache (N.). – *Stereovision and sensor fusion*. – MIT-Press, 1990.
- [BB81] Backer (H.H.) et Binford (T.O.). – Depth from edge and intensity based stereo. *In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, Canada*, pp. 631–636.
- [BB82] Ballard (D. H.) et Brown (C. M.). – *Computer Vision*. – Prentice-Hall, 1982.
- [BB91] Baron (P.) et Briot (M.). – Structuration d'images par utilisation de groupements perceptuels pertinents. *In: Actes du 13^{ème} Colloque sur le Traitement du Signal et des Images (GRETSI), Juan-lès-Pins*, pp. 1133–1135.
- [Ber77] Berger (M.). – *Géométrie 1. Action de groupes, espaces affines et projectifs*. – Paris, CEDIC/FERNAND NATHAN, 1977.
- [BF82] Barnard (S.T.) et Fischler (M.A.). – Computational Stereo. *ACM Computing Surveys*, vol. 14, n° 4, Dec. 1982, pp. 552–572.

- [BFBM87] Boissonnat (J.D.), Faugeras (O.D.) et Bras-Mehlman (E. Le). – Représentation de données stéréo par la triangulation de Delaunay. In : *Actes 6ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Antibes*, pp. 687–701.
- [Bko91] Bkouche (R.). – LA NAISSANCE DU PROJECTIF de la perspective à la géométrie projective. In : *Mathématiques et philosophie de l'antiquité à l'âge classique*. – Edition du Centre National de la Recherche Scientifique, 1991.
- [BMSFB88] Bras-Mehlman (E. Le), Schmitt (M.), Faugeras (O.D.) et Boissonnat (J.D.). – How the Delaunay Triangulation Can Be Used for Representing Stereo Data. In : *Proceedings of the Second International Conference on Computer Vision, Tampa, Florida*.
- [Bon91] Bonnin (P.). – *Méthode systématique de conception et de réalisation d'applications en vision par ordinateur*. – Thèse de doctorat de l'Université de Paris VII, 1991.
- [Bou88] Bouthemy (P.). – Modèles et méthodes pour l'analyse du mouvement dans une séquence d'images. *Technique et Science Informatique*, vol. 7, n° 6, 1988, pp. 527–546.
- [BY84] Brady (M.) et Yuille (A.). – An Extremum Principle for Shape from Contour. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, n° 3, 1984, pp. 288–301.
- [CB93] Chabbi (H.) et Berger (M.O.). – Recovering Planar Surfaces by Stereovision Based on Projective Geometry. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (USA)*. – à paraître.
- [Cha92] Chabbi (H.). – A New Approach for Finding 3D Planar Faces by Stereovision. In : *Proceedings of IAPR International Workshop on Structural and Syntactic Pattern Recognition, Bern (Switzerland)*, éd. par Bunke (H.), pp. 431–440. – World Scientific, August 1992.
- [Cha93] Chabbi (H.). – Checking 3D Planar Surfaces Using Projective Geometry. In : *Proceedings of 8th Scandinavian Conference on Image Analysis, Tromsø (Norway)*. – à paraître.
- [CM91] Chabbi (H.) et Masini (G.). – A Combined Use of Regions and Segments to Construct Facets. In : *Proceedings of the 6th International Conference on Image Analysis and Processing, Como (Italy)*, pp. 334–338.
- [Cox74] Coxeter (H.S.M.). – *Projective Geometry*. – University of Toronto Press, 1974.
- [CSZ91] Castan (S.), Shen (J.) et Zhao (J.). – Stéréovision utilisant les contours ISEF. In : *Actes des Deuxième Journées Nationales du GRECO-PRC, Toulouse*, pp. 425–435.

- [DA89] Dhond (U.R.) et Aggravwal (J.K.). – Structure from Stereo – A Review. *IEEE Trans. Syst. Man Cybern.*, vol. 19, n° 6, 1989, pp. 1489–1510.
- [DA91] Dhond (U.R.) et Aggravwal (J.K.). – A cost-Benefit Analysis of a Third Camera for Stereo Correspondence. *International Journal of Computer Vision*, vol. 6, n° 1, 1991, pp. 39–58.
- [Der87] Deriche (R.). – Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector. *International Journal of Computer Vision*, vol. 1, n° 2, 1987, pp. 167–187.
- [DH73] Duda (R.O.) et Hart (P.E.). – *Pattern Classification and Scene Analysis*. – Wiley-InterScience, 1973.
- [DWD92] DiMartino (J.C.) et Wrobel-Dautcourt (B.). – *Contribution de la stéréovision à la construction de facettes tridimensionnelles*. – Rapport interne n° 92-R-060, CRIN, 1992.
- [Fai90] Fairfield (J.). – Tobbogan Contrast Enhancement for Contrast Segmentation. In: *Proceedings of the 10th International Conference on Pattern Recognition, Atlantic City, NJ (USA)*, pp. 712–715. – Atlantic City, June 1990.
- [Fau88] Faugeras (O.). – Quelques pas vers la vision artificielle en trois dimensions. *Technique et Sciences Informatiques*, vol. 7, n° 6, 1988, pp. 547–590.
- [Fau92] Faugeras (O. D.). – *What Can Be Seen in Three Dimension with an Uncalibrated Stereo Rig?* In: *Proceedings of Second European Conference on Computer Vision, Santa Margherita Ligure (Italy)*, pp. 563–578.
- [Gag89] Gagalowicz (A.). – Coopération entre l'analyse et la synthèse d'images. In: *Actes 7ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Paris*, pp. 1727–1758.
- [GD91] Giraudon (G.) et Deriche (R.). – On Corner and Vertex Detection. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)(à paraître)*, pp. 650–655.
- [Gir87] Giraudon (G.). – *Chaînage efficace de contour*. – Rapports de Recherche n° 605, Rocquencourt, France, INRIA, 1987.
- [GPC91] Guisser (L.), Payrissat (R.) et Castan (S.). – A 3D Active Vision System with a Direct Parametrization of Surfaces. In: *Proceedings of the 7th International Scandinavian Conference on Image Analysis, Aalborg, Denmark*, pp. 210–217.
- [Gri81] Grimson (W.E.L.). – *From Images to Surfaces. A Computational Study of the Human Early Visual System*. – The Massachusetts Institute of Technology Press, 1981.
- [Hor86] Horn (B.K.). – *Robot Vision*. – New York, McGraw Hill, 1986.

- [HS89] Horaud (R.) et Skordas (T.). – Stereo Correspondence Through Feature Grouping and Maximal Cliques. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 11, n° 11, Nov. 1989, pp. 1168–1180.
- [Jar83] Jarvis (R.T.). – A Perspective on Range Finding Techniques for Computer Vision. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, n° 2, 1983, pp. 122–139.
- [Kan89] Kanatani (K.). – 3D Euclidiean Versus 2D Non-Euclidiean: Two Approaches to 3D Recovery from Images. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, n° 3, March 1989, pp. 329–332.
- [Knu75] Knuth (D.E.). – *Sorting and Searching*. – in *The Art of Computer Programming*, Vol. 3, Addison Welsey, Reading, Ma., 1975.
- [Koh81] Kohler (R.). – A Segmentation System Based on Thresholding. *Computer Graphics and Image Processing*, vol. 15, 1981, pp. 319–338.
- [leh88] lehmann (D.). – *Initiation à la géométrie*. – Presse universitaire de France, 1988.
- [LL86] Limozin-Long (P.). – *Vision stéréoscopique appliquée à la robotique*. – Thèse de doctorat de l'Université de Nice, 1986.
- [Low85] Lowe (D.G.). – *Perceptual Organization and Visual Recognition*. – Norwell, Massachusetts, Kluwer Academic Publishers, 1985.
- [LRD91] Lavest (J.M.), Rives (G.) et Dhome (M.). – Utilisation d'un objectif à focale variable en vision monoculaire en vue de la reconstruction 3D. In : *Actes 8ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Lyon*, pp. 293–301.
- [Lus87] Lustman (F.). – *Vision stéréoscopique et perception du mouvement en vision artificielle*. – Thèse de Doctorat. Université de Paris-Sud, 1987.
- [Lut90] Lutton (E.). – *Reconnaissance du point de prise de vue d'une photographie à partir d'un modèle de scène*. – Thèse de doctorat de l'Université de l'École Nationale Supérieur des Télécommunications Paris, 1990.
- [MA90] Mohr (R.) et Arbogast (E.). – It Can Be Done Without Calibration. *Pattern Recognition Letters*, no12, 1990, pp. 39–43.
- [Mar82] Marr (D.). – *Vision*. – San Francisco, Freeman, 1982.
- [Mar91] Martino (J.C. Di). – *Contribution de la stéréovision à la construction de facettes 3D*. – Rapport de DEA en Informatique, Université de Nancy I, 1991.
- [MBD87] Monga (O.) et B.Wrobel-Dautcourt. – Une méthode simple de contrôle de croissance de régions par les contours. In : *Actes 6ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Antibes*, pp. 503–508.

- [MMG91] Morin (L.), Mohr (R.) et Grosso (E.). – Positionnement relatif à partir d'invariants projectifs. *In: Actes 8ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Lyon*, pp. 311–321.
- [MMIQ91] Mohr (R.), Morin (L.), Inglebert (C.) et Quan (L.). – Geometric Solutions to Some 3D Vision Problems. *In: Integration and Control in Real Time Active Vision*, éd. par Crowley (J.L.), Granum (E.) et Storer (R.). – Springer Verlag, 1991.
- [MN88] Mohan (R.) et Nevatia (R.). – *Segmentation and Description Based on Perceptual Organisation*. – Technical report of USC-IRIS, 1988.
- [MN89] Mohan (R.) et Nevatia (R.). – Using Perceptual Organization to Extract 3-D Structures. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, n° 11, 1989, pp. 1121–1139.
- [Moh92] Mohr (R.). – Projective Geometry and Computer Vision. *In: à paraître. Handbook of Pattern Recognition and Computer Vision*, éd. par C.H. Chen (L.F. Pau) et Wang (P.S.P.). – World Scientific Publishing Compagny, 1992.
- [Mon88] Monga (O.). – *Segmentation d'image par croissance hiérarchique de régions*. – Thèse de doctorat de l'Université de Paris Sud Centre d'Orsay, 1988.
- [Mor77] Moravec (H.P.). – Towards Automatic Visual Obstacle Avoidance. *In: Proceedings of 5th International Joint Conference on Artificial Intelligence, Cambridge, MA (USA)*.
- [MP76] Marr (D.) et Poggio (T.). – Cooperative Computation of Stereo Disparity. *Science*, vol. 194, 1976, pp. 283–287.
- [MQVB92] Mohr (R.), Quan (L.), Veillon (F.) et Boufama (B.). – *Relative 3D Reconstruction using Multiples Uncalibrated Images*. – Rapport technique, LIFIA-IRIMAG, 1992.
- [MTB91] Meygret (A.), Thonnat (M.) et Berthod (M.). – Un algorithme de stéréovision adapté aux scènes comportant à la fois des objets naturels et des objets manufacturés. *In: Actes des Deuxième Journées Nationales du GRECO-PRC, Toulouse*, pp. 409–423.
- [MW84] Mohr (R.) et Wrobel (B.). – La correspondance stéréoscopique vue comme une recherche de chemin optimal dans un graphe. *In: Actes 4ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Paris*, pp. 71–79.
- [OPR78] Ohlander (R.), Price (K.) et Reddy (D. Raj). – Picture Segmentation Using A Recursive Region Splitting Method. *Computer Graphics and Image Processing*, vol. 8, 1978, pp. 313–333.

- [Par91] Paris (S.). – Modélisation structurelle par apprentissage. *In : Actes 8ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Lyon*, pp. 1345–1350.
- [PB92] Paris (S.) et Boufama (B.). – Concerning the Effect of the Indexing on the Recognition. *In : Proceedings of IAPR International Workshop on Structural and Syntactic Pattern Recognition, Bern (Switzerland)*, éd. par Bunke (H.), pp. 451–460. – World Scientific, 1992.
- [PFTV88] Press (W.H.), Flannery (B.P.), Teukolsky (S.A.) et Vetterling (W.T.). – *Numerical Recipes in C, The Art of Scientific Computing*. – Cambridge University Press, 1988.
- [PM89] Perona (P.) et Malik (J.). – Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, n° 7, July 1989, pp. 629–639.
- [QM89] Quan (Long) et Mohr (Roger). – Determining Perspective Structures Using Hierarchical Hough Transform. *Pattern Recognition Letters*, vol. 9, n° 4, May 1989, pp. 279–286.
- [QM91] Quan (L.) et Mohr (R.). – Towards Structure from Motion for Linear Features Through Reference Points. *In : Proceedings of the IEEE Visual Motion Workshop, Princeton(NJ)*, pp. 249–254.
- [Qua89] Quan (L.). – *Contribution de la vision monoculaire à la perception tridimensionnelle*. – Thèse de doctorat de l'Institut National Polytechnique de Lorraine, 1989.
- [RG91] Randriamasy (S.) et Gagalowicz (A.). – Region Based Stereo Matching Oriented Image Processing. *In : Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hawaii*, pp. 736–737.
- [RMK87] Remion (Y.), Maitre (H.) et Krahe (J.L.). – Recalage par zones de 2 vues stéréoscopiques d'un univers composé d'objets plans. *In : Proceedings of MARI 87, Paris*, pp. 174–179.
- [Sko88] Skordas (T.). – *Mise en correspondance et reconstruction stéréo utilisant une description structurelle des images*. – Thèse de doctorat de l'Institut National Polytechnique de Grenoble, 1988.
- [SN90] Sparr (G.) et Nielsen (L.). – *Shape and Mutual Cross-Ratio with Applications to Exterior, Interior and Relative Orientation*. *In : Proceedings of First European Conference on Computer Vision, Antibes (France)*, éd. par Faugeras (O.). pp. 607–609. – Springer Verlag.
- [Spa92] Sparr (G.). – *Depth Computations from Polyhedral Images*. *In : Proceedings of Second European Conference on Computer Vision, Santa Margherita Ligure (Italy)*, pp. 378–386.

- [Thi89] Thirion (E.). – *Interprétation et apprentissage géométrique en vision par ordinateur*. – Thèse de doctorat de l'Institut National Polytechnique de Lorraine, 1989.
- [Tho86] Thonnat (M.). – Semantic Interpretation of 3D Stereo Data: Finding the Main Structures. In: *Proceedings of the 8th International Conference on Pattern Recognition, Paris (France)*, pp. 1051–1054.
- [Tis83] Tisseron (C.). – *Géométries affines, projective et euclidienne*. – Paris, Hermann, 1983.
- [Tos87] Toscani (G.). – *Système de Calibration Optique et Perception du Mouvement en Vision*. – Thèse de Doctorat. Université de Paris-Sud, 1987.
- [TT88] Takahashi (H.) et Tomita (F.). – Planarity Constraint in Stereo Matching. In: *Proceedings of the 9th International Conference on Pattern Recognition, Rome (Italy)*, pp. 446–449.
- [VG91] Vezien (J-M.) et Gagalowicz (A.). – Reconstruction 3D basée sur une analyse en régions d'une paire d'images stéréoscopique. In: *Actes 8ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Lyon*, pp. 649–657.
- [Vin86] Vincent (A. Robert De Saint). – *Perception et modélisation de l'environnement d'un robot mobile: une approche par stéréovision*. – Thèse de doctorat de l'Université de Paul Sabatier de Toulouse, 1986.
- [Vin91] Vinet (L.). – Segmentation et mise en correspondance de régions de paires d'images stéréoscopiques. – Thèse de doctorat de l'Université de Paris IX Dauphine, 1991.
- [WD88] Wrobel-Dautcourt (B.). – *Perception de la distance par mise en correspondance de régions dans des images stéréoscopiques*. – Thèse de Doctorat, Institut National Polytechnique de Lorraine, 1988.
- [YKK86] Yachida (M.), Kiamura (Y.) et Kimachi (O.). – Trinocular Vision: New Approach for Correspondence Problem. In: *Proceedings of the 8th International Conference on Pattern Recognition, Paris (France)*, pp. 1041–1044.
- [ZK92] Zagrouba (E.) et Krey (C.). – A Cooperative Algorithm for 2D-Extraction Using Contours and Regions. In: *SPIE's OF TECHNOLOGY'92 - Intelligent Robots and Visual Communications, Hynes, Massachusetts (U.S.A.)*.
- [ZSC91] Zhao (J.), Shen (J.) et Castan (S.). – Trinocular Stereo Vision by Window Matching in 3D space. In: *Proceedings of the 7th International Scandinavian Conference on Image Analysis, Aalborg, Danemark*, pp. 202–209.

Index

Contrainte

- Contrainte 1 *Coplanarité d'une droite et d'un plan*, 94
- Contrainte 2 *Coplanarité d'un ensemble de points*, 96
- Contrainte 3 *Coplanarité d'un ensemble de droites*, 101
- Contrainte 4 *Coplanarité de points: méthode R. Mohr*, 105
- Contrainte 5 *Parallélisme entre facettes*, 147

Définition

- Définition 1 *Facette 2D*, 28
- Définition 2 *Zone support d'un segment*, 29
- Définition 3 *Lien segment-région*, 29
- Définition 4 *Segment inscrit*, 29
- Définition 5 *Facette inscrite*, 30
- Définition 6 *Plage d'intensité*, 45
- Définition 7 *Fonction Fus*, 47
- Définition 8 *Espace projectif*, 82
- Définition 9 *Homographie*, 83
- Définition 10 *Birapport de quatre points*, 83
- Définition 11 *Birapport d'un faisceau de quatre droites*, 84
- Définition 12 *Birapport d'un faisceau de quatre plans*, 84
- Définition 13 *Base projective*, 85
- Définition 14 *Système de base projective en correspondance*, 87
- Définition 15 *Plan de référence*, 88
- Définition 16 *Trace d'une droite*, 91
- Définition 17 *Trace d'un segment*, 94
- Définition 18 *Trace d'un ensemble de points*, 95

Définition 19 *Trace d'un ensemble de droites*, 99

Définition 20 *Trace d'une facette*, 117

Propriété

- Propriété 1 *Point de fuite*, 57
- Propriété 2 *Invariance des coordonnées projectives*, 88
- Propriété 3 *Coincidence entre p_j et $p_{i \rightarrow j}$* , 90
- Propriété 4 *Coincidence entre d_j et $d_{i \rightarrow j}$* , 90
- Propriété 5 *Construction de la trace d'une droite*, 93
- Propriété 6 *Position extrémité segment*, 95
- Propriété 7 *Construction de la droite à l'infini d'un plan*, 111
- Propriété 8 *Construction d'un point à l'infini*, 111
- Propriété 9 *Condition nécessaire pour la coplanarité*, 123
- Propriété 10 *Position d'un point*, 126
- Propriété 11 *Position d'un segment*, 126
- Propriété 12 *Position d'une facette*, 127

Théorème

- Théorème 1 *Invariant projectif*, 83
- Théorème 2 *Coplanarité de deux facettes*, 123
- Théorème 3 *Calcul du birapport par des déterminants*, 153

Liste des Figures

1.1	<i>Architecture simplifiée d'un système de vision artificielle.</i>	5
2.1	<i>Modèle à sténopé</i>	10
2.2	<i>Exemple d'extraction d'indices. A gauche : l'image en niveau de gris. Au milieu : les contours des régions. A droite : les segments.</i>	12
2.3	<i>Géométrie épipolaire</i>	14
2.4	<i>Exemple des segments 3D d'une scène [Mar91]</i>	17
2.5	<i>Architecture du constructeur de facettes 3D.</i>	24
3.1	<i>Architecture globale du constructeur de facettes 2D.</i>	27
3.2	<i>Facettes finales et facettes incertaines.</i>	28
3.3	<i>Correspondance région-segment.</i>	30
3.4	<i>Facettes pertinentes et facettes incertaines.</i>	30
3.5	<i>Construction d'une jonction-L ([Sko88]).</i>	32
3.6	<i>Fusion des premières facettes incertaines du système.</i>	33
3.7	<i>Cycle de déroulement de la phase-1 à l'étape i.</i>	35
3.8	<i>Décomposition d'une jonction.</i>	36
3.9	<i>Décomposition d'un segment.</i>	37
3.10	<i>Fermeture d'une facette.</i>	38
3.11	<i>Jonction créée à la Phase-2.</i>	39
3.12	<i>Effet de l'algorithme de renforcement d'image. a) Nappe de l'image initiale. b) Nappe de l'image clarifiée.</i>	42
3.13	<i>Intervalle des valeurs pertinentes contrôlant la segmentation en régions de part et d'autre du segment.</i>	43
3.14	<i>Principe du contrôle de la génération des cartes régions.</i>	44
3.15	<i>Distributions des intensités dans la zone support d'un segment relativement à l'image initiale et à l'image clarifiée.</i>	45
3.16	<i>Distributions des plages initiales des plages des histogrammes \mathcal{G} et \mathcal{D}.</i>	47
3.17	<i>Illustration de l'effet de la fonction Fus.</i>	48
3.18	<i>Décomposition d'un segment pour la recherche de seuils pertinents.</i>	50
3.19	<i>Image initiale et l'ensemble de segments extraits.</i>	52
3.20	<i>Résultats des différentes étapes de construction de facettes 2D.</i>	53
3.21	<i>Résultats (suite).</i>	54
3.22	<i>Facettes 2D de l'image de la caméra 1.</i>	55
3.23	<i>Facettes 2D de l'image de la caméra 2.</i>	55
3.24	<i>Facettes 2D de l'image de la caméra 3.</i>	55
3.25	<i>Architecture étendue du constructeur de facettes 2D.</i>	56

3.26	<i>Le point de fuite pour un ensemble de segments 3D parallèles.</i>	57
3.27	<i>Partant de l'image des segments en haut à gauche, nous récupérons trois groupes directionnels représentés chacun dans une image [Qua89]</i>	57
3.28	<i>Points de fuite des droites coplanaires d'un échiquier vu en perspective ([leh88] page 116).</i>	58
3.29	<i>Cas d'une facette non plane non détectée par la Propriété 1</i>	59
3.30	<i>Résultats de construction de facettes 2D[ZK92]</i>	61
4.1	<i>Contrainte épipolaire dans un système trinoculaire. Le point p_3 est à l'intersection des droites épipolaires E_{13} et E_{23} calculées respectivement à partir des points p_1 et p_2.</i>	66
4.2	<i>Contrainte de direction.</i>	70
4.3	<i>Contrainte de positionnement.</i>	70
4.4	<i>Résultat de la mise en correspondance globale.</i>	72
4.5	<i>Appariement d'une facette f de Im_1 qui se présente sous forme de deux facettes dans l'image Im_2.</i>	73
4.6	<i>Représentation du triplet d'images de segments. Seuls les segments portant un numéro font partie d'une facette.</i>	74
4.7	<i>Résultat d'appariement de facette 2D.</i>	75
4.8	<i>Les images des triplets de facettes.</i>	75
4.9	<i>Les triplets de facettes homologues.</i>	76
4.10	<i>Processus d'appariement de segments 2D avec la notion de facettes 2D.</i>	77
5.1	<i>Comment aborder l'étude des images 2D ?</i>	81
5.2	<i>Conservation du birapport.</i>	84
5.3	<i>Birapport d'un faisceau de droites concourantes.</i>	84
5.4	<i>Droite l_k correspondant à une valeur de birapport k donné.</i>	86
5.5	<i>Coordonnées projectives dans un plan.</i>	86
5.6	<i>Couple de bases projectives en correspondance.</i>	87
5.7	<i>Reconstruction de l'image d'un point P appartenant à un plan de référence.</i>	89
5.8	<i>$T_{2 \rightarrow 1}^\Pi$ appliquée à un point.</i>	90
5.9	<i>Représentation graphique de $T_{2 \rightarrow 1}^\Pi$ appliquée à une droite.</i>	91
5.10	<i>La détermination de la trace d'un segment S et de la droite L.</i>	92
5.11	<i>Détection de l'extrémité la plus proche de S à Π en utilisant la trace.</i>	94
5.12	<i>Trace d'un ensemble de 4 points coplanaires.</i>	96
5.13	<i>Exemple de non coplanarité de deux ensembles de points définissant une même droite de trace.</i>	99
5.14	<i>Exemple de non coplanarité d'un ensemble de droites ayant des points de trace alignés.</i>	100
5.15	<i>La non coplanarité d'un ensemble de droites est détectée en utilisant deux plans de références Π et Π'.</i>	102
5.16	<i>Contrainte de coplanarité de points utilisant la géométrie épipolaire.</i>	106
6.1	<i>Prétraitement des triplets de facettes homologues et construction des facettes 3D.</i>	110
6.2	<i>La construction : de la droite à l'infini D_∞^Π du plan Π et du point de fuite p_∞ de la droite Δ appartenant à Π.</i>	112

6.3	<i>Erreur dans la construction des traces. (a): les erreurs de reconstruction représentées en 3D. (b): leur conséquences sur la recherche des traces.</i>	113
6.4	<i>La droite de trace associée à un ensemble de segments coplanaires: (a) en théorie: elle passe par les points de traces, (b) en pratique: elle traverse chacune des zones de présence.</i>	116
6.5	<i>La construction de l'isobarycentre de deux points.</i>	117
6.6	<i>Un couple de points images issu de l'intersection des images de deux segments non coplanaires.</i>	118
6.7	<i>(a) La représentation 3D de trois facettes et de deux plans de références Π_1 et Π_2. (b) La représentation de la trace de ces trois facettes sur Π_1 et Π_2, f_1 et f_2 sont coplanaires: ils définissent le même couple de traces quant à f_3 elle définit un autre couple de traces.</i>	123
6.8	<i>Cas de facettes 3D appartenant au même faisceau de plan que Π_1.</i>	124
6.9	<i>Décomposition de $\delta_{\Pi_1}^{\Pi_j}$.</i>	125
6.10	<i>Caractérisation de la position d'un point par rapport à un plan.</i>	127
6.11	<i>La présentation de la position d'un segment reconstruit par rapport à sa position relative au plan de référence.</i>	128
6.12	<i>Construction de deux droites du plans support d'une facette F à partir de ses deux traces T_F^{Π} et $T_F^{\Pi'}$.</i>	130
6.13	<i>Erreur sur la direction de l'image d'un segment S, parallèle au plan Π.</i>	132
6.14	<i>Triplets de facettes 2D homologues.</i>	133
6.15	<i>Trace de facettes coplanaires au plan de référence.</i>	133
6.16	<i>Couple de traces caractéristiques d'une facette plane.</i>	134
6.17	<i>Points de traces d'une facette non plane.</i>	134
6.18	<i>La scène de synthèse traitée.</i>	135
6.19	<i>Les triplets d'images de synthèses.</i>	136
6.20	<i>Trace calculée sur l'image de synthèse.</i>	136
6.21	<i>Résultats de la reconstruction des facettes de l'image de synthèse.</i>	136
6.22	<i>Triplet de facettes homologues: images réelles.</i>	137
6.23	<i>Traces des facettes de l'image réelle.</i>	137
6.24	<i>Résultat de la reconstruction des facettes 3D.</i>	138
6.25	<i>Triplet d'images des segments.</i>	139
6.26	<i>Résultat de la reconstruction des segments 3D.</i>	139
7.1	<i>Parallélisme en 3D de facettes planes.</i>	148
C.1	<i>Différents repères.</i>	156
D.1	<i>Reconstruction des facettes de l'image de la mire.</i>	157
D.2	<i>Reconstruction des facettes coplanaires.</i>	158

**AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL POLYTECHNIQUE DE
LORRAINE**

o o o

VU LES RAPPORTS ETABLIS PAR :

**Monsieur RICHETIN Marc, Professeur, Université Blaise Pascal Aubière,
Monsieur CASTAN Serge, Professeur, Université Paul Sabatier Toulouse,
Monsieur HATON Jean-Paul, Professeur, CRIN/Université Nancy I.**

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Mademoiselle CHABBI Houda

à soutenir devant l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, une thèse intitulée :

"Construction de facettes 3D par stéréovision intégrant des principes de géométrie projective"

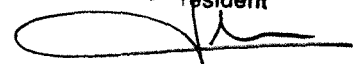
en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : **"INFORMATIQUE"**

Fait à Vandoeuvre le, **1^o Février 1993**

Le Président de l'I.N.P.L.,
M. LUCIUS



CHEVRIER

Résumé

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la vision artificielle et l'objectif recherché est de développer un processus de reconstruction stéréoscopique des facettes 3D des scènes observées.

Le principe fondamental que nous avons suivi pour réaliser la construction de facettes 3D a été d'éviter dans la mesure du possible, tout calcul direct sur les segments 3D. Nous avons alors proposé un protocole d'utilisation de l'information région et l'information contour pour construire des indices 2D : *facettes 2D* par l'intermédiaire desquels nous inférons l'existence d'un plan 3D. Utilisant ensuite des techniques de géométrie projective, nous vérifions en 2D la validité de l'hypothèse de planarité relative à chacun de ces indices. La recherche des paramètres de chacune des facettes 3D de la scène utilise alors toute l'information acquise en 2D concernant les propriétés 3D à vérifier par ces facettes à construire. Cette reconstruction permet ainsi de maintenir la propriété de planarité ou encore de coplanarité entre facettes améliorant la localisation 3D de ces primitives.