UNIVERSITÉ DE PICARDIE JULES VERNE
CENTRE DE ROBOTIQUE, D'ELECTROTECHNIQUE ET D'AUTOMATIQUE

THESE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE PICARDIE JULES VERNE

Discipline : Robotique

présentée et soutenue publiquement
par

David FOFI

le 18 septembre 2001

Titre :

Navigation d'un Véhicule Intelligent à l'aide d'un Capteur de Vision en Lumière Structurée et Codée

Directeurs de thèse : El Mustapha Mouaddib, Joaquim Salvi

JURY

M. Florent CHAVAND, Professeur à l'Université d'Evry	Président
Mme Edwige PISSALOUX, Professeur à l'Université de Rouen	Rapporteur
M. Jean DEVARS, Professeur à l'Université Pierre et Marie Curie – Paris 6	Rapporteur
M. Joaquim SALVI, Maître de Conférences à l'Université de Girona (Espagne)	Examinateur
M. Claude PEGARD, Professeur à l'Université de Picardie Jules Verne	Examinateur
M. El Mustapha MOUADDIB, Professeur à l'Université de Picardie Jules Verne	Directeur

Table des Matières

Liste des Illustrations

Liste des Tableaux

Remerciements

J'aimerais tout d'abord remercier les personnes qui ont suivi et dirigé ces travaux de recherche : El Mustapha Mouaddib, pour le sujet qu'il m'a proposé et pour m'avoir guidé tout au long de ces années, et Joaquim Salvi, pour l'aide et le soutien précieux qu'il m'a apportés, ainsi que pour m'avoir accueilli plusieurs semaines dans son laboratoire, en Espagne.

Je remercie également l'ensemble du jury, et en particulier les rapporteurs, Edwige Pissaloux et Jean Devars, pour la pertinence de leurs remarques et la justesse de leurs appréciations, et Florent Chavand pour m'avoir fait l'honneur de le présider.

Je n'oublie évidemment pas l'ensemble des collègues de l'équipe *Perception en Robotique*, Claude Pégard en tête, qui m'a permis d'enseigner au sein du département informatique de l'IUT d'Amiens, et les permanents, cités ici dans un strict ordre d'ancienneté : Eric Brassart, Laurent Delahoche, Pascal Vasseur, Djemâa Kachi et Ouiddad Igbida-Labani, Bruno Marhic. Je suis très reconnaissant à Pierre Détaille pour l'aide technique qu'il m'a apporté et pour sa grande disponibilité. Je remercie les thésards de l'équipe Cyril Cauchois, Arnaud Clérentin, Cyril Drocourt et tout particulièrement Arnaud Dupuis et Josselin Harp pour l'amitié dont ils ont fait preuve. Je joins à cette liste les nouveaux arrivants, Cédric Demonceaux et Mohammed Afif, et le récemment parti, Nicolas Hutin.

J'ai eu la chance, pendant ces années de thèse, de pouvoir effectuer un stage de recherche de deux mois au sein du laboratoire de robotique et vision dirigé par Joan Batlle, à Girona, en Espagne ; qu'il en soit vivement remercié, de même que tous les membres de son équipe, pour l'accueil chaleureux qu'ils m'ont réservé. Qu'ils m'excusent de ne pouvoir les citer tous nommément.

Je voudrais citer également l'ensemble des stagiaires ; les erasmus : Maria Sanz, Juan-Carlos Calvo Polanco et Pablo Lopez Mendetia, ce dernier pour les prises de vue et son aide sur l'analyse du mouvement ; l'équipe *VisioTech* : Alexandre Lima, qui a grandement contribué à l'implémentation des algorithmes de traitement de l'image en C++ et aux résultats expérimentaux y afférents présentés dans ce mémoire, Jean-Christophe Fretel et Nicolas Piskorski ; aussi, Guillaume Martin et Benjamin Champion. Un grand merci à tous pour l'ambiance amicale qu'ils ont su créer lors de leur passage au laboratoire.

Enfin, je remercie de manière toute particulière mes parents, mon frère, ma famille et mes amis.

Introduction Générale

Tout organisme vivant prélève de l'information dans le milieu qui l'entoure. Il est tout à fait évident qu'il s'agit là d'une fonction indispensable à son adaptation et à sa survie. Aussi peut-on l'observer chez les organismes les plus simples : une bactérie dispose d'analyseurs l'informant sur la composition chimique du milieu dans lequelle elle se développe. Le mot "information" dans son acception scientifique a, bien entendu, un sens beaucoup plus large que dans la langue usuelle. Il est employé pour désigner tout ce qui lève une incertitude, une indétermination. On peut alors parler d'information en décrivant le fonctionnement d'une machine : si, dans la machine, un interrupteur électrique peut être ouvert ou fermé, l'événement qui place l'interrupteur dans l'une ou l'autre de ces positions apporte de l'information à la machine. Une théorie mathématique de l'information ainsi définie a été élaborée par C. E. Shannon et W. Weaver en 1949.
Chez les animaux supérieurs et chez l'homme, l'information prélevée par les organes des sens circule dans le système nerveux sous forme de signaux se manifestant par des phénomènes électriques que l'on peut enregistrer. Mais la fonction du système nerveux ne se limite pas à transmettre les signaux, c'est-à-dire de l'information. Elle consiste essentiellement à *traiter* cette information, c'est-à-dire à la transformer. Les informations provenant d'un organe sensoriel comportant une multiplicité de cellules receptrices sont regroupées, comparées, structurées, filtrées. Elles sont traitées en fonction d'autres informations provenant des stimulations antérieures de cet organe ou d'autres modalités sensorielles, ou stockées dans le système nerveux et évoquées en l'absence de toute stimulation actuelle. L'ensemble de ce traitement extrêmement complexe et constamment en cours est orienté par les *attitudes* du sujet, c'est-à-dire par l'état de préparation dans lequel il se trouve au moment où la stimulation intervient, et aussi par ses besoins, ses *motivations* à ce moment-là. Ainsi, le recueil et le traitement de l'information sensorielle ne constituent pas des processus isolés de l'ensemble de la conduite. Ils sont au contraire orientés par les phases antérieures et par l'objet de cette conduite. Les actions qu'ils déclenchent et dont ils contrôlent le déroulement leur apportent des informations *en retour* (en Anglais, *feed-back*) qu'ils intègrent : la vision que j'ai d'un certain objet peut déclencher et contrôler le geste que je fais pour le saisir, mais cette activité motrice et ses résultats complètent mon information sur la position, la forme ou la nature de l'objet.

Ce mémoire de thèse est centré sur la perception visuelle ; mais, au lieu de s'intéresser aux aspects biologiques ou psychologiques de la chose, ce sont les aspects robotiques et informatiques qui sont étudiés : on appelle cette discipline *vision par ordinateur*. Dans la mesure où la comparaison entre un organisme vivant et une machine peut être faite, il s'agit de retranscrire, dans un language

compréhensible par la machine, les mécanismes de la vision biologique : la localisation et l'identification principalement. Bien sûr, malgré les progrès considérables de la communauté scientifique dans le domaine de la vision par ordinateur, les réponses apportées par les systèmes de vision artificielle demeurent partielles et partiales, loin encore (mais s'en rapprochera-t-elle jamais ?) de la précision, de la vitesse et de la richesse de la vision animale.

Deux paradigmes trouvent crédit au sein de la communauté des chercheurs en vision par ordinateur. Le premier, que nous appellerons *neuromimétique*, s'inspire directement de l'observation du processus de vision biologique et tente de l'imiter. La principale difficulté provient du fait que le système de perception visuelle humain est encore mal connu dans ses détails, ce qui conduit souvent à des modèles trop qualitatifs pour être exploitables algorithmiquement. Sans compter que la nature a produit un très grand nombre de solutions au problème de la perception visuelle, dont l'efficacité (et la pérennité) dépend de l'adaptation du mode de vision au milieu où l'animal évolue. C'est l'un des grands enseignements de la nature, révélé par les éthologues, qu'un mode perceptif ne peut être détaché du milieu d'évolution de l'organisme ; la science, au contraire, aurait tendance à chercher des solutions génériques... Le second consiste à apporter des réponses *mathématiques* au problème de la vision sans se soucier de la plausibilité neuro-biologique de ces réponses. Des algorithmes, des méthodes quantitatives, numériques, exploitables par l'informatique, peuvent en être tirés ; les retombées pratiques, technologiques sont directes, contrôlables et analysables. Les solutions mathématiques à la perception visuelle permettent aussi d'interroger la nature, par analogie et comparaison. Même dans ce cas, le monde biologique reste une source d'inspiration importante. Le travail présenté dans ce mémoire suit le second paradigme. L'application visée par ces travaux est la navigation, la plus autonome possible, d'un robot mobile à l'aide d'un capteur de vision d'un type un peu particulier, composé d'une source lumineuse appelée *lumière structurée et codée*.

0.1 Cadre, Objectifs et Contributions

Le travail réalisé pour cette thèse a été effectué au sein du laboratoire CREA (*Centre de Robotique, d'Electrotechnique et d'Automatique*) de l'Université de Picardie Jules Verne et, partiellement, au *Computer Vision and Robotics Group* de l'Université de Girona, en Espagne, dans le cadre d'une collaboration née en 1995. L'axe robotique du CREA, animé par El Mustapha Mouaddib, est dédié à la perception ; ses principaux thèmes de recherche sont l'étude et le développement d'un capteur de vision omnidirectionnel (SYCLOP) pour la localisation et la navigation de robots mobiles, la coopération multi-capteurs, la reconnaissance des formes, l'analyse de texture. C'est à la suite de la collaboration entreprise avec l'Espagne que la lumière structurée et codée fut agrégée aux thèmes de recherche du laboratoire. Une première thèse, rédigée par Joaquim Salvi et co-encadrée par

messieurs El Mustapha Mouaddib et Joan Batlle, a vu le jour en 1997 [SAL97]. Elle a abouti à une nouvelle technique de codage de la lumière structurée et a fourni de précieux résultats en matière de reconstruction tri-dimensionnelle à partir de la calibration forte du capteur.
L'objectif qui nous était fixé était l'application de la vision en lumière structurée à la navigation des véhicules intelligents. En d'autres termes, les questions qui nous étaient posées étaient : comment tirer parti des particularités de la vision en lumière structurée pour la perception d'un robot mobile ? Quels sont les avantages que l'on peut tirer d'un tel capteur, mais quelles en sont également les limites ? Etant donnés ces avantages et ces limites, peut-on envisager d'améliorer les techniques de navigation existantes par l'utilisation d'un capteur de vision en lumière structurée et codée ? Dans quelles conditions ?
Cette étude a abouti à quelques utiles contributions, dans le domaine propre à la lumière structurée et codée, pouvant être exploitées en robotique mobile :

- Nous proposons d'abord une méthode de décodage du motif structurant permettant de résoudre le problème de la mise en correspondance à partir d'une seule prise d'image.
- Nous étendons les algorithmes de reconstruction non-calibrée à la vision en lumière structurée, après une analyse complète ayant permis de définir les contraintes imposées par la projection d'un patron lumineux aux méthodes de reconstruction. Nous proposons également un test de colinéarité spatial et un test de coplanarité opérant uniquement à partir des coordonnées pixels des points.
- Nous donnons une méthode de détection d'obstacles de type quantitatif permettant de construire une carte de l'espace libre.
- Nous étudions le problème de l'analyse du mouvement en lumière structurée et montrons qu'il est possible d'obtenir une information sur le déplacement du capteur par rapport à la scène (ou d'un objet par rapport au capteur) à partir de de l'équation des plans qui composent la scène.

Le deuxième et le quatrième point, s'ils ont été grandement étudiés pour la stéréovision, restaient jusqu'alors largement inexplorés (et inexploités) en vision en lumière structurée. L'algorithme de décodage, outre son utilité pratique, démontre la pertinence du codage utilisé : c'est en effet le choix des primitives, le codage par triplets uniques et l'indépendance des éléments verticaux et horizontaux qui permet un décodage fiable et robuste du motif.

0.2 Organisation du Mémoire

Le mémoire s'articule autour des différents points qu'il nous a semblé utile d'analyser en vue d'une complète adaptation d'un capteur de vision en lumière

structurée et codée au problème de la navigation d'un robot mobile. Ceci nous a conduit à étudier différents domaines de la vision par ordinateur et du traitement d'images : nous avons, à chaque fois que cela a été possible, fait une analyse de l'existant, reliée aux enjeux et problèmes qui nous intéressent et à la manière de les résoudre. Les méthodes retenues sont toutes illustrées par des résultats expérimentaux commentés et analysés.

Dans le premier chapitre, nous exposons les principes et les caractéristiques de la vision en lumière structurée ; vue souvent comme un cas particulier de la stéréovision, nous en montrons les avantages, les points faibles et les différences essentielles qui existent entre la vision par capteur passif et la vision par capteur actif. Nous présentons enfin les principales applications qu'a trouvées jusqu'alors la vision en lumière structurée dans le domaine industriel ou scientifique.

Nous détaillons, dans le second chapitre, les traitements bas-niveau mis en œuvre en vue de l'obtention de données pertinentes dont recèlent les images. Les outils utilisés sont bien connus des traiteurs d'image : morphologie mathématique et transformée de Hough, en premier lieu. Un algorithme de décodage du patron est également proposé. A partir de considérations colorimétriques simples, dont nous rappelons quelques bases en préambule, nous proposons une technique de reconnaissance de la couleur projetée à partir de la couleur apparente en une seule prise d'image.

Le troisième chapitre traite de la reconstruction tri-dimensionnelle et de la géométrie qu'elle sous-tend. Nous détaillons la modélisation d'un capteur de vision et la géométrie épipolaire des systèmes binoculaires. Après avoir rappelé comment il était possible de calibrer un système de vision et comment, à partir d'un système calibré, il était possible de reconstruire la scène observée, nous proposons un état de l'art détaillé sur les techniques de reconstruction sans calibration. Pour conférer au capteur de vision des degrés de liberté supplémentaires, de sorte que la mise au point, le zoom et l'ouverture du diaphragme soient librement modifiables, la vision non-calibrée paraît être incontournable. Si l'on veut s'inspirer de la formidable capacité adaptative de la perception humaine (l'oeil est un capteur auto-focus, dont l'ouverture de diaphragme s'adapte à la luminosité ambiante, et la fovéation permet d'incrémenter la résolution en des régiones pertinentes de l'image rétinienne), il est évident que la calibration forte est à proscrire.
Nous décrivons d'abord une méthode de reconstruction projective adaptée à nos besoins et aux contraintes induites par la projection du motif structurant. Puis, nous proposons une méthode originale de reconstruction qui utilise les contraintes géométriques générées par la projection du patron de lumière pour redresser euclidiennement une scène projectivement reconstruite. A cela, nous ajoutons des outils permettant de s'affranchir de toute intervention humaine lors de la

reconstruction comme les tests de colinéarité et de coplanarité et un test de validité du modèle affine.

Dans le quatrième chapitre, nous proposons une étude des fonctions visuelles utiles à la navigation d'un robot mobile. Nous présentons d'abord une méthode de détection d'obstacles basée sur l'extraction des quasi-verticales de l'image. Nous montrons que ceci équivaut à caractériser les obstacles par la pente au point considéré. Nous complétons cette méthode par une technique de cartographie de l'espace libre, obtenue à partir de règles simples permises par la projection du patron de lumière. Finalement, nous nous penchons sur le problème de l'analyse du mouvement qui constitue une différence essentielle entre la stéréovision et la vision en lumière structurée. Nous montrons les problèmes soulevés par la projection, et nous tentons d'apporter quelques réponses. Enfin, nous proposons une méthode d'analyse du déplacement du robot dans la scène à partir de la mise en correspondance des plans qui la composent.

Nous résumons, dans une conclusion générale, les méthodes et stratégies retenues et nous essayons de dégager la réelle pertinence et les apports de la vision en lumière structurée au problème de la navigation d'un robot mobile. Nous tentons de définir ce qui relève de l'applicable et nous soulignons les points faibles de la vision en lumière structurée. Nous donnons enfin les perspectives scientifiques et technologiques ouvertes par ce travail.

0.3 Notations et Terminologie

Les points 3-D sont représentés par des lettres majuscules placées en italique ; ainsi *P*, *Q*, etc... L'image d'un point 3-D est donnée par la même lettre, toujours en italique, mais cette fois-ci, en minuscule : le projeté de *P* est *p*. Le vecteur des coordonnées d'un point *P* (ou *p*) est désigné par **P** (ou **p**, respectivement). D'une manière générale, les matrices et vecteurs sont représentés par des lettres romanes, en caractères gras.
Nous parlerons indifféremment de diapositive, de patron de lumière, de motif structurant ou d'image projetée pour désigner l'image issue du projecteur. Inversement, nous parlerons d'image capturée, d'image perçue, d'image codée ou simplement d'image quand il s'agira d'une image acquise par la caméra. On qualifiera de patron ou de motif *perçu* les donnée émises par le projecteur telles qu'elles sont capturées par la caméra. La grande majorité des travaux entrepris en vision par ordinateur ont pour support la stéréovision. Nous conserverons, quand les résultats demeurent valides, la terminologie usuelle en parlant de première et seconde image ; quand, en revanche, une différence essentielle apparaît entre stéréovision et vision en lumière structurée, les précautions de vocabulaire seront précisées.

$\mathbf{A}$	Matrice des paramètres intrinsèques
$\mathbf{C}$	Centre optique : centre de projection de la caméra
e, $\mathbf{e}$	Epipole de la première image, vecteur des coordonnées de l'épipole
e', $\mathbf{e}'$	Epipole de la seconde image, vecteur des coordonnées de l'épipole
$\mathbf{E}$	Matrice essentielle
$\mathbf{F}$	Matrice fondamentale
$\mathbf{I}$	Matrice identité
m, $\mathbf{m}$	Un point 2-D, vecteur des coordonnées du point
$\mathbf{m} \leftrightarrow \mathbf{m}'$	Couple de points homologues (appariement)
M, $\mathbf{M}$	Un point 3-D, vecteur des coordonnées du point
$\tilde{\mathbf{M}}, \tilde{\mathbf{P}}, \tilde{x}$	Vecteur, matrice ou scalaire exprimés dans un repère projectif
$\mathbf{P}$	Matrice de projection
$\mathbf{P}^{(c)}$	Matrice de projection de la caméra
$\mathbf{P}^{(p)}$	Matrice de projection du projecteur
$\wp^n$	Espace projectif de dimension n
$\mathbf{R}$	Matrice de rotation
$\mathbf{t}$	Vecteur de translation
Π	Un plan
Π_∞	Le plan de l'infini
$\mathbf{\Omega}$	La conique absolue
$\langle x, y \rangle$	Droite passant par les points x et y

Table 1. Principaux symboles utilisés

Chapitre 1 :
Vision 3-D et Lumière Structurée

1.1 Introduction

Un robot mobile, ou véhicule intelligent (ce n'est ici qu'une question de terminologie) a besoin d'un certain nombre de capteurs pour, comme les organes des sens chez l'homme, acquérir de l'information sur son environnement. On sait l'importance de la vue dans le système sensoriel humain, c'est donc naturellement celle-ci qui a été l'objet du plus grand nombre d'études et qui a permis le développement du plus grand nombre d'applications en sciences pour l'ingénieur.
L'objectif de ce chapitre est de positionner la vision en lumière structurée au sein des techniques de vision par ordinateur permettant d'inférer une information tri-dimensionnelle de la scène. Nous passons rapidement sur les systèmes monoculaires, avant de détailler le capteur-étalon de la vision 3-D : la stéréovision. Sa description nous permet d'introduire tout naturellement le concept de vision en lumière structurée : il s'agit en fait d'un capteur stéréoscopique pour lequel une caméra a été remplacée par une source lumineuse. Puisque ce qui nous intéresse principalement ici, c'est la lumière structurée dite codée (projection d'un patron bi-dimensionnel portant un code permettant de résoudre le problème de la mise en correspondance), nous nous arrêtons sur les différentes techniques de codage et nous rappelons une classification de ces codages. Dans la dernière partie de ce chapitre, nous présentons les principales applications qu'a trouvé la vision en lumière structurée, en robotique terrestre, extra-terrestre, sous-marine, en métrologie et en imagerie médicale.
Outre une présentation générale de ce qu'est la vision par ordinateur et un état de l'art détaillé sur la lumière structurée et codée, nous avons la volonté d'exposer, au sein de ce chapitre, les problèmes ouverts par la vision en lumière structurée dans le cadre de son utilisation en robotique mobile. C'est dans ce but que nous présentons les applications développées jusque-là pour ce type de capteur, en essayant d'en dégager les points encore inexplorés et en soulevant les problèmes pratiques rencontrés.

1.2 La Vision 3-D

1.2.1 La Vision Monoculaire

La vision monoculaire ne permet pas, *a priori*, de restituer la profondeur de la scène observée. Elle fournit une seule représentation bi-dimensionnelle, l’image, de l’espace tri-dimensionnel, ce qui n'est pas suffisant, on le sait, pour déterminer les distances et la profondeur. L'idée, pour recouvrer cette information à partir

d'une seule caméra, est d'utiliser le mouvement de celle-ci (on se ramène alors, sous certaines conditions, à un système binoculaire) ou les mouvements des objets qui peuplent la scène. D'autres techniques permettent d'obtenir une information sur la profondeur par l'analyse des textures ou des ombres.

La vision dynamique

Cette technique utilise le mouvement des objets afin de les détecter et de les identifier ; elle est généralement appliquée pour des scènes routières où l'objectif est de détecter des véhicules, de les suivre, etc. [SMI93].
A partir des variations spatio-temporelles de l'intensité lumineuse, appelées *flot optique*, on va approximer la projection du champ de vecteurs vitesse 3-D sur le plan image. Cette projection est couramment appelée *champ de vecteurs vitesse image* ou *mouvement apparent* [LUO88], [ALO90].
L'utilisation du flot optique permet une estimation des composantes tri-dimensionnelles du mouvement, malheureusement, les problèmes de mise en correspondance et la sensibilité au bruit en font un outil dont l'application est complexe.

La vision active

Il existe des mécanismes, chez les mammifères, permettant d'appréhender un objet de différentes manières, sous différents points de vue, pour mieux l'observer et le reconnaître. On peut tourner autour, ou le faire pivoter, on peut le rapprocher ou s'en éloigner, on peut encore concentrer son acuité ou au contrainte la relacher. A cela s'ajoute une auto-adaptation innée de l'oeil qui fait que l'iris fixera son ouverture de manière à recevoir une intensité lumineuse suffisante, ou que la courbure du cristallin permettra une mise au point précise sur la rétine. La vision active, dans son principe, tend à reproduire ces phénomènes d'adaptation et de contrôle. En pratique, elle consiste à agir sur les paramètres de la caméras (focale, ouverture, position, gain, etc.) pour parvenir à une image de meilleure qualité, ou pour se fixer un point de vue plus pertinent.
La méthode consistant à déplacer la caméra afin d'obtenir des vues multiples d'une scène sous des angles différents est probablement la plus utilisée [CRO90]. En connaissant les paramètres du déplacement entre chaque acquisition d'image, il est possible de calculer la structure tri-dimensionnelle de la scène, comme on le ferait en stéréovision : cette technique est appelée *shape from motion*.
Il est également possible, sans déplacer la caméra, d'obtenir différentes images d'une même scène en jouant sur le zoom ou sur la mise au points de la caméra. On parle alors *de depth from zooming* ou de d*epth from focus* [KRO86], [NAY94].

L'analyse de texture

Dans un environnement fortement texturé, il est possible d'estimer des paramètres des surfaces comme leur orientation ou leur distance à la caméra à partir de l'analyse des textures. Il existe un lien entre l'apparence de l'élément textural, son orientation, sa taille, sa déformation et la configuration des surfaces sur lesquelles il repose ; le gradient de densité de la texture contient notamment une information riche sur le positionnement et l'éloignement relatif des surfaces les unes par rapport aux autres [GIB50]. Cette technique est baptisée le *shape from texture* [WIT81], [GAR93].
Si l'environnement est composé de surfaces peu texturées, une solution consiste à appliquer artificiellement une texture par projection d'un patron lumineux (nous verrons dans les sections suivantes, des exemples de ces patrons).

L'analyse de l'ombre

Le niveau de gris d'un pixel dans une image représente la brillance de la scène en ce point. Cette brillance dépend de trois facteurs : l'intensité de l'illumination incidente, les propriétés de réflectance de la surface et son orientation spatiale [KOE80]. Connaissant la brillance en un point, on veut retrouver l'intensité de l'illumination et l'orientation de la surface. C'est un problème hautement indéterminé puisque le nombre d'inconnus est beaucoup plus élevé que le nombre de données. Cependant grâce à des hypothèses très contraignantes, on peut obtenir une information partielle sur l'orientation de la surface à partir des variances de brillance dans l'image de cette surface. On appelle cette technique le *shape from shading* [LEE85], [HOR89].

1.2.2 La Vision Omnidirectionnelle

Un champ de vision large, pouvant atteindre trois cent soixante degrés, a paru utile à nombre de chercheurs en vision par ordinateur qui ont concentré leurs efforts sur les capteurs, et les algorithmes y afférents, offrant cette possibilité. On distingue, en vision omnidirectionnelle, les capteurs en rotation et les capteurs catadioptriques (une caméra associée à un miroir). En cohérence avec le thème donné à cette section, nous ne présentons ici que les applications ayant trait à la vision 3-D et à ses outils, calibration et reconstruction.

Capteur en rotation

Pour cette approche, la caméra CCD qui observe l'environnement est en rotation autour d'un axe. Par juxtaposition des images ainsi obtenues, on peut directement construire une vue panoramique de la scène [ISH92]. Chacune de ces images est du type perspective, directionnel, et peut donc être traitée conventionnellement. Le temps requis pour traiter l'ensemble des images, le flou provoqué par le

mouvement de rotation de la caméra et le simple fait de devoir déplacer la caméra pour obtenir la vue omnidirectionnelle, sont les inconvénients majeurs de cette approche.
Des évolutions sont bien sûr venues compléter ce schéma de base du capteur en rotation. L'utilisation de deux caméras linéaires superposées, offrant l'avantage d'une meilleure résolution, a été proposée [BEN96], [BEN97]. Une procédure de calibration originale y est décrite, où les distorsions sont prises en compte. L'utilisation de deux caméras linéaires en mouvement permet d'opérer sur des vues panoramiques comme s'il s'agissait d'images stéréoscopiques classiques.

Capteur catadioptrique

L'obtention d'une image omnidirectionnelle peut s'effectuer, et c'est la seconde approche, en une seule prise de vue. On adjoint pour cela un miroir, sphérique, conique, parabolique ou hyperbolique, à la caméra CCD. Cette fois, il est impératif de redresser les images anamorphosées en images perspectives si l'on désire obtenir une vue panoramique. Notons qu'il n'est pas toujours possible, selon la forme du miroir et notamment avec un miroir conique, de redresser ces images.
Les techniques de calibration doivent prendre en compte la réflexion de la ligne de vue sur la surface du miroir ; à titre d'exemple, on peut se référer à [CAU99], [CAU00] pour la calibration d'un capteur conique. On peut utiliser ces capteurs stéréoscopiquement, en prenant deux vues omnidirectionnelles de la scène [BUN01]. On peut également, comme dans l'article cité, y adapter la géométrie épipolaire.

1.2.3 La Stéréovision

La stéréovision est le procédé le plus proche des mécanismes de reconstitution de la profondeur de la vision humaine. Elle consiste à observer une même scène à l'aide d'au moins deux caméras légèrement décalées l'une de l'autre [AYA89]. Typiquement, lorsque deux caméras sont utilisées, on parle de système binoculaire ; quand trois caméras sont utilisées, on parle de système trinoculaire [BOR91] ; il est très rare qu'un système stéréoscopique se compose de quatre caméras ou plus.
Connaissant la géométrie exacte du système, estimée dans l'étape de *calibration*, et après avoir identifier les points dans chacune des images correspondant au même point de l'espace (cette étape, primordiale, est appelée *mise en correspondance*), il est possible de reconstruire l'espace tri-dimensionnel observé, par triangulation (Figure 1).

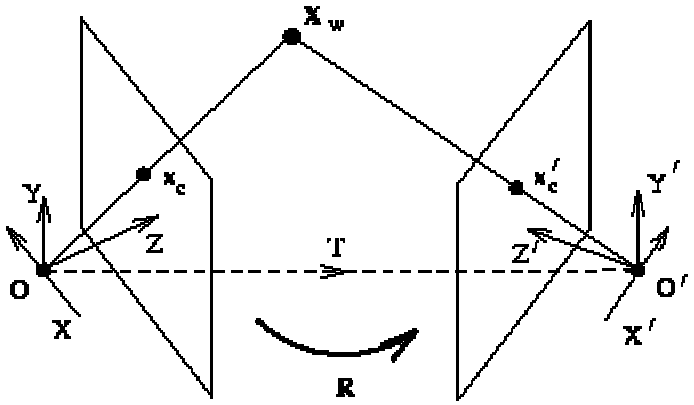


Figure 1. Géométrie d'un système binoculaire

Nous détaillerons, dans la suite de ce mémoire, la modélisation et la calibration d'une caméra ; arrêtons-nous sur le délicat problème de la mise en correspondance. Prenons le cas typique d'un système binoculaire. Il s'agit d'apparier les points homologues de la première et de la seconde image. Puisqu'ils représentent tous les deux des images d'un même point, des ressemblances détectables doivent permettre de les identifier. Si ce problème peut paraître simple à première vue, il est en fait l'un des plus complexes en vision par ordinateur et ce pour diverses raisons :

- Puisque les caméras sont décalées, les deux images de la scène peuvent présenter de grandes différences aussi bien en termes de morphologie que d'illumination.
- Un point de l'une des images peut se retrouver sans homologue dans l'autre image du fait que les deux champs de vision ne se recouvrent pas complètement (Figure 2) ou par la faute d'un objet provoquant une occlusion pour une caméra (Figure 3).

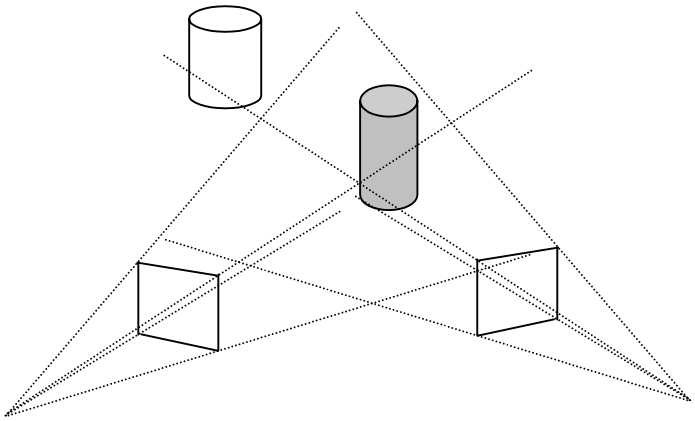


Figure 2. Recouvrement de champs : seuls les points de l'objet grisé peuvent être traités

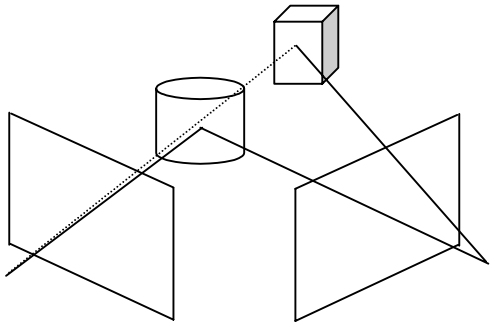


Figure 3. Occlusion : le cube n'est pas visible pour la caméra de gauche

- Dans une scène faiblement texturée, il est difficile de répérer des indices visuelles permettant d'effectuer la mise en correspondance. Une surface plane, éclairée de manière homogène, pourra difficilement être traîtée par les algorithmes de mise en correspondance.
- Plusieurs objets de forme et de taille similaires peuvent apparaître dans les images.

En général, les points d'intérêts considérés sont les arêtes de contour ou les coins. Des contraintes sont utilisées pour réduire l'espace de recherche. La contrainte la plus célèbre est certainement la contrainte épipolaire, que nous verrons dans le chapitre 3, elle permet de réduire l'espace de recherche du plan de l'image à une droite de l'image. Les autres contraintes sont la contrainte d'orientation, d'ordre, d'unicité et de disparité.

1.2.4 La Vision en Lumière Structurée

Une lumière structurée est une source lumineuse modulée dans l'espace, le temps, l'intensité et/ou la couleur. La vision en lumière structurée consiste à projeter une telle source lumineuse sur l'environnement, lui-même observé par une ou plusieurs caméras CCD ; il s'agit en fait de remplacer, dans un système stéréoscopique classique, une caméra par un projecteur de lumière. Dans sa forme la plus simple, la lumière structurée est un faisceau laser projetant un *point de surbrillance* sur l'environnement (voir Figure 4). Puisqu'un seul point est projeté, la mise en correspondance est immédiate. En revanche, un balayage de la scène suivant les deux axes *x* et *y* est nécessaire si l'on désire obtenir une mesure globale de celle-ci. Une deuxième solution consiste à projeter un *plan de lumière* formant un segment sur les objets illuminés (voir Figure 5). Le plan de lumière est obtenu en déviant un faisceau laser à l'aide d'une lentille cylindrique. Le balayage suivant l'axe horizontal demeure nécessaire. Cette technique fut utilisée par Shirai en Suwa en 1971 pour la reconnaissance d'objets polyhédriques [SHI71]. Plus près de nous, Strauss [STRA92] propose un système de perception tri-dimensionnel basé sur ce capteur. Cette fois encore, la mise en correspondance est univoque. On peut facilement démontrer que pour un couple de points appariés, seuls trois coordonnées sont nécessaires à la triangulation [BAT98]. De manière plus intuitive, on sait que l'intersection d'un plan (ici, de lumière) et d'une droite (la ligne de vue) donne un point.
Pour éviter toute action mécanique, coûteuse en temps, rendue nécessaire par le balayage, des motifs bi-dimensionnels (ou *patrons* ou *motifs structurants*) couvrant le champ de vision de la caméra, ont été proposés : faisceau de lignes (Figure 6), grille (Figure 7) ou matrice de points. Les premiers travaux en ce domaine datent de 1971, et sont l'œuvre de Will et Pennington [WILL71] qui baptisèrent leur technique *grid coding* ; citons également le travail de Hu, Jain et Stockman [HU86] et de Stockman et Hu [STO86]. L'inconvénient de ces systèmes vient du fait que la mise en correspondance n'est plus univoque. En effet, dans le cas d'une grille par exemple, puisque rien ne permet de distinguer dans l'image un nœud d'un autre nœud, il est difficile de trouver un unique homologue aux points capturés par la caméra. On peut certes réduire la complexité en prenant en compte les contraintes topologiques énoncées par Hu et Stockman [HU89] :

- Deux objets ne peuvent occuper la même place dans l'espace.
- Une continuité dans l'image correspond à une continuité dans l'espace physique.
- La disparité est limitée.
- Les objets sont uniques et leur position par rapport au capteur est connue approximativement.
- Les points consécutifs dans une grille conservent leurs relations topologiques (ordre, voisinage, etc.)

Ces contraintes n'empêchent cependant pas la mise en correspondance de rester tributaire de la segmentation et d'être peu robuste en présence d'occlusions ou de points sans correspondants ; toutes les ambiguïtés ne sont pas levées ; enfin, la limitation de la disparité réduit considérablement la résolution du capteur. On sait également que l'analyse des déformations du patron *perçu* par rapport au patron *projeté* permet, dans une stratégie de type *shape from texture*, de déduire des informations de forme sur la scène observée ; ce n'est pas l'option retenue ici.
Pour dépasser ces problèmes de mise en correspondance, des techniques de codage de la lumière structurée ont été développées au cours de ces vingt dernières années. Une revue exhaustive et une classification de ces différentes techniques a paru voici quelques années [BAT98]. Nous en présentons quelques exemples significatifs dans la section suivante.

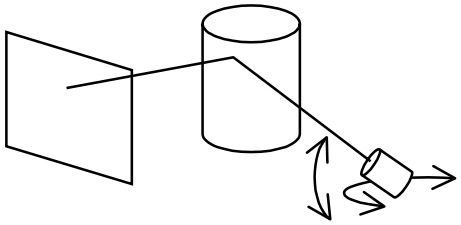


Figure 4. Point de surbrillance

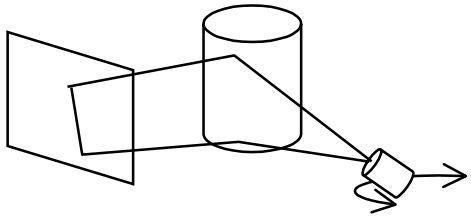


Figure 5. Plan de lumière

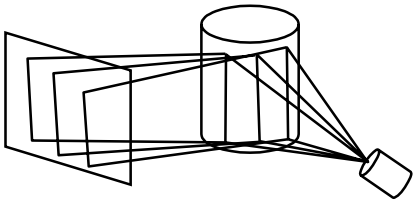


Figure 6. Faisceau de lignes

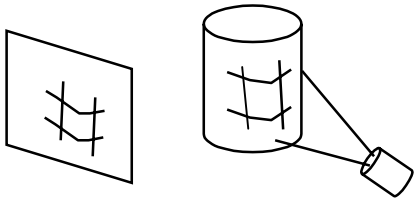


Figure 7. Grille de lumière

1.2.5 Une Classification des Codages

La lumière structurée et codée est une technique basée sur la projection de patrons de lumière. Le codage du patron permet de résoudre de manière efficace le problème de la mise en correspondance. Chaque élément (lignes, points, etc.) de la lumière projetée porte un code permettant de déterminer sa position au sein du patron. La reconnaissance de ce code dans l'image garantit une mise en correspondance unique. Nous avons déjà mentionné le fait que trois coordonnées sont suffisantes pour reconstruire un point ; en conséquence, le patron de lumière

peut être codé uniquement le long de ses lignes ou de ses colonnes, étant bien entendu qu'un codage le long des deux axes garantit une codification plus robuste. Battle, Mouaddib et Salvi [BAT98] ont proposé une classification des différentes techniques de codage basée sur trois critères. Le premier est lié à la *dépendance temporelle* du codage :

- **Statique** : la projection du patron est limitée à des scènes statiques. Cette contrainte apparaît quand plusieurs patrons doivent être projetés sur la scène (le code est une résultante des valeurs obtenues sous les différentes illuminations). Un quelconque mouvement dans l'intervalle de temps séparant deux projections met inévitablement en échec la mise en correspondance.
- **Dynamique** : le codage ne nécessite qu'une seule projection et le patron peut être utilisé pour l'analyse de scènes mobiles.

Le deuxième est lié à la *nature* de la lumière émise :

- **Binaire** : chaque point du patron prend la valeur 0 ou 1 ; ces valeurs correspondent à l'opacité ou à la transparence du patron en ce point.
- **En niveaux de gris** : un niveau de gris est associé à chaque point du patron (il représente son degré d'opacité). Une prise d'image sans projection de lumière, permettant d'évaluer et ainsi d'atténuer l'effet de réflexion des surfaces, est souvent requise avec un tel codage. Les codages en niveaux de gris sont, de ce fait, des codages "statiques".
- **En couleurs** : une couleur est associée à chaque point du patron. Ce codage est plutôt réservé à des scènes composées d'objets de couleurs neutres : des surfaces de couleurs fortement saturées étant susceptibles d'absorber les couleurs projetées.

Le troisième est lié à la *dépendance aux discontinuités en profondeur* :

- **Périodique** : les codes sont répétés périodiquement tout au long du patron. Cette technique permet de réduire le nombre de bits de codification. Cependant, les discontinuités des surfaces doivent être de taille supérieures à la moitié de la période de codage, par respect du théorème de Shannon.
- **Absolu** : le code est unique pour chaque élément du patron.

1.2.6 Etat de l'Art

Posdamer et Altschuler

Le patron lumineux proposé par Posdamer et Altschuler [POS82] est une matrice de $n \times n$ points dont chaque colonne peut être indépendamment illuminée (Figure 8). Ce système donne la possibilité de créer plusieurs motifs et la projection successive de tous ces motifs fournit un code en chacun des points (c'est un codage *temporel* réservé à l'analyse de scènes *statiques* qui forme une séquence *binaire*). Le nombre de motifs à projeter est donné par le nombre de colonnes à coder. Le premier motif projeté est la matrice dont tous les points sont illuminés.

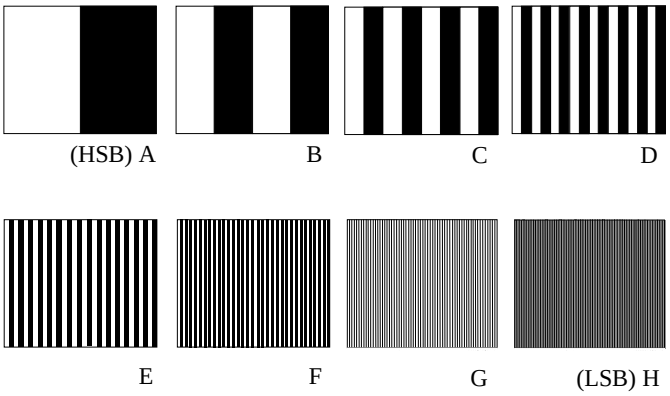


Figure 8. Posdamer et Altschuler, 1982

Pour rendre ce système exploitable sur des scènes en mouvement, Altschuler, Atschuler et Taboada [ALT81] proposent de projeter simultanément chaque motif sur des fréquences différentes ; les *n* motifs sont observés par *n* caméras auxquelles on a adjoint un filtre optique. Du fait de l'encombrement d'un tel système, tous les points ne peuvent être illuminés par tous les motifs et observés par toutes les caméras à la fois.

Le Moigne et Waxman

Le patron proposé par Le Moigne et Waxman [LEM84] offre un codage absolu suivant les deux axes, adapté aux scènes dynamiques, à partir d'une émission binaire de la lumière sous forme de grille. Certaines mailles de la grille sont opaques et servent de balises, de points de repères pour le décodage (Figure 9).

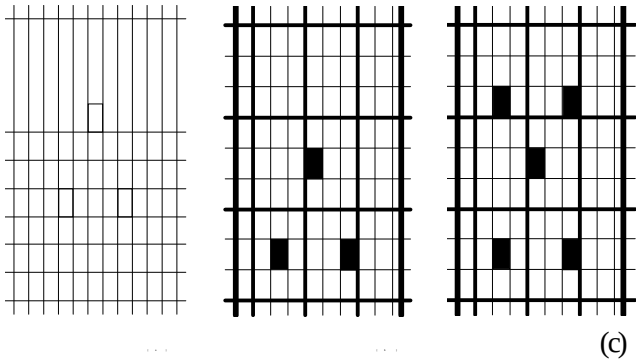


Figure 9. Le Moigne et Waxman, 1984

Dans la configuration proposée par les auteurs, le projecteur est déplacé par rapport à la caméra le long de l'axe *y*. De ce fait, les lignes verticales apparaissent dans l'image avec un minimum de déformation. Ces verticales sont utilisées comme guides pour détecter les mailles opaques. Pour chaque balise détectée, deux différents codages, ou étiquetages, sont calculés : l'un partant du côté droit de la balise, l'autre du côté gauche. Une fusion des différents étiquetages est ensuite effectuée pour obtenir un codage global du motif. Ce procédé permet un codage plus robuste et plus fiable du patron. Reste que si une ou plusieurs balises de la grille ne sont pas visibles dans l'image, le décodage devient difficile, voire impossible.

Carrihill et Hummel

Carrihill et Hummel [CAR85] proposent de projeter un degradé constant de niveaux de gris sur la scène à mesurer (Figure 10). Pour atténuer l'effet de réflectance des surfaces, une prise d'image sous illumination constante (la scène est illuminée par un patron d'égale opacité sur toute sa surface) est effectuée. Le décodage des colonnes est obtenu par soustraction des deux images capturées. Le codage de Carrihill et Hummel est absolu, en niveaux de gris et réservé aux scènes statiques.

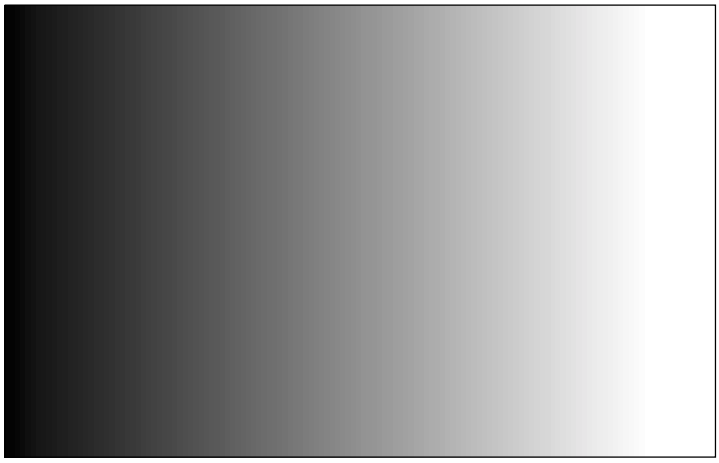


Figure 10. Carrihill et Hummel , 1985

Chazan et Kiryati [CHA95] ont proposé une méthode permettant d'augmenter la précision des mesures. Le patron de Carrihill et Hummel est utilisé en première approximation. Il est divisé en deux patrons similaires dans la deuxième étape ; en quatre, dans la troisième ; etc.

Maruyama et Abe

Le patron proposé par Maruyama et Abe [MAR93] est composé de lignes verticales, aléatoirement coupées, de manière à former une succession de

segments de tailles variables sur chacune d'elles (Figure 11). Une taille standard L_0 est définie pour les segments et chacun d'eux est alongé ou raccourci d'une tolérance aléatoire Δ. L'appariement des segments est effectué à partir de la mise en correspondance de leurs extrémités le long des droites épipolaires ; la coupe étant aléatoire, il est possible que plusieurs segments aient le même homologue dans l'image (Figure 12). C'est alors le voisinage des segments équivoques qui permettra de lever l'ambiguïté. Les auteurs utilisent un algorithme de croissance de régions à cette fin.

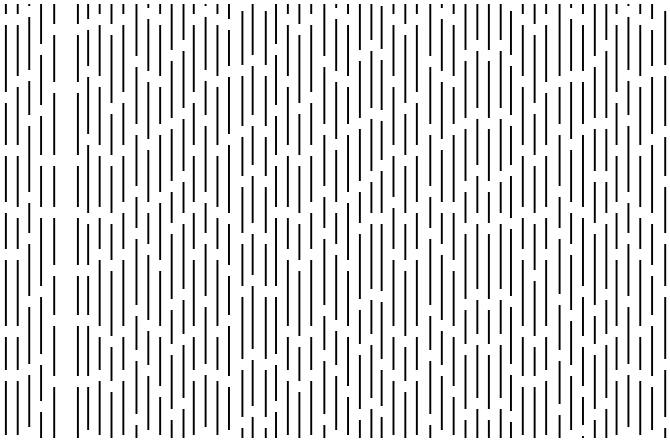


Figure 11. Maruyama et Abe, 1993

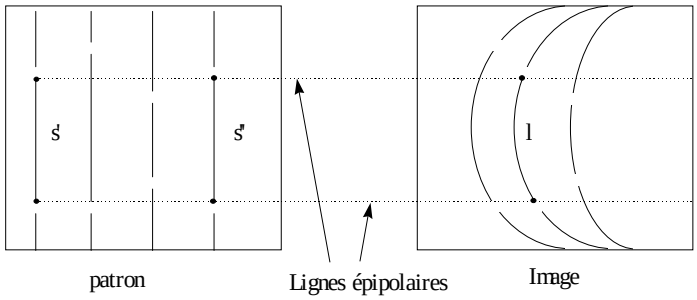


Figure 12. Un exemple d'appariement multiple

Morita, Yakima et Sakata

Comme nous l'avons mentionné au début de ce chapitre, il est possible de projeter une matrice de points plutôt que des lignes. Morita, Yakima et Sakata utilisent ce procédé [MOR88]. Le codage du patron de lumière qu'ils proposent est tel que chaque sous-patron est unique au sein d'une période de codification. Pour obtenir les coordonnées de chaque point dans l'image, la projection d'un patron

entièrement illuminé est nécessaire (Figure 13) : ce codage est réservé aux scènes statiques. En déplaçant le projecteur uniquement le long de l'axe *x* par rapport à la caméra, l'identification des points est simplifiée puisque leur position sur le plan image n'est décalée qu'horizontalement. Malheureusement, ce système souffre de plusieurs inconvénients : non-détection d'un point causée par une occlusion, déplacement horizontal d'un point si deux objets présentent un écart de profondeur trop important, permutation de deux points dans certains cas d'occlusion. Tous ces inconvénients rendent la mise en correspondance très délicate.

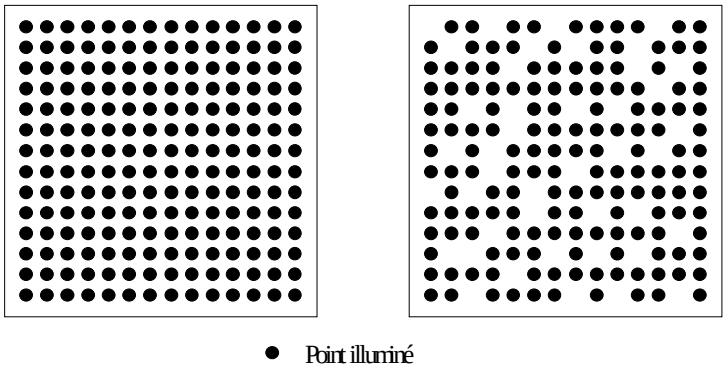


Figure 13. Morita, Yakima et Sakata, 1988

Vuylsteke et Oosterlinck

Le patron développé par Vuylsteke et Oosterlinck [VUY90] permet la mesure de scènes dynamiques ; il est basé sur un codage binaire de la lumière le long des colonnes du motif. Basiquement, il possède une structure en damier au sein de laquelle les carrés opaques et transparents s'alternent. A cette structure s'intercalent des points noirs ou blancs au sommet des carrés (Figure 14) de manière à moduler l'apparence et donc le codage de chaque colonne.

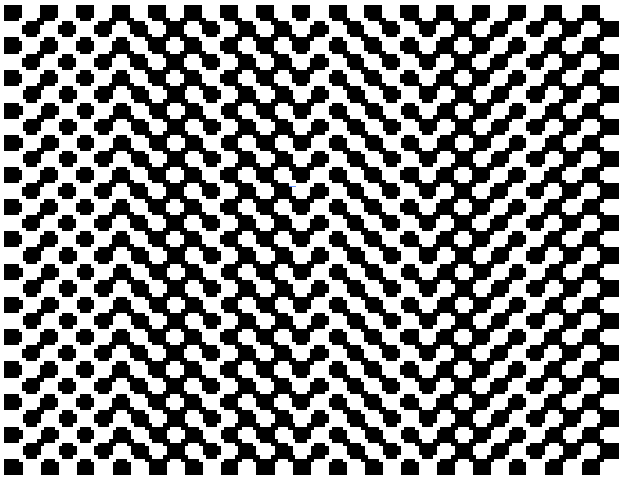


Figure 14. Vuylsteke et Oosterlinck, 1990

L'alphabet ainsi obtenu se compose de quatre lettres selon que, pour un voisinage 2×2, la première case en haut à gauche est noire (+) ou blanche (-) et que les points qui s'y intercalent sont eux aussi noirs (0) ou blancs (1). La figure suivante montre les quatre configurations possibles :

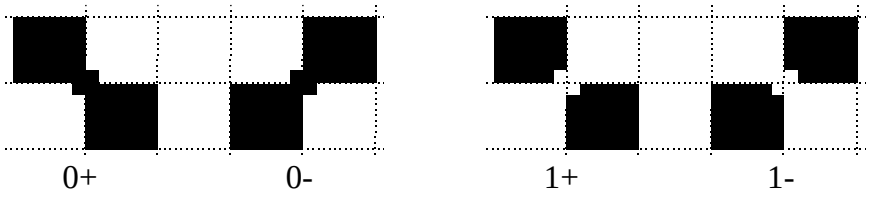


Figure 15. Détail du codage de Vuylsteke et Oosterlinck

Boyer et Kak

En 1987, Boyer et Kak [BOY87] proposent un motif lumineux codé par la couleur. Il s'agit d'une séquence de lignes verticales colorées en rouge, vert ou bleu (Figure 16). Notons qu'une variante de ce motif, intercalant un espace opaque entre chaque ligne, a également été proposée par les auteurs ; la couleur blanche peut être adjointe comme quatrième primitive de codage. Comme pour tous les motifs colorés, son utilisation est principalement réservée à la mesure d'objets de couleur neutre.

Pour un patron composé de *n* lignes verticales, les auteurs proposent de le diviser en *m* sous-patrons composés chacun de *k* lignes. L'une des difficultés provient du fait qu'il n'existe aucun moyen de savoir où un sous-patron commence et où il

finit, ce qui rend la mise en correspondance délicate en présence d'occlusion ou de lignes projetées en dehors du champ de vision de la caméra.

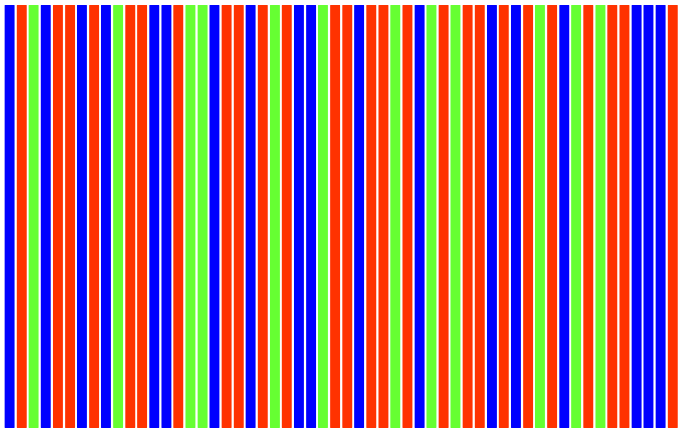


Figure 16. Boyer et Kak, 1987

Tajima et Iwakawa

Le système proposé par Tajima et Iwakawa [TAJ90] est lui aussi basé sur un codage par la couleur des lignes verticales du motif. Cependant, chaque ligne est projetée avec une longueur d'onde différente, le motif apparaissant alors comme un dégradé de couleurs allant du bleu au rouge, à la manière d'un arc-en-ciel (Figure 17). La scène est observée par une caméra monochromatique et non pas par une caméra couleur ; deux images de la scène sont capturées avec chacune un filtre différent. Les auteurs ont démontré que la relation entre les deux images ne dépendait ni de la couleur des objets, ni de l'illumination mais uniquement de la longueur d'onde utilisée lors de la projection. Puisque deux images sont requises, cette méthode est réservée à la mesure de scènes statiques, bien entendu, rien n'empêche d'utiliser ce même patron, observé par une caméra couleur, pour la mesure de scènes en mouvement.

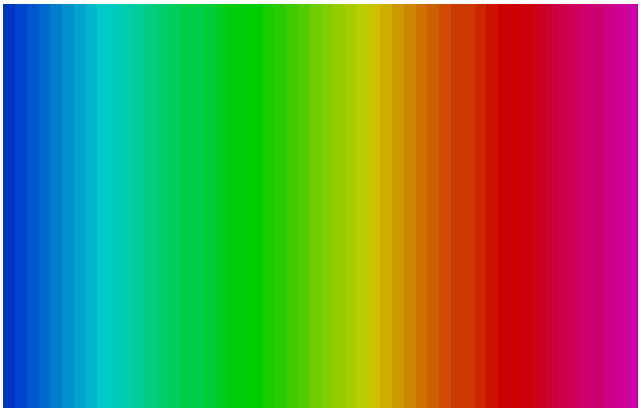


Figure 17. Tajima et Iwakawa, 1990

Wust et Capson

En 1991, Wust et Capson [WUS91] proposent un patron coloré formé d'une juxtaposition de motifs rappelant celui de Tajima et Iwakawa (Figure 18). Pratiquement, il est formé par le mélange des trois mêmes primitives de couleur (rouge, vert, bleu), mais projetées à des intensités différentes. L'intensité de chaque composante évolue sinusoïdalement le long du motif et chaque sinusoïde présente un décalage de phase en quadrature par rapport à la précédente. Le vert est décalé de 90° par rapport au rouge, et le bleu est décalé de 90° par rapport au vert (Figure 19).

Figure 18. Wust et Capson, 1991

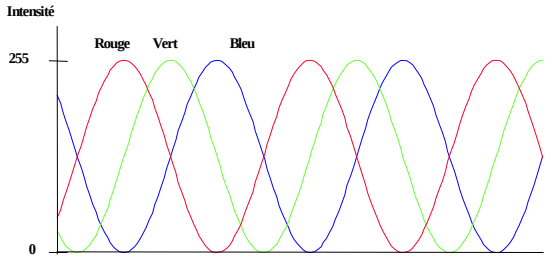


Figure 19. Les sinusoïdes de couleur composant le patron

L'information de profondeur est obtenue directement par l'analyse du décalage de phase, dans une technique proche de celle utilisée dans les méthodes de moiré. Le décodage de la couleur pour l'obtention de la coordonnée-colonne du point projeté n'est donc pas nécessaire. Le codage périodique limite l'utilisation du motif à des surfaces dont les discontinuités sont au plus égales à une demi-période. Hung [HUN93] a proposé un patron similaire, dont le codage est cependant issu d'une sinusoïde de niveaux de gris.

Griffin, Narasimhan et Yee

Griffin, Narasimhan et Yee [GRI92] proposent également un patron de couleur. Il s'agit cette fois-ci de la projection d'une matrice de points. Le codage d'un point est donné par sa couleur et par la couleur de ses quatre voisins (nord, sud, est et ouest). Chaque code est unique et le nombre de primitives de couleur est limité (Figure 20). La théorie des graphes fournit aux auteurs les outils pour assurer l'unicité du codage et pour, à partir d'un nombre de primitives fixé, déduire la taille maximale que pourra avoir la matrice.

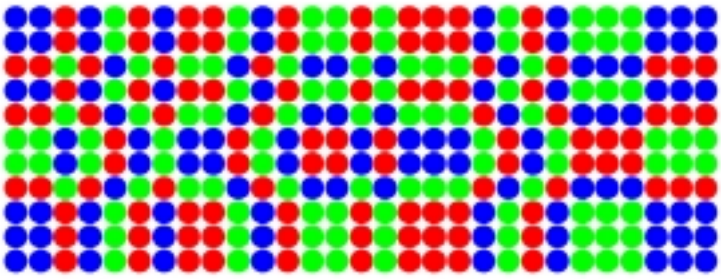


Figure 20. Griffin, Narasimhan et Yee, 1992

Salvi, Batlle et Mouaddib

Salvi, Batlle et Mouaddib [SAL98] ont proposé une grille composée de lignes horizontales et verticales régulièrement espacées et codée suivant les deux axes. Le codage est absolu et s'obtient à partir d'une unique projection, ce qui le rend utilisable pour la mesure de scènes dynamiques. Chaque ligne du motif est colorée

de manière à ce que le triplet qu'elle forme avec ses deux plus proches voisins n'apparaisse qu'une fois dans l'ensemble du motif. Le codage des lignes horizontales et verticales repose sur des primitives de couleur différentes, ce qui permet de les segmenter indépendamment. Le rouge, le vert et le bleu servent à colorer les horizontales ; le cyan, le magenta et le jaune, les verticales. Les primitives ont été choisies bien espacées dans le cône HSI pour faciliter leur discrimination.
Pour augmenter le nombre de points utiles du patron, il suffit d'incrémenter le nombre de primitives de couleur, au prix, bien sûr, d'un décodage plus malaisé. En utilisant six primitives, la taille maximale de la grille est de 29×29.

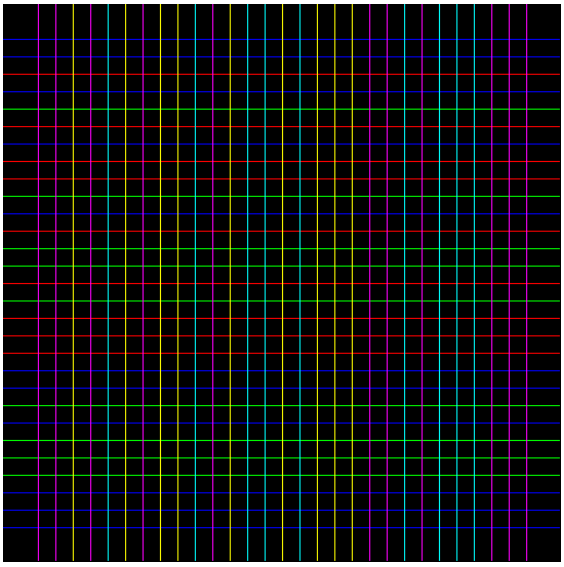


Figure 21. Salvi, Batlle et Mouaddib, 1997

1.3 Les Applications de la Vision en Lumière Structurée

De nombreuses applications, technologiques, industrielles, médicales, profitent des avantages de la vision en lumière structurée : résolution du problème de mise en correspondance, structuration artificielle des surfaces observées, faible coût, etc. Nous allons décrire quelques-unes de ces applications en soulignant les apports de la lumière structurée, les avantages qu'elle procure par rapport à la stéréovision, en montrant aussi les problèmes qu'elle génère et les problématiques de recherche qu'elle soulève. Nous avons choisi trois catégories représentatives des classes de problèmes générés par les systèmes de vision : la précision, la dynamique et l'absence de texture.

1.3.1 La Profilométrie

La profilométrie est la technique permettant la caractérisation de la géométrie d'une surface par mesure de son profil. La caractérisation de la géométrie et micro-géométrie d'une surface est, d'après les normes nationales et internationales, ramenée à celle de ses *profils* représentés par des *profilogrammes*. On admet de classer conventionnellement sous quatre numéros d'ordre les écarts géométriques de profil (Table 2).

Ordre	**Ecart géométrique**	
1	Ecart de forme (flèche)	Géométrie
2	Ondulation	
3	Strie, sillon	Micro-géométrie
4	Arrachement, piqûre, fente, etc.	

Table 2. Classement des écarts géométriques

A partir des profilogrammes, on peut calculer un certain nombre de paramètres qui seront choisis selon la fonction de surface. Le calcul des paramètres est effectué à partir de la ligne moyenne du profil redressé.
Longtemps, seuls les procédés mécaniques tels que les palpeurs étaient en mesure de fournir une telle information ; la lenteur du procédé, la connaissance différée et lacunaire des mesures sont autant de désavantages pour un monde industriel désireux d'allier productivité et contrôle. En ce sens, l'apparition des techniques de vision, qui permettent de surcroit une mesure sans contact, présentent l'avantage d'une automatisation complète, d'une précision et d'une fiabilité accrues. Garcia, Garcia, Obeso et Fernandez [GAR95] ont proposé un système de mesure des déformations des plaques métalliques utilisant une source laser et une caméra linéaire CCD. Puisque l'on observe une surface homogène, non-texturée, l'utilisation d'un capteur stéréoscopique est proscrite : la mise en correspondance serait inexorablement mise en échec.

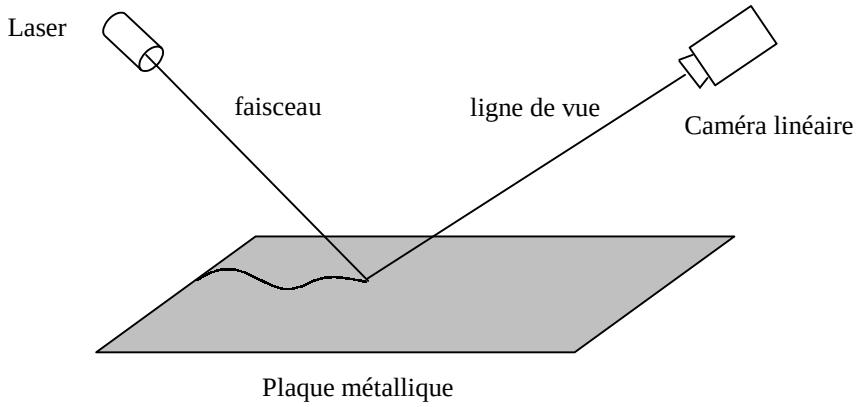


Figure 22. Profilométrie laser

La courbure de la surface en un point devie le faisceau laser selon un seul axe. C'est pourquoi une caméra linéaire est suffisante et présente même l'avantage d'une résolution très fine (on trouve couramment des caméras linéaires de 1000, 2000, 5000 pixels). En contrepartie, la position de la source laser et de la caméra doit être ajustée très précisément pour que la ligne de vue atteigne la barrette CCD de la caméra. La calibration forte du capteur permet de calculer l'information métrique désirée à partir de la position du faisceau laser dans l'image linéaire. La position du faisceau est obtenue par la détection du pic de l'histogramme.

Une autre application à la profilométrie a été développée, cette fois-ci par notre laboratoire [MOU96], pour la mesure des défauts de planéité du verre : ondulations (de l'ordre du dixième de millimètre en amplitude) et flèche. La transparence du verre et les effets spéculaires qu'il peut présenter empêchent l'utilisation d'une source laser. Pis, l'image qui se forme à la surface du verre est d'intensité trop faible pour être perçue par la caméra. Un dispositif de mesure en réflexion a donc été élaboré. Une motif structurant binaire, présentant une partie opaque et une partie transparente, est projeté sur la surface du verre et réfléchi sur un écran en verre dépoli. C'est cet écran qui est filmé par la caméra. Les avantages sont, outre d'obtenir un signal exploitable, d'utiliser l'effet bras de levier généré par la réflexion sur le verre pour amplifier les déviations du faisceau. En revanche, l'hypothèse posant que les déviations sont uniquement provoquées par la pente du verre au point mesuré et pas par sa distance au projecteur doit être prise pour pouvoir résoudre le problème. Cette approximation est similaire au modèle de projection perspective faible présenté en annexe A.

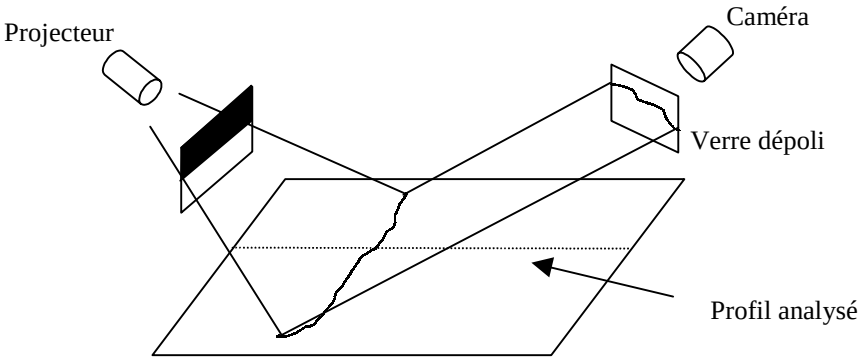


Figure 23. Mesure des déformations du verre

Une forte perte d'énergie lumineuse peut être constatée entre la projection et la capture. Pour garantir un signal exploitable, il est préférable d'enfermer le dispositif dans une chambre noire.

1.3.2 L'Imagerie Médicale

Pour les mesures anthropométriques, où les surfaces sont là encore très peu texturées (épiderme, os), on a souvent recours à la vision en lumière structurée. Sotoca, Buendia et Iñesta [SOT00] ont proposé un système de mesure des déformations pathologiques du dos. Une grille codée est projetée sur le dos des patients. Pour la mesure, une procédure originale de calibration est proposée ne nécessitant qu'une connaissance partielle de la géométrie du système (Figure 24).

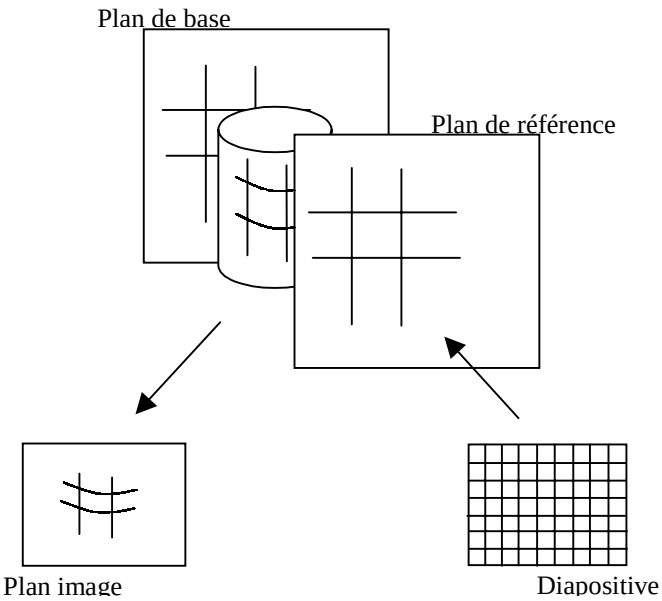


Figure 24. Procédure de calibration

Grâce à la connaissance de la position des points projetés sur un plan de base et un plan de référence, il est possible, par des calculs simples basés sur la similarité des triangles, d'obtenir une image de profondeur à partir des données images. Cette procédure est bien adaptée à la mesure d'objets de grande surface.
Silva, Paiva, Restivo, Campilho et Laranja Pontes [SIL99] ont appliqué la vision en lumière structurée à la mesure du volume mammaire pour la construction de prothèses. Le motif structurant qui illumine la scène est proche de celui proposé par Posdamer et Altschuler. La caméra et le projecteur sont calibrés et les points tri-dimensionnels estimés par intersection des plans de lumière et des lignes de vue. Pour calculer le volume mammaire, les points reconstruits sont regroupés trois à trois de manière à former un ensemble de triangles adjacents. Les sommets de ces triangles sont projetés sur un plan de référence et le volume des éléments prismatiques ainsi obtenus est additionné pour obtenir le volume total.

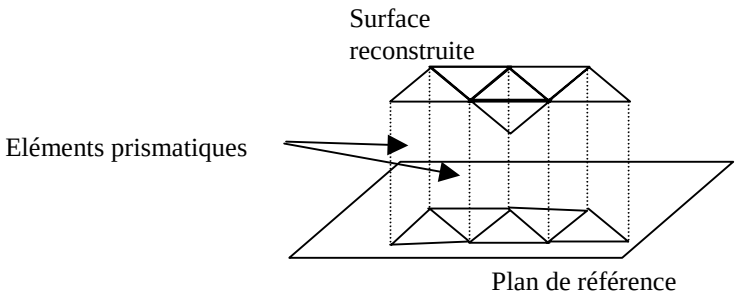


Figure 25. Calcul du volume

1.3.3 La Robotique Sous-Marine

Chantier, Clark et Umasuthan [CHA97A] ont présenté une méthode de calibration et de reconstruction adaptée aux contraintes de la vision sous-marine. Si l'on plonge le système de vision dans l'eau, il faut prévoir un dispositif étanche permettant de le protéger. La caméra et la source laser sont donc enfermées dans un caisson en plexiglass®. L'indice de réfraction de l'eau, du plexiglass et de l'air étant différents, les rayons lumineux seront déviés deux fois sur leur trajet (Figure 26).

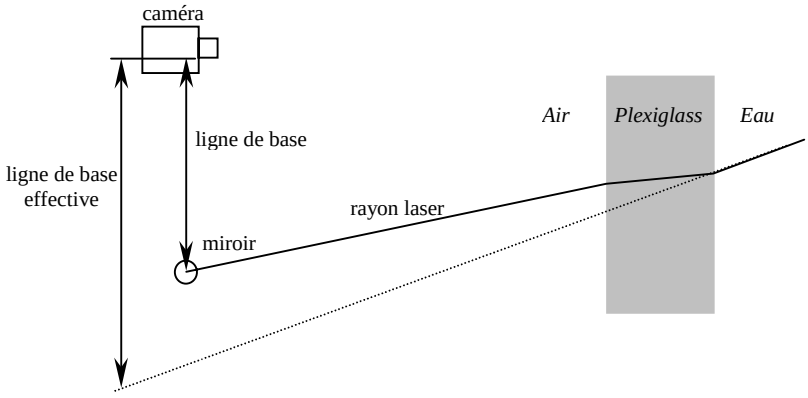


Figure 26. Dispositif de vision sub-aquatique

Appelons *ligne de base* la distance séparant la source laser de la caméra. Dans l'air, la ligne de base est constante et calculée une fois pour toute lors du processus de calibration. Ici, puisque la variation de l'indice de réfraction provoque une déviation des rayons lumineux, la ligne de base n'est plus constante et varie selon l'angle de balayage du faisceau. Ce nouveau degré de liberté doit être pris en compte lors de la calibration. Les effets de la turbidité sur les mesures sont également étudiés par les auteurs. On sait par exemple que dans un mileu sub-aquatique, le faisceau laser est soumis à la dispersion et à l'atténuation. Ces

paramètres sont modélisés et utilisés pour la définition des seuils pour la détection du faisceau dans l'image.
Forest, Salvi et Batlle [FOR00] ont également proposé l'utilisation d'un système de vision en lumière structurée pour des applications sous-marines. Le capteur, formé d'une caméra CCD et d'une source laser générant un plan de lumière rotatif, est monté sur un robot sous-marin et utilisé pour obtenir une image de profondeur de l'environnement. Le faisceau laser balaye la scène de manière à couvrir entièrement le champ de la caméra. Notons que la couleur du faisceau est verte et non pas rouge, comme on pourrait s'y attendre ; d'après les auteurs, cette couleur est moins affectée par l'effet de dispersion. Un codeur incrémental est utilisé pour connaître la position angulaire du plan de lumière. Une carte de traitement spécifique s'occupe de la segmentation de l'image : elle opère d'abord une conversion de l'espace RGB à l'espace HSI et permet l'extraction du trait laser par mémoire LUT. La synchronisation de la caméra et du laser est également gérée par cette carte. Une fois que l'intégralité de l'image, après différentes projections, est traitée, les données sont envoyées au calculateur pour la reconstruction tri-dimensionnelle. La caméra est modélisée et calibrée de manière classique. Huit paramètres sont nécessaires à la modélisation du plan de lumière : la position X, Y, Z du repère de la source lumineuse par rapport au repère monde, l'angle de rotation α par rapport à l'axe z du laser, l'angle β séparant le plan de lumière à l'axe z, les angles θ et γ de site et d'azimut, l'angle de biais δ corrigeant le désalignement mécanique de l'axe x du laser de l'origine du codeur incrémental.

L'obscurité des fonds marins fait que la vision en lumière structurée, en portant sa propre illumination, est particulièrement bien adaptée à son observation. Elle semble également plus robuste à la turbidité. Nous avons présenté deux exemples de vision sous-marine. Il existe bien entendu beaucoup plus de publications traitant du sujet ; citons l'étude de la propagation sous-marine d'un faisceau laser par Arnush [ARN72] et les travaux de Fox [FOX88] sur la formation d'images en lumière structurée dans un milieu turbide.

1.3.4 La Robotique Terrestre et Extra-Terrestre

La vision en lumière structurée trouve de nombreuses applications en robotique mobile. Krotkov l'a utilisée pour un robot marcheur [KRO91], l'enjeu étant la calibration du capteur pour permettre au robot de positionner convenablement ses pieds dans une stratégie de coordination oeil-jambe. Le problème est ainsi défini : comment identifier le déplacement reliant le repère attaché au robot à celui attaché à la source laser ? Des pastilles sont appliquées sur la jambe du robot et celle-ci est déplacée en différentes positions. Deux images sont nécessaires pour localiser les pastilles dans l'image et évaluer leur position : une image de profondeur et une image de réflectance. Nous n'allons pas détailler ici le processus de traitement des images et de calibration, retenons que la lumière structurée est principalement utilisée ici pour les traitements simples qu'elle permet.

Mais l'application la plus célèbre est certainement celle du *Pathfinfer*, véhicule d'exploration martienne, équipé d'un capteur en lumière structurée dédié à la détection d'obstacles [MAT95]. Le système que nous allons décrire ici est dérivé du *Pathfinder* dont il corrige les limitations en termes de temps de calcul [MAT97]. Il est composé de deux caméras et de deux sources laser ; chaque couple caméra/laser opère indépendamment, l'un sur la partie droite du champ de vision du robot, l'autre sur la partie gauche (Figure 27).

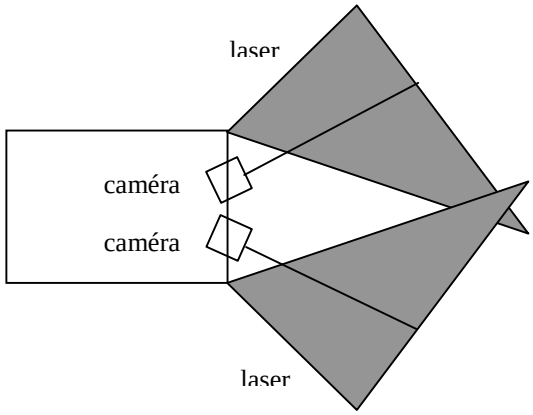


Figure 27. Configuration du capteur de Pathfinder

Les deux lasers sont diffractés en quinze faisceaux coplanaires et observés par l'une ou l'autre caméra. Les caméras sont déplacées horizontalement par rapport aux sources laser à laquelle elles correspondent. De ce fait, sur un terrain plat, les quinze points du faisceau sont alignés. Si un faisceau rencontre un obstacle, il sera dévié vers la droite ou vers la gauche selon l'élévation du terrain en ce point. L'obstacle pourra ainsi être détecté et la hauteur du terrain être évaluée.
Le système *Pathfinder* est fiable et précis, il permet la détection des obstacles de manière beaucoup plus rapide que par la stéréovision : la mise en correspondance est simple et rapide, les points à traiter sont peu nombreux.
Une autre application, proche de la première, est l'oeuvre de Henriksen et Krotkov [HEN97]. Cette fois, une source laser émet un faisceau vers un miroir rotatif projetant un plan de lumière sur le sol. Trois types d'obstacles sont considérés : les creux, les saillies et les inclinaisons transversales. Une fonction est associée à chaque type d'obstacle : elle est calculée à partir de l'élévation des points du trait laser et des seuils fixés. D'après les auteurs, leur système de vision en lumière structurée serait plus à même de détecter les obstacles du type creux qu'un système stéréoscopique. En effet, si l'on prend l'exemple d'un cratère, le trou du cratère est caché par ses parois et l'obstacle n'est pas détecté. Notons que le système de détection d'obstacles d'Henriksen et Krotkov s'insère dans une stratégie multi-capteurs pour la navigation quasi-autonome (les buts sont spécifiés par l'opérateur mais l'exécution est gérée par le robot).

1.4 Conclusion

De nombreuses techniques permettent la perception tri-dimensionnelle de l'environnement. Parmi elles, la stéréovision est la plus étudiée et la plus couramment employée, autant pour les résultats qu'elle apporte, que pour le cadre théorique qui la sous-tend. Un autre aspect non négligeable est l'analogie que l'on peut faire avec les mécanismes de la vision humaine. On sait pourtant que la stéréovision souffre d'un lourd handicap qui est le problème de la mise en correspondance ; problème dont la résolution devient illusoire dans un environnement faiblement texturé. La vision en lumière structurée permet d'alléger considérablement la combinatoire des algorithmes de mise en correspondance et de pallier à nombre des inconvénients qui leur sont attachés comme les occlusions et, justement, la mesure de surfaces peu texturées. Si la projection d'un point de surbrillance ou d'un plan de lumière permettent une mise en correspondance univoque et un traitement simple des images capturées, elles nécessitent le balayage de la scène pour obtenir une reconstruction dense de celle-ci. A l'inverse, la projection d'un motif structurant nécessite le codage de celui-ci ou des contraintes sévères prohibant son utilisation en robotique mobile. Pour un motif codé, le problème du décodage se substitue à celui de la mise en correspondance : il doit être robuste, peu sensible aux occlusions ou aux absorptions lumineuses des surfaces, adapté aux scènes dynamiques. Le second problème inhérent à ce type de capteur est celui de l'intensité lumineuse projetée : si l'illumination ambiante est faible ou, mieux, si le capteur évolue dans l'obscurité totale, aucun problème ne se pose ; dans le cas contraire, en revanche, on risque d'avoir à faire face à un signal inexploitable. L'utilisation d'une source laser résout partiellement ce problème, mais il n'existe pas, à notre connaissance, de source laser assez robustement codée pour une utilisation en robotique mobile. En conséquence, on trouve d'une part des applications de la lumière structurée dans des environnements contrôlés (du type industriel ou médical), et d'autre part des applications en robotique mobile n'utilisant pas les derniers développements de la vision en lumière structurée comme la projection de motif bi-dimensionnel et le codage. En effet, l'utilisation que fait la robotique mobile de la lumière structurée est presque exclusivement dédiée à la détection d'obstacles et basée sur la projection de points de surbrillance ou de plans de lumière. Le plus grand reproche que l'on peut faire à ces systèmes est de ne pas prendre en compte globalement l'environnement du robot. Il leur est par exemple impossible de détecter des obstacles qui ne se trouveraient pas en contact avec le sol. D'autre part, les problèmes tels que l'adaptation visuelle, la détection d'obstacles mobiles ou l'analyse du mouvement ne sont pas traités en lumière structurée. La principale motivation de notre travail était d'étudier ces différents problèmes pour éventuellement pallier à ces manques.

Chapitre 2 :
Segmentation des Images en Lumière Structurée

2.1 Introduction

Traiter une image, c'est exercer sur elle un certain nombre d'opérations et de transformations dans le but d'en tirer une information pertinente pour l'application que ces traitements doivent servir. En particulier, la segmentation d'une image consiste à décomposer des scènes complexes en des entités identifiables et plus aisément traitables que l'image entière. Les premières contributions à ce domaine de recherche apparaissent dès les années soixante. Le traitement des images est parfois vu comme une généralisation à deux dimensions du traitement du signal ; aujourd'hui, cette discipline, dont les ramifications vont de l'analyse à la reconnaissance des formes, de la compression à la restauration, a une existence propre qui emprunte à différentes théories mathématiques ou physiques, mais dont l'unité et l'autonomie n'est plus à démontrer. Les traitements que nous présentons dans ce chapitre peuvent être divisés en deux classes distinctes : la première est chargée de la segmentation des images, c'est-à-dire de l'extraction et de la paramétrisation des éléments du motif qui la composent ; la seconde, du décodage du motif par la reconnaissance des couleurs apparentes dans l'image parmi les six primitives projetées.
La vision en lumière structurée produit des images de morphologie particulière qui ne peuvent être traitées par les méthodes classiques (nous pensons à la détection de contours). C'est pourquoi la méthode que nous proposons est axée sur la squelettisation de l'image en niveaux de gris, qui tient lieu ici de détection de "contours". L'image est paramétrée, grâce à la transformée de Hough, à partir du squelette et les points d'intersection sont calculés. La couleur des points sur le lieu des segments extraits est convertie de RVB à Lab, puis injectée en entrée d'un algorithme de coalescence dont nous détaillons le mode opératoire. Au final, nous obtenons une image segmentée et décodée. Il est inutile de préciser que de la correcte estimation des coordonnées des points intersection et de leur décodage dépendront étroitement la précision et la fiabilité des traitements de plus haut niveau comme la reconstruction, l'analyse du mouvement, etc.

Dans la première section du chapitre, nous présentons les bases de la colorimétrie qui nous serviront à convertir nos images d'un espace de couleurs vers un autre espace de couleurs plus approprié. Nous présentons ensuite l'ensemble des traitements nécessaires à l'extraction de la structure telle qu'elle est perçue par la

caméra ainsi qu'un algorithme de décodage du motif. Nous illustrons chaque étape du traitement par des résultats expérimentaux. Enfin, nous évaluons la précision de la segmentation et la robustesse du décodage.

2.2 Eléments de Colorimétrie

2.2.1 Définition

La colorimétrie, ou métrique des couleurs, est la science qui permet de chiffrer et mesurer les couleurs. Elles y sont définies par trois paramètres physiques :

- *La teinte* qui représente une longueur d'onde, à laquelle est attribuée un nom de couleur correspondant (rouge, jaune, bleu,…)
- *Le degré de pureté*, ou rapport avec le blanc, qui détermine la quantité de blanc ajoutée à la teinte de base. Une forte quantité de blanc donnera une couleur lavée (le rose, par exemple) ; à l'inverse une quantité de blanc nulle donnera une couleur pure (le rouge).
- *Le facteur de clarté*, ou quantité de lumière transmise ; si une importante quantité lumineuse est transmise, la couleur est dite claire, sinon elle est dite foncée.

Nous parlerons plus couramment de *teinte*, *saturation* et *luminance* pour désigner respectivement ces trois indications. Les études inaugurées par Maxwell [MAX55, MAX60], Grassman [GRA53] et Helmholtz [HEL60] il y a un siècle, ont fondé les bases de cette discipline. Elle a abouti à la définition d'un *observateur de référence*, personnage fictif qui est censé représenter la moyenne des sujets normaux dans des conditions déterminées (luminance suffisante sous une source-étalon déterminée et adaptation normale de l'œil). La colorimétrie étudie essentiellement l'équivalence des diverses compositions de lumière qui donnent une même sensation colorée. Autrement dit, l'œil de l'observateur de référence est seulement un appareil de zéro qui juge l'identité visuelle.

2.2.2 Principe de la Trivariance Visuelle

Ce principe pose que le mélange, dans des proportions déterminées, de trois radiations indépendantes permet de reproduire toutes les impressions colorées possibles. En d'autres termes, trois grandeurs convenablement choisies sont nécessaires et suffisantes pour décrire toute lumière colorée. Ces trois grandeurs sont désignées comme étant les *excitations*. Les couleurs peuvent donc être représentées par des vecteurs dans un espace tri-dimensionnel. Les couleurs de base d'un espace sont appelées couleurs *primaires* ; celles qui, mélangées aux couleurs primaires donnent du blanc, sont appelées couleurs *secondaires*. Notons

que deux distributions spectrales d'énergie différentes peuvent donner naissance à une même couleur. Cette propriété est connue sous le nom de *métamérisme*.

2.2.3 Espace des Couleurs

Un *espace des couleurs* est une représentation géométrique des couleurs dans un espace généralement tridimensionnel. En 1931, la Commission Internationale de l'Eclairage (CIE) a recommandé, à cet effet, l'emploi d'un système de repérage conventionnel utilisant trois couleurs primaires spectralement définies avec les longueurs d'ondes dominantes. En vision par ordinateur, on connaît l'espace RVB (Rouge, Vert, Bleu) et son complémentaire, l'espace CMJ (Cyan, Magenta, Jaune) — on utilise également l'espace TSL (pour Teinte, Saturation, Luminance - Figure 28) et ses variantes (TSI, TSV, etc.). Les colorimètres travaillent principalement dans l'espace XYZ, issu de l'espace RVB dont il est une transformée linéaire. Cet espace corrige quelques inconvénients du système RVB, comme nous le verrons plus loin.
L'espace RVB, tout comme l'espace XYZ, n'est pas un espace à *chromaticité constante* (ou espace des couleurs uniforme), c'est-à-dire un espace au sein duquel les distances représentent approximativement des différences de couleurs perçues par l'œil. Pour remédier à cet inconvénient, la CIE a adopté l'espace Lab [MCL76] qui entend copier la réponse logarithmique de l'œil telle qu'elle est décrite par la loi de Weber-Fechner (Figure 30) ; les espaces TSL, TSI et TSV sont également à chromaticité constante. L'espace Lab est souvent représenté par une sphère comme sur la Figure 29.
En psychométrie, la loi de Weber-Fechner s'exprime ainsi :

$$R = Log\left(S \diagup S_0 \right) \qquad (1)$$

Où S est l'excitation (le stimulus) et R la réponse de l'œil.

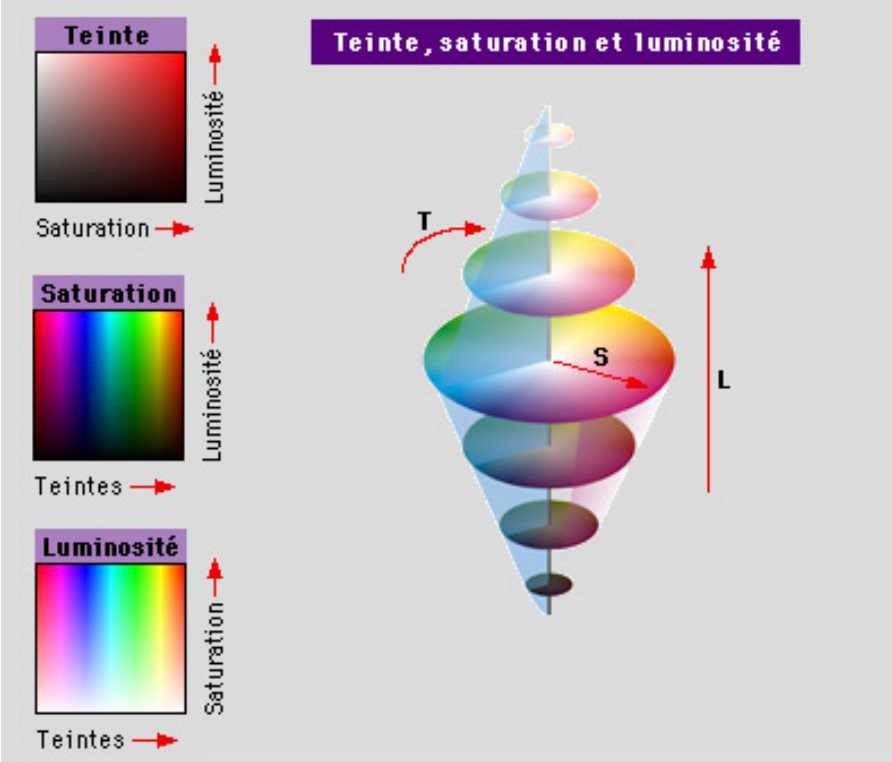


Figure 28. L'espace des couleurs TSL

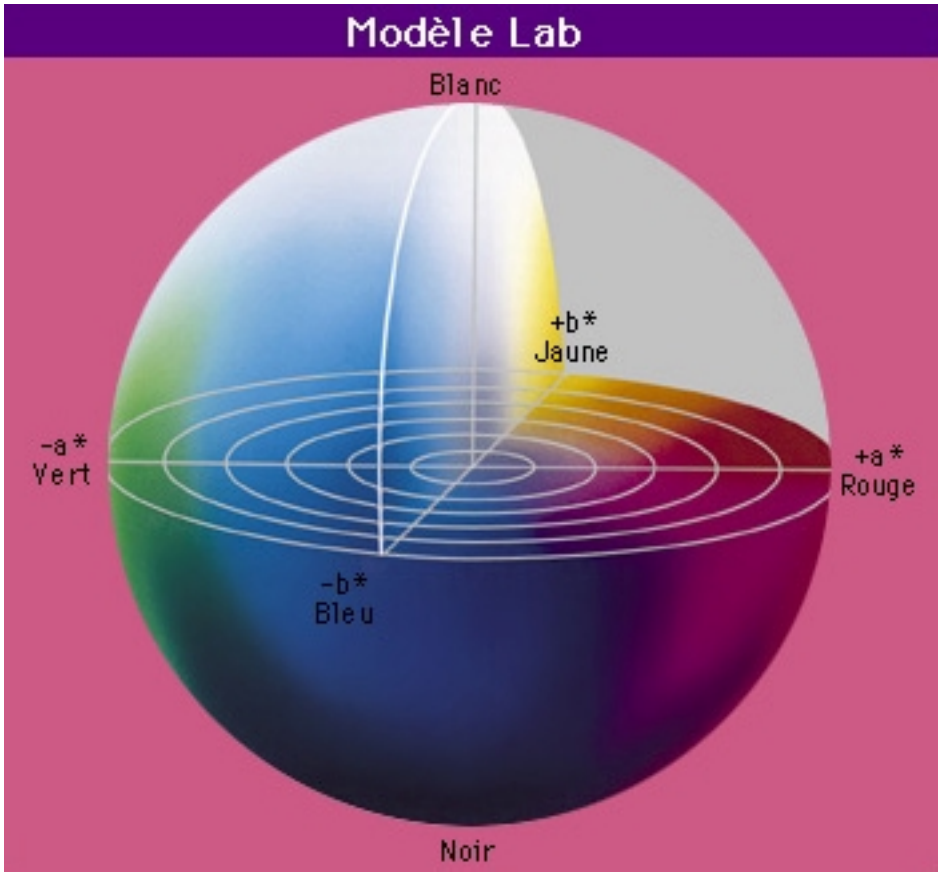


Figure 29. L'espace des couleurs CIE-Lab

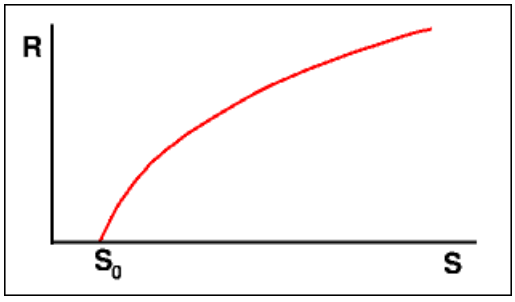


Figure 30. Loi de Weber-Fechner

Cette loi s'exprime ainsi : la sensation est proportionnelle au logarithme de l'excitation ; ce qui revient à constater que pour éclaircir un gris clair, il faut lui ajouter beaucoup de blanc, alors que pour éclaircir du même écart un gris foncé, une trace de blanc sera suffisante.

2.2.4 Lois Fondamentales de Grassmann

Les lois de Grassman [GRA53] sont un ensemble de règles pour la reproduction des couleurs qui s'appliquent dans le cas où la luminance n'est ni trop forte, ni trop faible :

- L'oeil humain ne peut pas retrouver les composantes d'un mélange de couleurs.
- *Loi d'additivité* : la luminance d'un mélange de couleurs est égale à la somme des luminances des composantes.
- *Loi de proportionnalité* : si deux plages lumineuses produisent la même impression colorée, cette égalité d'impression subsistera lorsque la luminance de l'une et de l'autre sera multipliée par un même nombre.
- *Loi de mélange des couleurs* : deux mélanges lumineux qui, juxtaposés, produisent la même impression colorée, se comportent aussi de façon identique dans le processus des mélanges. Ils peuvent donc se substituer l'un à l'autre.

2.3 Segmentation des Images en Lumière Structurée

2.3.1 Principe

Nous disposons d'images en lumière structurée colorées desquelles nous devons extraire les points d'intersection du motif perçu, qui sont les points d'intérêts dont nous nous servirons pour obtenir une information tri-dimensionnelle, et les couleurs des droites qui leur servent de support qui nous permettront de résoudre le problème de la mise en correspondance. L'idée est de séparer les informations de luminance et de chromaticité de l'image pour, sur la première, effectuer les traitements bas-niveau, et sur la seconde, analyser les couleurs pour décoder le motif. Le décodage des couleurs sera effectué sur le lieu des segments extraits (Figure 31). Nous avons vu dans la section précédente qu'il existait des espaces de couleurs qui approchaient plus convenablement la réponse de l'oeil à la perception des couleurs que d'autres. La première étape consistera donc à convertir les images RVB fournies par la caméra en images dont les couleurs sont exprimées dans un espace à chromaticité constante. Parmi les espaces de couleurs à chromaticité constante et permettant le découplage de la luminance et de la chromaticité, nous avons choisi l'espace Lab.

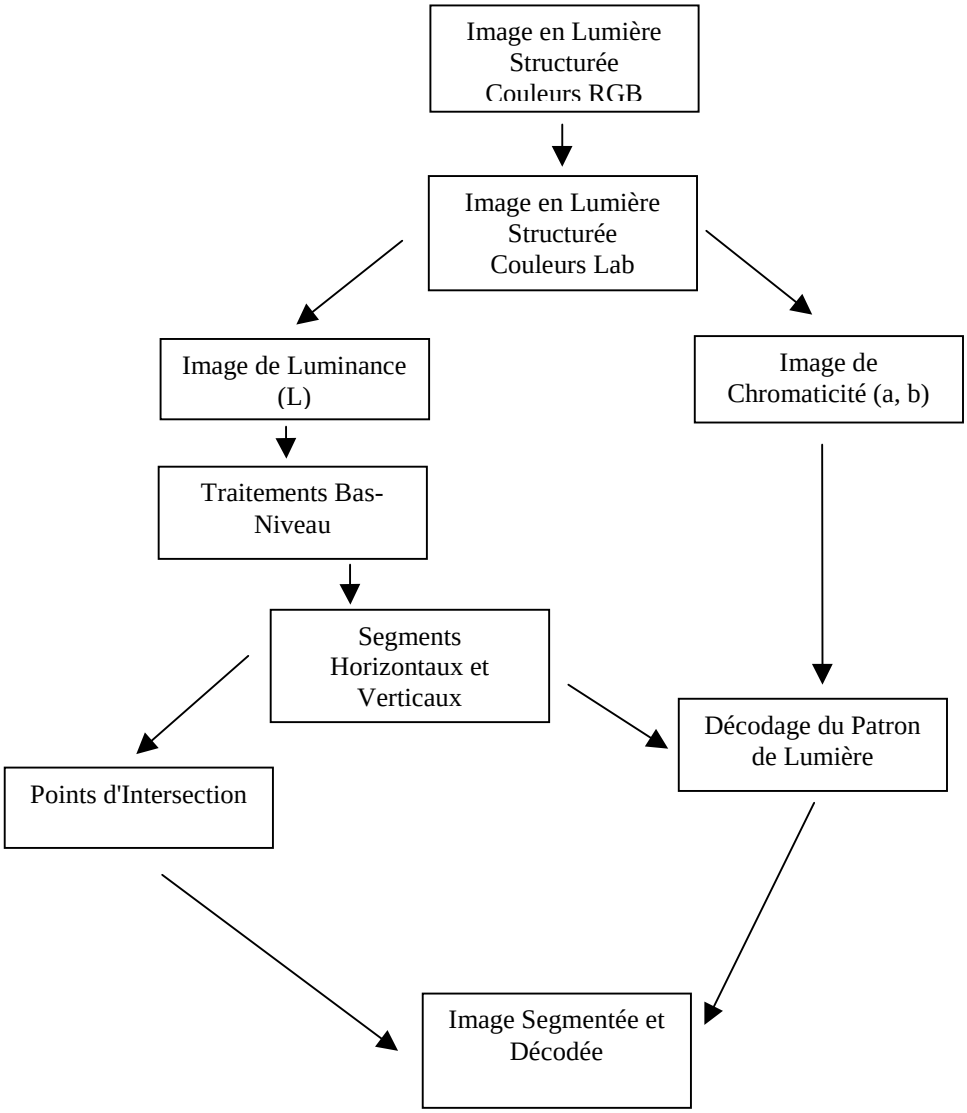


Figure 31. Algorithme de segmentation des images en lumière structurée

2.3.2 Conversion RGB-Lab

Pour convertir une couleur de l'espace RVB vers l'espace Lab, il faut passer par un espace intermédiaire, développé par la CIE en 1931, l'espace XYZ et dont les bases théoriques résident dans les travaux de Wright [WRI28] et Guild [GUI31]. Ses couleurs primaires ont la particularité d'être imaginaires et définies telles que :

- Les coordonnées couleurs soient toujours positives.
- La luminance de *X* et *Z* soient nulles, autrement dit *Y* concentre à elle seule l'information de luminance du signal.
- Le blanc soit représenté par un point défini par un mélange en quantités égales des trois primaires.

Le passage de l'espace RVB à l'espace XYZ est donné par :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} 0.618 & 0.177 & 0.205 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.056 & 0.944 \end{bmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2}$$

Maintenant, pour passer à l'espace Lab, dont la réponse logarithmique est approchée par la racine cubique, on dispose des équations suivantes :

$$\begin{cases} L = 116 \cdot \left(\dfrac{Y}{Y_0} \right)^{1/3} - 16 \quad \text{pour} \quad \dfrac{Y}{Y_0} > 0.008856 \\ L = 903.3 \cdot \left(\dfrac{Y}{Y_0} \right) \quad \text{pour} \quad \dfrac{Y}{Y_0} \le 0.008856 \\ a = 500 \cdot \left(f\left(\dfrac{X}{X_0} \right) - f\left(\dfrac{Y}{Y_0} \right) \right) \\ b = 200 \cdot \left(f\left(\dfrac{Y}{Y_0} \right) - f\left(\dfrac{Z}{Z_0} \right) \right) \end{cases} \tag{3}$$

où (X_0, Y_0, Z_0) sont les coordonnées du blanc de référence, et la fonction f est donnée par :

$$\begin{cases} f(t) = \sqrt[3]{t} \quad \text{pour} \quad t > 0.008856 \\ f(t) = 7.7787 \cdot t + \dfrac{16}{116} \quad \text{pour} \quad t \le 0.008856 \end{cases} \tag{4}$$

Le tableau ci-dessous présente quelques exemples de conversion de l'espace RVB vers l'espace Lab pour les trois couleurs primaires, leurs complémentaires, le blanc et le noir. L'échelle des couleurs RVB est originairement normalisée.

	R	**V**	**B**	**X**	**Y**	**Z**	**L**	**a**	**b**
Rouge	1	0	0	0.618	0.299	0.000	61.568	91.548	133.74
Vert	0	1	0	0.177	0.587	0.056	81.126	-137.9	90.942
Bleu	0	0	1	0.205	0.114	0.944	40.246	52.378	-99.22
Cyan	0	1	1	0.382	0.701	1.000	87.046	-81.37	-22.33
Magenta	1	0	1	0.823	0.413	0.944	70.386	96.213	-47.25
Jaune	1	1	0	0.795	0.886	0.056	95.413	-17.04	115.57
Blanc	1	1	1	1	1	1	100	0	0
Noir	0	0	0	0	0	0	0	0	0

Table 3. Conversion des couleurs

2.3.3 Traitements Bas-Niveau

Egalisation de l'histogramme

L'histogramme d'une image est la fonction qui donne, pour un niveau de gris, l'effectif correspondant au nombre de pixels présents dans l'image admettant ce niveau de gris ; il représente les fréquences relatives des différentes valeurs d'intensité lumineuse. Les techniques de modification de l'histogramme ont pour but une nouvelle répartition des niveaux de gris répondant au modèle que l'on s'est fixé. Parmi ces techniques, l'égalisation consiste à transformer l'histogramme de sorte que sa distribution soit *uniforme*. Elle a pour but d'estomper les trop grandes différences de luminosité de l'image. En pratique, les images ont été égalisées dans l'intervalle $[0;255]$; nous présentons les résultats sur un exemple d'image en lumière structurée ainsi que les histogrammes correspondants (Figure 32).

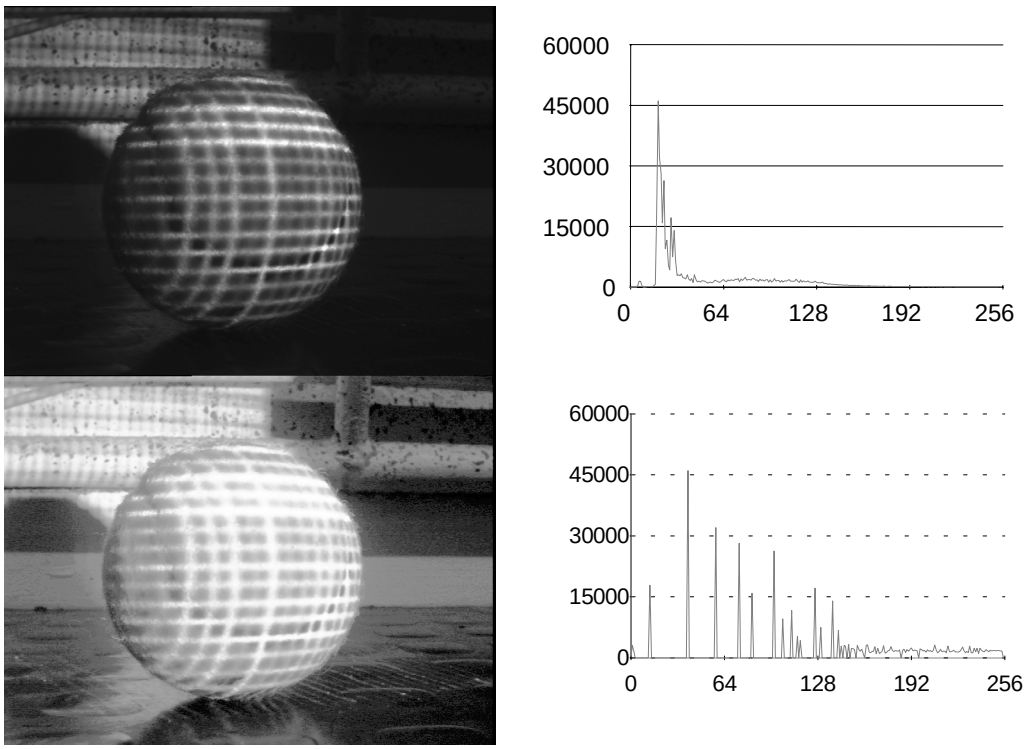


Figure 32. Egalisation d'histogramme. A gauche : l'image avant et après égalisation. A droite : les histogrammes correspondants

Nous constatons une meilleure répartition des niveaux de gris qui se traduit par l'apparition de pics dans l'histogramme. Sur l'image en niveaux de gris, on

remarque maintenant des détails qui restaient cachés dans l'image d'origine, un peu comme si la première image avait été prise de nuit et la seconde de jour.

Lissage de l'image

Les images capturées par la caméra sont inévitablement entachées de bruit. Si l'on se place dans l'hypothèse d'un bruit additif, l'image observée est la superposition de l'image idéale et d'un bruit *b*. Quand on ne dispose que d'une seule image, on effectue le plus souvent un moyennage local où chaque pixel est remplacé par la moyenne pondérée de ses voisins. Nous avons utilisé un masque de taille 3×3 :

$$\frac{1}{16}\cdot\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & \mathbf{4} & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

L'inconvénient majeur de ces techniques, on peut s'en rendre compte, est la perte de contraste sur les contours ; on verra plus bas que ce flou est relativement peu gênant pour la suite du traitement de nos images.

Seuillage de l'image

Le seuillage consiste à segmenter l'image en deux classes (on parle alors de binarisation) ou plus suivant la valeur de leurs niveaux de gris. Dans le cas de la binarisation, on va comparer chaque pixel de l'image à un seuil et l'on affectera la valeur noire à ce pixel si la valeur d'origine est inférieure au seuil et la valeur blanche dans le cas contraire.
A ce point du traitement, notre but est de discriminer le motif perçu du reste de l'image, donc, segmenter l'image en deux classes distinctes. Pour fixer le seuil de binarisation, on détecte habituellement les deux pics de l'histogramme correspondant aux zones sombres et claires de l'image ; on parle alors d'histogramme bi-modal. On constate, sur les images en lumière structurée, qu'un seul pic (celui des zones sombres) est visible (Figure 33). Une solution consiste à détecter le pic visible et à déterminer un seuil compris entre la valeur de ce pic et la valeur maximale des niveaux de gris rencontrés dans l'image. Empiriquement, nous avons pu établir que la fonction suivante, où NG_{pic} représente le niveau de gris du pic et $NG_{\max}$ le niveau de gris maximal, fournit un bon compromis pour l'estimation du seuil :

$$S = \frac{3 \cdot NG_{pic} + NG_{\max}}{4} \qquad (5)$$

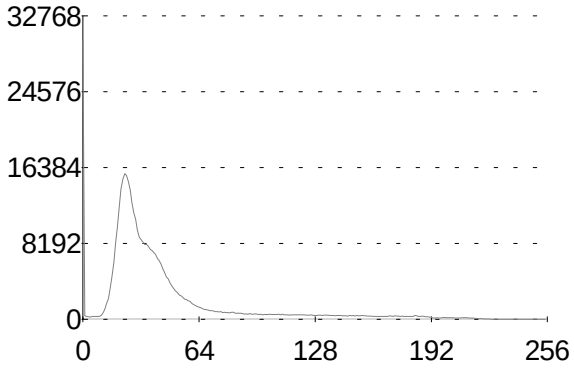


Figure 33. Histogramme mono-modal des images en lumière structurée

Malheureusement un tel seuillage ne prend pas en compte les différences de luminosité de la scène et, par conséquent, la luminosité locale aux différents sites de l'image. A cet effet, nous proposons une méthode de seuillage originale qui fixe adaptativement un seuil par rapport à la moyenne des niveaux de gris d'un voisinage donné.

Considérons un pixel *s* de l'image, et V_s un voisinage de ce pixel. La moyenne des intensités des pixels sur ce voisinage est donnée par :

$$\overline{m}(s) = \frac{1}{card(V_s)} \cdot \sum_{i \in v_s} I(i) \tag{6}$$

Etant donné que les régions que nous désirons discriminer sont d'orientation proche de la verticale ou proche de l'horizontale, le voisinage que nous avons choisi favorise ces deux directions. En fait, deux masques sont appliqués sur l'image d'origine ; les deux images résultant de ces opérations sont ensuite additionnées. Détaillons le processus. Pour chaque pixel, la moyenne est simultanément calculée sur un voisinage 16×16 horizontal et vertical ; notons $\overline{m}_H$ la moyenne horizontale et $\overline{m}_V$ la moyenne verticale. Nous obtenons deux images résultantes notées respectivement I_H et I_V, avec :

$$\begin{aligned} I_H(s) &= \begin{cases} 0 \text{ si } s < \overline{m}_H(s) \\ 1 \text{ si } s \geq \overline{m}_H(s) \end{cases} \\ I_V(s) &= \begin{cases} 0 \text{ si } s < \overline{m}_V(s) \\ 1 \text{ si } s \geq \overline{m}_V(s) \end{cases} \end{aligned} \tag{7}$$

où le 1 correspond au niveau de gris maximal et le 0 au niveau de gris minimal. Pour obtenir l'image seuillée finale, nous additionnons les deux images. La Figure 34 montre une image en niveaux de gris accompagnées des images seuillées

suivant les deux méthodes décrites ici. On peut constater que la seconde méthode fait ressortir des éléments de motif que la première méthode néglige. Elle est, en revanche, plus sensible au bruit que la première.

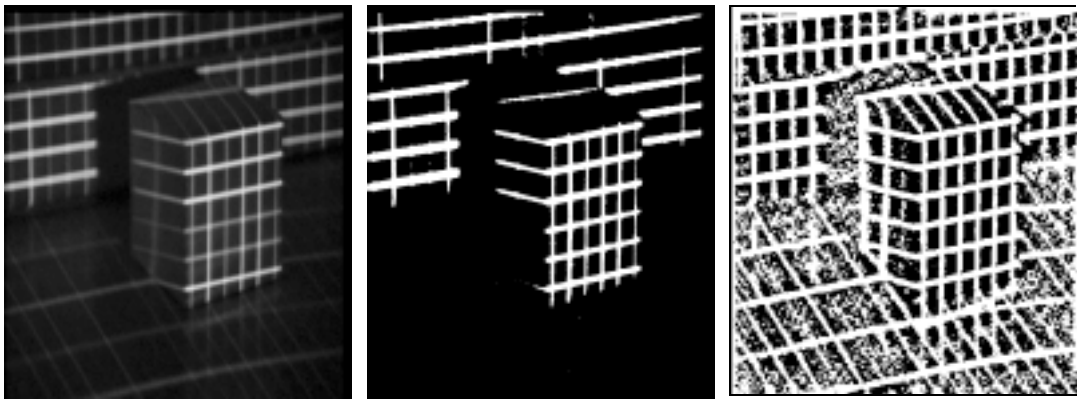


Figure 34. Seuillage. A gauche : image originale. Au centre : seuillage global. A droite : seuillage dynamique

Squelettisation

Le squelette d'une forme est défini comme l'ensemble des points dont la distance au point le plus proche du contour est localement maximum. Le squelette d'un cercle est le centre du cercle ; le squelette d'un carré est donné par ses diagonales. A ce stade du traitement, nous disposons d'une image binaire représentant, en traits épais, les éléments du motif perçu. Le squelette de chacun de ces traits est une ligne, dont l'épaisseur est d'un pixel, situé au centre du trait épais. Une méthode bien connue de squelettisation est donnée par la *morphologie mathématique*. Cette discipline, basée sur la théorie des ensembles, offre un ensemble d'opérateurs jouant sur la structure des images [SER82]. Le principe d'une opération morphologique est de chercher dans le voisinage de chaque pixel de l'image initiale une certaine configuration de points, appelée *élément structurant*. Si la configuration est trouvée, le pixel correspondant sera activé (le type d'activation dépendant de l'opérateur), dans le cas contraire, le pixel sera déterminé d'une façon complémentaire à l'activation.
Pour extraire le squelette, c'est l'opération d'*amincissement* qui sera utilisée. Considérons un élément structurant S formé de deux parties disjointes S^0 et S^1. En se référant à l'alphabet de Golay [GOL69], l'élément structurant de squelettisation est L (et parfois M) donné, respectivement pour un pavage carré et hexagonal de l'image, par :

0	0	0			0		0	
?	1	?		?		1		?
1	1	1			1		1	

Figure 35. L'élément structurant *S*

$$Sq(X) = (X \circ L)_{\infty} \tag{8}$$

Le squelette est homotope à l'ensemble *X* d'origine et d'épaisseur minimale. La squelettisation est anti-extensive ($Sq(X) \subset X$) et idempotente ($Sq(Sq(X)) = Sq(X)$). Il faut noter que le squelette est sensible au bruit : des barbules apparaissent de part et d'autres de celui-ci. La Figure 36 présente quelques résultats de squelettisation obtenus sur des images en lumière structurée.

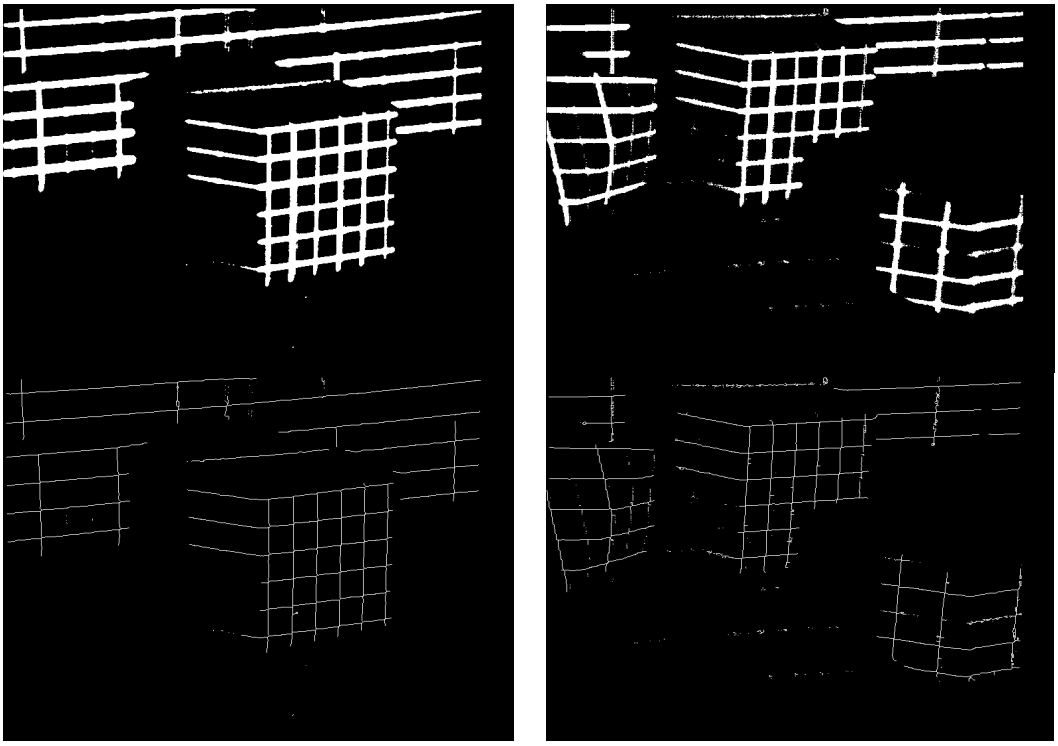


Figure 36. Squelettisation. En haut : images binarisées. En bas : images squelettisées

2.3.4 Extractions des Segments

Détection des droites par la transformée de Hough

Il est souvent préférable, plutôt que d'avoir une image de type *bitmap*, de paramétrer les éléments qui la composent en droites, segments et courbes ; en fait, de passer d'une représentation où la primitive est le pixel, à une représentation d'un niveau supérieur, moins sensible au bruit et de manipulation plus aisée. Dans un environnement polygonal ou plan par morceaux, il semble évident que le segment est une primitive appropriée ; de surcroit, l'approximation polygonale des courbes peut conduire à une représentation simple de l'image sans pour autant être trop simpliste en termes de précision pour les mesures ou pour une éventuelle reconstruction de la scène. En ce sens, la transformée de Hough [HOU62] est un outil couramment utilisé. Originairement, elle permet de détecter les droites au sein d'une image à partir d'un changement de représentation paramétrique de celles-ci. Elle fut ensuite généralisée à des courbes de degré supérieur. Il existe d'autres techniques d'approximation polygonale dont on trouvera un bilan dans [GAR92] ; il nous a semblé, cependant, que la transformée de Hough s'adaptait plus convenablement à la morphologie des images en lumière structurée :

- Si l'on connait *a priori* le positionnement relatif de la caméra et du projecteur, on peut en déduire la direction approximative qu'auront les droites verticales et horizontales projetées dans l'image et ainsi réduire la taille de l'accumulateur.
- La transformée de Hough, de par son fonctionnement statistique, possède le double avantage de combler les discontinuités des droites perçues et d'être robuste au bruit ; ce qui offre une marge de manoeuvre assez large dans le choix du seuil de binarisation ou du type de seuillage utilisé.

Considérons un ensemble de points du squelette m_1, m_2, ..., m_n, de coordonnées respectives $(x_i, y_i)_{1 \le i \le n}$. Imaginons un vecteur dont la valeur de la $j^{\text{ème}}$ composante est égale au nombre de points pour lesquels $y_i / x_i = j$. Si une majorité de points sont alignés sur une droite de pente a, alors la $a^{\text{ème}}$ composante devrait présenter un maximum et nous aurons détecté une ligne dans l'image, même dans le cas où les n points ne sont pas connexes. Généralisons ce résultat. L'équation générale d'une droite, dans l'espace affine, est donnée par :

$$y = a \cdot x + b \tag{9}$$

On sait que cette représentation ne convient pas aux droites verticales. On lui préférera donc la suivante :

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \qquad (10)$$

Dans l'espace des paramètres (ρ,θ), une droite est représentée par un point. Il passe une infinité de droites d'équations (10) par un point (x_i, y_i). Pour une second point (x_j, y_j), on aura une autre infinité de droites de paramètres (ρ,θ). Pour un ensemble de points alignés, ces droites se couperont en un unique point de l'espace des paramètres ; ces paramètres sont ceux de la droite liant les points. L'algorithme tiré de ce raisonnement, appelé *transformée de Hough*, peut s'écrire :

1. Quantifier l'espace des paramètres (ρ,θ) entre deux minima et deux maxima.
2. Initialiser un tableau à deux entrées, appelé *accumulateur*, à zéro.
3. Pour chaque point de contour, ou dans notre application, du squelette, (x_i, y_i), incrémenter l'élément $Accu(\rho_k, \theta_l)$ si :

$$\rho_k = x_i \cdot \cos\theta_l + y_i \cdot \sin\theta_l \qquad (11)$$

4. Les maxima locaux du tableau correspondent aux segments de droites. Les valeurs des éléments du tableau indiquent le nombre de points alignés.

Il convient maintenant d'étudier la particularité des images en lumière structurée pour fixer les bornes et le pas de l'espace des paramètres. Si l'on se restreint au positionnement préconisé dans la section 4.2.2 (et dont les commentaires quant à la validité et à la perte de généralité demeurent valables), on sait que les verticales projetées garderont un aspect quasi-vertical dans l'image et que les horizontales projetées garderont un aspect quasi-horizontal. De ce fait, nous scinderons l'ensemble des paramètres θ en deux intervalles, l'un proche de zéro, l'autre proche de $\pi/2$. Sur la série d'images de test, il est apparu qu'une tolérance de $\pm 10°$ $\left(\pm \pi/18\right)$ était suffisante et ce, aussi bien pour les verticales que pour les horizontales. Le pas d'échantillonnage fixé est de 1°. En pratique, on utilise deux accumulateurs, l'un pour la détection des quasi-verticales, l'autre pour la détection des quasi-horizontales ; le second étant obtenu en ajoutant $\pi/2$ aux valeurs du premier. Le paramètre ρ est choisi de sorte que l'ensemble de l'image soit balayée. Pour une image 512×512, par exemple, on prendra ρ allant de 0 à la diagonale, soit $\sqrt{2} \cdot 512 \cong 724$. Pour extraire les maxima locaux, nous passons une fenêtre 10×10 sur les accumulateurs et nous ne gardons, au sein de cette fenêtre, que la valeur du tableau la plus élevée. La Figure 37 présente les maxima locaux extraits des résultats précédents.

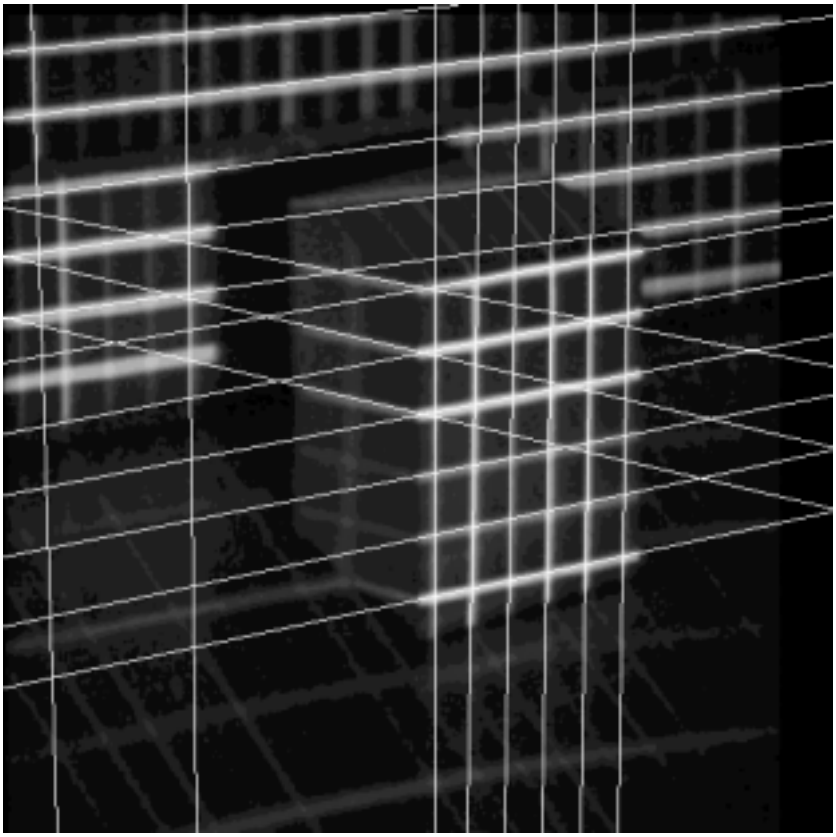


Figure 37. Extractions des droites par transformée de Hough

Extrémités des segments

Si nous calculions l'intersection des droites extraites en l'état, on trouverait bien évidemment des points d'intersection qui n'ont pas d'existence réelle. Il nous faut donc réduire les droites obtenues par la transformées de Hough au(x) segment(s) qui les compose(nt) avant de calculer l'intersection. Pour cela chaque droite est balayée jusqu'à ce qu'une extrémité soit rencontrée. Quelques précautions sont à prendre : puisque la transformée de Hough opère un échantillonnage sur l'orientation des droites, il faut tester la présence des extrémités sur un voisinage de deux ou trois pixels dans la direction perpendiculaire à l'orientation estimée. Pour éliminer les segments parasites provoqués par le bruit ou les barbules, on rejettera les segments dont la taille est inférieure à un seuil donné.

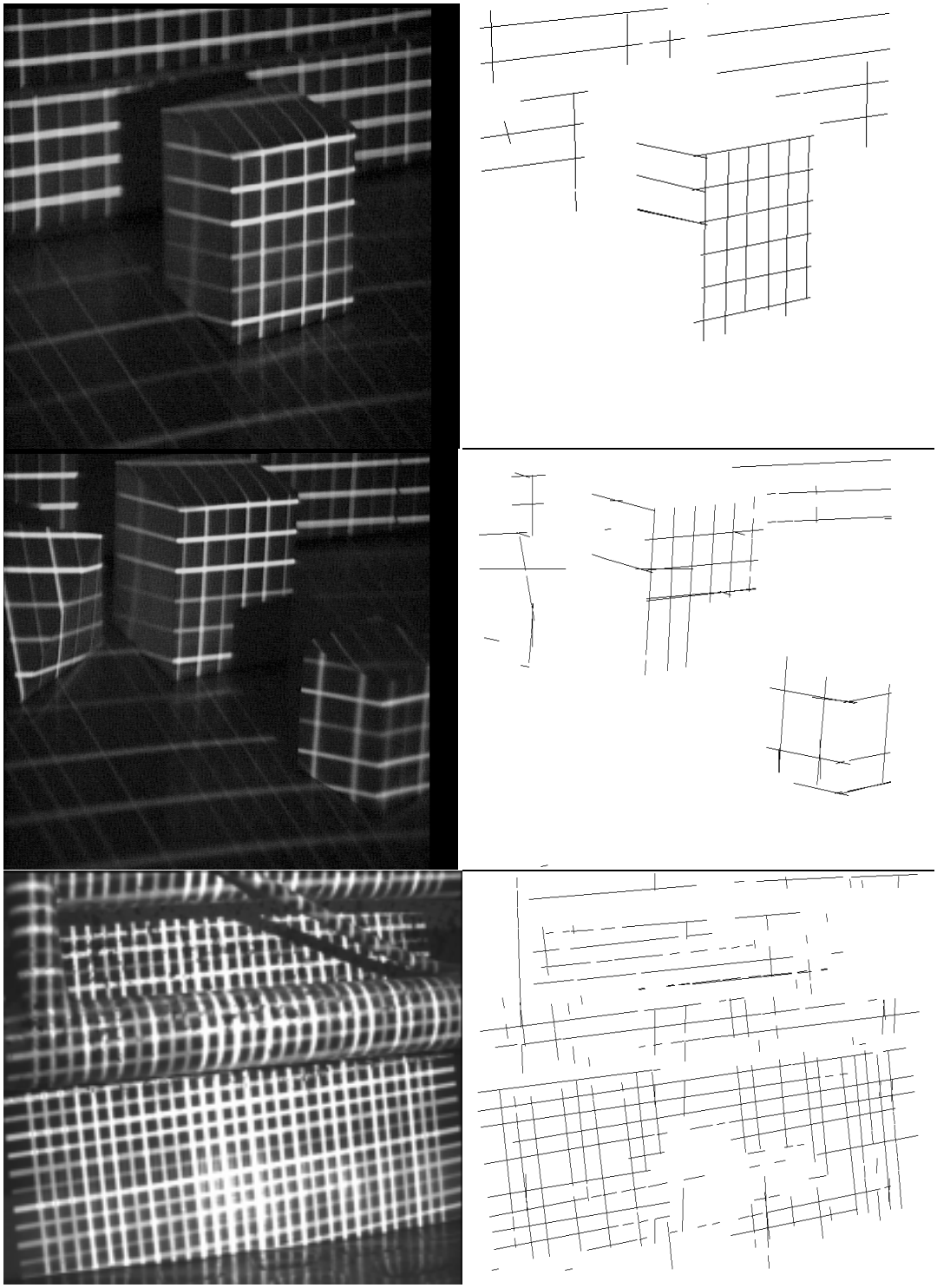


Figure 38. Extraction des segments

Au vu de ces premiers résultats, les limites qualitatives des traitements effectués peuvent déjà être pointées. Les résultats offerts sur des scènes planes ou planes

par morceaux sont relativement bons ; en revanche, ils se dégradent très vite si la scène est composée de surfaces complexes. Les courbes se scindent en plusieurs segments dont le nombre et la taille dépend à la fois de l'échantillonnage des paramètres de Hough et des seuils fixés pour la binarisation, le rejet des droites et des segments, etc. D'une manière générale, plus l'image est complexe, plus elle est sensible à la variation des seuils. Une mise au point imprécise est un autre facteur important de dégradation des résultats. Notons que pour la Figure 38, la binarisation a volontairement été très sélective.

2.3.5 Points d'Intersection

On va maintenant calculer les points d'intersection de chaque droite quasi-horizontale avec chaque droite quasi-verticale. Puisque nous avons scindé l'accumulateur de la transformée de Hough en deux, l'un consignant les points alignés sur les horizontales et sur les droites proches de l'horizontale, et l'autre se chargeant de l'opération similaire pour les droites proches de la verticale, ce calcul ne pose pas de problème. On sait que la géométrie projective permet de calculer l'intersection de deux droites à partir d'un simple produit vectoriel. Soient (a_H, b_H, c_H) les paramètres d'une droite quasi-horizontale et (a_V, b_V, c_V), les paramètres d'une droite quasi-horizontale, que l'on obtient à partir des paramètres de Hough par :

$$\begin{cases} a_H = \cos(\theta_H) \\ b_H = \sin(\theta_H) \\ c_H = \rho_H \end{cases} \qquad \begin{cases} a_V = \cos(\theta_V) \\ b_V = \sin(\theta_V) \\ c_V = \rho_V \end{cases} \tag{12}$$

L'intersection de ces deux droites est donnée par :

$$\begin{pmatrix} a_H \\ b_H \\ c_H \end{pmatrix} \times \begin{pmatrix} a_V \\ b_V \\ c_V \end{pmatrix} = \begin{pmatrix} b_H \cdot c_V - b_V \cdot c_H \\ c_H \cdot a_V - c_V \cdot a_H \\ a_H \cdot b_V - a_V \cdot b_H \end{pmatrix} = \begin{pmatrix} x \\ y \\ t \end{pmatrix} \tag{13}$$

Si *t* est nul, l'intersection des droites est rejetée à l'infini ; ce qui signifie que les droites sont parallèles. Dans tous les autres cas, on obtient un point d'intersection dont les coordonnées sont x/t et y/t . L'extrémité des segments est utilisée par éliminer les points d'intersection abusivement détectés.

2.3.6 Décodage du Patron de Lumière

Processus de formation de la couleur

Nous sommes partis de la constatation suivante : un observateur humain parvient sans peine à déterminer, à partir de la couleur apparente, la couleur originairement projetée sur la scène parmi les six primitives qui composent le motif et ce, pour une large gamme d'illuminations et de surfaces. Notre objectif est de parvenir à un résultat similaire par traitement numérique de la couleur.
Cependant, la couleur étant une *sensation* liée impérativement à trois dimensions : les caractéristiques photométriques de l'objet éclairé, la nature de la lumière qui l'éclaire, l'observateur, il semble par conséquent très difficile d'identifier convenablement la couleur projetée à partir de la couleur apparente : il faudrait, en toute rigueur, connaître les caractéristiques de réflectance, d'absorption et de transparence de tous les objets qui composent la scène ; il faudrait également calibrer la caméra de manière à connaître la relation qui lie les couleurs réelles aux couleurs de l'image ; il faudrait enfin mesurer les couleurs qui éclairent la scène. Récemment, Caspi, Kiryati et Shamir [CAS98] ont modélisé très précisément le processus de formation de la couleur dans l'image (ou *couleur apparente*) à partir d'une couleur initialement projetée (ou *couleur originale*).

$$\underbrace{\begin{pmatrix} R \\ G \\ B \end{pmatrix}}_{\mathbf{C}} = \underbrace{\begin{bmatrix} \Lambda_{RR} & \Lambda_{RG} & \Lambda_{RB} \\ \Lambda_{GR} & \Lambda_{GG} & \Lambda_{GB} \\ \Lambda_{BR} & \Lambda_{BG} & \Lambda_{BB} \end{bmatrix}}_{\mathbf{\Lambda}} \cdot \underbrace{\begin{bmatrix} k_R & 0 & 0 \\ 0 & k_G & 0 \\ 0 & 0 & k_B \end{bmatrix}}_{\mathbf{K}} \cdot \underbrace{P\left\{\begin{pmatrix} r \\ g \\ b \end{pmatrix}\right\}}_{\mathbf{c}} + \underbrace{\begin{pmatrix} R_0 \\ G_0 \\ B_0 \end{pmatrix}}_{\mathbf{C}_0} \qquad (14)$$

Où **C** représente les composantes *R*, *G*, *B* en un pixel donné de l'image en lumière structurée ; $\mathbf{C}_0$ représente les mêmes composantes au même point mais sous l'illumination ambiante ; **c** représente la couleur d'entrée, c'est-à-dire celle avec laquelle est construite le patron de lumière en ce point ; $P\{\ \}$est un opérateur non-linéaire de transformation entre la consigne et l'intensité lumineuse effectivement projetée ; $\mathbf{\Lambda}$ est la matrice de couplage projecteur-caméra ; **K** est la matrice de réflectance de la surface au point considéré obtenue en résolvant l'équation pour une entrée $\mathbf{c} = (255 \quad 255 \quad 255)^T$. Trois images sont nécessaires à la résolution complète de cette équation : une image en lumière structurée, une image en lumière ambiante et une image en lumière blanche uniforme. Cette méthode n'est pas adaptée à l'usage que nous entendons faire du capteur de vision en lumière structurée : en effet, elle nous ferait perdre le bénéfice du codage "dynamique".

Algorithme de décodage du patron

Dans l'espace CIE-Lab, l'*écart colorimétrique* s'exprime par la formule (15). L'information de teinte étant concentrée dans les variables *a* et *b*, on va choisir une distance, appelée *écart chromatique*, ne faisant intervenir que ces deux composantes (16).

$$\Delta E = \sqrt{\Delta L^2 + \Delta a^2 + \Delta b^2} \tag{15}$$

$$\Delta C = \sqrt{\Delta a^2 + \Delta b^2} \tag{16}$$

où Δ représente une différence.
Des mesures ont été effectuées sur une dizaine d'images où les composantes *a* et *b* des primitives de couleur ont été calculées. Nous avons ainsi cherché à vérifier le pouvoir discriminant d'un tel espace. La Figure 39 présente les résultats de mesures effectuées sur une seule de ces images.

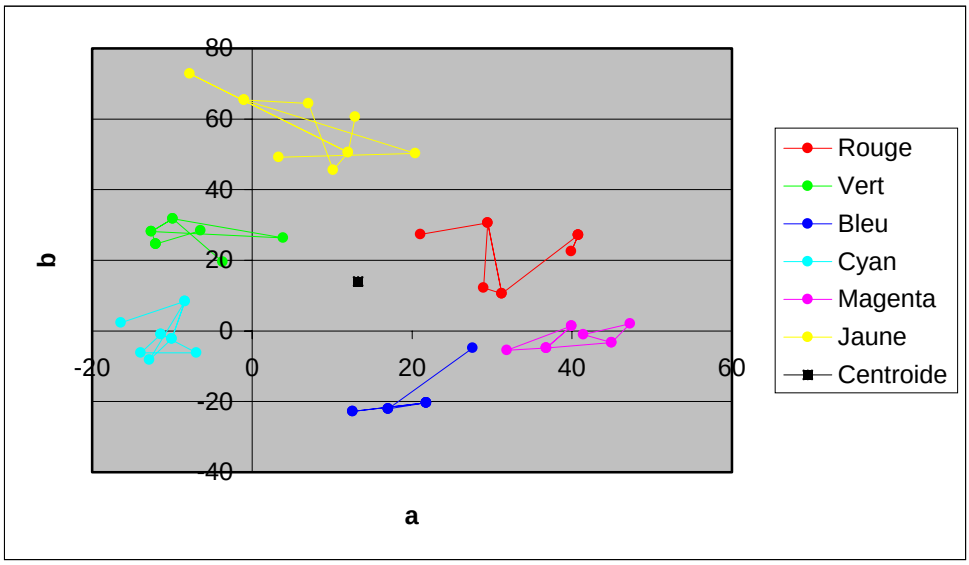


Figure 39. Classification des primitives de couleur

Une première constatation s'impose : les classes, si elles sont bien discriminées, sont très éloignées des mesures de référence théoriques ; elles semblent toutes se rapprocher du point d'achromatisme (0, 0), ce qui réduit leur pouvoir discriminant. Sur cet exemple, les coordonnées du barycentre des points sont approximativement (13, 14). En revanche, le positionnement relatif des six classes est respecté : le jaune est bien la classe dont la composante *b* est la plus forte, le bleu dont la composante *b* est la plus faible, le vert se trouve bien entre le jaune et le cyan, etc.

L'idée est d'utiliser un algorithme de coalescence [DUB90] pour partitionner les mesures en six classes représentant chacune une primitive de couleur et d'utiliser le positionnement relatif pour initialiser l'algorithme. Rappelons d'abord le principe de la coalescence : sur la base d'une partition en *n* classes, on affecte les mesures aux différentes classes de manière à minimiser, par exemple, le critère d'erreur quadratique. Le nombre de classes doit être connu *a priori*. Les étapes de l'algorithme sont les suivantes :

1. Choisir *n* centres de classe.
2. Créer une nouvelle partition en affectant chaque vecteur au centre de la classe qui est la plus proche.
3. Calculer les centres de gravité des classes ainsi obtenues.
4. Répéter les étapes 2 et 3 jusqu'à obtenir un extremum pour le critère retenu.
5. Eventuellement, revoir le nombre de classes en regroupant les classes peu remplies ou en éclatant les classes trop importantes ou en éliminant les points ou classes aberrantes.

On a pu constater avec ce type d'algorithme que le partitionnement final dépend de l'initialisation des centres de classe. Au lieu de choisir arbitrairement *n* points parmi les mesures comme centres de classe initiaux, nous utilisons le positionnement relatif des six classes, connu *a priori*, pour déduire les coordonnées des six centres de classe. On calcule tout d'abord le centre de gravité de tous les points mesurés dans l'espace (*a*, *b*) ; notons $\overline{m}$ ce point. On extrait ensuite des mesures le point dont la composante *b* est la plus élevée ; on suppose que ce point est membre de la classe jaune. On calcule l'écart chromatique *d* entre ces deux points. Puisque l'on connait les valeurs théoriques des six primitives de couleur (Table 3), on va placer le centre m_i de classe *i* de la manière suivante :

$$m_i = \frac{d}{\left\| a_i^2 + b_i^2 \right\|} \cdot \begin{pmatrix} a_i \\ b_i \end{pmatrix} + \overline{m} \qquad (17)$$

où a_i et b_i représente les valeurs théoriques. La première classe est initialisée dans la direction théorique du rouge, la deuxième dans la direction théorique du bleu, etc. Grâce à l'initialisation, à la connaissance *a priori* du positionnement relatif des classes et à l'algorithme de coalescence, on obtient directement la classification des points mesurés dans les six classes représentant chacune une primitive de couleur.

L'algorithme peut être affiné en deux points :

- Il est possible d'éliminer les mauvaises classifications en prenant en compte le fait que tous les points appartenant à un même élément de motif (vertical ou horizontal) sont de même couleur.
- Si l'on connait *a priori* la géométrie du système, il est possible de partitionner indépendamment les primitives utilisées pour le codage des verticales et des horizontales.

2.4 Résultats Expérimentaux

2.4.1 Extraction des Points

L'ensemble du processus de traitement des images en lumière structurée a été implémenté. L'objet de cette section est de comparer les résultats obtenus par la méthode que nous proposons avec des résultats obtenus par extraction manuelle des points d'intersection et aussi, avec des résultats obtenus par corrélation d'une imagette de référence. Nous avons testé notre algorithme sur quatre images d'une grille 5×5 prises sous différents angles et à différentes distances. Pour la segmentation manuelle, nous avons simplement cliqué sur le point qui nous semblait correspondre au point d'intersection ; la mise en correspondance d'imagette a été réalisée en utilisant la fonction de *pattern recognition* du logiciel Matrox Inspector® : une intersection du motif est utilisée comme modèle (imagette 20×20) et balaye l'image, l'indice de ressemblance est fixé à 60% et l'imagette de référence peut subir une rotation de ±5°. Les quatre coins du motif, à cause de leur forme particulière, ne sont jamais reconnus par cette méthode. En outre, sur des images moins polygonales, où les intersections apparaissent avec de plus grandes distortions, cet algorithme offre un taux de reconnaissance très faible.
Nous désignons, dans les tableaux suivant, la méthode que nous proposons par **LS**, la segmentation manuelle par **Manu** et la corrélation d'imagettes par **Insp**. Les images, sur lesquelles les tests ont été effectuées, sont présentées sur la partie gauche des figures avec les résultats de la segmentation donnés sur la partie droite.

Premier exemple

La grille 5×5 est capturée par la caméra sans distorsion sensible. On peut s'attendre, dans ces conditions, à ce que la corrélation d'imagette donne des résultats relativement fiables. On constate que la distance moyenne entre les coordonnées des points d'intersection extraits selon les trois différentes méthodes est de l'ordre du pixel. Mieux, la distance moyenne entre les points de la méthode **Insp** et ceux de la méthode **LS** sont de l'ordre du demi-pixel.

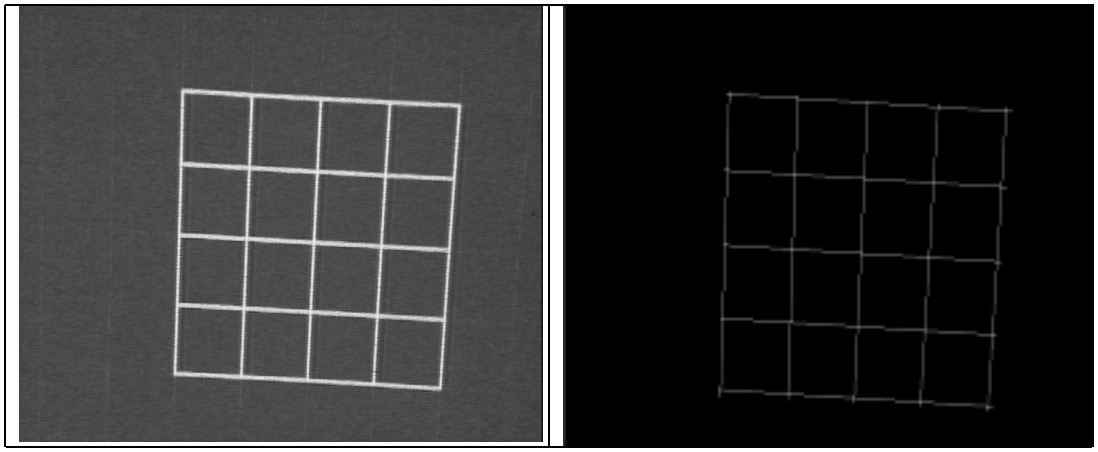


Figure 40. Test de segmentation. Image 1.

	LS-Manu		LS-Insp		Manu-Insp	
Moyenne	-0,30	-1,19	-0,36	-0,41	-0,14	0,71
Ecart-type	0,91	0,73	0,90	0,53	0,36	0,56

Distance moyenne	1,22	0,55	0,73

Table 4. Comparaison des méthodes de mise en segmentation. Image 1.

Deuxième exemple

Sur la deuxième image, la grille semble filmée de plus loin. On devrait s'attendre, théoriquement, à des résultats moins précis, mais en pouvant toujours compter sur la fiabilité de la méthode **Insp**. L'écart entre la méthode **Insp** et **LS** reste dans le même ordre de grandeur que sur l'exemple précédent. En revanche, les résultats obtenus par la méthode **Manu** semble s'écarter des méthodes automatiques. On peut l'expliquer par le fait que, la grille étant plus petite dans l'image, les points extraits manuellement l'ont été avec une précision plus grossière.

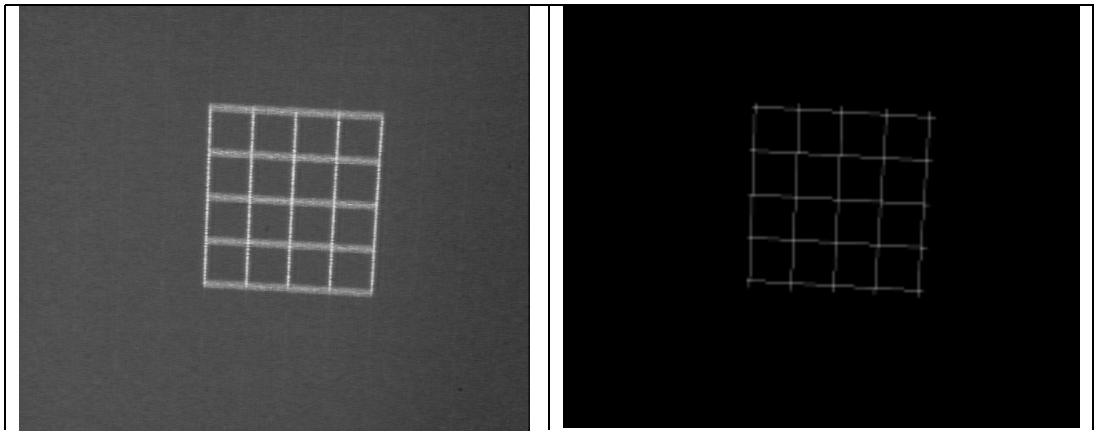


Figure 41. Test de segmentation. Image 2.

	LS-Manu		LS-Insp		Manu-Insp	
Moyenne	-0,55	-1,38	0,23	0,35	0,71	1,86
Ecart-type	0,69	0,69	0,51	0,69	0,56	0,91

Distance moyenne	1,49	0,42	1,99

Table 5. Comparaison des méthodes de segmentation. Image 2.

Troisième exemple

La grille est cette fois filmée de côté ; les carrés deviennent parallélogrammes. On peut envisager que la méthode **Insp** offre des résultats moins précis que sur les exemples précédents. En revanche, et c'est là son utilité, il n'y a aucune raison de croire que les résultats obtenus manuellement soient moins précis sur ce type d'image. On peut d'ailleurs constater que l'écart entre la méthode **LS** et **Manu** reste de l'ordre du pixel (plus faible ici que précédemment) et que la méthode **Insp** s'écarte significativement de deux autres méthodes. On peut donc penser que notre méthode demeure valide sur ce type d'image.

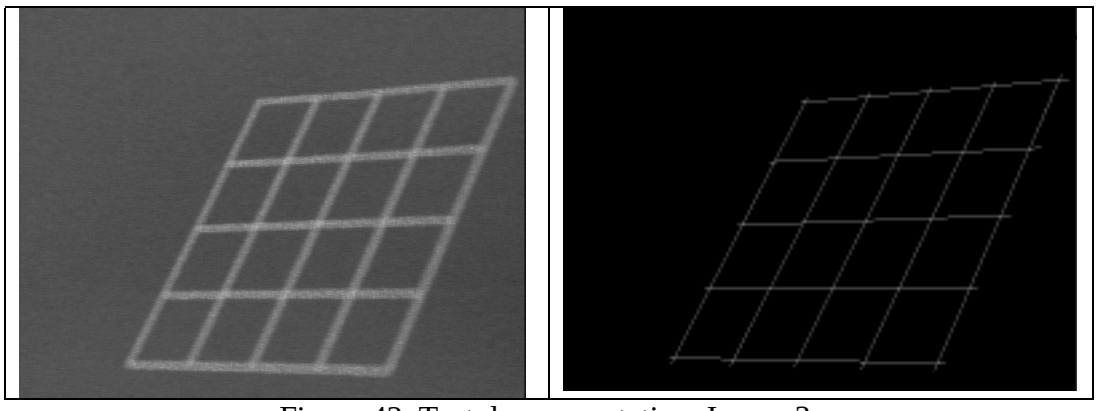


Figure 42. Test de segmentation. Image 3.

	LS-Manu		LS-Insp		Manu-Insp	
Moyenne	0,99	0,20	2,45	1,54	1,05	1,43
Ecart-type	7,76	2,58	8,24	2,31	1,07	1,69

Distance moyenne	1,01	2,89	1,77

Table 6. Comparaison des méthodes de segmentation. Image 3.

Quatrième exemple

La grille subit cette fois une transformation supplémentairte : les carrés deviennent ici des trapèzes. Les remarques notées précédemment sur la méthode **Insp** restent valables et sont même renforcées. Les conclusions quant aux écarts respectifs des trois méthodes et la validité de la méthode **LS** peuvent être tenues pour valides sur cet exemple également.

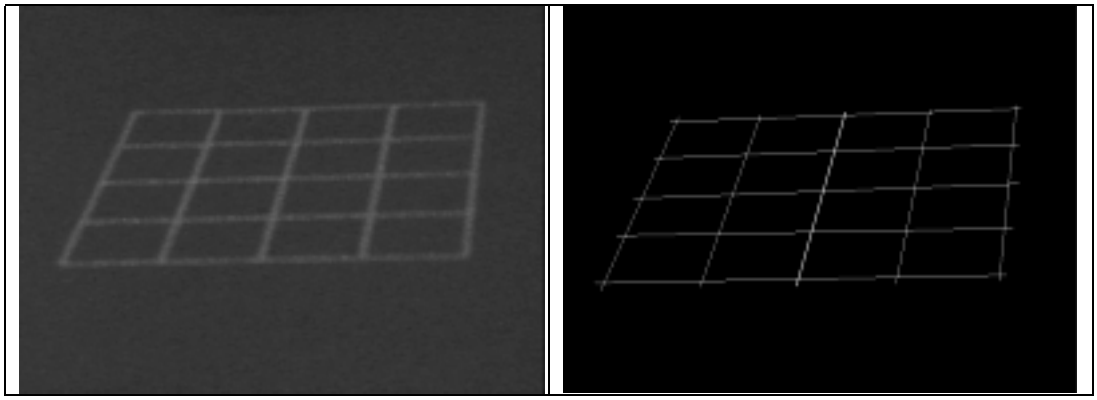


Figure 43. Test de segmentation. Image 4.

	LS-Manu		LS-Insp		Manu-Insp	
Moyenne	-0,50	0,24	0,60	0,77	1,42	0,08
Ecart-type	1,39	1,63	0,80	1,51	1,08	1,51

Distance moyenne	0,56	0,97	1,42

Table 7. Comparaison des méthodes de segmentation. Image 4.

Statistiques globales et conclusion

D'une manière générale, on peut constater que les écarts entre les trois méthodes comparées ici sont relativement faibles ; notre méthode ne s'écartant pas plus d'un pixel des deux autres, statistiquement. On sait aussi que la méthode **Insp** ne peut être utilisée sur les images en lumière structurée dont nous disposons. Il est

illusoire, bien entendu, de s'attacher à l'une ou l'autre des méthodes et prétendre qu'elle fournit une mesure "absolue" ou même "plus précise". Cependant, ces tests nous ont permis de mesurer la dérive de notre méthode par rapport à une méthode déja connue (**Insp**) et à une autre qui possède l'avantage d'être robuste au bruit, aux déformations et aux changements d'illumination, nous donnant une mesure certes difficilement analysable, mais toujours "proche de la vérité" (**Manu**). Les résultats obtenus, sur des scènes polygonales, valident notre méthode en termes d'efficience et de précision.

	LS-Manu		LS-Insp		Manu-Insp	
Moyenne	-0,09	-0,53	0,73	0,56	0,76	1,02
Ecart-type	2,69	1,41	2,61	1,26	0,77	1,17

Distance moyenne	0,54	0,92	1,27

Table 8. Comparaison des méthodes de segmentation. Statistiques globales.

2.4.2 Décodage du Patron de Lumière

Dans cette section, nous présentons des résultats obtenus pour l'algorithme de décodage. Chaque figure présentée est scindée en deux parties : la colonne de gauche représente les résultats obtenus sur une image et la colonne de droite, les résultats obtenus sur une autre image ; on y trouve, de haut en bas, l'image en lumière structurée, la classification donnée par l'algorithme et le mouvement des centres de classe. Analysons les résultats de la Figure 44, à gauche. Nous avons mesuré dix points différents par primitive de couleur ; les points ont été choisis de manière à couvrir toute l'étendue de l'image et le plus d'éléments de motif possible. Aucun point du sol dont l'obscurité est trop importante n'a cependant été mesuré. Les soixantes points mesurés ont été classés convenablement. Il semble n'y avoir qu'un seul point dans la classe magenta, en fait, les dix points mesurés sont très proches les uns des autres. Ceci est dû au fait qu'on rencontre peu d'éléments de motif de cette couleur dans l'image et que les points mesurés sont très rapprochés dans l'image et, conséquemment, dans l'espace des couleurs. Le mouvement des centres de classe nous montre que l'initialisation permet d'aller du "plus" discriminant (plus on s'éloigne de l'origine, plus les couleurs peuvent être discriminées facilement) vers le "moins" discriminant ; c'est un avantage puisque le mouvement inverse serait susceptible de faire dévier les membres d'une classe vers une classe voisine. On constate également que les centres de classe restent à tout moment assez éloignés les uns des autres.

Sur la Figure 44, à droite cette fois, la même démarche a été entreprise à la seule différence que des points appartenant à des éléments de motif mal définis (sur le sol, par exemple) ont été pris en compte. On constate un mouvement des centres

de classe similaire au précédent exemple bien que plus proches du centre de gravité des points (donc, au pouvoir discriminant affaibli). On peut interpréter le mauvais classement du point cyan parmi les bleus (le point cerclé de la figure centrale) comme une conséquence de ce fait. Pour le reste, les résultats sont similaires à ceux de l'exemple précédent.

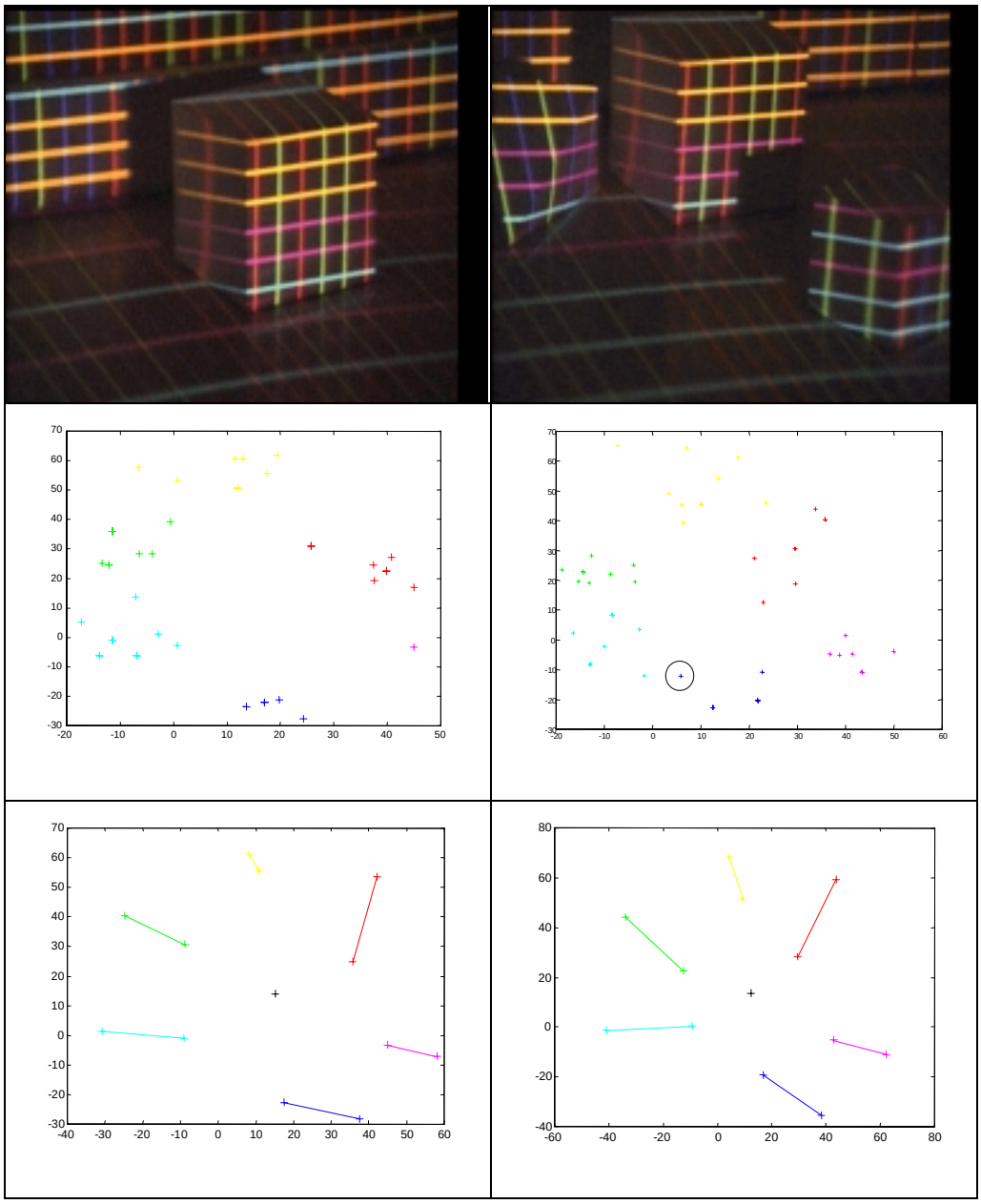


Figure 44. Décodage sur des images aux couleurs bien discriminées

Voyons maintenant les résultats obtenus en Figure 45, à gauche. A l'oeil, on constate que le bleu et le magenta sont difficilement discriminables et qu'il en est de même pour le vert et le jaune. Les résultats donnés par l'algorithme corroborent cette impression visuelle. Si la classe rouge est entièrement reconnue, deux points verts sont classés parmi les jaunes, huit points magentas passent chez les bleus et deux points magentas chez les cyans. En analysant le mouvement des centres de classe, on voit que le centre de la classe jaune se dirige clairement vers la classe verte, ce qui explique les confusions générées par l'algorithme. D'autre part, le magenta est complètement absorbé par le bleu et n'est pas reconnu. En fait, l'algorithme donne des résultats tout à fait comparables à l'interprétation visuelle que l'on peut avoir des images. Pour rendre la reconnaissance des couleurs plus robustes, on peut introduire une heuristique à l'algorithme. En effet, si l'on connait la géométrie du système, on peut en déduire par exemple, que les verticales de l'image sont rouges, vertes ou bleues et en aucun cas cyanes, magentas ou jaunes ; et inversement que les horizontales sont cyanes, magentas ou jaunes et en aucun cas rouges, vertes ou bleues. Les erreurs obtenues par l'algorithme avec six primitives de couleur seraient corrigées dans leur presque intégralité en ne prenant que trois primitives de couleur (on obtiendrait des résultats similaires à ceux de la figure précédente).
Sur la Figure 45, à droite, on dispose d'une image proche de l'image voisine : le jaune et le vert sont très proches, le bleu et le magenta également ; seuls le rouge et le cyan sont facilement reconnaissables. Les résultats donnent trois points verts classés parmi les jaunes ; tous les points magentas absorbés par la classe bleue ; un point rouge classé parmi les magentas. Si l'on se penche sur le mouvement des centres de classe, on voit que, comme précédemment, le jaune se rapproche du vert et que le bleu absorbe le magenta. L'effet de cette absorption est de dévier la classe magenta vers le rouge, ce qui explique la mauvaise classification d'un des points rouges. Pour le reste, les mêmes conclusions que sur l'exemple précédent peuvent être tirées. En décorrelant la reconnaissance du rouge, du vert et du bleu à celle du cyan, du magenta et du jaune, on devrait obtenir une classification robuste et fiable.

En conclusion, il est possible d'avancer que le décodage du motif est réalisable et fiable même si tout le processus de formation de la couleur n'est pas pris en compte. Les équivoques éventuelles entre les couleurs, comme celles de la Figure 45, peuvent être corrigées par des hypothèses simples et peu contraignantes sur la géométrie du système. L'algorithme de coalescence, initialisé grâce à la connaissance *a priori* du positionnement relatif des classes, donne un classement rapide des couleurs du motif perçu.

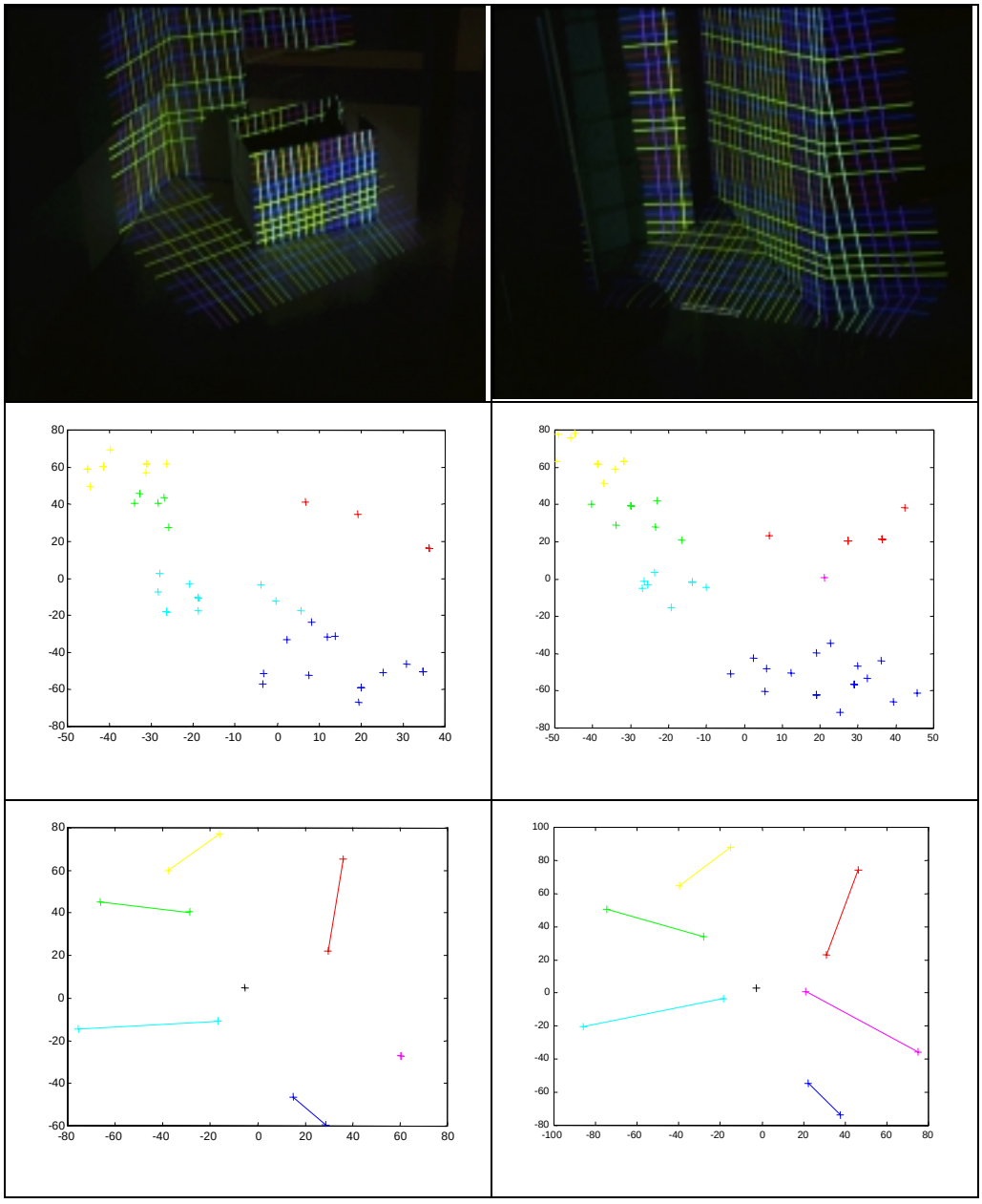


Figure 45. Décodage sur des images aux couleurs mal discriminées

2.5 Conclusion

Ce chapitre décrit la chaîne des traitements nécessaires à la segmentation et au décodage des images en lumière structurée. Nous avons montré que la précision donnée pour l'extraction des points d'intérêt du motif était satisfaisante et qu'un décodage fiable était possible sans prise en compte du processus complet de formation de la couleur. La méthode de segmentation est bien évidemment perfectible, mais nous avons pu établir qu'elle était suffisante pour traiter la plupart des images que le robot acquiert lors d'une mission. Dans la stratégie de détection d'obstacles sur le plan du sol[1] que nous utilisons et qui sera décrite au quatrième chapitre, les images obtenues possèdent une morphologie adaptée à ce type de traitements : nous rencontrerons souvent un ou deux objets, filmés d'assez près pour que l'approximation polygonale soit valide, et un pan de mur. Ajoutons qu'un objet trop éloigné, puisque l'énergie lumineuse du projecteur se dissipe avec la distance, ne sera pas convenablement perçu par le système de vision. Il faut également considérer le fait que la performance d'un tel capteur est meilleure pour des champs de vision restreints et sur des distances de prise de vue limitées. Nous avons notamment proposé une méthode de seuillage dynamique originale adaptée aux images en lumière structurée. La perte d'intensité lumineuse, la différence d'albedo des surfaces, leur teinte, fait que les éléments de motif apparaissent avec beaucoup de différences d'une région à l'autre de l'image, même voisines. Un seuillage global ne permet donc pas de faire ressortir l'ensemble du motif, ainsi, une perte d'information importante altère les images. Un seuillage moyennant, sur les lignes et les colonnes de l'image, permet d'atténuer les disparités d'intensité lumineuse de l'image, et d'adapter la valeur du seuil en chaque point à un voisinage restreint. Des différences notables, entre le seuillage global et le seuillage adaptatif que nous proposons, peuvent être soulignées. Une remarque peut être faite pour l'ensemble du processus de segmentation des images : les différents seuils (accumulateur de Hough, fenêtre de filtrage de Hough, taille des segments, etc.) devront être ajustés au type d'images que l'on désire analyser et aux conditions de prise de vue.

Les différentes expérimentations effectuées montrent qu'il est possible de décoder le motif structurant à partir d'une seule image en utilisant un algorithme de coalescence classique. L'idée est d'abord de convertir l'image RVB fournie par la caméra dans un espace des couleurs proche de la perception humaine, le CIE-Lab, puis d'utiliser une connaissance *a priori* sur la position relative des classes de couleurs dans cet espace pour initialiser l'algorithme. Nous avons montré que dans certaines conditions de prise de vue, l'algorithme pouvait être mis en défaut. Il est par exemple courant que le jaune et le vert ou le bleu et le magenta soient difficilement distinguables. En théorie, chaque couleur primaire est susceptible d'être confondue avec l'une des deux couleurs secondaires qui l'avoisinent, et vice-

[1] Ground plane obstacle detection

versa. Mais, puisque les couleurs primaires sont utilisées pour coder les éléments verticaux du motif et les couleurs secondaires pour coder les éléments horizontaux, il est possible de découpler leur décodage et, ainsi, d'en rendre l'algorithme très robuste. Un point important de la méthode de traitement de nos images est la séparation de la segmentation, réalisée sur une image en niveaux de gris, et du décodage, réalisée sur une image des couleurs. On pourrait ainsi imaginer, dans un futur proche, un traitement des images sur une architecture parallèle de segmentation et de décodage.

Chapitre 3 :
La Reconstruction 3-D et sa Géométrie

3.1 Introduction

On dit qu'un capteur stéréoscopique est calibré quand il est possible d'inférer une reconstruction euclidienne de la scène à partir des mesures qu'il fournit ; il en est de même pour un capteur de vision en lumière structurée. L'objectif de ce chapitre est d'étudier les possibilités offertes par la lumière structurée pour la reconstruction, d'en débusquer les points faibles et d'en proposer des solutions pratiques ; sans perdre de vue les enjeux robotiques de cette étude. Pour conférer une quelconque autonomie à un robot mobile, il faut s'affranchir, autant que faire se peut, des étapes nécessitant l'intervention d'un opérateur (comme la calibration) et permettre au capteur de pouvoir s'auto-adapter aux changements de l'environnement ou à des techniques d'observation et d'exploration. Ceci nous a conduit à l'élaboration d'une méthode originale d'auto-calibration (ou de reconstruction sans-calibration, selon la terminologie utilisée) basée sur la stratification des géométries : dans un premier temps une reconstruction projective de la scène est effectuée, puis cette reconstruction est redressée dans l'espace euclidien grâce aux propriétés géométriques de la projection d'un patron de lumière. Nous mettons en évidence quelques propriétés de notre patron de lumière permettant de classer un ensemble de points en "colinéaires dans l'espace", "coplanaires" ou "quelconques" à partir de la conservation du birapport dans l'image.

Nous allons décrire, dans une première partie de ce chapitre, le cadre conceptuel et les méthodes développées pour la reconstruction euclidienne d'une scène à partir d'un capteur de vision en lumière structurée. Tout d'abord, nous détaillerons comment un tel système est modélisé et calibré. Nous nous arrêterons sur la calibration de la source lumineuse, qui peut être spécifique, et nous expliquerons une technique simple de triangulation pour l'obtention des coordonnées tri-dimensionnelles de la scène observée. Nous introduirons également les principes de la géométrie épipolaire, son rôle joué dans les techniques d'auto-calibration en faisant un outil précieux, à mi-chemin entre la calibration forte et l'auto-calibration (on parle d'ailleurs de calibration faible pour les systèmes dont la géométrie épipolaire est connue).
La seconde partie de ce chapitre traite de la reconstruction à partir d'un capteur de vision non-calibré (en fait, non-calibré *fortement*). Nous passons d'abord en revue les principales méthodes de reconstruction sans calibration, qui permettent une reconstruction projective de la scène, ainsi que quelques méthodes de reconstruction affine et d'auto-calibration. Puisqu'il s'agit d'adapter les outils de la

stéréovision à la lumière structurée, nous avons voulu prendre en compte un grand nombre de méthodes ne nécessitant ni mire de calibration, ni intervention humaine. Ces méthodes ont été évaluées en deux points : leur adaptation possible à la vision en lumière structurée, leur compatibilité avec la vision adaptative. Ceci nous a permis de définir un certain nombre de contraintes inhérentes à la lumière structurée et de dégager une méthode (il y en a d'autres) permettant de reconstruire projectivement la scène à partir de notre capteur (il s'est avéré que les méthodes d'auto-calibration "directes" ne sont pas exploitables). Cette reconstruction projective est ensuite redressée en une reconstruction euclidienne grâce à la génération de contraintes euclidiennes que permet la projection d'un patron de lumière. Des résultats expérimentaux viennent valider la méthode en fin de chapitre. Un rappel des éléments de géométrie pouvant être utiles à la compréhension de ce chapitre est proposé en Annexe B.

3.2 Modélisation, Calibration et Reconstruction

3.2.1 La Caméra

Il existe de nombreux modèles pour décrire le comportement géométrique et/ou optique d'une caméra. Nous les avons regroupés en Annexe A. Dans les sections suivantes, nous décrirons le modèle projectif dit *sténopé*. C'est le plus utilisé par la communauté en vision par ordinateur car il offre un très bon compromis précision/simplicité.

Le modèle sténopé

Une caméra est définie par un centre optique C et un plan rétinien $\Re$. Le point principal P_0 est la projection orthogonale de C sur $\Re$. La distance $P_0 C$ est appelée *distance focale*. L'image m d'un point M sur la rétine est donnée par l'intersection de la ligne de vue $\langle C, M \rangle$ avec $\Re$. Si nous considérons $\Re$ comme un plan projectif et l'espace ambiant comme un espace projectif, cette transformation peut s'exprimer linéairement en les coordonnées homogènes [FAU93].
Détaillons plus avant ce modèle. La scène que nous désirons analyser, mesurer ou reconnaître a pour système de coordonnées le repère du monde $\{w\}$. Le processus de formation des images que nous allons détailler nous permet de déterminer la relation mathématique entre le repère de monde et le repère de l'image $\{i\}$ où les points sont exprimés en pixels. Un repère $\{c\}$, dont l'origine est O_c , appelé *centre optique* (*ou centre de projection* ou *point focal*), est attaché à la caméra. (O_c , x_c) et (O_c , y_c) sont les axes parallèles au *plan image* (ou au *plan rétinien*[2]), correspondant aux directions des lignes et colonnes des pixels. L'axe (O_c, z_c),

[2] La différence entre le plan image et le plan rétinien vient du fait que dans le plan image, les coordonnées sont exprimées en pixels et dans le plan rétinien, on dispose encore de coordonnées métriques.

orienté vers la scène et perpendiculaire au plan image, est appelé *axe optique*. Pour passer du repère $\{w\}$ au repère $\{i\}$, trois étapes sont nécessaires :

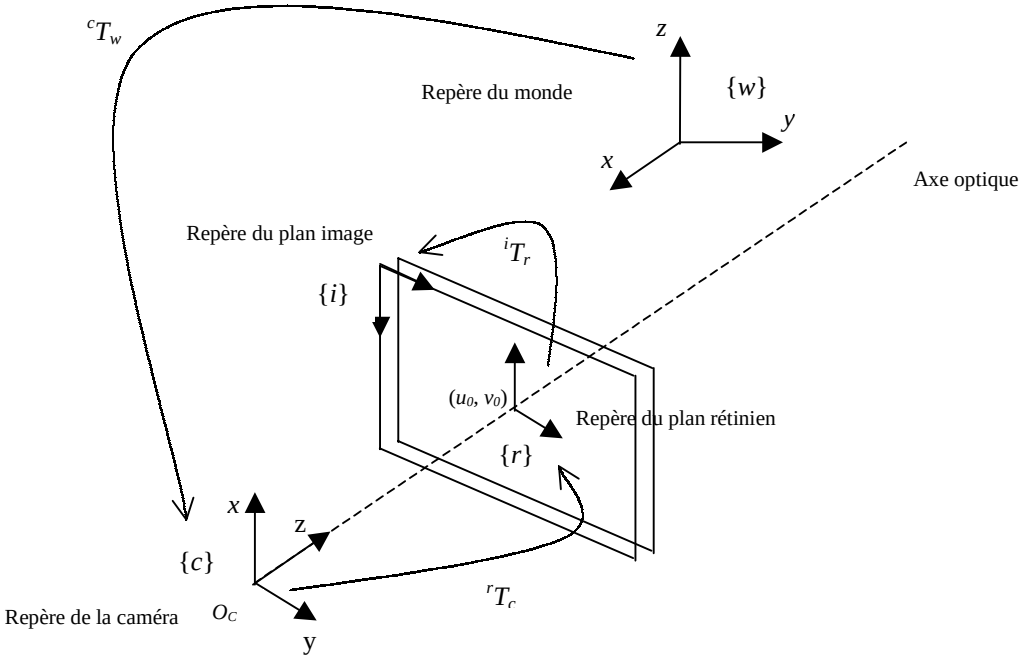


Figure 46. Géométrie de la prise de vue

- *Un mouvement rigide* : pour passer du repère du monde $\{w\}$ au repère attaché à la caméra $\{c\}$ trois rotations et trois translations par rapport aux trois axes sont nécessaires :

$$^c\mathbf{T}_w = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{18}$$

où $\mathbf{R} = \mathbf{R}_z \cdot \mathbf{R}_y \cdot \mathbf{R}_x = \begin{bmatrix} \cos\lambda & \sin\lambda & 0 \\ -\sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$ est la matrice combinant les trois rotations et $\mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$ est le vecteur de translation. Cet ensemble de paramètres regroupe les paramètres extrinsèques.

- *Une projection 3-D / 2-D* : cette projection dépend du modèle de la caméra (voir Annexe A). Le modèle sténopé étant une bonne approximation du comportement géométrique d'une caméra, il est couramment utilisé en vision par ordinateur. La matrice de projection est alors :

$$ {}^{r}\mathbf{T}_{c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/\gamma & 0 \end{bmatrix} \tag{19} $$

Elle décrit la transformation qui permet le passage du repère du monde au repère de la rétine $\{r\}$.

- *Un changement de coordonnées* : il transforme les points 2-D de l'observation de coordonnées métriques en coordonnées pixels ; il gère le passage du repère {r} au repère {i}.

$$ \mathbf{A} = {}^{i}\mathbf{T}_{r} = \begin{bmatrix} k_u & -k_u cot\theta & u_0 \\ 0 & k_v \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{20} $$

où $k_u \times k_v$ represente la taille d'un pixel et (u_0 , v_0) les coordonnées du point principal (projection du centre optique sur la rétine suivant l'axe optique). θ est l'angle formé par les deux axes du plan image ; on prend souvent l'hypothèse que ces axes sont perpendiculaires et on pose $\theta = \pi/2$. On obtient donc :

$$ \mathbf{A} = {}^{i}\mathbf{T}_{r}\,{}^{r}\mathbf{T}_{c} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{21} $$

où $\alpha_u = \gamma k_u$ et $\alpha_v = \gamma k_v$ sont respectivement les facteurs d'échelle horizontale et verticale. Les quatre paramètres de cette matrice sont les paramètres intrinsèques.

En résumé, avec $\mathbf{P} = {}^{i}\mathbf{T}_{r} \cdot {}^{r}\mathbf{T}_{c} \cdot {}^{c}\mathbf{T}_{w}$, on obtient pour un point *M* de l'espace projeté en *m* sur le plan image :

$$ \lambda \cdot \mathbf{m} = \mathbf{P} \cdot \mathbf{M} \text{ ou } \mathbf{m} \approx \mathbf{P} \cdot \mathbf{M} \tag{22} $$

$$\begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} = \mathbf{P} \cdot \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \tag{23}$$

où λ représente le facteur d'échelle.

La calibration forte

La calibration consiste à estimer les paramètres intrinsèques et extrinsèques d'un modèle de caméra à partir d'un ensemble de points 3-D et de leur image. Il s'agit, globalement, d'estimer les éléments de la matrice **P** (22). La technique que nous allons décrire est appelée *calibration forte*, en opposition aux techniques de *calibration faible* et d'*auto-calibration* que nous verrons dans la section 3.4.
Pour estimer les paramètres de la caméra, il faut mettre en correspondance n points 3-D avec leur projection sur le plan image et identifier la matrice qui permet le passage d'un jeu de coordonnées à l'autre. Autrement dit, estimer la matrice **P** de l'équation (22) à partir de la mise en correspondance $\mathbf{m}_i \leftrightarrow \mathbf{M}_i, 1 \le i \le n$. On utilise généralement une mire de calibration (Figure 47) à laquelle est attaché le repère de la scène. Les coordonnées des sommets des quadrilatères de la mire sont mesurées de manière très précise ; ce sont les points 3-D dont on aura besoin. La mire est observée par une caméra et ceci nous donne les projections correspondant aux points 3-D.

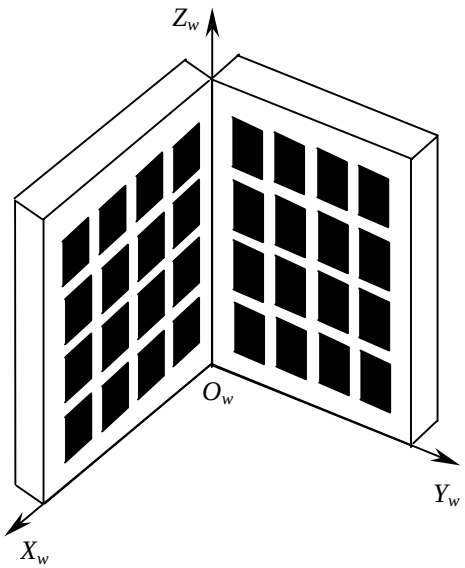


Figure 47. Mire de calibration

Soit le point $\mathbf{M}_i$ de coordonnées $\begin{pmatrix} x_i & y_i & z_i & 1 \end{pmatrix}^T$ et son projeté sur le plan image $\mathbf{m}_i$ de coordonnées $\begin{pmatrix} u_i & v_i & 1 \end{pmatrix}^T$. On a :

$$\lambda \cdot \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{24}$$

En développant cette équation et en éliminant le facteur d'échelle, on obtient :

$$p_{11}x_i - p_{31}u_i x_i + p_{12}y_i - p_{32}u_i y_i + p_{13}z_i - p_{33}u_i z_i + p_{14} - p_{34}u_i = 0 \tag{25}$$

$$p_{21}x_i - p_{31}v_i x_i + p_{22}y_i - p_{32}v_i y_i + p_{23}z_i - p_{33}v_i z_i + p_{24} - p_{34}v_i = 0 \tag{26}$$

De nombreux auteurs, dont Faugeras et Toscani [FAU86], supposent connu le dernier élément de la matrice de projection et le posent égal à 1. Les équations s'en trouvent simplifiées. Plaçons les inconnues dans un vecteur ligne, si *i* varie de 1 à *n*, on a :

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1x_1 & -v_1y_1 & -v_1z_1 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 & -u_2z_2 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v_2x_2 & -v_2y_2 & -v_2z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 & 0 & 0 & 0 & 0 & -u_nx_n & -u_ny_n & -u_nz_n \\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -v_nx_n & -v_ny_n & -v_nz_n \end{bmatrix} \cdot \begin{pmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{pmatrix} \tag{27}$$

Pour résoudre le système d'équations (27), six points 3-D non-coplanaires et leurs projetés sur le plan image sont nécessaires, puisque chaque point nous donne deux équations. En pratique, on utilisera évidemment plus de six points pour réduire les effets des erreurs de mesure. On peut citer ici les principales méthodes de calibration : celle de Hall [HAL82] et de Faugeras-Toscani [FAU86], de Tsai [TSA87] et de Weng [WEN92] qui prennent en compte les distorsions

Les distorsions

Si le modèle décrit plus haut est suffisant pour de nombreuses applications, il ne prend pas en compte plusieurs caractéristiques des objectifs à lentilles : le flou et les distorsions, notamment. Nous rappelons que la distorsion est un phénomène optique qui provoque la déformation des lignes droites d'une image. Pour le flou, il faut abandonner le modèle sténopé au profit du modèle à lentille, mince ou épaisse (voir Annexe A). Pour modéliser les distorsions, pouvant être provoquées par des imperfections dans l'assemblage ou la conception des lentilles, une simple adjonction de paramètres au modèle sténopé est suffisante [TSAI87, WEN92]. On en compte deux types principaux : les distorsions *radiale* et *tangentielle*.
Considérons les coordonnées $(\hat{u},\hat{v})$ d'un point mésuré dans l'image, et les coordonnées idéales (u,v) de ce point si la projection suivait parfaitement le modèle décrit précédemment. Les distorsions de la lentille ont pour effet de déplacer la projection idéale comme indiqué par l'équation suivante :

$$\begin{cases} u = \hat{u} + d_u \\ v = \hat{v} + d_v \end{cases} \tag{28}$$

Modéliser les distorsions revient à estimer les équations qui régissent ce déplacement ; en d'autres termes, déterminer à quoi d_u et d_v sont égaux.

- *La distorsion radiale* : elle est principalement due à un défaut de courbure de la lentille. Elle déplace, symétriquement par rapport à l'axe optique, les projections vers l'intérieur ou, au contraire, vers l'extérieur, en donnant à l'image un aspect *coussinet* ou *barillet* (Figure 48). Elle est modélisée par les équations suivantes, où r est la distance radiale qui sépare la projection observée du point principal et les k_i représentent les coefficients de la distorsion :

$$\begin{cases} d_u = \hat{u} \cdot \left(k_1 r^2 + k_2 r^4 + \ldots + k_n r^{2n}\right) \\ d_v = \hat{v} \cdot \left(k_1 r^2 + k_2 r^4 + \ldots + k_n r^{2n}\right) \end{cases} \tag{29}$$
$$\text{avec} \quad r = \sqrt{\hat{u}^2 + \hat{v}^2}$$

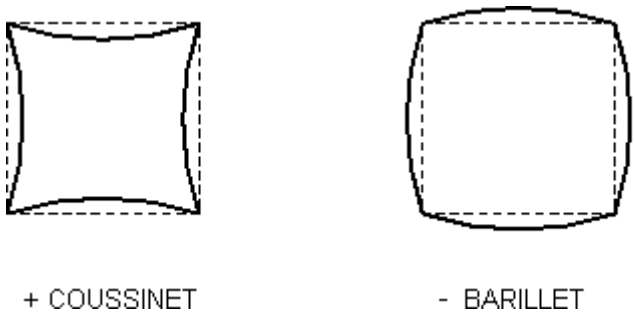


Figure 48. Distortions radiales

Souvent, on s'arrêtera au coefficient d'ordre 1 de la distorsion radiale, puiqu'il semble que la prise en compte de coefficients d'ordre supérieur n'apporte pas d'amélioration très significative.

- *La distorsion tangentielle* : elle permet de corriger les effets de décentrage des lentilles (non-colinéarité des axes optiques des lentilles qui composent l'objectif). De nombreux auteurs, dont Tsai [TSA87], considèrent que la distorsion tangentielle est négligeable par rapport à la distorsion radiale ; elle n'est donc pas modélisée. De surcroit, l'élaboration d'un modèle trop compliqué pour décrire les distorsions ne conduit pas forcément à une précision accrue, mais peut provoquer une instabilité numérique au moment du calcul des paramètres de calibration.

3.2.2 Le Projecteur

Nous appelons projecteur la source émettant la lumière structurée sur la scène. Il en existe de différents types (projecteur classique, vidéo-projecteur, laser) et de différentes formes. Sans prétendre à l'exhaustivité, nous en donnons quelques modèles fondamentaux et nous présentons quelques méthodes permettant de les calibrer.

Des modèles pour le projecteur

Géométriquement, un projecteur peut être vu comme une caméra travaillant "à l'envers" ; le sens de la ligne de vue étant inversé : de la scène vers le plan image pour la caméra, de l'image vers la scène pour le projecteur. Le modèle de la caméra détaillé plus haut, statique (le temps n'est pas pris en compte), décrit alors tout aussi bien le comportement du projecteur. C'est donc, dans le cas général, la même équation qui va servir à modéliser le projecteur et la caméra (22). Si toutefois l'on projette un unique faisceau laser (point de surbrillance) ou un plan de lumière, on peut considérablement simplifier ce modèle.
Prenons le cas du point de surbrillance. Les seuls paramètres qu'il est nécessaire d'estimer pour la reconstruction sont ceux du rayon lumineux allant éclairer un

point de la surface de l'objet à mesurer ; ce sont les paramètres d'une droite dans l'espace :

$$\begin{cases} a_1 x + b_1 y + c_1 z + d_1 = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \end{cases} \quad (30)$$

En plaçant judicieusement le projecteur par rapport à la caméra, il est possible de simplifier ces équations. Imaginons que le faisceau soit confiné dans un plan formé par deux des trois axes du repère de la caméra. On peut ainsi ôter une dimension au modèle et ne considérer que l'équation d'une droite dans un plan. Par exemple, si le laser appartient au plan défini par les axes x_c et z_c :

$$x_c = az_c + b \quad (31)$$

De même, en considérant un repère placé sur le faisceau, l'axe des *z* longeant celui-ci, les points mesurés dans ce repère sont de la forme $(0 \quad 0 \quad z \quad 1)^T$. Et seul le déplacement par rapport à la caméra ou au repère du monde reste à estimer [CHA97B].

De la même manière, si c'est un plan de lumière que l'on projette, on aura à estimer les paramètres de ce plan dans l'espace pour modéliser la projection :

$$ax + by + cz + d = 0 \quad (32)$$

Cette équation doit être exprimée dans le repère du monde pour pouvoir être combinée avec le modèle de la caméra [BOL81]. La matrice de déplacement séparant le projecteur de la caméra doit être incluse au modèle. Souvent, puisqu'un balayage de la scène est rendu nécessaire par ce type d'illumination, le déplacement du repère caméra au repère projecteur est variable et dépend de l'angle (ou des angles) d'orientation de ce dernier, ce qui peut conduire à des modèles très élaborés [FOR00].
Pour les motifs structurants représentant un faisceau de lignes parallèles, on considère parfois celui-ci comme étant un patron de lumière (il est dans ce cas modélisé à la manière d'une caméra) [SAL98] ou comme étant une projection de plusieurs plans de lumière (dans ce cas, chaque plan est modélisé isolément) [MCI95]. Notons aussi que les distorsions peuvent être également ajoutées au modèle du projecteur quand celui-ci est décrit par la même équation qu'une caméra.

La calibration du projecteur

La calibration du projecteur, comme celle de la caméra, consiste à estimer les paramètres du modèle. L'étape la plus délicate ici est la mesure des points 3-D sur lesquels le motif est projeté. Relever à la main les coordonnées d'une trace lumineuse est loin de garantir une précision suffisante, mais c'est souvent le seul moyen dont on dispose. On trouve, dans la littérature, assez peu de mode opératoire pour la mesure de ces points. On peut, comme dans [SAL98], recouvrir une mire de calibration d'un papier millimétré sur lequel on viendra pointer les traces lumineuses. Cette méthodologie reste valable quelle que soit la forme du motif et semble la plus employée. Il existe cependant d'autres méthodes, plus spécifiques, que nous allons citer brièvement.
Pour les plans de lumière (simple ou multiple), il est très courant de calibrer le projecteur et la caméra dans une même opération [FAU88]. En combinant les équations (22) et (32), on obtient une expression de la forme :

$$\mathbf{AX} = \mathbf{B} \tag{33}$$

Avec :

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ p_{11} - up_{31} & p_{12} - up_{32} & p_{13} - up_{33} \\ p_{21} - vp_{31} & p_{22} - vp_{32} & p_{23} - vp_{33} \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} d \\ up_{34} - p_{14} \\ vp_{34} - p_{24} \end{bmatrix} \quad \text{et} \quad \mathbf{X} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{34}$$

Où p_{ij} est l'élément de l'$i^{ème}$ ligne et $j^{ème}$ colonne de la matrice **P**. En inversant **A**, on peut obtenir l'expression des coordonnées tri-dimensionnelles en fonction des coordonnées pixels et des paramètres de calibration à estimer.
En supposant que la projection suit un modèle orthographique, Proesmans, Van Gool et Oosterlinck [PRO96] ont démontré que la reconstruction tri-dimensionnelle de la scène pouvait être obtenue si l'angle entre la direction de projection et de prise de vue était connu. La "calibration", très particulière ici, consistera à observer une mire de calibration entièrement blanche et dont l'angle formé par les deux plans est précisément connu. Nous ne reviendrons pas sur la méthode de Sotoca, Buendia et Iñesta [SOT00] qui permet la calibration pour la mesure de larges surfaces à partir de la projection du patron de lumière sur deux plans, puisqu'elle a été présentée dans le première chapitre.
Nous finirons par la méthode proposée par Huynh, Owens et Hartman [HUY99], développée pour la projection d'un plan de lumière, mais qui semble adaptable à

des motifs plus complexes. Quatre ensembles de trois points colinéaires sont placés sur les différents plans d'une mire de calibration ; les coordonnées de ces douze points étant parfaitement connues. Quand le plan de lumière vient couper ces quatre droites, l'intersection nous donne un quatrième point : le birapport des trois points et de l'intersection sur la mire (de ce fait, dans le repère du monde) est égal au birapport de l'image de ces quatre points. On dispose ainsi d'une mesure 3-D des traces lumineuses relativement précises (à la stabilité du birapport près). Cette méthode ne peut bien sûr être appliquée que lorsque la projection du motif structurant ne cache pas les points de la mire de calibration (comme c'est le cas avec le patron lumineux que nous utilisons).

3.2.3 L'Ensemble Caméra-Projecteur

La géométrie épipolaire

Si l'on considère maintenant deux caméras (ou une caméra et un projecteur), des propriétés géométriques remarquables entre les images capturées par l'une et l'autre caméra peuvent être établies [FAU93]. Ces propriétés sont indépendantes de la scène observée et des caméras qui l'observent. C'est la notion de *géométrie épipolaire*, essentielle en vision par ordinateur.
Soit un pixel *m* de la première image, projection du point objet *M*, et la ligne de vue $\langle C, M \rangle$ qu'il détermine avec le centre optique. Le pixel *m'*, projection du même point *M* dans la seconde image, se trouve nécessairement sur l'image de $\langle C, M \rangle$ (voir Figure 49). Cette droite est appelée *droite épipolaire* de *m* et est notée l'_m . De la même manière, l'image de $\langle C, M \rangle$ dans la première image est la droite épipolaire de *m'* notée $l_{m'}$.
La projection de *C'* dans la première image nous donne le point *e* et la projection de *C* dans la seconde image, le point *e'* . Ces deux points sont appelés *épipoles.* Ils sont à l'intersection du faisceau des droites épipolaires de l'une et de l'autre image.

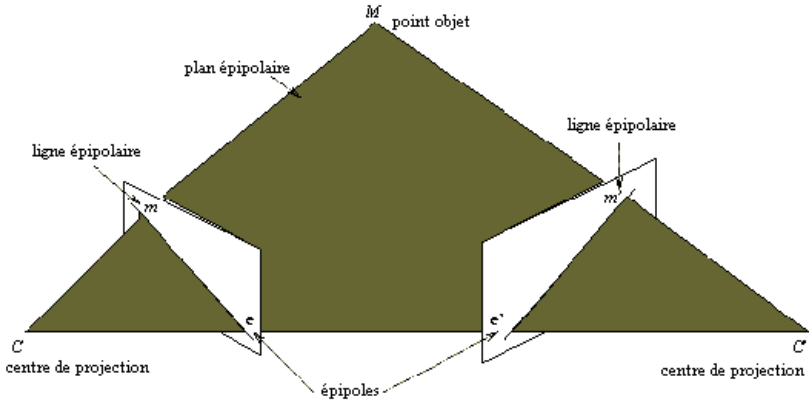


Figure 49. La géométrie épipolaire

Par définition, les coordonnées des épipoles sont données par les équations suivantes :

$$\mathbf{e} = \mathbf{P} \cdot \begin{pmatrix} C' \\ 1 \end{pmatrix} \mathbf{e}' = \mathbf{P}' \cdot \begin{pmatrix} C \\ 1 \end{pmatrix} \qquad (35),(36)$$

La matrice fondamentale

Plaçons la première caméra à l'origine du repère de la scène. La seconde caméra (ou le projecteur) est déplacée par rapport à ce repère d'une rotation **R** et d'une translation **t**. Les équations de projection deviennent alors :

$$\mathbf{m} = \mathbf{A} \cdot [\mathbf{I} \quad \mathbf{0}] \cdot \mathbf{M} \qquad (37)$$

$$\mathbf{m}' = \mathbf{A}' \cdot [\mathbf{R} \quad \mathbf{t}] \cdot \mathbf{M} \qquad (38)$$

En éliminant **M** dans les équations, on obtient :

$$\mathbf{m}'^{T} \cdot \mathbf{A}'^{-T} \cdot [\mathbf{t}]_{\times} \cdot \mathbf{R} \cdot \mathbf{A}^{-1} \cdot \mathbf{m} = 0 \qquad (39)$$

$$\mathbf{F} = \mathbf{A}'^{-T} \cdot [\mathbf{t}]_{\times} \cdot \mathbf{R} \cdot \mathbf{A}^{-1} \qquad (40)$$

$$\mathbf{m}'^{T} \cdot \mathbf{F} \cdot \mathbf{m} = 0 \qquad (41)$$

Où $[\mathbf{t}]_{\times}$ est la matrice anti-symétrique de **t**, telle que $[\mathbf{t}]_{\times} \cdot \mathbf{R} = \mathbf{t} \times \mathbf{R}$ ($\times$ est le produit vectoriel).

$$\mathbf{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \Rightarrow [\mathbf{t}]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \qquad (42)$$

La matrice **F** ainsi obtenue est appelée *matrice fondamentale*. C'est une matrice 3×3, de rang 2 (son détermimant est nul) définie à un coefficient multiplicatif près. *Toute la géométrie épipolaire est contenue dans cette matrice*. Elle relie un point de la première image à sa ligne épipolaire dans la seconde image :

$$\mathbf{l}'_m = \mathbf{F}\mathbf{m} \qquad (43)$$

Si *m* coïncide avec l'épipole *e*, la ligne épipolaire n'est pas définie (**F** n'est pas de rang 3), nous avons donc :

$$\mathbf{F}\mathbf{e} = 0 \qquad (44)$$

Un lien peut être établi entre la matrice fondamentale et la *matrice essentielle* de Longuet-Higgins [LON81]. Cette dernière ne prend pas en compte les paramètres intrinsèques du capteur et exprime uniquement le déplacement d'un repère caméra à l'autre : elle traduit la relation d'un plan image à l'autre dans le cas où les points sont exprimés en coordonnées normalisées.
Si **F** est la matrice fondamentale de la première image vers la seconde et **F'**, la matrice fondamentale de la seconde image vers la première, on peut facilement démontrer que :

$$\mathbf{F}' = \mathbf{F}^T \text{ et } \mathbf{F} = \mathbf{F}'^T \tag{45}$$

Toutes les équations précédentes peuvent être récrites en plaçant ou en ôtant les ' sur les points et les droites, et en transposant la matrice fondamentale.

Algorithme des huit points

Il existe de nombreuses méthodes algébriques pour estimer les coefficients de la matrice fondamentale [LUO93]. Nous allons présenter l'algorithme dit des *huit points*, qui a l'avantage d'être linéaire et qui offre des résultats tout à fait satisfaisant. Si nous développons l'équation (41), nous obtenons :

$$\begin{pmatrix} x_i' & y_i' & 1 \end{pmatrix} \cdot \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = 0 \tag{46}$$

En reformulant cette équation, il est possible d'obtenir :

$$\mathbf{U}_n \mathbf{f} = \mathbf{0} \tag{47}$$

Avec :

$$\mathbf{U}_n = \begin{pmatrix} u_1 & u_2 & \dots & u_n \end{pmatrix}^T \tag{48}$$

$$\mathbf{f} = \begin{pmatrix} F_{11} & F_{12} & F_{13} & F_{21} & F_{22} & F_{23} & F_{31} & F_{32} & F_{33} \end{pmatrix}^T \tag{49}$$

Et :

$$u_i = \begin{pmatrix} x_i x_i' & y_i x_i' & x_i' & x_i y_i' & y_i y_i' & y_i' & x_i & y_i & 1 \end{pmatrix} \tag{50}$$

Ce système admet la solution $\mathbf{f} = \mathbf{0}$, dite *solution triviale*. Cette solution n'est évidemment pas appropriée. Pour l'éviter, on peut imposer des contraintes aux

coefficients de la matrice fondamentale. Le plus simple consiste à fixer l'un de ces coefficients à 1 ; si ce coefficient est effectivement différent de zéro, nous ne rencontrerons aucun problème dans la résolution du système, dans le cas contraire, la matrice fondamentale que l'on obtiendra sera très mal conditionnée. Une solution à ce problème consiste à fixer tour à tour chaque coefficient de la matrice fondamentale à 1 et à ne retenir que le meilleur résultat. La meilleure matrice fondamentale étant définie comme étant celle qui minimise la distance entre les points et leur ligne épipolaire.
Fixons $F_{33} = 1$. L'équation (47) devient alors :

$$\mathbf{U}'_n \mathbf{f}' = -\mathbf{1}_n \tag{51}$$

Avec les modifications correspondantes :

$$\mathbf{U}'_n = \begin{pmatrix} u'_1 & u'_2 & \dots & u'_n \end{pmatrix}^T \tag{52}$$

$$\mathbf{f}' = \begin{pmatrix} F_{11} & F_{12} & F_{13} & F_{21} & F_{22} & F_{23} & F_{31} & F_{32} \end{pmatrix}^T \tag{53}$$

$$u'_i = \begin{pmatrix} x_i x'_i & y_i x'_i & x'_i & x_i y'_i & y_i y'_i & y'_i & x_i & y_i \end{pmatrix} \tag{54}$$

On peut alors résoudre l'équation (51) :

$$\mathbf{f}' = -\left(\mathbf{U}'^T_n \mathbf{U}'_n\right)^{-1} \mathbf{U}'^T_n \mathbf{1}_n \tag{55}$$

L'algorithme des huit points tel qu'il est décrit jusque-là est très sensible au bruit et très instable. Si l'on observe les éléments de u'_i, pour une image 512×512 par exemple, on trouve des valeurs de type *xx'* de l'ordre de 100^2, des valeurs de type *x* de l'ordre de 100 et la dernière valeur fixée à 1. Il est évident que la matrice $\mathbf{U}'^T_n \mathbf{U}'_n$ est très mal conditionnée. Une solution très efficace, proposée par Hartley [HAR95], s'articule autour de deux étapes :

- Les points sont translatés de manière à ce que leur barycentre soit à l'origine du nouveau système de coordonnées (une translation différente pour chaque image).
- Les points sont mis à l'échelle isotropiquement de manière à ce que la distance moyenne à l'origine soit égale à $\sqrt{2}$.

Regroupons ces deux transformations au sein de la matrice **T** pour la première image et de la matrice **T'** pour la seconde image. On obtient :

$$\hat{\mathbf{m}} = \mathbf{T}\mathbf{m} \tag{56}$$

$$\hat{\mathbf{m}}' = \mathbf{T}'\mathbf{m}' \tag{57}$$

On calcule la matrice fondamentale normalisée $\hat{\mathbf{F}}$ pour les couples de points $\hat{\mathbf{m}} \leftrightarrow \hat{\mathbf{m}}'$. On peut facilement établir les relations suivantes entre les deux matrices fondamentales :

$$\hat{\mathbf{F}} = \mathbf{T}'^{-T}\,\mathbf{F}\mathbf{T}^{-1} \tag{58}$$

$$\mathbf{F} = \mathbf{T}'^{T}\,\hat{\mathbf{F}}\mathbf{T} \tag{59}$$

Finalement, l'équation (59) nous permet de calculer les coefficients de la matrice fondamentale d'origine. Hartley démontre que cette simple transformation des coordonnées permet d'obtenir des résultats proches de ceux obtenus grâce aux méthodes itératives.

3.2.4 Reconstruction par Triangulation

Une fois la caméra et le projecteur calibrés, il est possible d'obtenir la reconstruction tri-dimensionnelle de la scène observée. On connait les coordonnées des points capturés par la caméra et les coordonnées de leurs homologues dans le patron (grâce au décodage de celui-ci). Il reste, à partir de ces coordonnées et des matrices de projection, à calculer les lignes de vue correspondantes, et le point d'intersection de ces lignes de vue, pour obtenir les coordonnées spatiales du point observé : c'est ce qu'on appelle la *triangulation*.
Les équations obtenues après la calibration sont les suivantes :

$$\lambda^{(c)} \cdot \begin{pmatrix} u^{(c)} \\ v^{(c)} \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11}^{(c)} & p_{12}^{(c)} & p_{13}^{(c)} & p_{14}^{(c)} \\ p_{21}^{(c)} & p_{22}^{(c)} & p_{23}^{(c)} & p_{24}^{(c)} \\ p_{31}^{(c)} & p_{32}^{(c)} & p_{33}^{(c)} & p_{34}^{(c)} \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{60}$$

$$\lambda^{(p)} \cdot \begin{pmatrix} u^{(p)} \\ v^{(p)} \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11}^{(p)} & p_{12}^{(p)} & p_{13}^{(p)} & p_{14}^{(p)} \\ p_{21}^{(p)} & p_{22}^{(p)} & p_{23}^{(p)} & p_{24}^{(p)} \\ p_{31}^{(p)} & p_{32}^{(p)} & p_{33}^{(p)} & p_{34}^{(p)} \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{61}$$

L'exposant $^{(c)}$ désigne la caméra et $^{(p)}$ le projecteur. En développant ces deux systèmes et en consignant les coordonnées du point à reconstruire dans une matrice, on obtient une équation de la forme :

$$\mathbf{PM} = \mathbf{F} \tag{62}$$

Avec :

$$\mathbf{M} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \mathbf{F} = \begin{pmatrix} p_{34}^{(c)} u^{(c)} - p_{14}^{(c)} \\ p_{34}^{(c)} v^{(c)} - p_{24}^{(c)} \\ p_{34}^{(p)} u^{(p)} - p_{14}^{(p)} \\ p_{34}^{(p)} u^{(p)} - p_{24}^{(p)} \end{pmatrix} \quad (63), (64)$$

$$\mathbf{P} = \begin{bmatrix} p_{11}^{(c)} - p_{31}^{(c)} u^{(c)} & p_{12}^{(c)} - p_{32}^{(c)} u^{(c)} & p_{13}^{(c)} - p_{33}^{(c)} u^{(c)} \\ p_{21}^{(c)} - p_{31}^{(c)} v^{(c)} & p_{22}^{(c)} - p_{32}^{(c)} v^{(c)} & p_{23}^{(c)} - p_{33}^{(c)} v^{(c)} \\ p_{11}^{(p)} - p_{31}^{(c)} u^{(p)} & p_{12}^{(p)} - p_{32}^{(p)} u^{(p)} & p_{13}^{(p)} - p_{33}^{(p)} u^{(p)} \\ p_{21}^{(p)} - p_{31}^{(p)} v^{(p)} & p_{22}^{(p)} - p_{32}^{(p)} v^{(p)} & p_{23}^{(p)} - p_{33}^{(p)} v^{(p)} \end{bmatrix} \quad (65)$$

Finalement, les coordonnées du point sont obtenues par l'équation :

$$\mathbf{M} = \left(\mathbf{P}^T \mathbf{P}\right)^{-1} \mathbf{P}^T \mathbf{F} \quad (66)$$

3.2.5 En Conclusion

La calibration forte est restée pendant longtemps la seule technique viable de reconstruction pour les capteurs à triangulation. Elle souffre pourtant de plusieurs inconvénients majeurs. Tout d'abord le dispositif à mettre en place est lourd et nécessite l'intervention d'un opérateur humain. Le processus de calibration doit impérativement être exécuté hors-ligne, préalablement à toute mesure. Enfin, si l'un des paramètres du capteur, intrinsèque ou extrinsèque, vient à changer, tout le processus de calibration doit être réitéré. Son utilisation en robotique mobile s'en trouve largement contrainte : en effet, il peut être souhaitable, dans de nombreuses applications, que le système de vision adapte automatiquement ses paramètres à l'environnement physique dans lequel il se trouve. Dans ce contexte, l'ouverture du diaphragme devra s'ajuster à l'illumination ambiante pour fournir au calculateur des images ni sous-exposées, ni sur-exposées ; la mise au point devra s'ajuster en fonction de la distance des objets observés ; le zoom pourra être utilisé comme équivalent robotique de la fovéation pour concentrer la résolution du capteur sur des régions pertinentes de l'image.
L'intérêt de faire appel à des approches du type auto-calibration, que nous présentons dans la suite de ce chapitre, s'en trouve directement démontrée.

3.3 La Reconstruction Non-Calibrée : Une Nécessité pour la Vision en Lumière Structurée

3.3.1 Etat de l'Art

La vision non-calibrée a connu un véritable essor dès la fin des années quatre-vingt. Conscients des limites de la calibration forte, dans des applications où la machine de vision doit adapter son comportement aux variations de l'environnement (notamment en robotique mobile), les chercheurs se sont penchés sur le problème délicat qui consiste à déduire une représentation tri-dimensionnelle du monde à partir des seules données présentes dans les images. En utilisant uniquement les coordonnées des points images et, éventuellement, la géométrie épipolaire, il est établi qu'une reconstruction Euclidienne (c'est-à-dire une reconstruction à une transformation Euclidienne près) de la scène ne peut être obtenue. Suivant le modèle de caméra choisi, on pourra au mieux en calculer une reconstruction affine ou projective. Des contraintes, relatives à la structure tri-dimensionnelle de la scène, au mouvement de la caméra ou à l'invariance des paramètres intrinsèques, doivent être prises en compte si l'on veut recouvrer la structure Euclidienne de la scène.
Dans la section qui suit, nous allons présenter les méthodes de reconstruction permettant d'obtenir une structure non-métrique (affine ou projective) de la scène, puis, dans celle qui lui suivra, les méthodes permettant d'estimer la structure Euclidienne ou méthodes d'auto-calibration.

3.3.1.1 Structure Non-Métrique

Méthode du plan de référence

Les premiers travaux en matière de reconstruction non-calibrée furent initiés par Koenderink et van Doorn [KOE89]. La méthode qu'ils proposèrent, en 1989, permet de recouvrer la structure affine d'une scène à partir de deux images (au moins) de celle-ci. L'hypothèse posée par les auteurs est que le comportement de la caméra peut être approché par un modèle de projection parallèle. Un invariant de forme est calculé à partir de la construction géométrique présentée en Figure 50 : un *plan de référénce* et un *point de référence Q* indépendant de ce plan.

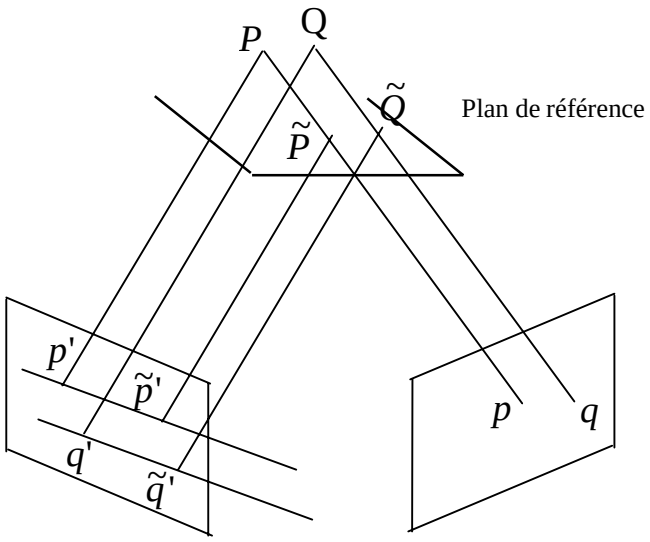


Figure 50. La méthode de Koenderink et van Doorn

P représente un point quelconque de la scène, *p* et *p'* les projetés de ce point sur les deux plans images et $\tilde{P}$ le projeté de *P* sur le plan de référence en suivant la ligne de vue de la première caméra ; enfin, $\tilde{p}'$ est la projection de $\tilde{P}$ sur le second plan image (*p'* et $\tilde{p}'$ sont confondus si *P* appartient au plan de référence). Les coordonnées de $\tilde{p}'$ peuvent être déterminées en considérant la transformation affine donnée par la projection sur le plan image de trois points appartenant au plan de référence. Considérons maintenant le point de référence *Q*, les deux trapezoïdes $\{P, \tilde{P}, p', \tilde{p}'\}$ et $\{Q, \tilde{Q}, q', \tilde{q}'\}$ sont reliés par une similarité, dont on peut tirer l'invariant suivant :

$$\gamma_P = \frac{\left|P - \tilde{P}\right|}{\left|Q - \tilde{Q}\right|} = \frac{\left|p' - \tilde{p}'\right|}{\left|q' - \tilde{q}'\right|} \tag{67}$$

Lee et Huang ont proposée, en 1990, une méthode similaire à celle de Koenderink et van Doorn [LEE90]. En 1992, Shashua [SHA92] a proposé une méthode généralisant celle de Koenderink et van Doorn au modèle de projection perspective. Deux plans de références sont cette fois-ci nécessaires (voir Figure 51). Le birapport est utilisé comme invariant de forme ; pour un point *P* et connaissant les coordonnées des épipoles e_1 et e_2, on a :

$$\alpha_P = \frac{\left|P-\tilde{P}\right|}{\left|P-\hat{p}\right|}\cdot\frac{\left|e_1-\hat{p}\right|}{\left|e_1-\tilde{P}\right|} = \frac{\left|p'-\tilde{p}'\right|}{\left|p'-\hat{p}'\right|}\cdot\frac{\left|e_l-\hat{p}'\right|}{\left|e_l-\tilde{p}'\right|} \qquad (68)$$

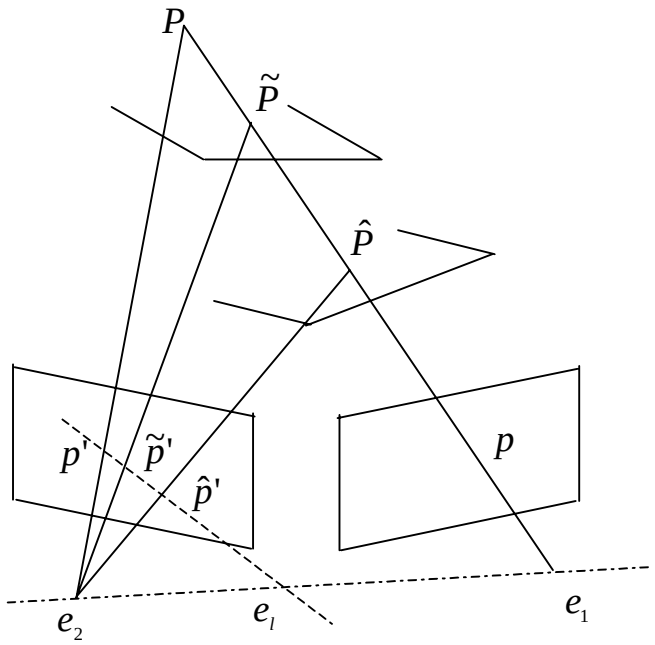


Figure 51. La méthode de Shashua

La matrice Gramienne

La méthode de Weinshall [WEI93], toujours basée sur un invariant de forme, permet quant à elle d'obtenir une représentation Euclidienne de l'environnement. En dépit de ce fait, nous l'avons classée parmi les méthodes non-métriques. En effet, la base sur laquelle repose la méthode, l'invariant de forme, est affine et c'est seulement dans un second temps que la structure est redressée dans l'espace Euclidien.

Quatre points E_0, E_1, E_2, E_3 sont choisis comme base affine à laquelle on affecte les coordonnées de la base affine canonique. Un point P quelconque de la scène sera exprimé dans cette base de la manière suivante :

$$\forall P, \exists \lambda_1, \lambda_2, \lambda_3 \,/\, P = \lambda_1 E_1 + \lambda_2 E_2 + \lambda_3 E_3 \qquad (69)$$

Considérons maintenant les projetés p_0, p_1, p_2, p_3 de ces points sur un des plans images, et leurs coordonnées respectives $(0,0)$, (x_1,y_1), (x_2,y_2) et (x_3,y_3) ; les coordonnées de p, projeté de P, sont données par :

$$\begin{cases} x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 \\ y = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 \end{cases} \quad (70)$$

Ainsi chaque point donne deux équations et trois inconnues (λ_1, λ_2, λ_3). En considérant deux images, on obtient quatre équations pour les mêmes trois inconnues : la représentation affine de l'environnement est calculable.
Prenons maintenant quatre autres points Q_1, Q_2 et Q_3 avec Q_0 placé à l'origine du repère. La matrice Gramienne de ces points est donnée par :

$$\mathbf{G} = \begin{bmatrix} \overrightarrow{Q_0Q_1} \cdot \overrightarrow{Q_0Q_1} & \overrightarrow{Q_0Q_1} \cdot \overrightarrow{Q_0Q_2} & \overrightarrow{Q_0Q_1} \cdot \overrightarrow{Q_0Q_3} \\ \overrightarrow{Q_0Q_2} \cdot \overrightarrow{Q_0Q_1} & \overrightarrow{Q_0Q_2} \cdot \overrightarrow{Q_0Q_2} & \overrightarrow{Q_0Q_2} \cdot \overrightarrow{Q_0Q_3} \\ \overrightarrow{Q_0Q_3} \cdot Q_0Q_1 & \overrightarrow{Q_0Q_3} \cdot \overrightarrow{Q_0Q_2} & \overrightarrow{Q_0Q_3} \cdot \overrightarrow{Q_0Q_3} \end{bmatrix} \quad (71)$$

Elle contient l'ensemble des connaissances sur la structure Euclidienne de ces quatre points. Sur la diagonale, on trouve la longueur de chaque vecteur, et partout ailleurs, les angles formés par tous ces vecteurs. Une particularité de **G** est qu'elle est invariante pour toutes les transformations Euclidiennes. En supposant **G** connue et en prenant un modèle perspective faible pour la caméra, il est possible de calculer les coordonnées Euclidiennes de ces points dans un repère dont l'origine est Q_0. Weinshall a montré que trois images étaient nécessaires au calcul de **G**. On dispose à présent des outils pour : (i) reconstruire la structure affine d'une scène et (ii) redresser Euclidiennement cette structure par l'intermédiaire du Gramien.

Décomposition en valeurs singulières

La méthode proposée par Tomasi et Kanade [TOM91] est basée sur la décomposition en valeurs singulières de la matrice des inconnues (les inconnues étant les coordonnées des points à reconstruire). Elle permet d'estimer à la fois la structure de la scène et la rotation de la caméra sous l'hypothèse que le modèle orthographique approche le comportement de celle-ci. Les coordonnées de *m* points, suivis dans *n* vues, sont rassemblés dans une matrice dite *matrice des mesures* :

$$\mathbf{W} = \begin{bmatrix} \mathbf{U} \\ \hline \mathbf{V} \end{bmatrix} \quad (72)$$

Où **U** (respectivement **V**) est une matrice $n \times m$ contenant les coordonnées horizontales (respectivement verticales) des points, notées u_{nm} (respectivement v_{nm}). En soustrayant, à chaque coordonnée, la moyenne des coordonnées de la même ligne, on obtient la matrice $\tilde{\mathbf{W}}$, appelée *matrice d'enregistrement des*

mesures. En plaçant l'origine du système de coordonnées au centroïde des *m* points, on obtient :

$$\tilde{\mathbf{W}} = \mathbf{R} \cdot \mathbf{S} \tag{73}$$

Où **R** représente la rotation de la caméra et **S**, la structure de la scène (autrement dit, les coordonnées des points à reconstruire). Puisqu'en pratique, on dispose de mesures bruitées, c'est la technique de décomposition en valeurs singulières qui est utilisée pour extraire **R** et **S**.
Quelques années plus tard, Poelman et Kanade [POE94] ont accru la précision de cette méthode en la généralisant au modèle para-perspective. Triggs [TRI96] et Sturm et Triggs [STU96] sont allés plus loin en généralisant cette même méthode au modèle perspective. Une phase d'estimation de la profondeur (plus précisément, le facteur d'échelle de l'équation de projection) est requise dans ce cas.

Géométrie épipolaire affine

Récemment, Zhang et Xu [ZHA97A] ont proposé une méthode d'estimation de la structure affine d'une scène à partir de la géométrie épipolaire affine du capteur. Dans ce but, les auteurs ont présenté une expression générale de la matrice fondamentale valide aussi bien pour le modèle affine que pour le modèle perspectif. La matrice fondamental affine possède seulement quatre degrés de liberté ce qui permet de formuler les équations de reconstruction de manière très simple.

$$\mathbf{F_A} = \begin{bmatrix} 0 & 0 & a_{13} \\ 0 & 0 & a_{23} \\ a_{31} & a_{.32} & a_{33} \end{bmatrix} \tag{74}$$

Le problème revient à déduire les matrices de projection $\mathbf{P_A}$ et $\mathbf{P'_A}$ à partir de la matrice fondamentale $\mathbf{F_A}$. A partir des équations (75) et (76), on obtient l'équation (77).

$$\mathbf{m}^T \mathbf{F_A} \mathbf{m}' = 0 \tag{75}$$

$$\mathbf{m} = \mathbf{P_A} \mathbf{M} \text{ et } \mathbf{m}' = \mathbf{P'}_\mathbf{A}\, \mathbf{M} \tag{76}$$

$$\mathbf{M}^T \mathbf{P}_\mathbf{A}^\mathbf{T} \mathbf{F_A} \mathbf{P'}_\mathbf{A}\, \mathbf{M} = 0 \tag{77}$$

En développant cette dernière, on obtient :

$$\begin{cases} a_{13}P_{11} + a_{23}P_{21} + a_{31}P'_{11} + a_{32}P'_{21} = 0 \\ a_{13}P_{12} + a_{23}P_{22} + a_{31}P'_{12} + a_{32}P'_{22} = 0 \\ a_{13}P_{13} + a_{23}P_{23} + a_{31}P'_{13} + a_{32}P'_{23} = 0 \\ a_{13}P_{14} + a_{23}P_{24} + a_{31}P'_{14} + a_{32}P'_{24} = -a_{33} \end{cases} \quad (78)$$

Les éléments de $\mathbf{P_A}$ et $\mathbf{P'_A}$ peuvent ainsi être déterminés. Notons que la géométrie épipolaire affine avait été précédemment décrite par Shapiro, Zisserman et Brady dans [SHA94].

Estimation des paramètres

L'approche consistant à estimer l'ensemble des paramètres a été utilisée par Mohr et son équipe [MOH93] et par Hartley [HAR94]. Elle consiste à estimer simultanément les coordonnées tri-dimensionnelles des points objets et les éléments des matrices de projection. Pour un point objet **P** et sa projection **p** sur le plan image, on a :

$$\mathbf{p} \approx \mathbf{M} \cdot \mathbf{P} \quad (79)$$

L'estimation des paramètres revient à résoudre les équations de manières à minimiser cette erreur :

$$\varepsilon = d(\mathbf{p}, \mathbf{M} \cdot \mathbf{P}) \quad (80)$$

où *d()* est une distance. Si l'on ne dispose d'aucune connaissance *a priori* sur la scène observée, cinq points sont arbitrairement choisis comme base projective. La différence entre la méthode proposée par Mohr et celle proposée par Hartley réside dans le passage de la reconstruction projective vers la reconstruction Euclidienne. La contrainte utilisée par Mohr est appliquée à la géométrie de la scène, Hartley pour sa part prend l'hypothèse que les paramètres intrinsèques de la caméra sont invariants. Dans les deux cas, cette approche conduit à un système d'équations non-linéaires dont la résolution est délicate. On sait par exemple que la convergence de ce type d'algorithme ne peut pas être guarantie. Cependant, elle offre un cadre théorique très général, valide aussi bien pour le modèle affine que pour le modèle projectif. La méthode de Mohr possède également l'avantage de ne pas contraindre les matrices de projection ce qui permet son utilisation dans des stratégies de vision adaptative.

La calibration faible

Quand la géométrie épipolaire est connue, on dit que la caméra ou le système de vision est faiblement calibré. En 1992, Faugeras [FAU92A] et, de manière indépendante, Hartley [HAR92], ont démontré qu'à partir d'un système de vision pluri-oculaire (ou d'une caméra en mouvement) faiblement calibré, il était possible d'estimer la structure projective de la scène, à une transformation projective près. Détaillons la méthode de Faugeras pour un système stéréoscopique. On affecte à cinq points de la scène E_1, E_2, E_3, E_4 et E_5, non quatre à quatre coplanaires, les coordonnées de la base projective canonique de l'espace. On affecte aux projections des quatre premiers de ces points, que l'on désigne par e_1, e_2, e_3 et e_4, les coordonnées de la base projective du plan. Les matrices de projection $\mathbf{P}_1$ et $\mathbf{P}_2$ peuvent alors s'exprimer de manière très simplifiée :

$$\mathbf{P}_1 = \begin{bmatrix} \rho_1 & 0 & 0 & \rho_4 \\ 0 & \rho_2 & 0 & \rho_4 \\ 0 & 0 & \rho_3 & \rho_4 \end{bmatrix} \mathbf{P}_2 = \begin{bmatrix} \sigma_1 & 0 & 0 & \sigma_4 \\ 0 & \sigma_2 & 0 & \sigma_4 \\ 0 & 0 & \sigma_3 & \sigma_4 \end{bmatrix} \tag{81}$$

Si l'on formule maintenant la projection du cinquième point et que l'on pose $x_1 = \begin{pmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{pmatrix}$, on obtient :

$$\rho_5 x_1 = P_1 E_5 \Rightarrow \begin{cases} \rho_1 + \rho_4 = \rho_5 \alpha_1 \\ \rho_2 + \rho_4 = \rho_5 \beta_1 \\ \rho_3 + \rho_4 = \rho_5 \gamma_1 \end{cases} \tag{82}$$

En remplaçant x_1 par x_2 et ρ_i par σ_i , on obtient les équations homologues pour la seconde caméra. Le nombre d'inconnues est donc réduit à deux. En calculant les coordonnées des épipoles, le système d'équations est complètement déterminé et la reconstruction projective peut être effectuée.

Représentation canonique

Luong et Viéville [LUO94] ont proposé une répresentation unifiée, baptisée représentation canonique, pour la modélisation du système de prise de vue. Nous allons donner ici les résultats obtenus pour deux vues, mais retenons que les auteurs ont généralisé ces résultats à n vues. Ils parviennent à une formulation simple des matrices de projection qui dépend de la géométrie, projective, affine ou Euclidienne, considérée.
Pour la géométrie Euclidienne, on a la relation classique reliant les deux vues :

$$\begin{cases} \mathbf{P} = \mathbf{A}[\mathbf{I} \quad \mathbf{0}] \\ \mathbf{P}' = \mathbf{A}'[\mathbf{R} \quad \mathbf{t}] \end{cases} \tag{83}$$

où **A** et **A'** sont les matrices des paramètres intrinsèques des caméras ; **R** et **t** sont les matrices de rotation et le vecteur de translation de la seconde caméra par rapport à l'origine du repère scène, fixé à l'origine du repère de la première caméra.
Si l'on passe à présent dans l'espace affine, les équations sont quelque peu modifiées. Les matrices des paramètres intrinsèques disparaissent et c'est l'homographie de l'infini qui définit l'orientation de la seconde caméra par rapport à la première :

$$\begin{aligned} \mathbf{P} &= [\mathbf{I} \quad \mathbf{0}] \\ \mathbf{P}' &= [\mathbf{H}_\infty \quad \mathbf{e}'_N] \end{aligned} \tag{84}$$

où $\mathbf{H}_\infty$ est l'homographie de l'infini de la seconde image vers la première et $\mathbf{e'}_\mathbf{N}$ est l'épipole normalisé de la seconde image. Cette relation reste conforme à ce que nous avons vu en annexe : les transformations affines laissent l'hyperplan de l'infini globalement invariant.
Passons maintenant dans l'espace projectif. La première caméra est toujours fixée à l'origine du repère monde. Pour obtenir la matrice de projection de la seconde caméra, on a la relation suivante :

$$\begin{aligned} \mathbf{P} &= [\mathbf{I} \quad \mathbf{0}] \\ \mathbf{P}' &= [\mathbf{S} \quad \mathbf{e}'_N] \end{aligned} \tag{85}$$

où **S** est la projection de la première image vers la seconde donnée par :

$$\mathbf{S} = -\frac{\mathbf{1}}{\|\mathbf{e'}_\mathbf{N}\|^2}[\mathbf{e'}_\mathbf{N}]_\times \mathbf{F} \tag{86}$$

On constate que la seconde matrice de projection dépend uniquement de la géométrie épipolaire du système (les coordonnées de l'épipole de la seconde image et **S** qui est une fonction de cet épipole et de la matrice fondamentale). A partir de deux vues et du calcul de la matrice fondamentale, il est donc possible d'estimer les matrices de projection et la structure projective de la scène. Notons que les matrices de projection Euclidiennes sont calculées à une similitude près, les matrices de projection affines à une affinité près et les matrices de projection projectives à une collinéation près.

La méthode du birapport

Les coordonnées projectives (x_1, x_2, x_3) d'un point P du plan, par rapport à un repère formé de quatre points (A, B, C, D), sont définies par :

$$\frac{x_1}{x_2} = \{CA; CB; CD; CP\} \qquad (87)$$

$$\frac{x_2}{x_3} = \{AB; AC; AD; AP\}$$

où {…} représente la mesure du birapport. Mohr et Morin [MOH91] ont proposé une méthode de reconstruction (en fait, de positionnement relatif de la caméra par rapport à la scène) basée sur l'invariance des coordonnées projectives. Considérons quatre points (a, b, c, d) comme étant une base projective du plan image. Les coordonnées projectives d'un point p quelconque appartenant au plan image peuvent être exprimées relativement à cette base. Considérons maintenant P_1, l'intersection de la ligne de vue $\langle O, p \rangle$ avec le plan ($ABCD$) – Figure 52. Puisque les coordonnées projectives sont invariantes aux transformations projectives et donc aux projections (transformations projectives d'un espace dans un de ses sous-espaces), les coordonnées de P_1 par rapport à (A, B, C, D) sont égales aux coordonnées de p par rapport à (a, b, c, d). Les coordonnées projectives de P_2 par rapport à (E, F, G, H) peuvent être déterminées de la même manière. La droite passant par P_1 et P_2 est la ligne de vue du point P se projetant en p. La connaissance des coordonnées d'au moins six points est requise (deux ensembles de quatre points avec deux points en commun).

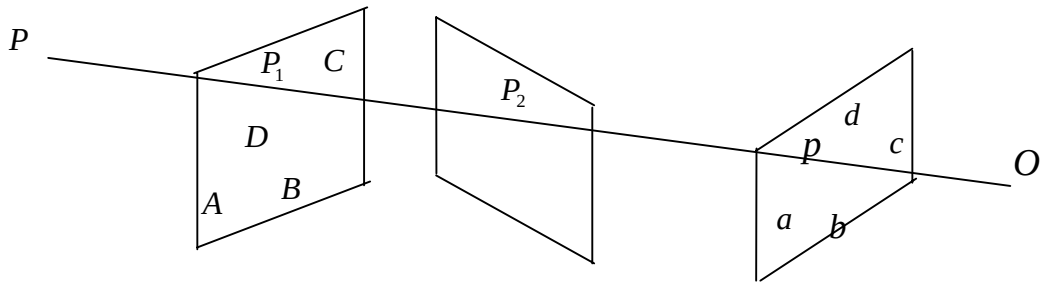


Figure 52. Retroprojection d'un point image

3.3.1.2 Structure Euclidienne : l'Auto-Calibration

Les équations de Kruppa

Le mathématicien autrichien Kruppa a, le premier, mis en évidence la relation entre les paramètres intrinsèques d'une caméra et la conique absolue. Faugeras, Maybank et Luong [FAU92B] ont redécouvert les équations de Kruppa et les ont utilisées pour l'auto-calibration. Rappelons d'abord l'équation de la conique absolue :

$$x_4 = 0 \qquad x_1^2 + x_2^2 + x_3^2 = 0 \tag{88}$$

Soit l'intersection d'une quadrique avec le plan de l'infini. Cette conique est invariante pour les similitudes (un déplacement et un facteur d'échelle), de ce fait, son image ω est indépendante de la position et de l'orientation de la caméra ; elle dépend uniquement des paramètres intrinsèques. Considérons la matrice de projection suivante :

$$\mathbf{P} = \mathbf{A}\left[\mathbf{R}|\mathbf{t}\right] \tag{89}$$

Si un point **M** appartient à la conique absolue, on peut facilement en déduire que $\mathbf{M} = \begin{pmatrix}\mathbf{X} & 0\end{pmatrix}^T$ et donc que l'équation de la conique est $\mathbf{X}^T\mathbf{X} = 0$. En projetant ce point sur le plan image, on a $\mathbf{m} = \mathbf{ARX}$ et, en inversant, $\mathbf{X} = \mathbf{R}^T\mathbf{A}^{-1}\mathbf{m}$. L'équation de ω peut ainsi être formulé par :

$$\mathbf{m}^T\mathbf{A}^{-T}\mathbf{R}\mathbf{R}^T\mathbf{A}^{-1}\mathbf{m} = \mathbf{m}^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{m} = 0 \tag{90}$$

Où $\mathbf{A}^{-T}\mathbf{A}^{-1}$ est la matrice de l'image de la conique absolue et $\mathbf{K} = \mathbf{A}\mathbf{A}^T$ est la matrice de l'image du dual de la conique absolue, appelée aussi *matrice des coefficients de Kruppa*. **K** est symétrique et définie-positive, on peut la mettre sous la forme :

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_2 & k_4 & k_5 \\ k_3 & k_5 & 1 \end{bmatrix} \tag{91}$$

Quand **K** est connue, la matrice **A** des paramètres intrinsèques peut en être extraites par l'intermédiaire de la décomposition de Choleski.
Hartley [HAR97] a démontré que les équations de Kruppa pouvaient être obtenues explicitement en décomposant la matrice fondamentale **F** en valeurs singulières :

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad \text{avec} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \mathbf{u}_3^T \end{bmatrix}, \ \mathbf{V} = \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \mathbf{v}_3^T \end{bmatrix} \text{ et } \mathbf{D} = \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{92}$$

Ce qui donne :

$$\frac{\mathbf{v}_2^T \mathbf{K} \mathbf{v}_2}{r^2 \mathbf{u}_1^T \mathbf{K} \mathbf{u}_1} = \frac{-\mathbf{v}_2^T \mathbf{K} \mathbf{v}_1}{rs \mathbf{u}_1^T \mathbf{K} \mathbf{u}_2} = \frac{\mathbf{v}_1^T \mathbf{K} \mathbf{v}_1}{s^2 \mathbf{u}_2^T \mathbf{K} \mathbf{u}_2} \tag{93}$$

Les équations de Kruppa donnent deux équations quadratiques indépendantes en les cinq paramètres de **K** pour chaque matrice fondamentale calculée, c'est-à-dire pour chaque déplacement du capteur effectué. Deux mouvements au moins sont donc nécessaires à leur résolution. Lors de ces mouvements, la matrice des paramètres intrinsèques doit demeurer constante.
Cette méthode connait beaucoup de variantes. Dans un article récent, Quan et Triggs [QUA00] proposent une méthode unifiée d'auto-calibration. Le nouveau cadre conceptuel dans lequel ils placent l'auto-calibration permet d'utiliser un modèle affine ou projectif, d'auto-calibrer des scènes planes ou non et le mouvement de la caméra peut être une pure rotation.

Constance des paramètres intrinsèques

Contrairement aux méthodes précédentes, qui donnaient directement une reconstruction Euclidienne de la scène, ces méthodes donnent la reconstruction Euclidienne à partir d'une reconstruction projective.
Une reconstruction projective est donnée par la séquence de matrices de projection suivante :

$$\mathbf{P}_{proj}^0 = \left[\mathbf{I} \middle| \mathbf{0}\right] \quad \mathbf{P}_{proj}^i = \left[\mathbf{Q}^i \middle| \mathbf{q}^i\right] \tag{94}$$

On sait que les transformations Euclidiennes forment un sous-groupe des transformations projectives, en d'autres termes, il existe une matrice 4×4 non-singulière telle que :

$$\mathbf{P}_{eucl}^i \approx \mathbf{P}_{proj}^i \mathbf{H} \tag{95}$$

La séquence des matrices de projection, exprimées dans un espace Euclidien cette fois, est donnée par :

$$\mathbf{P}_{eucl}^{0} = \mathbf{A}\left[\mathbf{I}\middle|\mathbf{0}\right] \quad \mathbf{P}_{eucl}^{i} = \mathbf{A}\left[\mathbf{R}^{i}\middle|\mathbf{t}^{i}\right] \tag{96}$$

On peut déduire des équations précédentes que **H** a la forme suivante :

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{v}^{T} & s \end{bmatrix} \tag{97}$$

Où **v** est un vecteur de trois éléments et *s* est un facteur d'échelle qui peut être fixé à un. De l'équation (96), il vient :

$$\mathbf{P}_{eucl}^{i} \approx \mathbf{P}_{proj}^{i}\mathbf{H} = \left[\mathbf{Q}^{i}\mathbf{A} + \mathbf{q}^{i}\mathbf{v}^{T}\middle|\mathbf{q}^{i}\right] \tag{98}$$

Si l'on considère que :

$$\mathbf{P}_{eucl}^{i} = \mathbf{A}\left[\mathbf{R}^{i}\middle|\mathbf{t}^{i}\right] = \left[\mathbf{A}\mathbf{R}^{i}\middle|\mathbf{A}\mathbf{t}^{i}\right] \tag{99}$$

On obtient finalement :

$$\mathbf{Q}^{i}\mathbf{A} + \mathbf{q}^{i}\mathbf{v}^{T} \approx \mathbf{A}\mathbf{R}^{i} \tag{100}$$

C'est l'équation de base de cette classe de méthodes. La matrice **A** donne cinq inconnues et le vecteur **v** trois inconnues. **R** est une matrice de rotation et les matrices **Q** et **q** sont données par la reconstruction projective. On peut citer, dans ce domaine, les travaux de Hartley [HAR94], les travaux d'Heyden et Aström [HEY96] .

Contraintes Euclidiennes

Revenons à l'équation (95), la méthode des contraintes Euclidiennes consiste à générer des informations géométriques de la scène pour contraindre les éléments de la matrice **H**. Les principaux travaux sur cette approche ont été menés par Boufama, Mohr et Veillon [BOU93]. **H** est une matrice 4×4 qui admet donc quinze degrés de liberté. Si les coordonnées Euclidiennes de cinq points, au moins, sont connues, il est aisé d'estimer **H** (3 coordonnées × 5 points = 15 équations). Si les coordonnées de cinq points ne sont pas disponibles, il est possible de générer d'autres contraintes comme l'alignement ou la distance entre deux points, le parallélisme ou l'orthogonalité de deux segments, etc.

Posons $w_{ij,1\le i,j\le 4}$, l'élément (i, j) de la matrice $\mathbf{W} = \mathbf{H}^{-1}$. Pour un point $\mathbf{M}_{proj}$ exprimé dans un repère projective et le même point $\mathbf{M}_{eucl}$ exprimé dans un repère Euclidien, on a :

$$\mathbf{M}_{eucl} \approx \mathbf{W}\mathbf{M}_{proj} \tag{101}$$

L'obtention de ces connaissances n'est pas toujours simple : dans un environnement peu structuré et *a priori* inconnu, il n'est pas toujours possible de trouver des droites parallèles ou orthogonales (dans l'espace), par exemple. Dans une application dédiée à la synthèse d'images faciales, Zhang, Isono et Akamatsu [ZHA98] ont exprimé les contraintes géométriques issues de la scène à l'aide de la distance de Mahalanobis carrée. La matrice de covariance est utilisée pour modéliser un domaine flou traduisant une imprécision dans la localisation des points. La reconstruction Euclidienne est obtenue en minimisant la somme des distances de Mahalanobis.

3.3.2 Le Cas de la Vision en Lumière Structurée

3.3.2.1 Les contraintes

Toutes les méthodes que nous avons présentées plus haut ne sont pas adaptables à la vision en lumière structurée. Voyons les contraintes que celles-ci imposent et qu'elles sont en conséquence les méthodes véritablement exploitables. D'abord, on peut remarquer qu'un mouvement du projecteur pendant le processus de mesure produit un glissement du motif projeté sur la scène. Partant, les points 3-D de l'image avant le mouvement du capteur sont perdus après le mouvement. Les méthodes, développées pour la stéréovision, utilisant plus de deux vues (ce qui se traduit en lumière structurée par une vue et une projection) sont donc à écarter. On a vu qu'un grand nombre de méthodes supposaient que les paramètres intrinsèques restent constants. Evidemment, cette hypothèse ne peut plus être tenue en vision en lumière structurée. Nous sommes limités à une vue et à une projection, nous venons de le voir. Affirmer que les paramètres intrinsèques restent constants revient donc à dire que les paramètres de la caméra et du projecteur ont mêmes valeurs. On peut difficilement assumer une telle hypothèse eu égard à l'hétérogénéité du capteur (une caméra et un projecteur). Dernier point enfin, si nous nous sommes orientés vers une méthode de reconstruction sans-calibration, c'est pour pouvoir agir, pendant la mission du véhicule, sur les paramètres tels que la mise au point, l'ouverture ou le zoom et ainsi conférer un comportement adaptatif aux mécanismes de vision du véhicule. On pourra alors ajuster la mise au point du projecteur à la distance des surfaces observées, adapter l'ouverture de diaphragme à la luminosité ambiante et utiliser le zoom comme un instrument de fovéation. De ce fait, la méthode de reconstruction ne devra pas contraindre les

matrices de projection de manière à laisser ces degrés de liberté au capteur. Le choix d'une méthode de reconstruction se trouve donc largement restreint. Nous avons identifié deux méthodes permettant de reconstruire projectivement la scène observée par notre capteur de vision en lumière structurée : la première, celle que nous avons expérimentée dans ce mémoire, est l'estimation des paramètres ([MOH93] détaillée dans la sous-section suivante) ; la seconde est la représentation canonique [LUO94]. Si nous avons choisi la première de ces méthodes, c'est parce qu'elle offre un cadre plus général et semble plus cohérente avec la méthode des contraintes Euclidiennes que nous présenterons par la suite.

3.3.2.2 La Reconstruction Projective

Considérons le problème général du recouvrement de la structure d'une scène à partir de *n* images et de *m* points. On sait qu'il est nécessaire d'avoir $n \geq 2$ et qu'il est suffisant de voir apparaître chaque point dans seulement deux images. On suppose que la mise en correspondance est effectuée à travers les *n* images, on a :

$$\mathbf{m}_{ij} \approx \mathbf{P}_j \cdot \mathbf{M}_i ; i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\} \tag{102}$$

où $\mathbf{m}_{ij} = \begin{pmatrix} u_{ij} & v_{ij} & w_{ij} \end{pmatrix}^T$ est le $i^{\text{ème}}$ point apparaissant dans la $j^{\text{ème}}$ image, $\mathbf{M}_i = \begin{pmatrix} x_i & y_i & z_i & t_i \end{pmatrix}^T$ et $\mathbf{P}_j$ est la matrice de projection associée à la $j^{\text{ème}}$ image. On pose :

$$\mathbf{P}_j = \begin{bmatrix} p_{11}^{(j)} & p_{12}^{(j)} & p_{13}^{(j)} & p_{14}^{(j)} \\ p_{21}^{(j)} & p_{22}^{(j)} & p_{23}^{(j)} & p_{24}^{(j)} \\ p_{31}^{(j)} & p_{32}^{(j)} & p_{33}^{(j)} & p_{34}^{(j)} \end{bmatrix} \tag{103}$$

En développant ces équations et en éliminant la dernière coordonnée homogène, on obtient les valeurs des coordonnées en pixels $U_{ij} = u_{ij} / w_{ij}$ et $V_{ij} = v_{ij} / w_{ij}$:

$$\begin{cases} U_{ij} = \dfrac{p_{11}^{(j)} x_i + p_{12}^{(j)} y_i + p_{13}^{(j)} z_i + p_{14}^{(j)} t_i}{p_{31}^{(j)} x_i + p_{32}^{(j)} y_i + p_{33}^{(j)} z_i + p_{34}^{(j)} t_i} \\ V_{ij} = \dfrac{p_{21}^{(j)} x_i + p_{22}^{(j)} y_i + p_{23}^{(j)} z_i + p_{24}^{(j)} t_i}{p_{31}^{(j)} x_i + p_{32}^{(j)} y_i + p_{33}^{(j)} z_i + p_{34}^{(j)} t_i} \end{cases} \tag{104}$$

Puisque nous avons *m* points, donnant chacun deux coordonnées, et *n* images, le nombre d'équations données par un tel système est $2 \times m \times n$. Chaque matrice de projection, formées de douze éléments mais définies à un facteur d'échelle près,

nous donne 11×*n* inconnues. Chaque point à reconstruire nous donne 3×*m* inconnues. Pour pouvoir envisager la résolution de ce système, il nous faut choisir les paramètres *n* et *m* assez grands ; plus précisément pour obtenir un système d'équations redondant, il faut :

$$2 \times m \times n > 3 \times m + 11 \times n - 15 \tag{105}$$

Si, comme dans le cas de la vision en lumière structurée, le nombre d'images est limité à deux (une projection et une acquisition), il faut que :

$$m > 7 \tag{106}$$

On peut penser que la résolution du système nous donne directement une reconstruction tri-dimensionnelle de la scène. Pourtant, si $\hat{\mathbf{P}}$ et $\hat{\mathbf{M}}$ sont les solutions pour, respectivement, une matrice de projection et les coordonnées d'un point à reconstruire, alors, $\hat{\mathbf{P}}\mathbf{W}^{-1}$ et $\mathbf{W}\hat{\mathbf{M}}$ sont également des solutions, comme nous le montre l'équation suivante :

$$\mathbf{m} = \left(\hat{\mathbf{P}}\mathbf{W}^{-1}\right) \cdot \left(\mathbf{W}\hat{\mathbf{M}}\right) = \hat{\mathbf{P}} \cdot \left(\mathbf{W}^{-1}\mathbf{W}\right) \cdot \hat{\mathbf{M}} = \hat{\mathbf{P}}\hat{\mathbf{M}} \tag{107}$$

Où **W** est une matrice 4×4, inversible, qui décrit une collinéation de l'espace. De ce fait, la reconstruction ne peut être obtenue qu'à une collinéation (ou transformation projective) près. Pour trouver une unique solution au système, il est nécessaire de fixer **W**, ce qui revient à fixer un repère projectif dans l'espace. Puisque une collineation de $\wp^3$ dans $\wp^3$ possède quinze degrés de liberté (les seize éléments qu'elle admet auxquels on ôte le facteur d'échelle), cinq points dans l'espace sont suffisant pour la déterminer : les cinq points de la base projective choisie. La géométrie projective, nous l'avons vu plus haut, ne reconnait ni les distances, ni les angles, on peut de ce fait choisir arbitrairement cinq points parmi ceux que l'on entend reconstruire (aux seules conditions qu'ils ne soient ni quatre à quatre coplanaires, ni trois à trois colinéaires) et leur affecter les coordonnées de la base projective canonique. La résolution du système d'équations nous donnera cette fois une reconstruction projective de la scène relativement à cette base.

Puisque qu'il nous faut choisir cinq points pour définir la base projective sous les contraintes que ces points ne soient ni quatre à quatre coplanaires, ni trois à trois colinéaires, nous présentons dans la section suivante deux tests permettant d'évaluer la colinéarité et la coplanarité d'un ensemble de points. Nous proposons ensuite une méthode permettant de valider le choix du modèle affine pour le capteur : si jusque-là, nous avons approché le comportement du capteur par le modèle sténopé, nous verrons que pour la génération de contraintes Euclidiennes, le modèle affine est plus approprié.

3.3.2.3 Quelques Propriétés

Cette section présente deux tests originaux permettant de classer un ensemble de points en tant que colinéaires, coplanaires ou quelconques. Un test de coplanarité a antérieurement été proposé par Patrick Gros [GRO93], le nôtre possède l'avantage de ne pas nécessiter le calcul des coordonnées des épipoles. Nous proposons enfin un test de validité du modèle affine basé sur la déformation du patron de lumière dans l'image. Ces tests ne nécessitent aucune calibration préalable du capteur.

Test de colinéarité spatiale

Considérons quatre points de l'espace *P*, *Q*, *R*, *S*, leurs correspondant respectifs *p*, *q*, *r*, *s* dans le patron de lumière et leurs projetés *p'*, *q'*, *r'*, *s'* sur le plan de l'image (Figure 53). Si *P*, *Q*, *R* et *S* sont alignés, on a :

$$\{P,Q,R,S\}=\{p,q,r,s\}=\{p',q',r',s'\} \tag{108}$$

où{…} désigne le birapport des quatre points. Le passage des points du patron de lumière aux points de l'espace s'obtient par une homographie ; de la même manière, le passage des points de l'espace aux points de l'image s'obtient également par une homographie. On sait que le birapport est conservé par ce type de transformations, on peut en déduire que les trois birapports sont égaux.

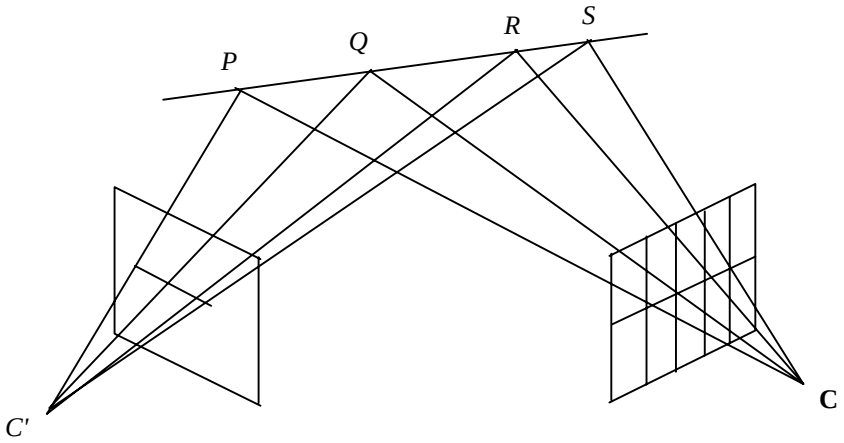


Figure 53. Test de colinéarité spatiale

Inversement, on peut déduire que si le birapport des quatre points du plan de l'image est égal au birapport de leur quatre homologues du patron de lumière, les

points *P*, *Q*, *R* et *S* appartiennent à la même droite de l'espace. Les coordonnées des points du patron étant parfaitement connus, la lumière structurée nous fournit une mesure exacte de ce birapport. Le test de colinéarité consistera à vérifier la valeur du birapport de quatre points au sein de l'image.

Test de coplanarité

Si nous considérons cette fois les deux configurations de points de la Figure 54, un test de coplanarité peut être imaginé. Si les points O_i, P_i, Q_i, R_i et S_i (i = 1 ou 2) sont projetés sur un plan, alors le birapport au sein du patron est égal au birapport des cinq points formés sur ce plan ; partant, le birapport que l'on retrouve dans l'image est égal aux deux autres. Là encore, la transformation des points du projecteur vers la caméra est obtenue par deux homographies successives.

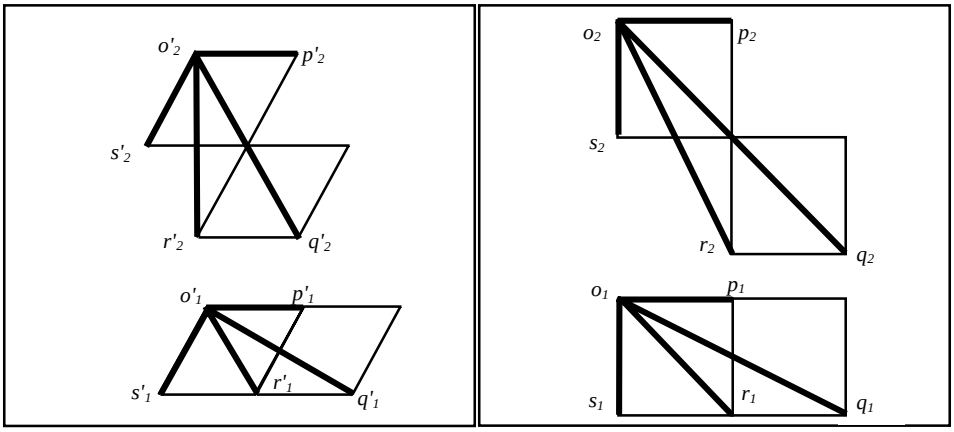


Figure 54. Test de coplanarité. (Gauche) Plan de l'image .(Droite) Patron de lumière.

On en déduit que si l'équation (109) est vérifiée, alors les points 3-D correspondant *O*, *P*, *Q*, *R* et *S* sont coplanaires.

$$\{o_i ; p_i , q_i , r_i , s_i\} = \{o'_i ; p'_i , q'_i , r'_i , s'_i\} \text{avec } i = 1 \text{ ou } 2 \qquad (109)$$

Stabilité du birapport

Nous avons testé la stabilité du birapport pour les configurations de points données pour le test de colinéarité et pour le test de coplanarité. Pour le premier, nous avons pris quatre points alignés pour lesquels les distances d entre deux points successifs sont égales. Nous avons ajouté un bruit sur les coordonnées des points : le bruit variant de 0 à 0.5 × d. Les résultats de nos expériences sont illustrés par la Figure 55. Pour le second, le protocole d'expérimentation reste

inchangé : seuls la configuration des points change et, par conséquent, la formule utilisée pour le calcul du birapport. Les résultats sont regroupés Figure 56.

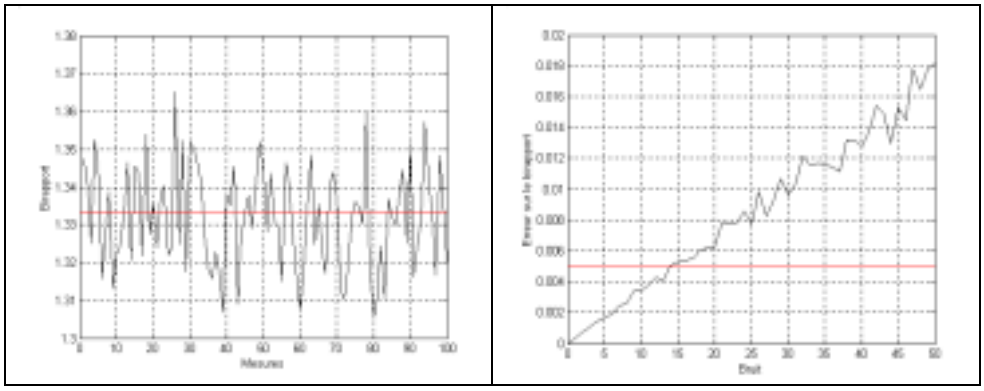


Figure 55. Stabilité du birapport : test de colinéarité. (Gauche) Mesures bruitées à ± 5%. (Droite) Evolution de l'erreur avec un bruit variant de 0 à 50%.

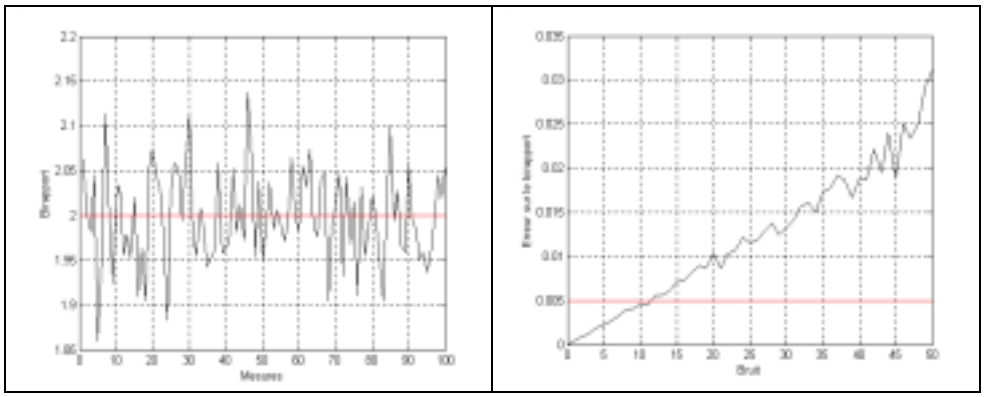


Figure 56. Stabilité du birapport : test de coplanarité. (Gauche) Mesures bruitées à ± 5%. (Droite) Evolution de l'erreur avec un bruit variant de 0 à 50%.

Pour être en mesure de comparer les birapports théoriques avec les birapports issus des images, nous avons utilisé une *distance projective* à partir de la méthode des birapports aléatoires dont on trouvera une explication en Annexe C. Nous avons empiriquement fixé le seuil de tolérance de l'erreur à 5×10^{-3}. Dans ces conditions, un bruit atteignant 15% de la distance *d* permet encore une discrimination valide des configurations de points colinéaires et coplanaires. Evidemment (et comme il peut être déduit des tests de stabilité effectués) plus la distance séparant les points dans l'image est grande, plus la mesure du birapport est robuste au bruit ; en pratique, on choisira donc des points le plus éloignés possible les uns des autres.

La validité du modèle affine

Dans la suite de ce chapitre, nous allons prendre l'hypothèse que le comportement du capteur peut être approché par un modèle affine. Si, dans certaines conditions de prise de vue (voir Annexe A), le modèle affine offre des résultats tout à fait comparables au modèle projectif, il n'en est pas toujours le cas. Notre but est d'obtenir, par l'analyse de l'image, un indice de confiance de la validité de ce modèle. Nous avons vu, dans les premières sections de ce chapitre, qu'une transformation affine avait la particularité de transformer un carré en parallélogramme ; de là, on déduit qu'un carré projeté sur une surface plane et observé par une caméra formera un parallélogramme sur le plan image si et seulement si le capteur de vision suit le modèle affine.
Considérons les quatre sommets d'un carré appartenant au patron de lumière et prenons l'hypothèse que ces quatre points sont projetés sur un même plan. Leur quatre homologues dans le plan image forment, on le sait, un quadrilatère. S'il s'agit d'un parallélogramme parfait, l'hypothèse du modèle affine peut être soutenue. A l'inverse, s'il s'agit clairement d'un quadrilatère quelconque, cette hypothèse doit être abandonnée. Qu'en est-il des formes intermédiaires ? Comment mesurer la ressemblance de la forme obtenue avec la forme parallélogramme et comment définir le seuil de validité du modèle affine ?
Concrètement, imaginons un parallélogramme dont les sommets, pris successivement et dans le sens des aiguilles d'une montre, sont les points *m*, *n*, *n'* et *m'*. On sait que, pour un parallélogramme, les longueurs de deux arêtes opposées sont égales. En mesurant la différence entre ces longueurs, on peut avoir une information sur la ressemblance de la forme analysée avec un parallélogramme idéal :

$$\phi = \frac{\left|\sqrt{(u_m - u_n)^2 + (v_m - v_n)^2} - \sqrt{(u_{m'} - u_{n'})^2 + (v_{m'} - v_{n'})^2}\right|}{\sqrt{(u_m - u_n)^2 + (v_m - v_n)^2}} + \frac{\left|\sqrt{(u_m - u_{m'})^2 + (v_n - v_{n'})^2} - \sqrt{(u_m - u_{m'})^2 + (v_n - v_{n'})^2}\right|}{(u_m - u_{m'})^2 + (v_n - v_{n'})^2} \qquad (110)$$

Où les $(u_x, v_x)_{x=m,n,m',n'}$ sont les coordonnées pixels de chaque point. Nous considérerons que le modèle affine est valide si la mesure donnée par l'équation (110) est proche de zéro.

3.3.2.4 Contraintes Euclidiennes

On sait que la géométrie Euclidienne est un cas particulier de la géométrie projective ; les transformations Euclidiennes forment un sous-groupe des transformations projectives. En d'autres termes, il existe une transformation projective **W** qui, pour un point *M* de l'espace, dont les coordonnées projectives sont rassemblées dans le vecteur $\tilde{\mathbf{M}} = \begin{pmatrix} \tilde{x} & \tilde{y} & \tilde{z} & \tilde{t} \end{pmatrix}^T$ et les coordonnées Euclidiennes dans le vecteur $\mathbf{M} = \begin{pmatrix} x & y & z & t \end{pmatrix}^T$ donne :

$$\mathbf{M} = \mathbf{W} \cdot \tilde{\mathbf{M}} \tag{111}$$

W est une collinéation de l'espace ; c'est une matrice 4×4 non-singulière, elle est définie, comme toutes entités projectives, à un facteur d'échelle près. La méthode des contraintes euclidiennes [BOU93] consiste à traduire des informations géométriques sur les points 3-D en contraintes mathématiques sur les éléments de **W**. Dans la suite de cette section, nous donnons une liste de contraintes obtenues par l'analyse des images en lumière structurée.

Fixer un point

A ce stade du processus, nous disposons de la matrice de projection de la caméra $\tilde{\mathbf{P}}^{(c)}$, de la matrice de projection du projecteur $\tilde{\mathbf{P}}^{(p)}$ et des coordonnées 3-D des *n* points $\tilde{\mathbf{M}}_i, 1 \le i \le n$. Tous ces éléments sont exprimés dans un repère *projectif*. Si nous supposons connues les coordonnées Euclidiennes d'au moins cinq points de l'espace, il est possible d'estimer la matrice **W**, chaque point nous donnant trois équations (une pour chaque coordonnée) . En développant l'équation (111) et en éliminant le facteur d'échelle, on obtient :

$$\begin{cases} x = \dfrac{w_{11}\tilde{x} + w_{12}\tilde{y} + w_{13}\tilde{z} + w_{14}\tilde{t}}{w_{41}\tilde{x} + w_{42}\tilde{y} + w_{43}\tilde{z} + w_{44}\tilde{t}} \\ y = \dfrac{w_{21}\tilde{x} + w_{22}\tilde{y} + w_{23}\tilde{z} + w_{24}\tilde{t}}{w_{41}\tilde{x} + w_{42}\tilde{y} + w_{43}\tilde{z} + w_{44}\tilde{t}} \\ z = \dfrac{w_{31}\tilde{x} + w_{32}\tilde{y} + w_{33}\tilde{z} + w_{34}\tilde{t}}{w_{41}\tilde{x} + w_{42}\tilde{y} + w_{43}\tilde{z} + w_{44}\tilde{t}} \end{cases} \tag{112}$$

Une contrainte sur le facteur d'échelle de **W** doit être ajoutée ; le plus simple consiste à fixer le dernier élément $w_{44} = 1$. Il est préférable cependant, si la valeur

réelle de w_{44} devait être proche de zéro, de normaliser la collinéation de cette manière :

$$\sum_{i,\,j} \left(w_{ij}\right)^2 = 1 \tag{113}$$

En robotique mobile, quand le véhicule est susceptible d'évoluer dans un environnement inconnu et, *a fortiori*, en vision en lumière structurée, où les noeuds de la structure doivent se projeter très exactement sur les points 3-D connus, cette information est rarement disponible et, en pratique, inefficiente. En revanche, cette contrainte nous permet de fixer l'origine du repère en posant les équations (112) égales à zéro.

Contraintes d'alignement

Chaque ligne horizontale du patron forme un plan de lumière dans l'espace qui peut être considéré comme un plan 3-D horizontal dans le repère centré sur le projecteur ; similairement, chaque ligne verticale du patron forme un plan de lumière qui peut être considéré comme un plan 3-D vertical dans ce même repère. Le motif perçu par la caméra correspond à l'intersection des plans de lumière générés par la projection avec les surfaces qui composent la scène. On peut en déduire que les points qui composent le motif perçu appartiennent à des plans horizontaux et/ou verticaux de l'espace et inférer des contraintes d'alignement. Si deux points m_1 et m_2 du plan image appartiennent à la même verticale du motif, alors les points M_1 et M_2 qui leur correspondent sont alignés verticalement dans l'espace, de sorte que $x_1 = x_2$ et $y_1 = y_2$, on en déduit que :

$$\begin{cases} \dfrac{w_{11}\tilde{x}_1 + w_{12}\tilde{y}_1 + w_{13}\tilde{z}_1 + w_{14}\tilde{t}_1}{w_{41}\tilde{x}_1 + w_{42}\tilde{y}_1 + w_{43}\tilde{z}_1 + w_{44}\tilde{t}_1} = \dfrac{w_{11}\tilde{x}_2 + w_{12}\tilde{y}_2 + w_{13}\tilde{z}_2 + w_{14}\tilde{t}_2}{w_{41}\tilde{x}_2 + w_{42}\tilde{y}_2 + w_{43}\tilde{z}_2 + w_{44}\tilde{t}_2} \\ \dfrac{w_{21}\tilde{x}_1 + w_{22}\tilde{y}_1 + w_{23}\tilde{z}_1 + w_{24}\tilde{t}_1}{w_{41}\tilde{x}_1 + w_{42}\tilde{y}_1 + w_{43}\tilde{z}_1 + w_{44}\tilde{t}_1} = \dfrac{w_{21}\tilde{x}_2 + w_{22}\tilde{y}_2 + w_{23}\tilde{z}_2 + w_{24}\tilde{t}_2}{w_{41}\tilde{x}_2 + w_{42}\tilde{y}_2 + w_{43}\tilde{z}_2 + w_{44}\tilde{t}_2} \end{cases} \tag{114}$$

De la même manière, si deux points m_1 et m_2 du plan image appartiennent à la même horizontale du motif, alors les points M_1 et M_2 qui leur correspondent sont alignés horizontalement dans l'espace avec $x_1 = x_2$ et $z_1 = z_2$; on a :

$$\begin{cases} \dfrac{w_{11}\tilde{x}_1 + w_{12}\tilde{y}_1 + w_{13}\tilde{z}_1 + w_{14}\tilde{t}_1}{w_{41}\tilde{x}_1 + w_{42}\tilde{y}_1 + w_{43}\tilde{z}_1 + w_{44}\tilde{t}_1} = \dfrac{w_{11}\tilde{x}_2 + w_{12}\tilde{y}_2 + w_{13}\tilde{z}_2 + w_{14}\tilde{t}_2}{w_{41}\tilde{x}_2 + w_{42}\tilde{y}_2 + w_{43}\tilde{z}_2 + w_{44}\tilde{t}_2} \\ \dfrac{w_{31}\tilde{x}_1 + w_{32}\tilde{y}_1 + w_{33}\tilde{z}_1 + w_{34}\tilde{t}_1}{w_{41}\tilde{x}_1 + w_{42}\tilde{y}_1 + w_{43}\tilde{z}_1 + w_{44}\tilde{t}_1} = \dfrac{w_{31}\tilde{x}_2 + w_{32}\tilde{y}_2 + w_{33}\tilde{z}_2 + w_{34}\tilde{t}_2}{w_{41}\tilde{x}_2 + w_{42}\tilde{y}_2 + w_{43}\tilde{z}_2 + w_{44}\tilde{t}_2} \end{cases} \quad (115)$$

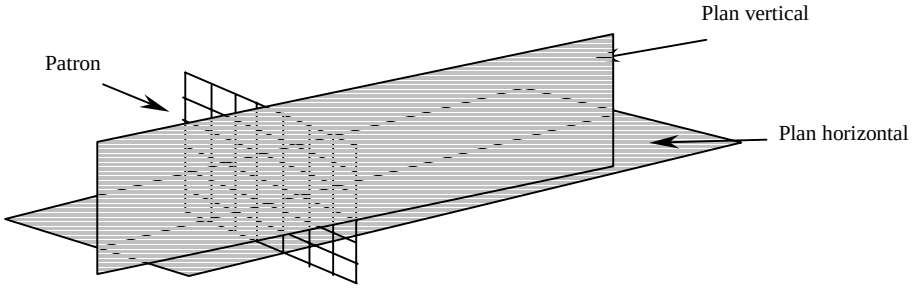


Figure 57. Contraintes d'alignement

Notons que l'alignement horizontal peut également se formuler avec $y_1 = y_2$ et $z_1 = z_2$. Les équations deviennent alors :

$$\begin{cases} \dfrac{w_{21}\tilde{x}_1 + w_{22}\tilde{y}_1 + w_{23}\tilde{z}_1 + w_{24}\tilde{t}_1}{w_{41}\tilde{x}_1 + w_{42}\tilde{y}_1 + w_{43}\tilde{z}_1 + w_{44}\tilde{t}_1} = \dfrac{w_{21}\tilde{x}_2 + w_{22}\tilde{y}_2 + w_{23}\tilde{z}_2 + w_{24}\tilde{t}_2}{w_{41}\tilde{x}_2 + w_{42}\tilde{y}_2 + w_{43}\tilde{z}_2 + w_{44}\tilde{t}_2} \\ \dfrac{w_{31}\tilde{x}_1 + w_{32}\tilde{y}_1 + w_{33}\tilde{z}_1 + w_{34}\tilde{t}_1}{w_{41}\tilde{x}_1 + w_{42}\tilde{y}_1 + w_{43}\tilde{z}_1 + w_{44}\tilde{t}_1} = \dfrac{w_{31}\tilde{x}_2 + w_{32}\tilde{y}_2 + w_{33}\tilde{z}_2 + w_{34}\tilde{t}_2}{w_{41}\tilde{x}_2 + w_{42}\tilde{y}_2 + w_{43}\tilde{z}_2 + w_{44}\tilde{t}_2} \end{cases} \quad (116)$$

Les contraintes d'alignement nous permettent d'ajuster la structure à un repère orthogonal centré sur le projecteur.

Contraintes du parallélogramme

Puisque que nous posons l'hypothèse que notre capteur de vision a un comportement affine, nous pouvons en déduire que chaque carré du motif projeté sur une surface plane de la scène apparaîtra dans l'image sous la forme d'un parallélogramme (Figure 58). Ce qui est remarquable ici, c'est que les points 3-D correspondant sont les sommets d'un parallélogramme dans l'espace. Les propriétés du parallélogramme nous permettent d'en déduire des contraintes de distance entre ces points (117) ainsi que des contraintes de parallélisme (118). Il faut préalablement vérifier que les points considérés appartiennent bien à un

même plan de l'espace. On sait que certaines configurations de points peuvent donner naissance à un parallélogramme dans l'image sans qu'elles ne correspondent à un parallélogramme dans l'espace ; si l'on teste la coplanarité des quatre sommets du parallélogramme, cette ambiguïté est levée.

$$\left\|\overrightarrow{MN}\right\| = \left\|\overrightarrow{M'N'}\right\| \quad \text{et} \quad \left\|\overrightarrow{MM'}\right\| = \left\|\overrightarrow{NN'}\right\| \tag{117}$$

$$\begin{aligned} (MN)//(M'N') &\Leftrightarrow \frac{\overrightarrow{MN}}{\left\|\overrightarrow{MN}\right\|} = \frac{\overrightarrow{M'N'}}{\left\|\overrightarrow{M'N'}\right\|} \\ (MM')//(NN') &\Leftrightarrow \frac{\overrightarrow{MM'}}{\left\|\overrightarrow{MM'}\right\|} = \frac{\overrightarrow{NN'}}{\left\|\overrightarrow{NN'}\right\|} \end{aligned} \tag{118}$$

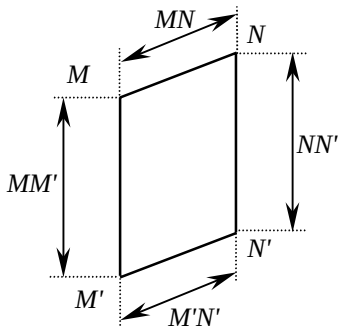


Figure 58. Contraintes du parallélogramme

On peut constater que ces deux jeux de contraintes sont redondants. Puisque la géométrie projective préserve l'alignement et la coplanarité, les équations (117) et (118) décrivent la même configuration de points. En les combinant, les contraintes de parallélisme peuvent être grandement simplifiées (120) :

$$\begin{cases} (x_N - x_M)^2 + (y_N - y_M)^2 + (z_N - z_M)^2 = (x_{N'} - x_{M'})^2 + (y_{N'} - y_{M'})^2 + (z_{N'} - z_{M'})^2 \\ (x_{M'} - x_M)^2 + (y_{M'} - y_M)^2 + (z_{M'} - z_M)^2 = (x_{N'} - x_N)^2 + (y_{N'} - y_N)^2 + (z_{N'} - z_N)^2 \end{cases} \tag{119}$$

$$\begin{cases} (x_N - x_M) = (x_{N'} - x_{M'}), (y_N - y_M) = (y_{N'} - y_{M'}), (z_N - z_M) = (z_{N'} - z_{M'}) \\ (x_{M'} - x_M) = (x_{N'} - x_N), (y_{M'} - y_M) = (y_{N'} - y_N), (z_{M'} - z_M) = (z_{N'} - z_N) \end{cases} \tag{120}$$

Il faut noter également qu'un seul parallélogramme détermine complètement le plan 3-D sur lequel il repose. De ce fait, il suffira d'implémenter les contraintes

d'un seul parallélogramme pour chaque surface plane de la scène. Les équations (120) sont similaires aux contraintes d'alignement ; si nous développons les équations (119), on obtient :

$$\begin{aligned}
&\left(\frac{w_{11}\tilde{x}_N + w_{12}\tilde{y}_N + w_{13}\tilde{z}_N + w_{14}\tilde{t}_N}{w_{41}\tilde{x}_N + w_{42}\tilde{y}_N + w_{43}\tilde{z}_N + w_{44}\tilde{t}_N} - \frac{w_{11}\tilde{x}_M + w_{12}\tilde{y}_M + w_{13}\tilde{z}_M + w_{14}\tilde{t}_M}{w_{41}\tilde{x}_M + w_{42}\tilde{y}_M + w_{43}\tilde{z}_M + w_{44}\tilde{t}_M}\right)^2 + \\
&\left(\frac{w_{21}\tilde{x}_N + w_{22}\tilde{y}_N + w_{23}\tilde{z}_N + w_{24}\tilde{t}_N}{w_{41}\tilde{x}_N + w_{42}\tilde{y}_N + w_{43}\tilde{z}_N + w_{44}\tilde{t}_N} - \frac{w_{21}\tilde{x}_M + w_{22}\tilde{y}_M + w_{23}\tilde{z}_M + w_{24}\tilde{t}_M}{w_{41}\tilde{x}_M + w_{42}\tilde{y}_N + w_{43}\tilde{z}_M + w_{44}\tilde{t}_M}\right)^2 + \\
&\left(\frac{w_{31}\tilde{x}_N + w_{32}\tilde{y}_N + w_{33}\tilde{z}_N + w_{34}\tilde{t}_N}{w_{41}\tilde{x}_N + w_{42}\tilde{y}_N + w_{43}\tilde{z}_N + w_{44}\tilde{t}_N} - \frac{w_{31}\tilde{x}_M + w_{32}\tilde{y}_M + w_{33}\vec{z}_M + w_{34}\tilde{t}_M}{w_{41}\tilde{x}_M + w_{42}\tilde{y}_M + w_{43}\tilde{z}_M + w_{44}\tilde{t}_M}\right)^2 = \\
&\left(\frac{w_{11}\tilde{x}_{N'} + w_{12}\tilde{y}_{N'} + w_{13}\tilde{z}_{N'} + w_{14}\tilde{t}_{N'}}{w_{41}\tilde{x}_{N'} + w_{42}\tilde{y}_{N'} + w_{43}\tilde{z}_{N'} + w_{44}\tilde{t}_{N'}} - \frac{w_{11}\tilde{x}_{M'} + w_{12}\tilde{y}_{M'} + w_{13}\tilde{z}_{M'} + w_{14}\tilde{t}_{M'}}{w_{41}\tilde{x}_{M'} + w_{42}\tilde{y}_{M'} + w_{43}\tilde{z}_{M'} + w_{44}\tilde{t}_{M'}}\right)^2 + \\
&\left(\frac{w_{21}\tilde{x}_{N'} + w_{22}\tilde{y}_{N'} + w_{23}\tilde{z}_{N'} + w_{24}\tilde{t}_{N'}}{w_{41}\tilde{x}_{N'} + w_{42}\tilde{y}_{N'} + w_{43}\tilde{z}_{N'} + w_{44}\tilde{t}_{N'}} - \frac{w_{21}\tilde{x}_{M'} + w_{22}\tilde{y}_{M'} + w_{23}\tilde{z}_{M'} + w_{24}\tilde{t}_{M'}}{w_{41}\tilde{x}_{M'} + w_{42}\tilde{y}_{N'} + w_{43}\tilde{z}_{M'} + w_{44}\tilde{t}_{M'}}\right)^2 + \\
&\left(\frac{w_{31}\tilde{x}_{N'} + w_{32}\tilde{y}_{N'} + w_{33}\tilde{z}_{N'} + w_{34}\tilde{t}_{N'}}{w_{41}\tilde{x}_{N'} + w_{42}\tilde{y}_{N'} + w_{43}\tilde{z}_{N'} + w_{44}\tilde{t}_{N'}} - \frac{w_{31}\tilde{x}_{M'} + w_{32}\tilde{y}_{M'} + w_{33}\vec{z}_{M'} + w_{34}\tilde{t}_{M'}}{w_{41}\tilde{x}_{M'} + w_{42}\tilde{y}_{M'} + w_{43}\tilde{z}_{M'} + w_{44}\tilde{t}_{M'}}\right)^2
\end{aligned} \qquad (121)$$

Les contraintes du parallélogramme nous permettent d'injecter dans le système une information de distance et une information d'angle.

Contraintes d'orthogonalité

L'orthogonalité est une caractéristique importante de la structure Euclidienne. La détection de plans orthogonaux permet, au moins partiellement de définir un repère Euclidien de l'espace 3-D. On constate que la projection de droites orthogonales $\langle O, M \rangle$ et $\langle O, N \rangle$ produit deux plans orthogonaux dans l'espace (Figure 59). Quand ces plans coupent des surfaces planes de l'environnement, ils produisent un segment lumineux en leur superficie qui sera capturé par la caméra. Nous avons donc deux droites $\langle O', M' \rangle$ et $\langle O', N' \rangle$ dans l'espace qui appartiennent à des plans orthogonaux. Puisque *O'* et *M'* appartiennent au même plan vertical et *O'* et *N'* appartiennent au même plan horizontal, on a :

$$x_{O'} = x_{M'} \text{ et } y_{O'} = y_{N'} \qquad (122)$$

Deux vecteurs sont orthogonaux si leur produit scalaire est nul, donc :

$$\langle O', M' \rangle \perp \langle O', N' \rangle \Leftrightarrow \overrightarrow{O'M'} \cdot \overrightarrow{O'N'} = 0 \tag{123}$$

En développant le second membre, on obtient :

$$(x_{O'} - x_{M'})(x_{O'} - x_{N'}) + (y_{O'} - y_{M'})(y_{O'} - y_{N'}) + (z_{O'} - z_{M'})(z_{O'} - z_{N'}) = 0 \tag{124}$$

En combinant avec (122), on obtient finalement :

$$\langle O', M' \rangle \perp \langle O', N' \rangle \Leftrightarrow z_{O'} = z_{M'} \text{ ou } z_{O'} = z_{N'} \tag{125}$$

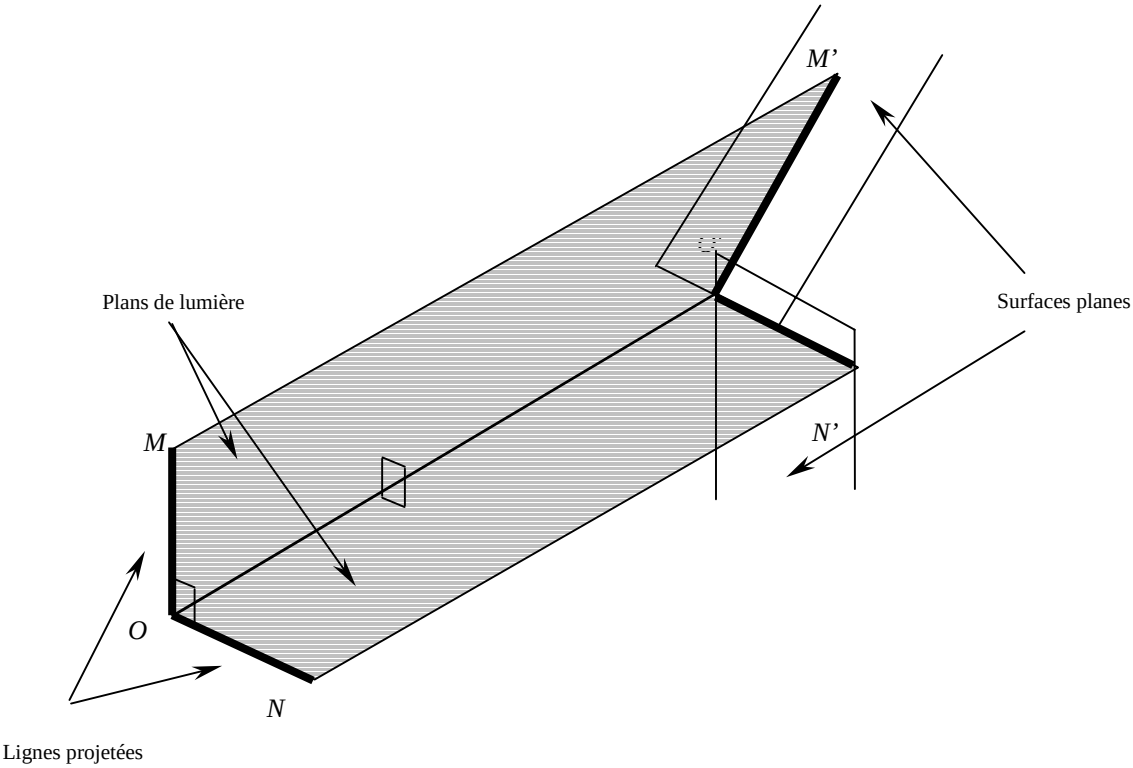


Figure 59. Contraintes d'orthogonalité

Si les conditions imposées par (125) sont vérifiées, nous obtenons une contrainte d'orthogonalité ; dans le cas contraire, nous obtenons une *contrainte d'orthogonalité réduite* :

$$(x_{O'} - x_{M'})(x_{O'} - x_{N'}) + (y_{O'} - y_{M'})(y_{O'} - y_{N'}) = 0 \tag{126}$$

Les équations à résoudre seront donc du type :

$$\begin{aligned}
&\left(\frac{w_{11}\tilde{x}_{O'}+w_{12}\tilde{y}_{O'}+w_{13}\tilde{z}_{O'}+w_{14}\tilde{t}_{O'}}{w_{41}\tilde{x}_{O'}+w_{42}\tilde{y}_{O'}+w_{43}\tilde{z}_{O'}+w_{44}\tilde{t}_{O'}}-\frac{w_{11}\tilde{x}_{M'}+w_{12}\tilde{y}_{M'}+w_{13}\tilde{z}_{M'}+w_{14}\tilde{t}_{M'}}{w_{41}\tilde{x}_{M'}+w_{42}\tilde{y}_{M'}+w_{43}\tilde{z}_{M'}+w_{44}\tilde{t}_{M'}}\right)\cdot\\
&\left(\frac{w_{11}\tilde{x}_{O'}+w_{12}\tilde{y}_{O'}+w_{13}\tilde{z}_{O'}+w_{14}\tilde{t}_{O'}}{w_{41}\tilde{x}_{O'}+w_{42}\tilde{y}_{O'}+w_{43}\tilde{z}_{O'}+w_{44}\tilde{t}_{O'}}-\frac{w_{11}\tilde{x}_{N'}+w_{12}\tilde{y}_{N'}+w_{13}\tilde{z}_{N'}+w_{14}\tilde{t}_{N'}}{w_{41}\tilde{x}_{N'}+w_{42}\tilde{y}_{N'}+w_{43}\tilde{z}_{N'}+w_{44}\tilde{t}_{N'}}\right)+\\
&\left(\frac{w_{21}\tilde{x}_{O'}+w_{22}\tilde{y}_{O'}+w_{23}\tilde{z}_{O'}+w_{24}\tilde{t}_{O'}}{w_{41}\tilde{x}_{O'}+w_{42}\tilde{y}_{O'}+w_{43}\tilde{z}_{O'}+w_{44}\tilde{t}_{O'}}-\frac{w_{21}\tilde{x}_{M'}+w_{22}\tilde{y}_{M'}+w_{23}\tilde{z}_{M'}+w_{24}\tilde{t}_{M'}}{w_{41}\tilde{x}_{M'}+w_{42}\tilde{y}_{M'}+w_{43}\tilde{z}_{M'}+w_{44}\tilde{t}_{M'}}\right)\cdot\\
&\left(\frac{w_{21}\tilde{x}_{O'}+w_{22}\tilde{y}_{O'}+w_{23}\tilde{z}_{O'}+w_{24}\tilde{t}_{O'}}{w_{41}\tilde{x}_{O'}+w_{42}\tilde{y}_{O'}+w_{43}\tilde{z}_{O'}+w_{44}\tilde{t}_{O'}}-\frac{w_{21}\tilde{x}_{N'}+w_{22}\tilde{y}_{N'}+w_{23}\tilde{z}_{N'}+w_{24}\tilde{t}_{N'}}{w_{41}\tilde{x}_{N'}+w_{42}\tilde{y}_{N'}+w_{43}\tilde{z}_{N'}+w_{44}\tilde{t}_{N'}}\right)+\\
&\left(\frac{w_{31}\tilde{x}_{O'}+w_{32}\tilde{y}_{O'}+w_{33}\tilde{z}_{O'}+w_{34}\tilde{t}_{O'}}{w_{41}\tilde{x}_{O'}+w_{42}\tilde{y}_{O'}+w_{43}\tilde{z}_{O'}+w_{44}\tilde{t}_{O'}}-\frac{w_{31}\tilde{x}_{M'}+w_{32}\tilde{y}_{M'}+w_{33}\tilde{z}_{M'}+w_{34}\tilde{t}_{M'}}{w_{41}\tilde{x}_{M'}+w_{42}\tilde{y}_{M'}+w_{43}\tilde{z}_{M'}+w_{44}\tilde{t}_{M'}}\right)\cdot\\
&\left(\frac{w_{31}\tilde{x}_{O'}+w_{32}\tilde{y}_{O'}+w_{33}\tilde{z}_{O'}+w_{34}\tilde{t}_{O'}}{w_{41}\tilde{x}_{O'}+w_{42}\tilde{y}_{O'}+w_{43}\tilde{z}_{O'}+w_{44}\tilde{t}_{O'}}-\frac{w_{31}\tilde{x}_{N'}+w_{32}\tilde{y}_{N'}+w_{33}\tilde{z}_{N'}+w_{34}\tilde{t}_{N'}}{w_{41}\tilde{x}_{N'}+w_{42}\tilde{y}_{N'}+w_{43}\tilde{z}_{N'}+w_{44}\tilde{t}_{N'}}\right)=0
\end{aligned} \qquad (127)$$

Il suffira d'ôter le membre en *z* pour obtenir les contraintes d'orthogonalité réduite.

3.3.2.5 Résolution des Equations

Nous avons vu comment poser les équations permettant de reconstruire projectivement une scène et celles, générées par l'analyse des images, permettant de redresser cette reconstruction dans un espace euclidien. Nous détaillons maintenant comment il est possible de résoudre ces équations et quelles contraintes mathématiques il faut y ajouter.

Reconstruction projective

La reconstruction projective consiste, on l'a vu, à résoudre le système d'équations (102). Au sens des moindres carrés, il s'agit donc de minimiser la distance séparant les points images et projetés mesurés avec les points images et projetés obtenus en estimant les matrices de projection et les points 3-D :

$$\xi^2 = \sum_{ij}\left(U_{ij} - \frac{m_{11}^{(j)}x_i + m_{12}^{(j)}y_i + m_{13}^{(j)}z_i + m_{14}^{(j)}t_i}{m_{31}^{(j)}x_i + m_{32}^{(j)}y_i + m_{33}^{(j)}z_i + m_{34}^{(j)}t_i} \right)^2 + \sum_{ij}\left(V_{ij} - \frac{m_{21}^{(j)}x_i + m_{22}^{(j)}y_i + m_{23}^{(j)}z_i + m_{24}^{(j)}t_i}{m_{31}^{(j)}x_i + m_{32}^{(j)}y_i + m_{33}^{(j)}z_i + m_{34}^{(j)}t_i} \right)^2 \quad (128)$$

avec i=1,...,n (nombre de points à reconstruire) et j={c ou p} respectivement pour la caméra ou le projecteur. Pour obtenir une unique solution, nous avons vu qu'il fallait fixer une base projective, c'est-à-dire ajouter cinq équations du type (128) en remplaçant les coordonnées (x_i, y_i, z_i, t_i) par les coordonnées des points de la base projective canonique.

Enfin, puisque l'on travaille dans l'espace projectif, tous les vecteurs et toutes les matrices rencontrés sont définis à un facteur d'échelle près. De ce fait, les vecteurs des coordonnées des points, ainsi que les matrices de projection devront être mis à l'échelle. Pour chaque point objet de coordonnées homogènes (x_i, y_i, z_i, t_i), on posera donc :

$$x_i^2 + y_i^2 + z_i^2 + t_i^2 = 1 \quad (129)$$

Pour les matrices de projection, on va choisir la contrainte la plus simple, qui consiste à fixer l'un des éléments à 1 :

$$m_{34}^{(c)} = 1 \text{ et } m_{34}^{(p)} = 1 \quad (130)$$

Le système composés des équations (128), (129) et (130) est résolu par l'algorithme de Levenberg-Marquardt [MAR63].

Résolution des contraintes euclidiennes

Pour obtenir la reconstruction euclidienne, c'est le système d'équations (111) qu'il faut résoudre. Les équations sont celles données par les contraintes Euclidiennes générées : leur nombre et leur type dépend donc du type d'image soumise à la mesure. Une contrainte sur le facteur d'échelle de la collinéation à estimer doit être ajoutée. Là encore, c'est l'algorithme de Levenberg-Marquardt qui permet la résolution d'un tel système.

3.4 Résultats Expérimentaux

3.4.1 Test de Colinéarité et de Coplanarité

Les résultats de cette section ont été obtenus à partir d'images capturées dans des conditions de prise de vue réalistes (les images qui apparaissent dans cette section sont, pour permettre plus de clarté, les négatifs des images réellement obtenues). Dans un premier temps, nous avons évalué le test de colinéarité spatiale. La Figure 60 présente trois configurations distinctes de points. La première, en partant du haut, montre quatre points alignés appartenant à un même plan : le birapport mesuré est très proche du birapport théorique et la colinéarité est détectée. Pour la deuxième image, le patron a été projeté sur le coin d'un cube, deux points ont été choisis sur une face et les deux autres sur l'autre face. Les points sont bien reconnus comme n'étant pas colinéaires. Dans le troisième exemple où les points ont été choisis clairement désalignés, l'erreur projective atteint même une valeur de l'ordre de 10^{-1}.

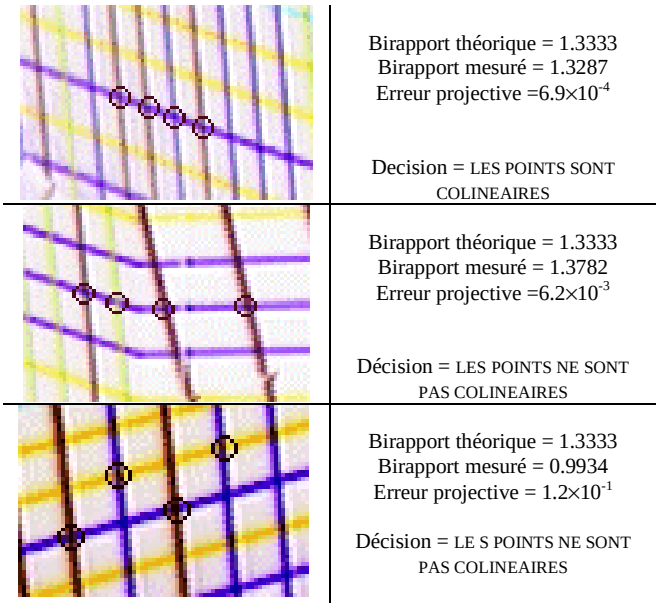


Figure 60. Résultats du test de colinéarité

Nous avons, dans un second temps, évalué le test de coplanarité : la Figure 61 en montre trois exemples. Dans le premier, toujours en partant du haut, le test détecte une configuration planaire de points. Dans le deuxième, on peut voir que le patron est projeté sur une surface irrégulière et le test détecte bien la non-coplanarité des points choisis. Enfin, dans le troisième exemple, les points sont projetés sur le coin d'un cube et l'erreur projective obtenue devient alors importante.

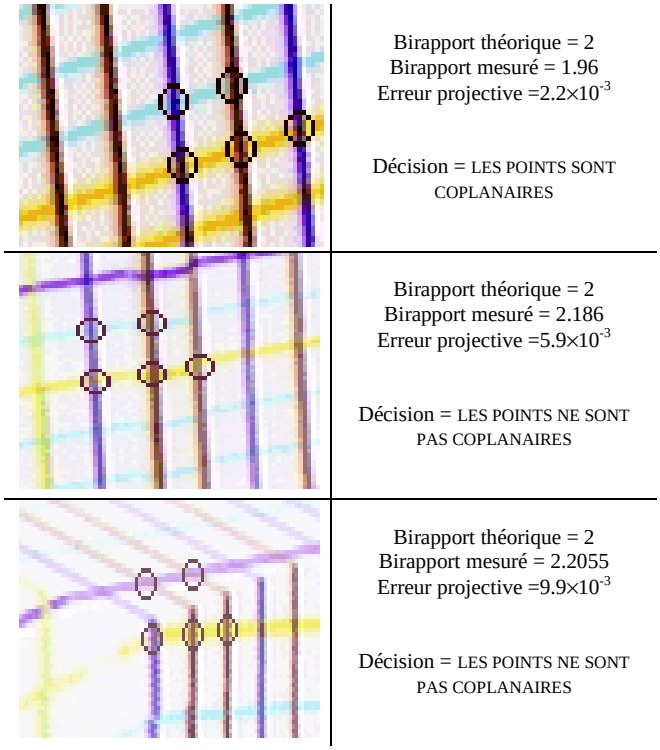


Figure 61. Résultats du test de coplanarité

Les résultats obtenus montrent que la conservation du birapport permet de détecter un ensemble de points coplanaires, ou colinéaires dans l'espace. Le seuil fixé expérimentalement pour discriminer les classifications semble correct et robuste aux erreurs provoquées par l'instabilité du birapport.

3.4.2 Reconstruction à partir d'Images Synthétiques

Coordonnées Euclidiennes de cinq points

On prend ici l'hypothèse que les coordonnées Euclidiennes de cinq points de l'espace sont connues. On fixe le repère de la première caméra à l'origine du repère monde. La matrice de projection ne dépend plus dans ce cas que des paramètres intrinsèques : seulement quatre paramètres doivent être estimés pour obtenir cette matrice. Les coordonnées du point principal sont initialisées avec les coordonnées du centre de l'image ; les coordonnées des points objets sont initialisées avec les coordonnées du barycentre de l'ensemble des points à reconstruire. Bien évidemment, sur des données synthétiques non-bruitées, l'écart entre les points reconstruits et les points à reconstruire est presque nul ; il n'est dû

qu'à l'erreur d'arrondi. Dans le but d'évaluer la robustesse de la méthode, nous avons ajouté un bruit uniforme aux coordonnées pixels. On sait que pour un système de vision en lumière structurée, l'image projetée (la diapositive, par exemple) est parfaitement connue puisque construite : aucun bruit n'est donc ajouté aux coordonnées des points projetés. La reconstruction a été effectuée sur quarante points. Le tableau suivant présente les résultats de dix de ces points, un bruit uniforme a été ajouté aux coordonnées pixels des images de ces points.

Coordonnées réelles			**Erreurs sur les coord. estimées**		
X	***Y***	***Z***	***ΔX***	***ΔY***	***ΔZ***
100	-50	4000	0.518	-0.267	3.95
300	-50	2000	-0.65	-0.242	-1.5
700	-50	4000	0.614	-0.33	6.43
500	-400	4020	-1.132	-1.768	-4.332
300	50	4000	0.091	0.397	2.597
500	50	2000	0.076	-0.119	0.449
900	50	4000	0.13	0.171	2.007
300	-430	3000	0.505	-0.911	5.079
450	75	2500	0.76	-1.154	4.016
705	-120	1000	0.603	-0.827	0.829

Table 9. Erreur de reconstruction — bruit uniforme ±1 pixel

Le second tableau présente les résultats pour les mêmes dix points, cette fois, c'est un bruit uniforme plus important qui a été ajouté aux coordonnées des points images.

Coordonnées réelles			**Erreurs sur les coord. estimées**		
X	***Y***	***Z***	***ΔX***	***ΔY***	***ΔZ***
100	-50	4000	-3.254	1.824	-25.173
300	-50	2000	2.036	-1.664	7.267
700	-50	4000	-7.926	0.819	-26.925
500	-400	4020	-4.309	2.186	-20.012
300	50	4000	-0.919	-0.833	-3.775
500	50	2000	-1.179	0.121	-3.286
900	50	4000	-4.718	-1.120	-12.154
300	-430	3000	-2.174	4.496	-13.745
450	75	2500	1.791	-1.306	6.928
705	-120	1000	1.852	-0.577	-0.703

Table 10. Erreur de reconstruction — bruit uniforme ±2 pixels

On a pu remarquer qu'en utilisant les coordonnées Euclidiennes de cinq points de l'espace, les résultats se dégradaient très rapidement avec le bruit. Cette méthode semble très sensible à la position des cinq points utilisés. La Figure 62 illustre la dégradation des résultats en présence de bruit.

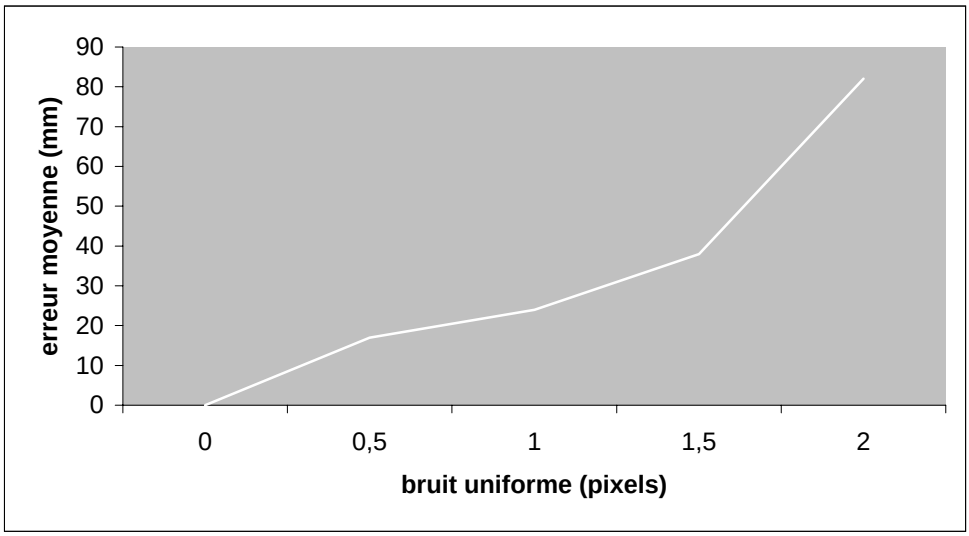


Figure 62. Erreur de mesure en fonction du bruit

Contraintes euclidiennes

Cette fois, la reconstruction est effectuée en deux étapes distinctes : d'abord, une reconstruction projective en affectant à cinq points de la scène les coordonnées de la base projective canonique ; ensuite, une reconstruction Euclidienne en détectant les contraintes géométriques telles que le parallélisme et l'orthogonalité qui relient les points à reconstruire.
Pour valider la reconstruction projective, les points reconstruits dans l'espace projectif sont retro-projetés au travers des matrices de projection (elles aussi exprimées dans un espace projectif) sur les plans images. L'erreur résiduelle est évaluée entre les points théoriques et les points retro-projetés (respectivement représentés par des cercles et des croix dans la). La conclusion est que la reconstruction projective offre de bons résultats et ce, en quelques itérations (typiquement entre dix et trente). Cependant, pour assurer la convergence de l'algorithme, le positionnement relatif des points choisis comme base projective doit correspondre grossièrement au positionnement de la Figure 63 (où les ei correspondent aux points de la base projective canonique). Si, par exemple, *e5* se trouve de l'autre côté du repère formé par les quatre autres points, l'algorithme ne pourra pas converger.

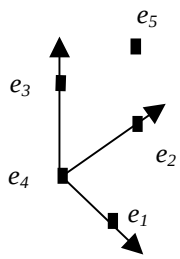


Figure 63. Les cinq points de la base

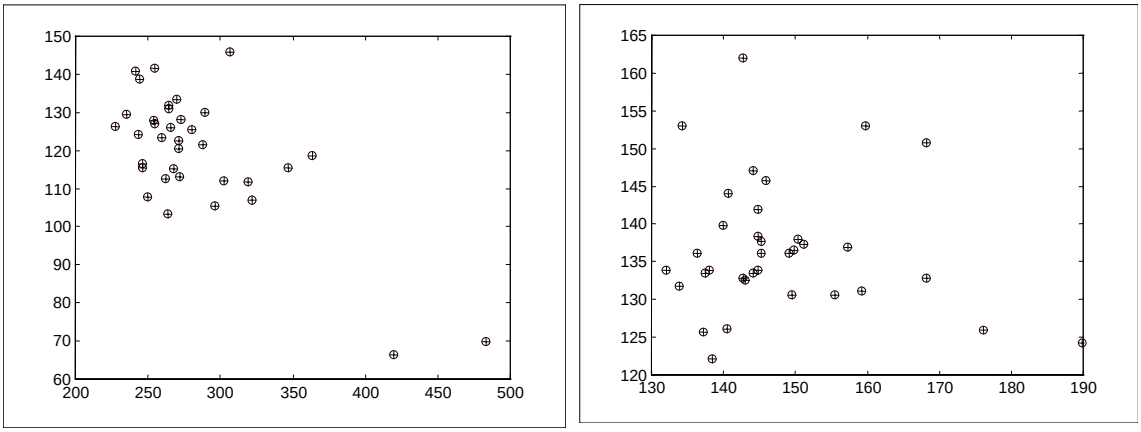


Figure 64. Validation de la reconstruction projective – plans images de gauche et de droite

Nous avons utilisé différentes contraintes Euclidiennes telles que fixer l'origine, le parallélisme, les distances, etc. En re-calant et en mettant à l'échelle la reconstruction calculée, il est possible de valider les résultats en superposant ce que l'on obtient avec ce que l'on devrait obtenir (Figure 65). Sur cette figure, les cercles représentent les points 3-D, les croix représentent la base projective choisie et les astérisques représentent les points 3-D mesurés. Sur cet exemple, l'erreur absolue moyenne est inférieure à 7 mm et l'erreur absolue maximale est de 35 mm.

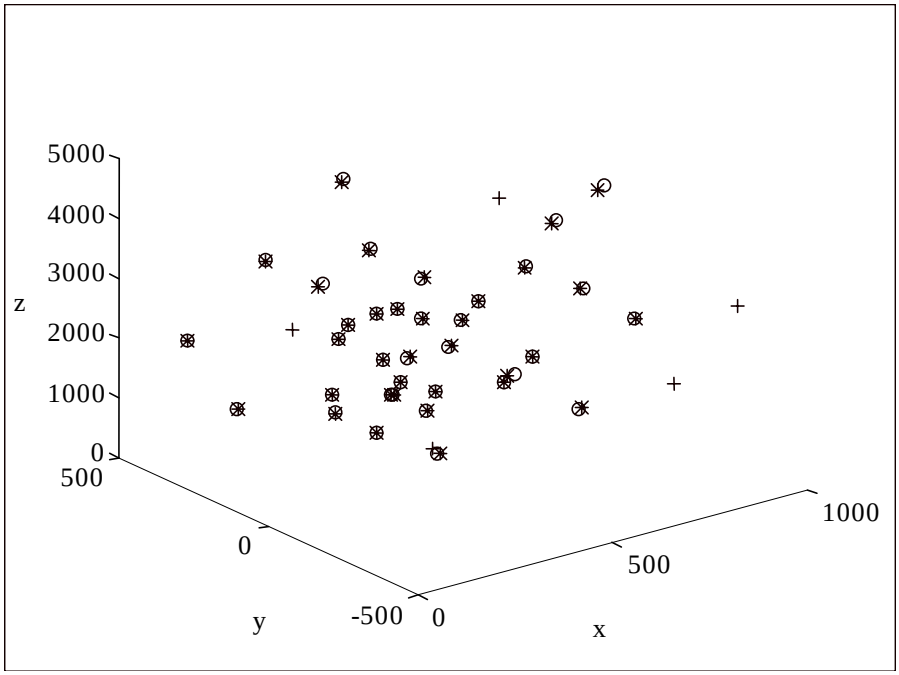
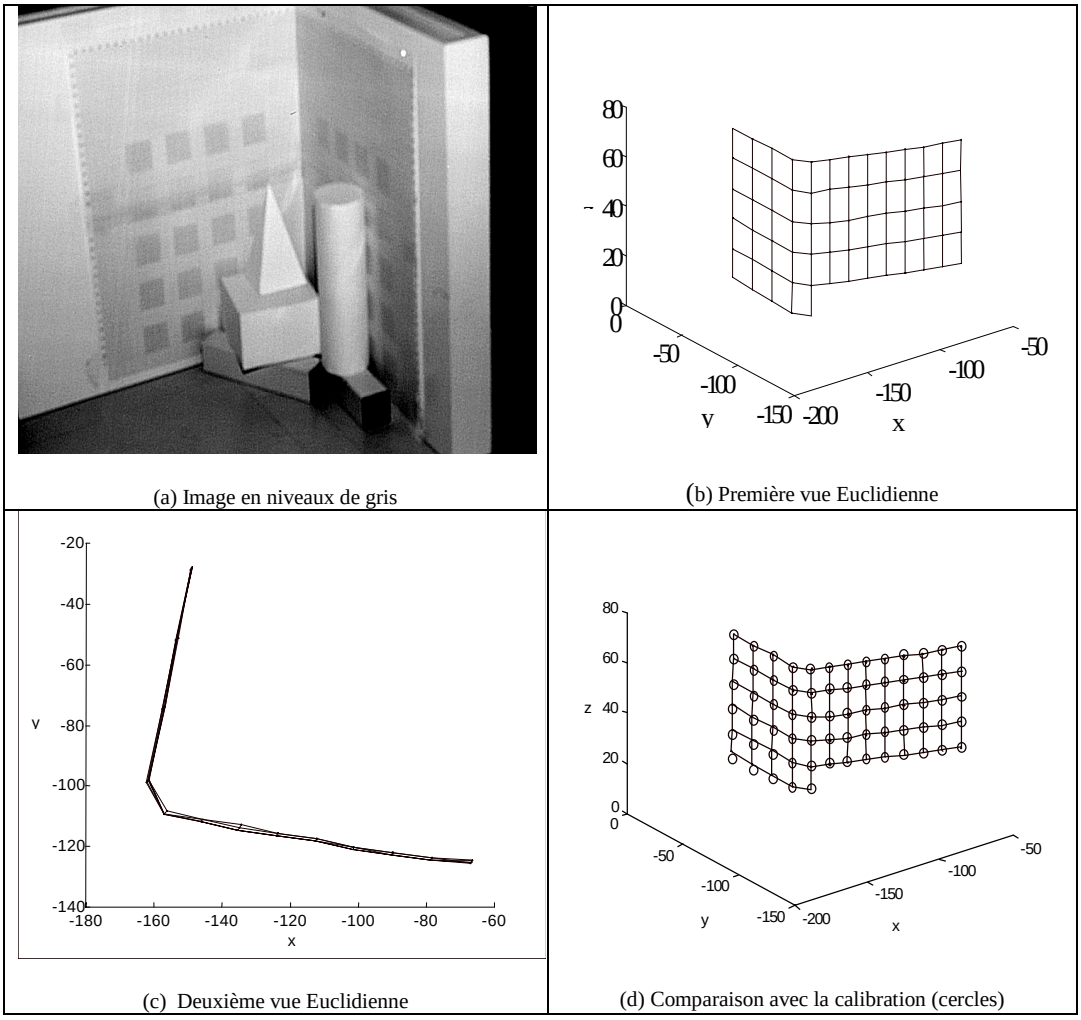


Figure 65. Reconstruction Euclidienne

3.4.3 Reconstruction à partir d'Images Réelles

Nous avons tout d'abord reconstruit soixante-dix points d'une scène réelle qui représente un cube (Figure 66a). La reconstruction Euclidienne a été calculée à partir d'une reconstruction projective en appliquant les contraintes définies en 3.5.3. Pour illustrer la précision de la méthode, nous avons comparé la reconstruction non-calibrée à une reconstruction obtenue après calibration. Les résultats sont données en Figure 66d : ils sont tout à fait acceptables pour l'utilisation que nous entendons faire du capteur de vision.



(a) Image en niveaux de gris

(b) Première vue Euclidienne

(c) Deuxième vue Euclidienne

(d) Comparaison avec la calibration (cercles)

Figure 66. Reconstruction Euclidienne par contraintes géométriques

La Figure 67 présente des résultats de reconstruction Euclidienne à partir d'une image acquise dans des conditions réalistes, c'est-à-dire sans se préoccuper de l'achromatisme des objets la composant et en respectant le positionnement du capteur pour une utilisation en robotique mobile. Seuls les traits dessinés ont été reconstruits. On peut noter que le parallélisme, l'orthogonalité, les distances relatives ont été correctement restitués. Quelques itérations seulement ont été nécessaires au passage de la reconstruction projective à la reconstruction Euclidienne.

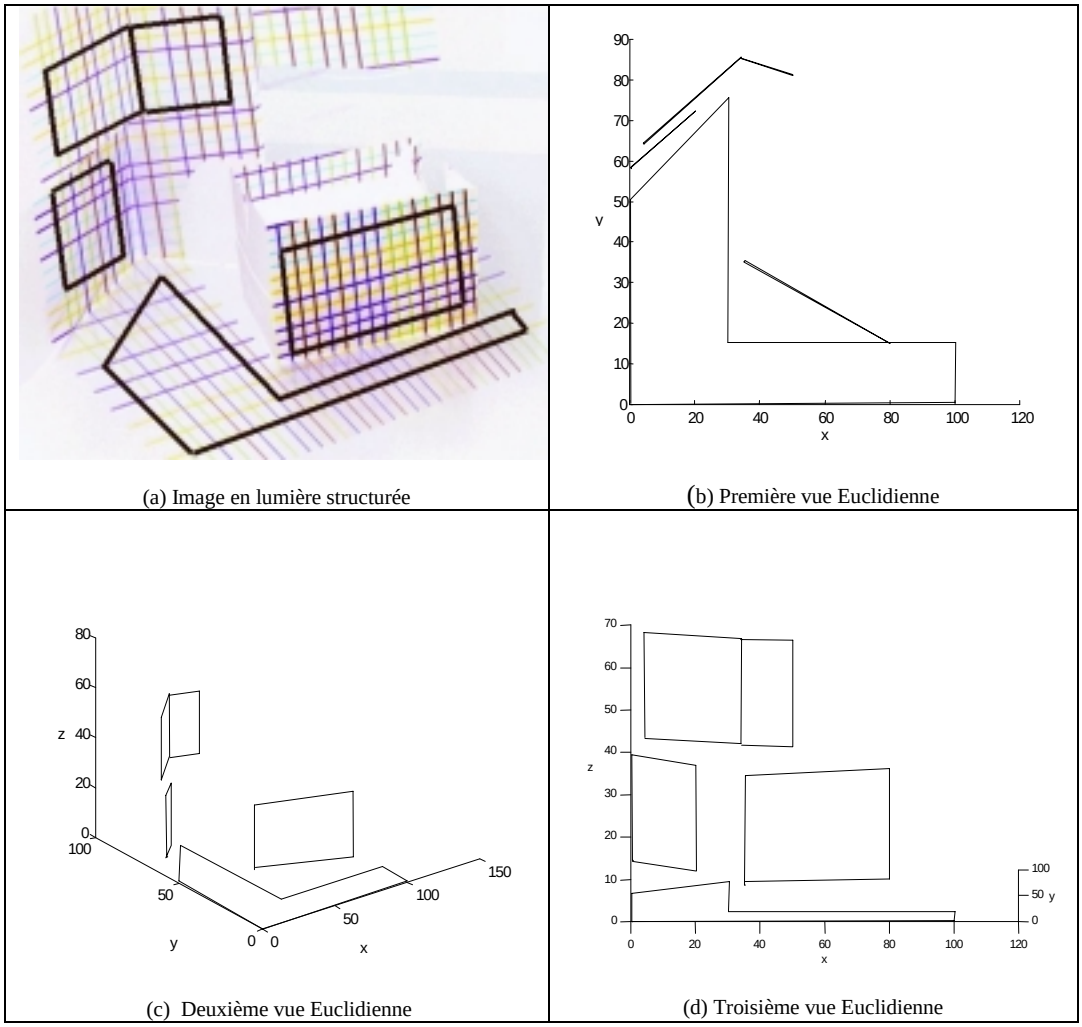


Figure 67. Reconstruction Euclidienne par contraintes géométriques

3.5 Conclusion

Nous nous sommes inspirés, pour ce chapitre, du rapport entre les principales géométries de la vision par ordinateur : la géométrie projective qui permet de modéliser la prise de vue de manière élégante et la géométrie Euclidienne qui permet de reconstruire et mesurer l'espace observé de manière facilement interprétable (ceci nous a conduit à présenter la géométrie affine qui permet le passage de l'une à l'autre, voir Annexe B). Nous nous sommes particulièrement penchés sur la modélisation des capteurs de vision et sur les techniques de reconstruction tri-dimensionnelle. Nous avons montré que la reconstruction à partir d'un capteur calibré rendait difficile son utilisation en robotique mobile quand on désire conférer un comportement adaptatif au capteur. Nous nous sommes donc naturellement orientés vers une technique de reconstruction non-

calibrée : nous avons montré qu'il était possible d'adapter quelques-unes de ces techniques à la vision en lumière structurée et nous avons donné les contraintes que celle-ci imposait. Nous avons également pris garde que cette technique de reconstruction projective permette de modifier les paramètres de mise au point, d'ouverture et de zoom lors de la mesure. Finalement, nous avons proposé une méthode de reconstruction Euclidienne basée sur la seule analyse des images en lumière structurée. Puisqu'il existe un rapport entre les points projetés, les plans de lumière générés par la projection et les points capturés, il est possible de déduire des contraintes, utilisables pour redresser une reconstruction projective dans l'espace Euclidien, de l'analyse des images reliant les points tri-dimensionnels. Ces résultats expérimentaux ont permis de valider la méthode.

Les résultats que nous présentons reposent sur une reconstruction projective de la scène effectuée à partir de la méthode d'estimation des paramètres. Cette méthode nécessite la définition d'une base projective de l'espace : cinq points, non quatre à quatre coplanaires et non trois à trois colinéaires, pour assurer leur indépendance linéaire. Pour cela, nous proposons deux tests simples, basés sur la conservation du birapport, pour détecter si quatre points sont colinéaires et si cinq points sont coplanaires. Nous présentons des résultats expérimentaux qui valident l'efficience de ces tests. Ces tests sont également utilisés pour détecter si les quatre sommets d'un parallélogramme détecté dans l'image sont bien coplanaires dans l'espace (certaines surfaces illuminées, filmées sous certains angles, peuvent donner un parallélogramme dans l'image sans que les sommets soient coplanaires). On peut également utiliser le test de colinéarité et de coplanarité comme contraintes Euclidiennes reliant les points de l'espace. Enfin, étant donné que les contraintes Euclidiennes générées par l'analyse des images en lumière structurée prennent l'hypothèse que le comportement géométrique du capteur est affine, nous proposons un test permettant de valider l'utilisation d'un tel modèle. Tous les tests proposés ne requièrent qu'une simple analyse d'images : ils respectent donc la contrainte de non-calibration que nous nous sommes fixés. L'avantage principal, par rapport à des méthodes similaires dédiées à la stéréovision, vient du fait que les contraintes Euclidiennes générées dépendent des relations entre les points du motif structurant et pas directement des relations entre les points objets. Au surplus, la projection du motif garantit un nombre suffisant de contraintes, même si l'environnement observé est peu structuré.

Chapitre 4 :
Fonctions Visuelles pour la Navigation

4.1 Introduction

Après avoir traité de la segmentation des images en lumière structurée, puis de la reconstruction tri-dimensionnelle de la scène obtenue à partir de telles images, nous proposons, dans ce dernier chapitre, des algorithmes et méthodes plus spécifiquement orientés "navigation". Nous avons décomposé cette étude en deux grandes parties : la première est dédiée à la détection des obstacles, la seconde à l'analyse du mouvement et plus précisément l'analyse de la cinématique de la scène.

La fonction visuelle la plus élémentaire, et certainement la plus indispensable, pour un robot mobile amené à se mouvoir dans un environnement inconnu ou connu partiellement, est sans aucun doute la détection des obstacles. Elle est à la base des algorithmes d'évitement, de calcul de trajectoire, et surtout, elle est la première garante de la sécurité du robot et de son environnement. Nous proposons, en guise d'introduction, un inventaire des méthodes de détection d'obstacles, qui dépendent du type d'image dont on dispose et de l'application à développer, que l'on trouve dans la littérature. De là, nous déduisons les critères qui nous ont semblé essentiels pour la détection des obstacles : rapidité, simplicité, généricité. La méthode que nous proposons est directement issue du traitement des images. Nous montrons que la détection d'une quasi-verticale dans l'image équivaut à la détection d'un obstacle dans l'environnement du robot. L'extraction des droites de l'image, exposée dans le deuxième chapitre de ce mémoire, nous indique donc directement la présence ou non d'un obstacle (puisque la transformée de Hough nous donne explicitement l'orientation des droites extraites). Nous proposons ensuite une extension de cette méthode basée sur des calculs géométriques très simples en vue d'obtenir une carte de l'espace libre de l'environnement. Nous présentons pour finir des résultats expérimentaux relatifs à ces deux algorithmes.

Il nous a semblé d'autre part utile d'aborder le problème de l'analyse du mouvement dans ce mémoire. On peut, grâce à elle, connaître le déplacement du robot dans l'environnement ou déduire la présence d'obstacles mobiles. Elle peut également fournir des informations pour l'initialisation d'un algorithme de localisation, pour le calcul de trajectoire et l'évitement d'obstacles ou pour la construction itérative d'une carte de l'environnement. En premier lieu, nous proposons une étude qui permet de positionner la vision en lumière structurée dans la problématique d'analyse du mouvement : nous soulignons le problème du mouvement en général, et des mouvements singuliers en particulier ; nous définissons le mouvement apparent dans des images en lumière structurée et nous en déduisons une méthode de détection du mouvement. Il nous a paru difficile

d'aller plus loin dans cette voie, c'est pourquoi, en second lieu, nous proposons une méthode d'estimation du déplacement du robot par mise en correspondance de plans. Ce sont les résultats expérimentaux pour cette dernière méthode qui viennent clore la partie d'analyse du mouvement de ce chapitre.

4.2 La Détection des Obstacles

La détection d'obstacles est une fonction-clef en robotique mobile ; elle est une des fonctions perceptives élémentaires pour un robot devant évoluer dans un environnement inconnu ou partiellement connu. En effet, à moins d'un environnement statique ou donnant lieu à des mouvements exactement déterminés, elle est un moment incontournable des mécanismes perceptifs et, en ce sens, elle va provoquer, orienter, contraindre l'action du robot. Si l'on définit l'autonomie d'un robot comme étant la capacité à se mouvoir (système automoteur) alliée à celle d'acquérir un certain nombre d'informations sur son environnement et à des capacités décisionnelles lui permettant d'interpréter ces informations et d'agir ou réagir en conséquence, on comprend que la détection d'obstacles en est une condition nécessaire.
Comme le rappellent fort justement Zhang, Weiss et Hanson [ZHA97B], formuler une définition précise et rigoureuse de ce qu'est un obstacle est étonnamment difficile ; intuitivement, il s'agit d'un objet, quel qu'il soit, obstruant le mouvement d'une personne, d'un véhicule ou d'un robot. Les apparences qu'il peut recouvrir sont des plus diverses, virtuellement infinies, d'où la difficulté de le *caractériser* de manière générique (sans être trop général) et précise (sans être trop spécifique). L'importance de la caractérisation est primordiale puisque nous verrons qu'elle conditionne la phase de traitements du système de détection en termes de portabilité, charge de calcul et qualité de l'information.

4.2.1 Etat de l'Art

Les techniques de détection d'obstacles reposent sur une caractérisation *a priori* des obstacles que le système est susceptible de rencontrer. Cette caractérisation dépend de l'information perçue par le système, mais également du contexte environnemental et des objectifs fixés. Les attributs caractéristiques pris en compte sont de différents types : *photométriques* (niveau de gris, histogramme, portion d'image), *morphologique* (élément de contour ou toute autre information sur la forme de l'obstacle projetée sur l'image), *physique* (information sur la forme propre de l'obstacle indépendamment de sa projection sur l'image), *cinématique* (obstacles mobiles) ; leur rôle étant de discriminer, dans l'image, les régions abritant des obstacles des régions dites *d'espace libre*.

Caractérisation photométrique

Il s'agit principalement de techniques de reconnaissance de formes ou de discrimation de textures. Les modèles ou classes d'objets mémorisés doivent répondre aux critères d'invariance, *i.e.* la représentation de l'objet doit être indépendante de sa position dans l'image, de son échelle, de son orientation, de sa projection sur le plan image (éventuellement) et des conditions d'illumination ; en fait, elle doit être la plus indépendante possible des conditions de prise de vue.
Efenberger et Graefe [EFE96] ont proposé une représentation pour la reconnaissance d'objets dans une scène naturelle. Les modèles d'objets à reconnaître sont des portions d'image sous-échantillonnées dont l'intervalle d'échantillonnage dépend de la distance de prise de vue, ce qui assure une représentation unique pour chaque objet (représenté par un nombre fixé de pixels) invariante en taille et en position. Une fonction de similarité rend compte de la ressemblance des objets rencontrés avec les modèles mémorisés (on met en correspondance une région de l'image et un modèle de manière à maximiser cette fonction) :

$$\Psi(i,k) = -\sum_j \sum_l \left| s(i+j,k+l) - m(j,l) \right| \tag{131}$$

Où (*i*,*k*) représente le coin supérieur gauche de la portion d'image à analyser ; *s*(*x*,*y*) le niveau de gris au point de coordonnées *x* et *y* ; *m*, l'image de référence. Pour rendre cet algorithme plus robuste aux conditions d'illumination (en particulier, pour garantir l'indépendance de la comparaison vis-à-vis des transformations $a \cdot x + b$), une seconde fonction de similarité a été utilisée :

$$\Phi(i,k) = \frac{\sum_j \sum_l s(i+j,k+l) \cdot m(j,l)}{\sqrt{\sum_j \sum_l s^2(i+j,k+l) \cdot \sum_j \sum_l m^2(j,l)}} \tag{132}$$

D'une manière générale, les algorithmes basés sur une telle représentation souffrent d'une relative lenteur, sont sensibles au bruit et aux conditions d'illumination. En particulier, plus l'objet est éloigné, moins son histogramme est étendu, et plus la sensibilité au bruit est grande. S'ils permettent une reconnaissance des objets (et pas seulement leur détection), ils nécessitent une connaissance *a priori* du type d'obstacle que le véhicule peut recontrer (piétons, voitures, camions, etc.) et une modélisation des obstacles pris sous différents angles est souvent rendue nécessaire.
Une version plus élaborée est présentée par Tsinas et Graefe [TSI93]. Dédiée à la reconnaissance d'obstacles en temps réel pour un véhicule intelligent, elle exploite

les capacités des réseaux de neurones. Un premier module se charge de la détection et du suivi des bords de route (segmentation, puis reconnaissance du bord de route par réseau de neurones). Une fenêtre est ensuite déplacée le long de ces bords, normalisée puis injectée à l'entrée d'un second réseau de neurones pour la reconnaissance d'objets. En sortie de réseau, on obtient une réponse oui/non quant à la détection d'un obstacle. L'apprentissage des différentes classes d'objets est effectué classiquement par un algorithme de rétro-propagation du gradient. L'utilisation d'un réseau de neurones (qui, par rapport à l'exemple précédent, remplace la procédure de mise en correspondance) accélère la phase de reconnaissance/détection et rend le système plus robuste au bruit.
Dans les travaux de Hötter, Mester et Müller [HOT96], on retrouve une modélisation de l'obstacle par ses caractéristiques photométriques. L'image est partitionnée en *cellules de détection* (*detection cell*) dans lesquelles des informations statistiques, comme l'espérance et la variance des niveaux de gris, sont calculées. Elles préjugent de la présence ou non d'un obstacle en leur sein. L'avantage de ce procédé provient du fait qu'on manipule des paramètres globaux (des moments) et non plus simplement des données locales (niveaux de gris) ; la robustesse au bruit s'en trouve renforcée. En revanche, ces caractéristiques ne suffisent pas, en général, à discriminer convenablement la scène et les obstacles, des moments d'ordre supérieur seraient nécessaires ; elles sont utilisées comme indices pour minimiser l'espace de recherche ou sont membres d'un ensemble pluriel de paramètres discriminants (remarque : leur calcul est couplé à un module de détection de mouvement et suivi d'un module de décision stochastique ; la décision est prise pour chaque cellule).

Caractérisation morphologique

Les algorithmes utilisés sont proches de ceux décrits dans la section précédente (apprentissage de classes d'objets ou codage, mise en correspondance, fonctions de similarité). La différence principale réside dans le choix des primitives : ce sont, en général, des segments ou des éléments de contour. Il s'agit là encore plus de reconnaissance que de simple détection d'obstacles. Une fonction de similarité vérifie leur cohérence par rapport à un modèle codé ou une classe d'objets. Les algorithmes fondés sur une telle caractérisation bénéficient d'une plus grande robustesse au bruit (du fait des primitives choisies, elles-mêmes moins sensibles au bruit que les simples pixels). Ils partagent des limitations (dépendance vis-à-vis de l'angle de vue, connaissance a priori des obstacles) avec la caractérisation photométrique. Ce sont des propriétés de l'image qui sont modélisées et pas des caractéristiques intrinsèques de l'obstacle.
Efenberger et Graefe [EFE96] ont étendu leur méthode (voir ci-dessus) en utilisant des éléments de contour pour primitives. Ils ont ainsi pu comparer les performances de ces deux caractérisations. Ils montrent que les méthodes par éléments de contour sont plus rapides et moins sensibles au bruit, bien qu'elles soient susceptibles de provoquer de fausses alarmes.

Meygret [MEY90] propose une détection d'obstacles (véhicules automobiles en environnement routier) en évaluant leur ressemblance à un modèle préalablement codé. Ce modèle décrit la position et l'orientation relative de quelques segments pertinents rappelant la forme du type de véhicule à identifier. Les configurations proches des modèles seront considérées comme des obstacles (émission d'hypothèses 2-D). Cette reconnaissance est suivie d'une phase de coopération 3-D pour lever les ambiguïtés (validation 3-D). Le principal avantage d'un tel modèle, outre sa simplicité, est d'être invariant par projection perspective pour un large domaine de variation du point d'observation. Un espace de recherche est tout d'abord défini dans l'image en utilisant l'information 3-D disponible. Les segments horizontaux et parallèles sont groupés par des critères de similarité et de proximité. La reconnaissance d'un obstacle est, en dernier lieu, validée par le calcul de la largeur et de la hauteur réelle de l'objet.

Caractérisation physique

Zhang, Weiss et Hanson [ZHA97B] supposent que tous les obstacles sont de géométrie quasi-verticale (c'est une hypothèse assumée dans de nombreux travaux comme nous le verrons tout au long de cette section). Cela leur permet d'assimiler la détection d'obstacles à la résolution d'un système d'équations dépendant du vecteur de déplacement et des paramètres de calibration de la caméra. Les auteurs démontrent qu'un tel système admet une solution si et seulement si tous les points considérés appartiennent au plan du sol. Ils proposent deux algorithmes qualitatifs (réponse oui/non quant à la présence d'un obstacle dans la scène) pouvant indifféremment fonctionner en monovision ou en stéréovision et un algorithme quantitatif (reconstruction partielle de la scène) nécessitant la stéréovision. Le premier des algorithmes qualitatifs, baptisé *KGP* pour *Known Ground Plane*, suppose connue l'équation du plan du sol ; le second, *UGP* pour *Unknown Ground Plane*, considère uniquement que le sol est plan. Dans un environnement conforme aux hypothèses posées, les performances (robustesse au bruit par rapport à la taille des obstacles de l'algorithme *KGP* sont supérieures à celles de *UGP* (l'une des questions examinées par les auteurs est l'influence de l'a priori géométrique sur la performance des algortihmes de détection d'obstacles). Le troisième algorithme, quantitatif, estime l'équation du plan du sol et ne nécessite qu'une calibration partielle du système de vision ; il donne de meilleurs résultats que les deux précédents au prix d'un temps d'exécution plus long.
Nous allons décrire plus précisément le premier algorithme duquel découlent les deux autres. Cet algorithme utilise l'équation du flot optique en fonction du déplacement du robot et de ses paramètres de calibration :

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}=\begin{bmatrix}-xy & 1+x^2 & -y & K & 0 & -xK\\ -(1+y)^2 & xy & x & 0 & K & -yK\end{bmatrix}.\begin{pmatrix}\Omega_x\\ \Omega_y\\ \Omega_z\\ t_x\\ t_y\\ t_z\end{pmatrix} \tag{133}$$

Où $K = k_x x + k_y y + k_z$ représente l'équation du plan du sol, Ω_i la composante de rotation du mouvement et t_i la composante de translation. Ce modèle peut être condensé :

$$\dot{\mathbf{v}} = \mathbf{HV} \tag{134}$$

H est la matrice qui transforme le vecteur des paramètres du mouvement **V** en vecteur du flot optique $\dot{\mathbf{v}}$. Pour *n* points de l'image, on obtient un système d'équations du type :

$$\begin{pmatrix}\dot{\mathbf{v}}_1\\ \dot{\mathbf{v}}_2\\ \vdots\\ \dot{\mathbf{v}}_n\end{pmatrix}=\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\\ \vdots\\ \mathbf{H}_n\end{bmatrix}\cdot\mathbf{V} \tag{135}$$

Que l'on peut écrire :

$$\mathbf{b} = \mathbf{DV} \tag{136}$$

Et de là, on tire que les n points appartiennent tous au plan du sol si et seulement si le système (136) adment une solution ; autrement dit, si et seulement si :

$$Rang(\mathbf{D}) = Rang(\mathbf{Db}) \tag{137}$$

D'après la même hypothèse (les obstacles sont les points n'appartenant pas au plan du sol) mais de manière plus intuitive, Storjohann, Zielke, Mallot et von Seelen [STO90] utilisent l'inversion de la projection perspective (*inverse perspective mapping*) pour détecter la présence d'obstacles dans l'image. Le principe est simple : les points de l'image sont rétro-projetés sur un plan virtuel parallèle au sol (celui-ci est approché localement par un plan). Ceci a pour effet de corriger les

distorsions de l'images dues à la perspective, de reconstruire correctement les points appartenant au plan du sol (et pas les autres) et de redresser le champ du mouvement apparent. Si le sol est structuré (pavage, dallage, motif régulier) on peut, par soustraction d'une image de référence le représentant, déterminer les régions de l'image hors-sol, c'est-à-dire abritant des obstacles, pour un système monovisuel. En stéréovision, on possède deux images I_1 et I_2 et on procède de la sorte :

1. On projette I_1 sur un plan parallèle au sol ; on obtient l'image $I_1^{'}$.
2. On projette $I_1^{'}$, par transformation perspective directe, sur le plan de l'image I_2 ; on obtient $I_1^{''}$.
3. On compare I_2 à $I_1^{''}$, les différences correspondent aux sites de l'image abritant des obstacles.

Cette procédure ne nécessite ni mise en correspondance ni reconstruction, elle donne directement comme résultat une carte de l'espace libre de l'environnement perçu par le robot. Elle est, en revanche, très sensible au bruit. Ce même principe, traité différemment, a également été utilisé par Xie [XIE95].
A partir d'un système trinoculaire, Buffa, Faugeras et Zhang [BUF93] proposent une méthode donnant une carte de l'espace libre et le calcul de la trajectoire du robot au sein de l'environnement. Dans un premier temps, les images sont segmentées et la reconstruction de la scène est effectuée. Les segments reconstruits tri-dimensionnellement sont projetés sur le plan du sol. Par triangulation de Delaunay (dual du diagramme de Voronoï) et en prenant en compte la position du robot au moment de la prise de vue, ils construisent une carte de l'environnement indiquant la position et la forme des obstacles ainsi que les régions cachées. Ils complètent leur procédé par un calcul de trajectoire s'appuyant sur l'algorithme de Dijkstra.

Caractérisation cinématique

Un obstacle correspond cette fois à une région de l'image différant, au sens du mouvement apparent (norme et/ou direction) du reste de la scène. Ces méthodes nécessitent la connaissance a priori ou l'estimation des mouvements de la caméra (il faut, au minimum en connaître la direction). La détection des obstacles et l'analyse de leur mouvement sont effectués simultanément ; en revanche, les obstacles immobiles ne peuvent être détectés. L'idée générale est de compenser le mouvement dominant dans l'image (provoqué par le mouvement de la caméra) et de détecter les régions mal compensées (ou de prédire, à partir d'une image donnée, l'image successive connaissant le mouvement de la caméra). Celles-ci sont supposées abriter un obstacle mobile.
Pour un mouvement rectiligne de la caméra, le champ du mouvement apparent est radial. L'algorithme d'appariement d'histogramme normalisé d'Hashimoto et al.

[HAS96] part de cette constatation : connaissant la dimension et la position d'une région d'image (représentée par son histogramme normalisé), on sait que dans l'image successive cette même région sera positionnée sur la même radiale, réduite ou agrandie suivant que le robot avance ou recule, sauf si cette région correspond au lieu d'un obstacle mobile. La normalisation compense le facteur d'échelle reliant les deux portions d'images homologues. Avant de mettre en oeuvre cet algorithme, il faut choisir les régions de l'image qui sont à tester. L'estimation du flot optique pour l'ensemble des régions de l'image d'une taille donnée serait, pour un algorithme de détection d'obstacles, prohibitive.
Les régions mal compensées peuvent également être caractérisées par l'erreur résiduelle du *foyer d'expansion* (FOE) du mouvement (ou point de convergence du flot optique). Takeda, Watanabe et Onoguchi [TAK96] emploient une telle méthode. Les régions de l'image dans lesquelles cette erreur est importante sont interprétées comme des régions d'obstacle. Le FOE est estimé par la méthode des moindres carrés, connaissant l'expression du flot optique.

4.2.2 Détection d'Obstacles par Extraction des Verticales

Positionnement du capteur

Nous allons considérer, dans la suite de ce chapitre, que les axes *y* des repères caméra et projecteur sont parallèles au plan du sol. Cette hypothèse est moins restrictive que celle du modèle de stéréovision rectifié (Figure 68) ; d'autant moins que nous accordons une tolérance assez lâche au parallélisme. En effet, il suffit d'assurer ici que les verticales du patron de lumière, projetées sur une paroi verticale, restent verticales dans l'image ; ou mieux, que les verticales du patron, projetées sur une paroi quasi-verticale, apparaissent comme quasi-verticales dans l'image. Cette restriction ne constitue pas une perte de généralité pour la méthode. Il est plus aisé, pour la démonstration, de parler de verticales ou quasi-verticales que de droites d'orientation quelconque ; si le positionnement effectif du capteur ne respectait pas cette recommandation, il suffirait d'identifier l'orientation des droites de l'image qui correspond à la projection d'une verticale du patron sur une paroi verticale.

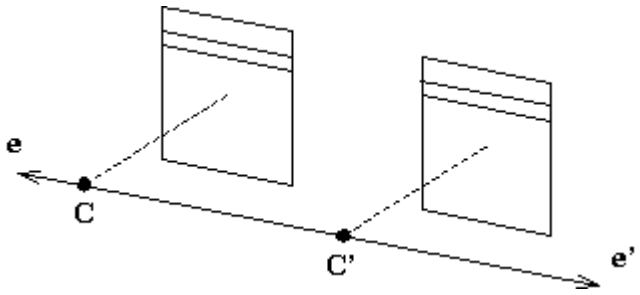


Figure 68. Géométrie d'un système rectifié

Nous allons maintenant préciser plus avant le positionnement de la caméra et du projecteur par rapport à la scène. Nous adoptons une stratégie de détection d'obstacles sur le plan du sol (*ground plane obstacle detection*) ; dans cette optique, la caméra et le projecteur sont légèrement penchés vers le sol de manière à observer l'environnement immédiat du robot, un peu à la manière d'un piéton observant ses pieds lors de la marche. On restreint, de cette manière, le champ de vision, on augmente la résolution spatiale et on réduit les problèmes liés à l'illumination et au traitement des images.
La méthode que nous proposons privilégie les environnements polygonaux ou plans par morceaux, elle ne leur est cependant pas *exclusivement* réservée. Nous montrerons que la détection d'obstacles de surface courbe donne, grâce notamment à l'approximation polygonale effectuée par la transformée de Hough sur le squelette, des résultats exploitables en termes de détection et de cartographie.

Méthodologie de la détection

L'analyse des images en lumière structurée montre que la présence d'un obstacle dans le champ de vision du robot provoque une distorsion locale du motif structuré. On remarque, en particulier, que les éléments du motif épousent les formes des surfaces sur lesquelles ils sont projetés. Si l'on considère les obstacles comme des objets quasi-verticaux (*i.e.* dont la pente est telle qu'elle empêche le robot de franchir la zone de l'espace où ils siègent), on peut en conclure que *la détection des lignes verticales ou quasi-verticales de l'image équivaut à la détection d'un ou plusieurs obstacles dans la scène* (Figure 69).
Nous voyons donc qu'à partir d'un traitement simple des images, il est possible d'obtenir une détection des obstacles, c'est-à-dire la réponse oui/non quant à la présence d'un obstacle dans le champ de vision. Certains auteurs qualifient ce type de détection comme détection *qualitative* des obstacles en oppositon à la détection *quantitative* donnant une information supplémentaire sur la forme des obstacles et leur position. La détection qualitative est suffisante pour stopper le robot s'il est face à un obstacle et le faire changer de direction, c'est ce qu'on demande

principalement à un algorithme de détection d'obstacles. Elle possède en outre l'avantage d'être très rapide et assez générale pour détecter un grand nombre d'obstacles de formes et de tailles variées. Cependant, si l'on connait la position des obstacles, leur forme, même partielle, il est possible d'élaborer une méthode de calcul de trajectoire permettant d'éviter au mieux les zones encombrées de l'environnement. En ce sens, la section qui suit peut être vue comme une extension de la méthode permettant d'obtenir une détection quantitative des obstacles.

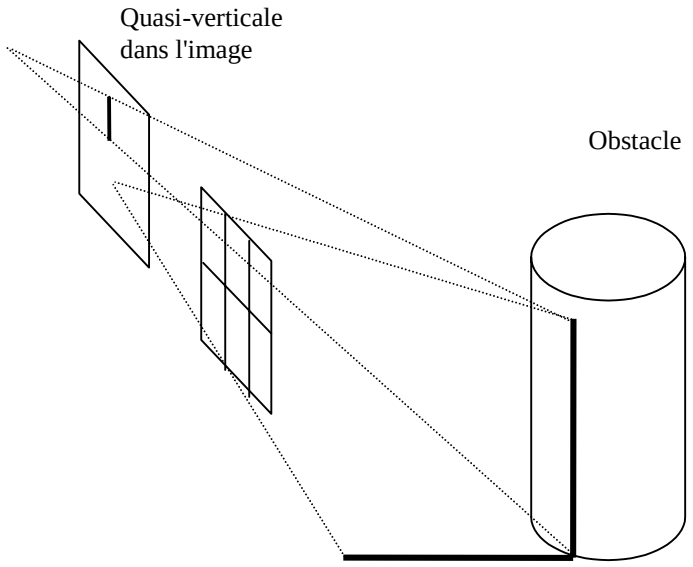


Figure 69. La détection des obstacles en lumière structurée

4.2.3 La Carte de l'Espace Libre

Il ne s'agit pas ici de cartographier l'ensemble de l'environnement au fur et à mesure des observations, mais, à partir d'une seule image, d'identifier les lieux qui abritent des obstacles pour les représenter sous la forme d'une carte bi-dimensionnelle. On appellera la carte ainsi obtenue la *carte de l'espace libre*. Imaginons que les obstacles aient été détectés par la méthode détaillée dans la section précédente. On va d'abord reconstruire (que ce soit à partir d'un capteur calibré ou non apporte peu de différence) les points appartenant aux quasi-verticales extraites. Chaque point est étiqueté relativement à l'élément vertical du motif auquel il appartient. Notre grille comptant 29 éléments verticaux, on attribue à chaque segment *i de l'image* une étiquette E_i dont la valeur est dans l'intervalle [1;29]. Deux points provenant de la même quasi-verticale de l'image auront donc la même étiquette.

On va rejeter, de l'ensemble de points ainsi constitué, tous les points qui ne sont pas nécessaires à la construction de la carte de l'espace libre. On va tout d'abord projeter ces points sur le plan du sol (on prend donc ici l'hypothèse que le sol est plan ou qu'il peut être localement approché par un plan) en appliquant la règle formulée ci-dessous :

$$\begin{cases} z = 0 \quad \text{si} \quad z \geq h \,\text{et}\, z \leq H \\ \text{sinon le point est rejeté} \end{cases} \tag{138}$$

Où *h* est un seuil bas permettant de rejeter les points correspondant aux petites proéminences ou aspérités du sol et *H* est un seuil haut correspondant à la hauteur minimale sous laquelle le robot peut passer.
Une seconde règle est ensuite appliquée qui permet de ne garder, pour un ensemble de points ayant la même étiquette, que celui dont la distance au robot est la plus courte :

$$\begin{aligned} &\forall (x_i, y_i), (x_j, y_j) \,\text{et}\, E_i = E_j \,; \\ &\sqrt{x_i^2 + y_i^2} < \sqrt{x_j^2 + y_j^2} \Rightarrow (x_j, y_j) \,\text{est rejeté} \end{aligned} \tag{139}$$

La dernière étape consiste à ordonner les points suivant leur étiquette et à relier deux points consécutifs si la distance qui les sépare est inférieure à la largeur du robot. On obtient de ce fait une première approximation de la forme des obstacles projetée sur le sol, que l'on peut encore simplifier en ajustant à ces groupes de points un segment ou une ligne brisée. Au final, nous obtenons donc une carte de l'espace libre de l'environnement perçu par le robot.

4.2.4 Résultats Expérimentaux

Détection des obstacles

La méthode de détection des obstacles que nous proposons découle directement du traitement des images décrit dans le deuxième chapitre de ce mémoire : il s'agit en fait d'extraire convenablement les droites verticales et quasi-verticales de l'image. Pour ces expériences, la tolérance autour de la verticale a été fixée à ±10° (les bornes de l'accumulateur de Hough pour la détection des quasi-verticales sont $\theta \in [-10°;10°]$).

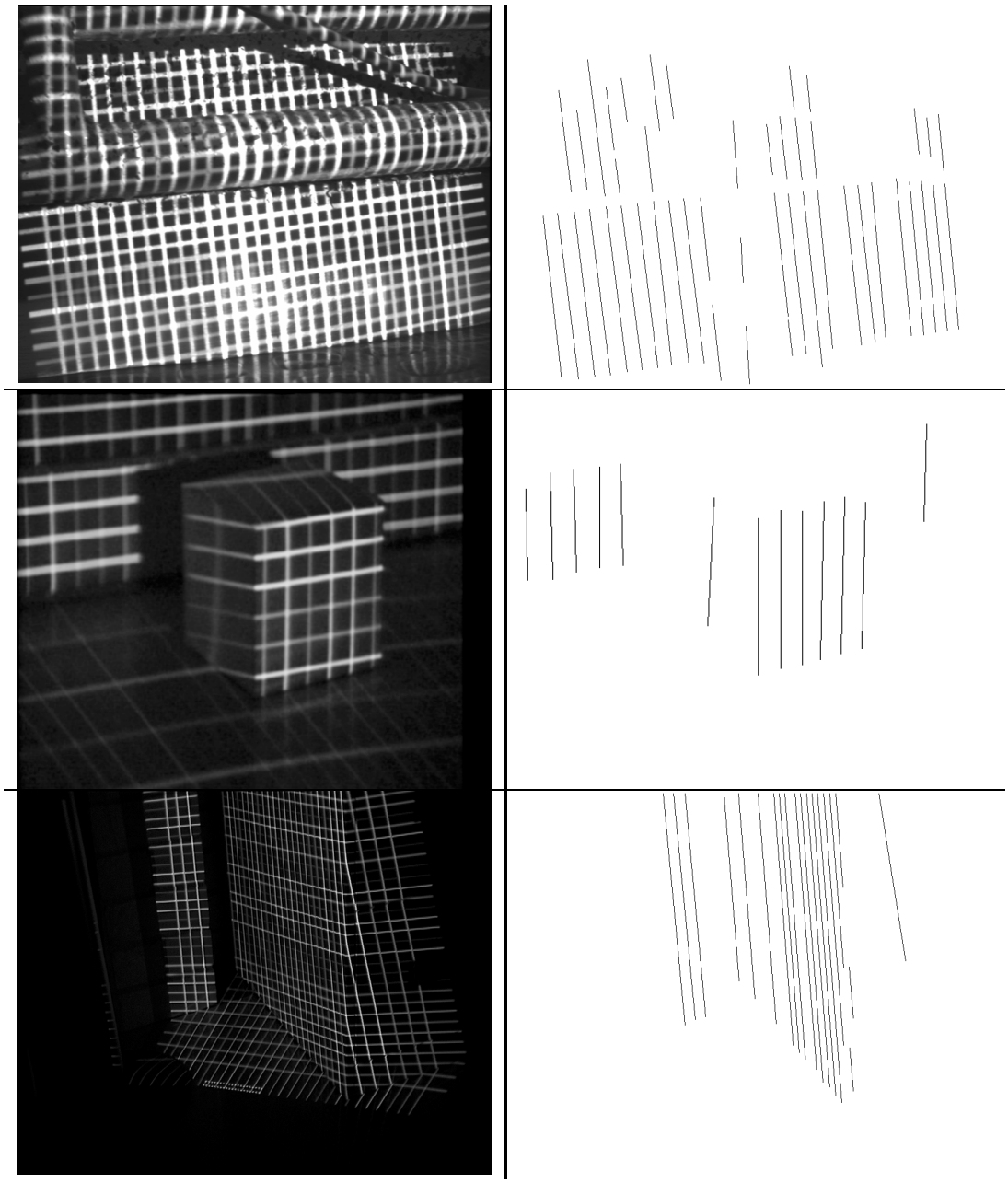


Figure 70. Détection des obstacles. A gauche : la scène. A droite : les obstacles détectés.

Si l'on observe une scène plane par morceaux, on peut admettre qu'il est relativement aisé, par la méthode proposée, de détecter efficacement les obstacles (murs, portes, etc.) De même, si l'on observe des objets cylindriques, comme sur la Figure 69. La Figure 70 présente des résultats obtenus sur quelques images. On voit que, même si certaines droites ne sont pas extraites, les obstacles, sur

lesquels, dans la grande majorité des cas, plusieurs droites sont projetées, sont tout de même détectés. Dans ce type d'environnement, pour ce type d'obstacles, la méthode est d'une grande efficacité. On détecte en outre très peu d'artefacts (en fait, les rares fois où un artefact apparaît, c'est quand les seuils pour la transformée de Hough sont mal choisis).
Qu'en est-il maintenant des objets courbes, de type sphérique par exemple ? Arriverait-on à détecter la présence d'une balle ou d'un ballon se présentant sur la trajectoire du robot ? C'est ce que nous avons voulu vérifier dans l'expérience suivante. On peut s'apercevoir que cette image possède l'inconvénient supplémentaire d'être très floue (les éléments du motif sont très mal définis, la lumière est absorbée par les objets) : il s'agit d'un cas extrême.

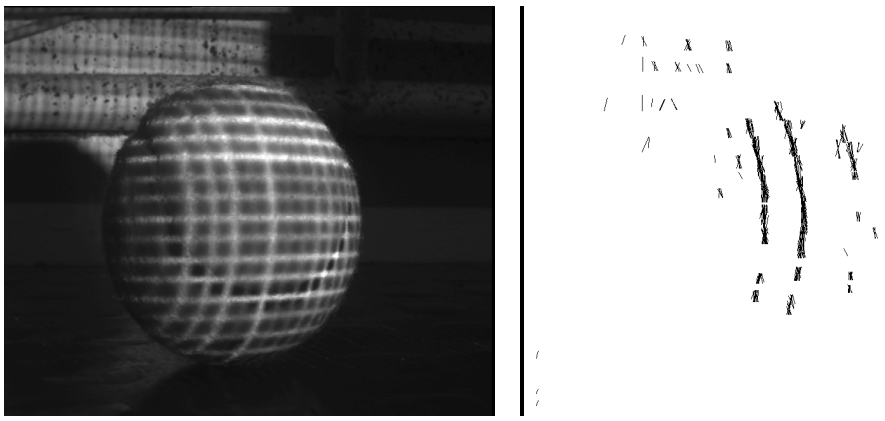


Figure 71. Détection d'un obstacle sphérique

On détecte, sur cette image, une multitude de petits segments quasi-verticaux : quelques-uns appartenant au mur du fond de la scène et, majoritairement, ceux appartenant aux principales lignes projetées sur la sphère. La segmentation est loin d'être satisfaisante (étant donnée la qualité de l'image, le contraire eût été surprenant) mais suffisante pour détecter l'obstacle.

Carte de l'espace libre

A partir des images des obstacles détectés (Figure 70, Figure 71), nous avons tenté de déduire une carte de l'espace libre par la méthode indiquée en 4.2.3. La seule extraction des droites par transformée de Hough n'est pas suffisante ici. Il faut reconstruire les points pour estimer leur élévation et donc effectuer l'ensemble des traitements (jusqu'à l'extraction des points d'intersection) et reconstruire la scène (au moins les points appartenant aux quasi-verticales extraites dans l'étape précédente). La construction de la carte de l'espace libre, si les étapes préalables ont été correctement effectuées, ne pose aucun problème particulier.

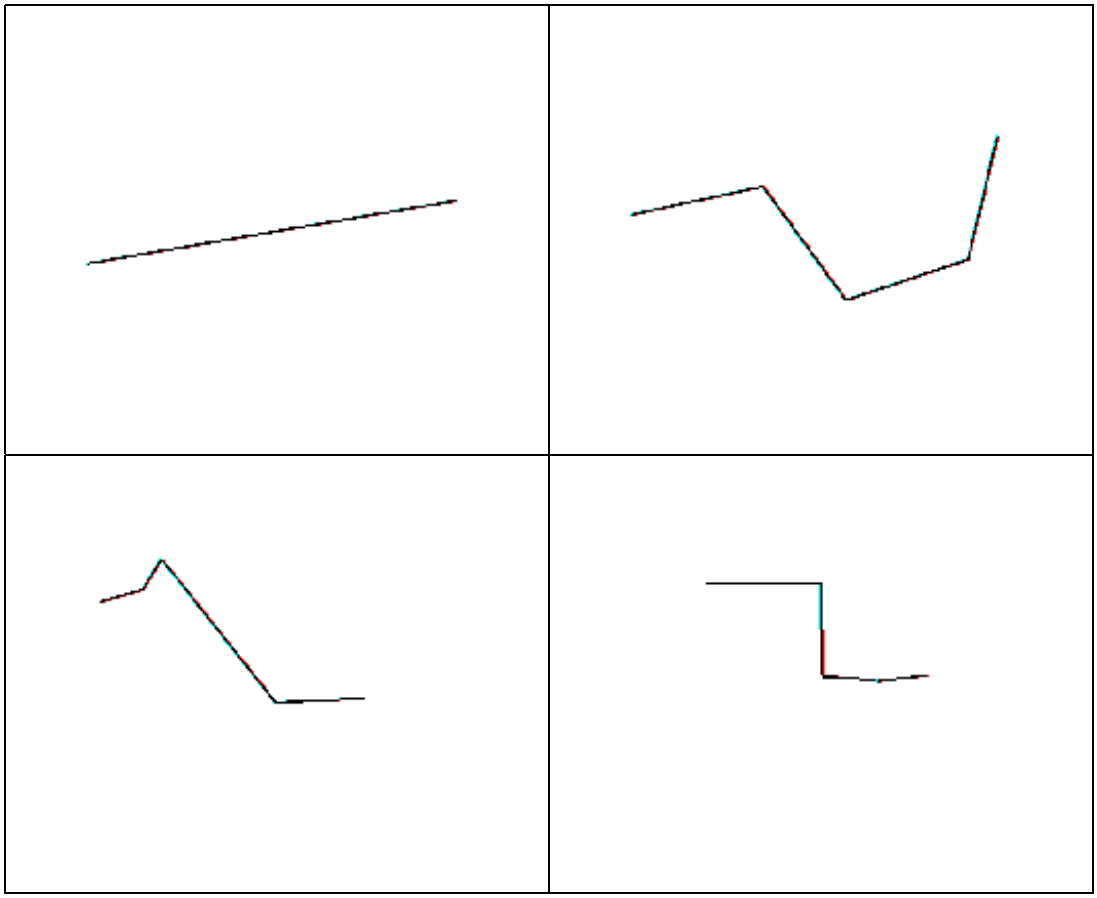


Figure 72. Cartes de l'espace libres. Exemples

Ouvrons une parenthèse d'ordre général sur la détection d'obstacles. Un artefact (la détection d'un obstacle qui n'existe pas en réalité) engendrera une perte de temps et d'énergie pour le robot, mais ne mettra pas en péril son intégrité physique, au contraire de la non-détection d'un obstacle effectif. Il vaut donc mieux toujours détecter un trop grand nombre d'obstacles plutôt qu'un nombre insuffisant. Au vu des résultats, on sait la méthode assez peu sujette aux artefacts ; mais on la sait aussi capable de rater la détection d'un obstacle en cas de forte disparité de luminance dans l'image. Pour éviter toute déconvenue, on pourra par exemple, connaissant la carte de l'espace libre, orienter lentement le robot vers une région où aucun obstacle n'a été détecté, puis analyser de nouveau la scène. La carte de l'espace libre ne donne en aucun cas l'instantané des obstacles se présentant dans le champ de vision du robot, mais une estimation de celui-ci dépendant des conditions de prise de vue, d'éclairage et des limites propres aux traitements effectués sur les images.

4.3 L'Analyse du Mouvement

L'analyse du mouvement est un domaine de recherche important en vision par ordinateur. Elle consiste à déduire, complètement ou partiellement selon l'application qu'elle sert, le mouvement des objets observés ou du capteur observant, à partir des différences ou des similarités entre des images prises successivement. Typiquement, ce sont les différences de luminosité dans l'image qui sont utilisées (*le flot optique*) : l'hypothèse de la stationnarité de la fonction d'intensité conduit à *l'équation de contrainte du mouvement apparent* qui permet de calculer en chaque point de l'image, la composante du flot optique perpendiculaire au gradient spatial de la fonction d'intensité. Des contraintes additionnelles doivent être utilisées pour calculer la composante colinéaire (ce problème est connu sous le nom de *problème de l'ouverture*), la plus connue étant la contrainte de lissage proposée par Horn et Schunck [HOR81]. D'autres approches, basées sur la mise en correspondance de primitives, sont également souvent employées : les contours [JAC80], les coins [SHA84] ou les segments de contour [ZHA88], par exemple. La stéréovision peut être considérée comme un cas particulier de cette dernière approche (capteur mobile dans un environnement statique) mais elle peut aussi être utilisée pour la mise en correspondance de primitives 3-D donnant directement, par triangulation, le mouvement 3-D des objets.
L'analyse du mouvement fournit plusieurs avantages pratiques à la navigation d'un robot mobile. Tout d'abord, elle peut servir à estimer le mouvement relatif du véhicule dans un environnement statique. On peut ainsi inférer une information grossière sur la localisation du robot, qui peut servir comme initialisation pour des algorithmes spécifiques. Elle peut également fournir de précieuses informations quant à la présence d'obstacles mobiles dans l'environnement. Si l'on parvient à extraire du champ du mouvement calculé, l'information propre au mouvement du robot, il est possible de détecter la présence, voire d'estimer le mouvement, des obstacles mobiles.
Dans la suite de ce chapitre, nous supposons que la caméra et le projecteur sont fixes l'un par rapport à l'autre et que le mouvement dominant est donné soit par le déplacement du capteur dans l'environnement, soit par le déplacement d'un objet observé par le capteur.

4.3.1 Le Problème du Mouvement

Le problème du mouvement (déjà mentionné en section 3.4.2 pour la reconstruction non-calibrée) est ce qui différencie le plus certainement la stéréovision de la vision en lumière structurée. Un mouvement du capteur, plus particulièrement du projecteur, provoque un glissement des points projetés sur les surfaces. En d'autres termes, les points 3-D illuminés *avant* le mouvement sont différents des points 3-D illuminés *après* le mouvement (Figure 73). Cette perte des points objets d'une prise de vue à l'autre fait que l'analyse du mouvement par

suivi de primitives tri-dimensionnelles n'est pas envisageable. De même, l'analyse du mouvement apparent ne peut être appliquée sans prendre en compte ce problème.

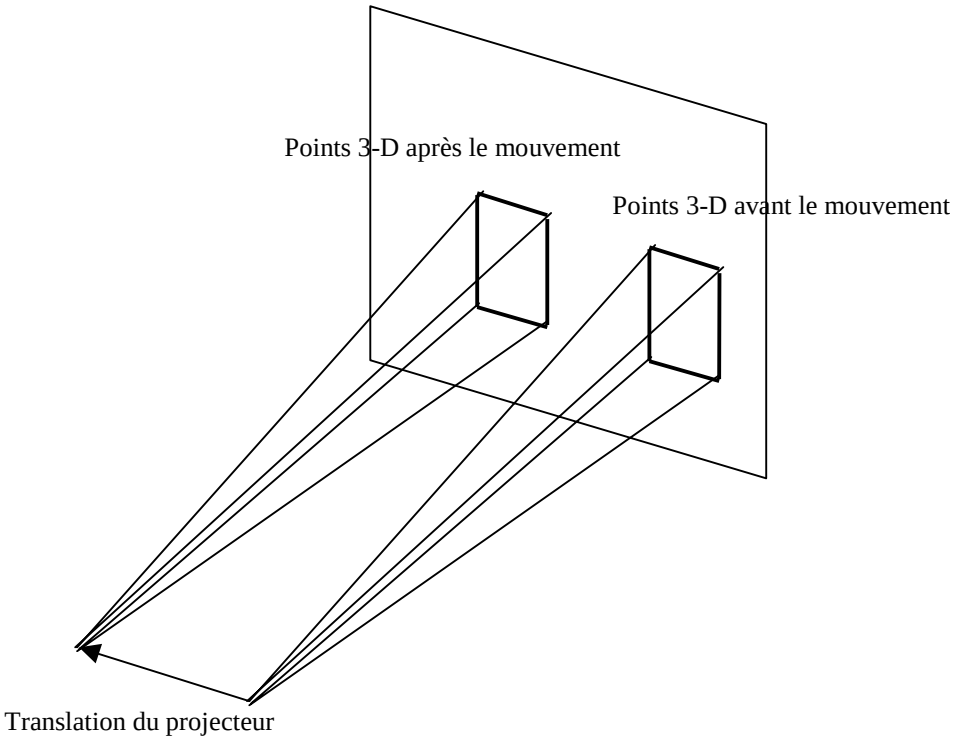


Figure 73. Illustration du glissement des points lors d'un mouvement en lumière structurée

Nous appellerons *mouvement apparent* le déplacement d'un même élément de motif (généralement, les éléments de motif considérés sont les points d'intersection) d'une image sur l'autre après un déplacement du capteur, sans perdre de vue que l'élément de motif n'illumine plus le même point 3-D.
Considèrons maintenant l'ensemble du capteur, formé de la caméra et du projecteur ; on constate que certains mouvements, quand certains types de surfaces sont illuminées, provoquent un mouvement apparent nul dans l'image : nous les appellerons *mouvements singuliers* . Quels sont-ils ? Nous en avons dénombré deux :

- Le patron de lumière est projeté sur une surface plane se déplaçant perpendiculairement à sa normale. Sur la Figure 74 par exemple, tous les points projetés sur la face avant du cube auront un mouvement apparent nul si le déplacement du capteur est perpendiculaire à la normale à cette face. Evidemment, le champ du mouvement apparent présentera de forte discontinuités aux bords des surfaces (quand la projection passera brusquement d'une surface à une autre).

- Le patron de lumière est projeté sur un objet tournant autour de son axe de révolution. L'exemple de la Figure 75 montre un cylindre en rotation autour de son axe de révolution (les effets seront équivalents si c'est la capteur qui tourne autour du cylindre) : on se rend aisément compte que le mouvement apparent du point projeté est nul.

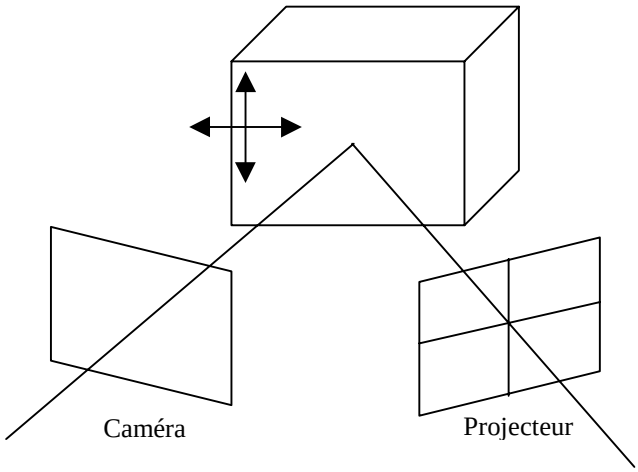


Figure 74. Mouvement singulier : déplacement perpendiculaire à la normale

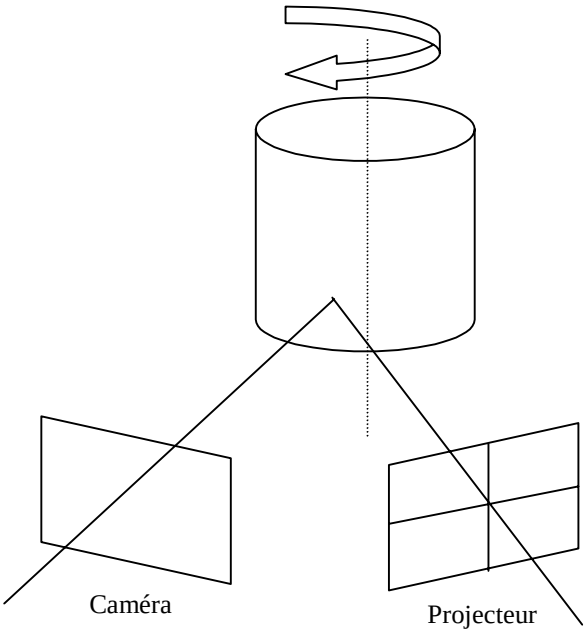


Figure 75. Mouvement singulier : rotation autour de l'axe de révolution

On retrouve des problèmes similaires en stéréovision : une sphère, dont la surface est uniforme en termes de couleur ou de texture, qui tourne sur elle-même donnera un mouvement apparent nul. Dans certains cas très précis, on obtiendra un champ de mouvement projeté sur le plan image très différent du champ de mouvement apparent (le cas, très particulier, d'une spirale enroulant un cylindre en rotation).

Une caractéristique, propre à la lumière structurée et que nous appellerons *effet de bord*, doit être mentionnée. Cet effet survient quand un point du patron est projeté sur une surface et que le déplacement du capteur fait que, brusquement, le rayon lumineux va se projeter sur une autre surface, comme on le voit sur la figure suivante :

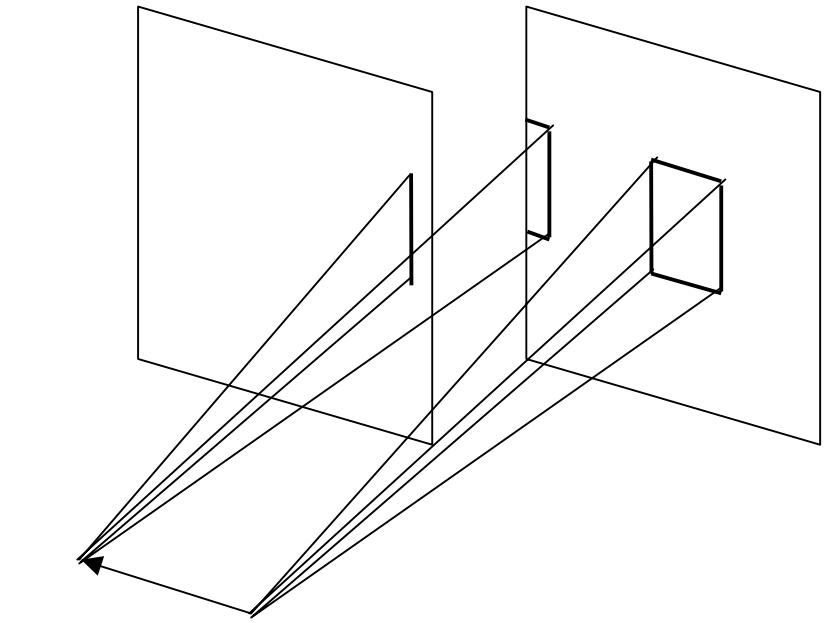


Figure 76. Effet de bord

On peut s'attendre dans ce cas à ce que le mouvement apparent des deux points ayant subis l'effet de bord soient très différents des autres points qui les avoisinent. On devrait donc observer, dans le champ du mouvement apparent, un ensemble de points ayant un comportement qui coupe nettement avec celui de leurs voisins ; des discontinuités qui pourraient éventuellement aider à la segmentation de l'image au sens du mouvement mais, en contrepartie, générer des artefacts pour la détection d'obstacles mobiles, entendus comme points de l'image dont le mouvement diffère du mouvement dominant.

Malgré ces problèmes, il est quand même possible de déduire, d'un capteur de vision en lumière structurée, des informations de mouvement du véhicule par

rapport à la scène. L'objet des sections suivantes est de fournir une étude détaillée de ce qu'il est possible d'obtenir et de comment l'obtenir.

4.3.2 La Détection du Mouvement

La détection du mouvement consiste en principe à détecter le mouvement d'un ou plusieurs objets dans l'image, ou le mouvement du capteur, par l'analyse du mouvement apparent : de manière élémentaire, il s'agit de classifier chaque point analysé en point en mouvement ou point immobile. En vision en lumière structurée, on peut d'ores et déjà s'apercevoir que les mouvements singuliers donnent un mouvement apparent nul là où pourtant on devrait effectivement détecter un mouvement. Présentons d'abord, sur quelques exemples, le champ du mouvement apparent. Les Figure 77 et Figure 78 présentent, à gauche, l'image de départ, au centre, l'image d'arrivée et à droite, le champ du mouvement apparent. Le capteur était fixe lors de la prise des images, seules les objets ont été déplacés. Les points rouges sur l'image du mouvement apparent représentent les points de départ du mouvement, les points verts représentent les points d'arrivée et les points noirs, les points où le mouvement apparent est nul.

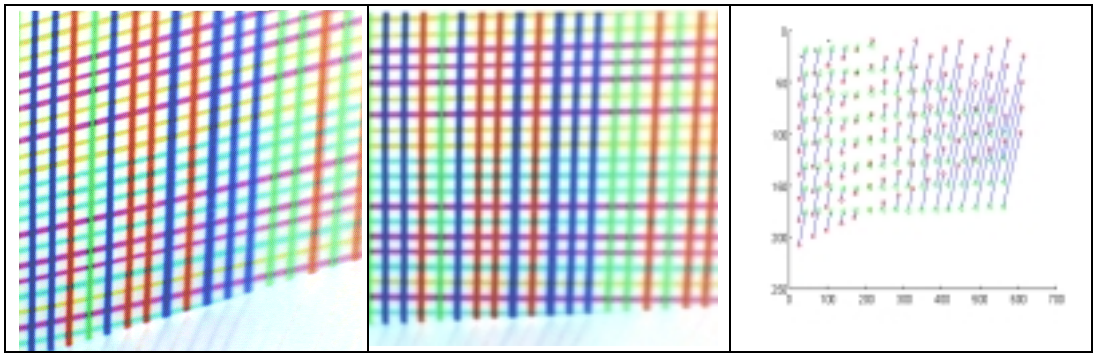


Figure 77. Champ du mouvement apparent. Exemple 1

Figure 78. Champ du mouvement apparent. Exemple 2

Les conclusions qui peuvent être tirées sont très similaires à celles de l'estimation du mouvement en vision monoculaire :

- Les mouvements sont bien détectés à l'exception des mouvements singuliers.
- On constate que la direction et la norme des vecteurs de mouvement obéissent à une loi qui permet de les paramétrer, ce qui laisse penser qu'une estimation du mouvement est envisageable, quand bien même serait-elle partielle ou lacunaire.

Par rapport à la vision monoculaire et aux problèmes classiques d'analyse du mouvement, on peut espérer que les mouvements singuliers aient une influence et une occurence plus faible que le problème de l'ouverture et les mouvements engendrant un mouvement apparent nul ; mais c'est une conjecture. On peut aussi espérer pouvoir trier, au sein du champ du mouvement apparent, le mouvement dominant des mouvements secondaires engendrés par le déplacement d'un obstacle mobile. Néanmois, du fait de la perte des points 3-D d'une image à l'autre, le champ du mouvement apparent reste difficilement interprétable. C'est pourquoi nous avons opté pour une analyse du mouvement 3-D des objets par mise en correspondance de primitives (les plans). Nous détaillerons cette méthode dans la section suivante. Avant, puisque nous avons soulevé, dans la partie précédente, le problème de l'effet de bord, illustrons cette caractéristique sur un exemple :

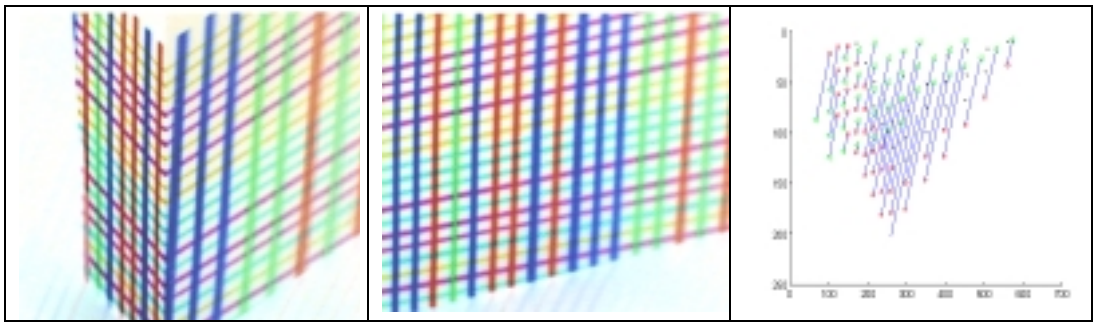


Figure 79. Champ du mouvement apparent avec effet de bord

On remarque que le sens des vecteurs de mouvement apparent s'inverse aux points où les deux plans de la première image se coupent et que leur norme est moindre. Mais il semble tout de même difficile de différencier les points étant restés sur le même plan des points ayant subi l'effet de bord et donc difficile d'utiliser cette information au sein d'un algorithme.

4.3.3 L'Estimation du Mouvement

Dans cette section, nous décrivons une méthode originale d'analyse du mouvement adaptée à la vision en lumière structurée. Celle-ci consiste à estimer le déplacement (3 rotations et 3 translations) du capteur et revient en fait à calculer la matrice essentielle reliant le même plan image (ou les deux plans du

patron de lumière, ce qui revient au même dans cette application) en deux positions différentes dans l'espace.

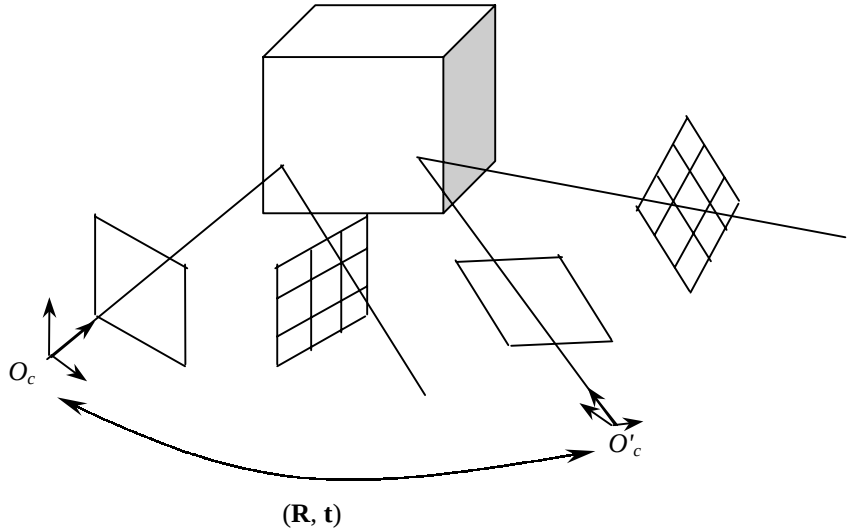


Figure 80. Principe de l'estimation du mouvement

Le problème en lumière structurée réside une nouvelle fois en la perte des points 3-D d'une image sur l'autre après un mouvement du capteur. Il est ici contourné en choisissant directement l'équation des plans comme primitive et pas les coordonnées des points appartenant aux plans.

Estimation de l'équation d'un plan

Considérons une scène plane par morceaux. Il est possible, à partir des points reconstruits dans l'espace, d'estimer les équations des plans $\mathbf{Q}^T\mathbf{M} = 0$ (avec $\mathbf{Q} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$) qui composent la scène. Soient trois points dans l'espace, de coordonnées $\mathbf{M}_i = \begin{pmatrix} x_i & y_i & z_i & 1 \end{pmatrix}^T{}_{1 \le i \le 3}$; la normale au plan formé par ces trois points est donnée par :

$$\mathbf{n} = \mathbf{M}_1\mathbf{M}_2 \times \mathbf{M}_1\mathbf{M}_3 = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{140}$$

Reste à calculer le dernier paramètre du plan :

$$d = -\mathbf{n} \cdot \mathbf{M}_1 \tag{141}$$

On peut, grâce au test de coplanarité proposé dans le chapitre précédent, regrouper les points appartenant au même plan pour estimer l'équation de celui-ci. Ce test nous permet en outre de définir le nombre de plans de l'image de départ et le nombre de plans de l'image d'arrivée : on pourra ainsi évaluer si tous les plans d'une image ont ou non un correspondant dans l'autre image.
La précédente méthode n'est valide que pour l'estimation du plan à partir de trois points. Si l'on dispose de *n* mesures bruitées, on pourra s'orienter vers la méthode des moindres carrés. Soit le système d'équations suivant :

$$\underbrace{\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{bmatrix}}_{\mathbf{X}} \cdot \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}}_{\mathbf{Q}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{0}} \tag{142}$$

On recherche $\hat{\mathbf{Q}} = \arg\min_{a,b,c,d} \|\mathbf{XQ}\|^2 = \arg\min_{a,b,c,d} (\mathbf{XQ})^T (\mathbf{XQ}) = \arg\min_{a,b,c,d} (\mathbf{Q}^T \mathbf{X}^T \mathbf{XQ})$. Il existe bien sûr la solution triviale $\hat{\mathbf{Q}} = (0 \quad 0 \quad 0)^T$. On peut néanmoins obtenir une solution non-triviale en choisissant le vecteur propre associé à la plus petite valeur propre de la matrice semi-définie positive $\mathbf{X}^T \mathbf{X}$.

Estimation du déplacement des plans

On veut à présent estimer le déplacement qui sépare deux plans mis en correspondance. Si l'on disposait du suivi des points 3-D, on pourrait évaluer ce déplacement, c'est-à-dire identifier la matrice **H** permettant d'aller d'un jeu de coordonnées à l'autre, par la mise en correspondance de quatre points. On ne dispose pas d'une telle information en vision en lumière structurée à cause du problème du mouvement évoqué plus haut. Il nous faut donc estimer la matrice **H** directement à partir des équations des plans. Soient deux jeux de coordonnées **M** et **M**' correspondant au même point en mouvement. On a :

$$\mathbf{M}' = \mathbf{H}\mathbf{M}$$

$$\text{avec} \quad \mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{143}$$

On sait que **M** appartient au plan d'origine et **M'** au plan déplacé, on en déduit donc que :

$$\mathbf{Q}^T \mathbf{M} = 0 \quad \text{et} \quad \mathbf{Q}'^T \mathbf{M}' = 0 \tag{144}$$
$$\text{et de là } \mathbf{Q}'^T \mathbf{H}\mathbf{M} = 0$$

Et donc que :

$$\mathbf{Q}'^T \mathbf{H} = \mathbf{Q}^T \quad \text{soit} \quad \mathbf{Q}'^T = \mathbf{Q}^T \mathbf{H}^{-1} \tag{145}$$

En reformulant cette dernière équation, on obtient finalement :

$$\mathbf{Q}' = \left(\mathbf{H}^{-1}\right)^T \mathbf{Q} \tag{146}$$

Où $\mathbf{H}^* = \left(\mathbf{H}^{-1}\right)^T$ est appelé le dual de **H**. C'est une matrice 4×4, de la forme :

$$\mathbf{H}^* = \begin{bmatrix} \mathbf{R} & 0 \\ -\mathbf{t}^T \mathbf{R} & 1 \end{bmatrix} \tag{147}$$

puisque **R** est une matrice de rotation, orthogonale, et donc $\mathbf{R}^{-1} = \mathbf{R}^T$.

L'équation (146) admet une infinité de solutions. Si l'on considère que le capteur évolue dans un environnement entièrement statique, on peut alors, en résolvant le système d'équations obtenu pour l'ensemble des plans observés, estimer les paramètres de $\mathbf{H}^*$ au sens des moindres carrés par exemple (chaque plan donne une équation du type de (146), quatre plans au moins sont nécessaires). On aura dans ce cas une estimation du déplacement du robot dans l'environnement. Si, de surcroit, on utilise un algorithme permettant de rejeter les données aberrantes, on pourra détecter les obstacles en mouvement et leur déplacement propre.

Pour rendre l'algorithme plus robuste, on peut utiliser les informations *a priori* que l'on possède sur la matrice de déplacement, notamment sur les rotations. On sait par exemple qu'une matrice de rotation $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}$ admet :

$$\begin{cases} \mathbf{r}_1^T \mathbf{r}_1 = 1 \\ \mathbf{r}_2^T \mathbf{r}_2 = 1 \\ \mathbf{r}_3^T \mathbf{r}_3 = 1 \\ \mathbf{r}_1^T \mathbf{r}_2 = 0 \\ \mathbf{r}_1^T \mathbf{r}_3 = 0 \\ \mathbf{r}_2^T \mathbf{r}_3 = 0 \end{cases} \tag{148}$$

où $\mathbf{r}_i^T = \begin{pmatrix} r_{i1} & r_{i2} & r_{i3} \end{pmatrix}$.
On peut grâce à elles contraindre l'estimation de **H**.

Extraction des paramètres du déplacement

On utilisera la représentation RPY (pour *Roll, Pitch, Yaw*) pour les rotations, où la rotation globale est une combinaison des trois rotations autour de chaque axe.
Soit :

$$\mathbf{R} = \mathbf{R}_{/z} \cdot \mathbf{R}_{/y} \cdot \mathbf{R}_{/x} \tag{149}$$

Avec :

$$\begin{aligned} \mathbf{R}_{/x} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \\ \mathbf{R}_{/y} &= \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \\ \mathbf{R}_{/z} &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{150}$$

De cette équation et de l'équation (143) , on peut facilement déduire que :

$$\begin{cases} \theta = \arctan\left(\dfrac{r_{32}}{r_{33}}\right) \\ \phi = -\arcsin\left(r_{31}\right) \\ \psi = \arctan\left(\dfrac{r_{21}}{r_{11}}\right) \end{cases} \tag{151}$$

Et ainsi déduire les paramètres du mouvement de l'estimation de la matrice **H** (les paramètres de translation sont directement obtenus : $t_x = H_{14}^*, t_y = H_{24}^*, t_z = H_{34}^*$).

Mise en correspondance des plans

Avant de pouvoir estimer le mouvement des plans qui composent la scène (ou le mouvement du capteur dans la scène), il faut mettre en correspondance les primitives de la première image avec celles de la seconde. L'hypothèse, couramment admise, d'un faible déplacement entre les deux prises d'image est posée ici. Les seules données à notre disposition sont les paramètres a_i, b_i, c_i et d_i de l'équation des plans i.
Dans un premier temps, deux plans Π_i et Π_j seront mis en correspondance si l'angle qui les sépare est minime, soit :

$$\Pi_i \leftrightarrow \Pi_j \Leftrightarrow \left(\hat{i}, \hat{j}\right) = \arg\min_{i,j} \Omega_{ij} \tag{152}$$

Avec :

$$\Omega_{ij} = \arccos\left(\frac{\mathbf{n}_i^T \cdot \mathbf{n}_j}{\left\|\mathbf{n}_i\right\| \cdot \left\|\mathbf{n}_j\right\|}\right) \tag{153}$$

Où $\mathbf{n}_i$ est la normale de Π_i et $\mathbf{n}_j$, la normale de Π_j. Bien sûr, le plan Π_i sera choisi parmi les plans de l'image d'origine et le plan Π_j parmi les plans de l'image obtenue après déplacement. Cette information dépendant uniquement de la normale possède l'avantage d'être invariante aux translations ; en revanche, elle ne permet pas de discriminer les plans parallèles. Pour ce faire, une seconde étape est ajoutée au processus de mise en correspondance. Pour tout plan Π_i de la première

image, on retiendra les plans Π_j et Π_k donnant les deux plus petits résultats pour (153). Si la distance $\Omega_j - \Omega_k < \varepsilon$, on considère que les plans sont parallèles et on tranche en évaluant l'expression suivante :

$$\Pi_i \leftrightarrow \Pi_j \Leftrightarrow \delta_{ij} < \delta_{ik} \quad \text{sinon } \Pi_i \leftrightarrow \Pi_k \tag{154}$$

Avec :

$$\begin{cases} \delta_{ij} = d_i - d_j \\ \delta_{ik} = d_i - d_k \end{cases} \tag{155}$$

Cette distance dépend, elle, à la fois des paramètres de translation et des paramètres de rotation (puisque la dernière ligne, celle qui correspond au paramètre *d*, de la matrice $\mathbf{H}^*$est $-\mathbf{t}^T\mathbf{R}$) et nous permettra faire un choix (celui qui minimise la distance δ) entre deux plans parallèles.

Synthèse de l'algorithme d'analyse du mouvement

Nous reprenons brièvement et de manière synthétique les principales étapes de l'algorithme d'analyse du mouvement. Ces étapes viennent après le traitement des images : on dispose donc des coordonnées pixels des points du motif.

1. Regroupement des points coplanaires à l'aide du test proposé dans le chapitre précédent, sur l'image de départ et celle d'arrivée.
2. Estimation des équations des plans des deux images.
3. Mise en correspondance des plans d'une image avec ceux de l'autre image.
4. Estimation du déplacement à partir des équations des plans mis en correspondance.
5. Extraction des paramètres du déplacement.

4.3.4 Résultats Expérimentaux

Cette section présente les résultats obtenus pour la mise en correspondance et l'estimation du déplacement. Ces résultats ont été obtenus à partir de données simulées.

Mise en correspondance des plans

Les coordonnées d'un cube d'arête 100 mm ont été définies, puis déplacées de trois translations et de trois rotations connues. On a donc au total six plans (on n'en trouve rarement plus dans une image en lumière structurée), deux à deux parallèles. Les coordonnées du cube et celles du cube déplacé ont ensuite été bruitées ; c'est à partir d'elles que les équations des plans ont été estimées. L'algorithme de mise en correspondance a été testé sur ces équations. Deux points ont été étudiés : l'efficacité et la robustesse de la mise en correspondance, l'influence du seuil ε sur la mise en correspondance.
Nous avons d'abord fixé ε à 0.05 et les paramètres de déplacement à :

t_x (mm)	t_y (mm)	t_z (mm)	θ (rad)	ϕ (rad)	ψ (rad)
-10	2	20	0.1	0.4	-0.3

Table 11. Paramètres du déplacement

Puis nous avons évalué le nombre d'échecs dans la mise en correspondance des plans en faisant croître le bruit (uniforme) de ±0 à ±10. Une série de 50 mesures a été prise pour chaque niveau de bruit. Le tableau suivant présente le nombre d'échecs maximum pour une série de mesures (pour les six plans du cube) et le nombre total d'échecs pour l'ensemble des mesures (6×50 plans).

Bruit (± mm)	0	1	2	3	4	5	6	7	8	9	10
Max	0	0	0	0	1	1	1	1	1	1	1
Echec	0	0	0	0	1	5	5	13	13	18	22

Table 12. Echec de la mise en correspondance pour un niveau de bruit donné (paramètres du déplacement constant)

Etant donné le fort rapport bruit / signal sur les dernières mesures du tableau, l'algorithme de mise en correspondance peut être considéré comme robuste. Il faut noter que l'algorithme n'empêche pas les appariements de type "un à plusieurs". Si l'on prend soin d'éviter ce type d'appariement (on peut par exemple ne garder que celui des deux plans appariés qui est le plus proche et, pour le plan rejeté, réitérer l'algorithme en éliminant le plan auquel il avait été préalablement apparié), il apparait que la plupart des échecs sont corrigés.
Toujours avec les mêmes paramètres de déplacement, nous avons cette fois fixé le bruit à ±4 et fait varier ε ; les résultats sont consignés dans le tableau suivant :

ε	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.7
Max	1	1	1	1	1	1	1	0	0	0	1
Echec	50	38	15	9	4	1	1	0	0	0	4

Table 13. Influence de ε sur la mise en correspondance

Si le seuil choisi est trop faible, tous les appariements sont "discutés" et le risque de faux appariement conséquemment accru. Inversement, s'il est trop grand, on met artificiellement en concurrence des plans d'orientation très différente et le nombre de faux appariements croît également. Le seuil optimal doit être compris entre une borne inférieure reliée au bruit de mesures sur les coordonnées 3-D des points (et donc, sur les coordonnées des images de ces points) et une borne supérieure reliée à la configuration de la scène observée (taille des objets, parallélisme, etc.). Sur les simulations que nous avons pu effectuer, un seuil ε compris entre 0.05 et 0.1 a donné de très bons résultats (mise en échec de l'appariement quasi-nulle).

Estimation du déplacement

Là encore, nous avons défini les sommets d'un cube dans l'espace et nous leur avons fait subir un déplacement, modélisé par une matrice **H**. On obtient un nouveau jeu de coordonnées qui représente les mêmes points après le mouvement. On bruite ensuite les deux jeux de coordonnées et on estime, à partir des mesures bruitées, les équations des plans qui composent le cube avant et après le mouvement. De là, on va estimer la matrice de déplacement à l'aide de la méthode exposée plus haut. Précisons que les arêtes du cube ont une longueur de 100 mm et que, pour les résultats consignés dans le tableau ci-dessous, le bruit sur les mesures est uniforme et d'amplitude ±1 mm.

	Paramètres du mouvement		
	Solution estimée	Solution simulée	Erreur abs / rel
t_x (mm)	50.2822	50	0.2822 / 0.56%
t_y (mm)	-20.2760	-20	0.2760 / 1.38 %
t_z (mm)	100.9388	100	0.9388 / 0.94%
θ (rad)	0.9712	1	0.0288 / 2.88%
ϕ (rad)	0.7343	0.7	0.0343 / 4.90%
ψ (rad)	0.1042	0.1	0.0042 / 4.2%

Table 14. Estimation des paramètres de déplacement

Comme on peut l'imaginer, la précision des résultats sur les paramètres de translation dépend du rapport bruit / translation. Pour évaluer son importance, nous avons effectué une série de 100 mesures pour chaque valeur de l'amplitude du bruit comprise entre 0 et 10. Nous nous sommes limités pour cela à une translation le long d'un seul axe, en l'occurence l'axe *z*, fixée à 100 mm et aucune rotation (le bruit atteignant donc jusqu'à 10% de la valeur de la translation).

Bruit (±)	1	2	4	6	8	10
M.A. (mm)	0.7250	1.2991	3.8966	5.1936	7.3580	10.3035
Ecart-type	0.5751	0.9809	2.5290	3.4801	4.9356	7.4546

Table 15. Erreur et écart-type pour une translation de 100mm (M.A. = erreur moyenne absolue)

Puisque la translation est de 100 mm, l'erreur moyenne absolue et l'erreur moyenne relative prennent exactement les mêmes valeurs, nous n'avons donc pas consigné cette dernière dans le tableau. Les résultats restent tout à fait acceptables pour une application robotique qui ne requiert pas des mesures d'une précision absolue. La précision obtenue au chapitre précédent pour la reconstruction tri-dimensionnelle de la scène garantit un bruit relativement faible et donc une mesure du déplacement relativement précise.
Nous avons également voulu évaluer l'influence du bruit sur l'estimation de la rotation. Cette fois, le déplacement se réduit à une seule rotation de $\pi/4$ autour de l'axe *z* ; les valeurs de bruit sont les mêmes que pour le tableau précédent.

Bruit (±)	1	2	4	6	8	10
M.A. (mm)	0.0053	0.0084	0.0216	0.0337	0.0396	0.0471
Ecart-type	0.0037	0.0063	0.0120	0.0221	0.0323	0.0323
M.R. (%)	0.6748	1.0695	2.7502	4.2908	5.0420	5.9970

Table 16. Erreur et écart-type pour une rotation de $\pi/4$ (M.A. = erreur moyenne absolue, M.R. = erreur moyenne relative)

On remarque ici que la rotation est encore moins sensible au bruit que la translation ; la précision est donc tout à fait satisfaisante.
Les mesures d'erreur de ces deux tableaux ont été reportées sur le graphique suivant :

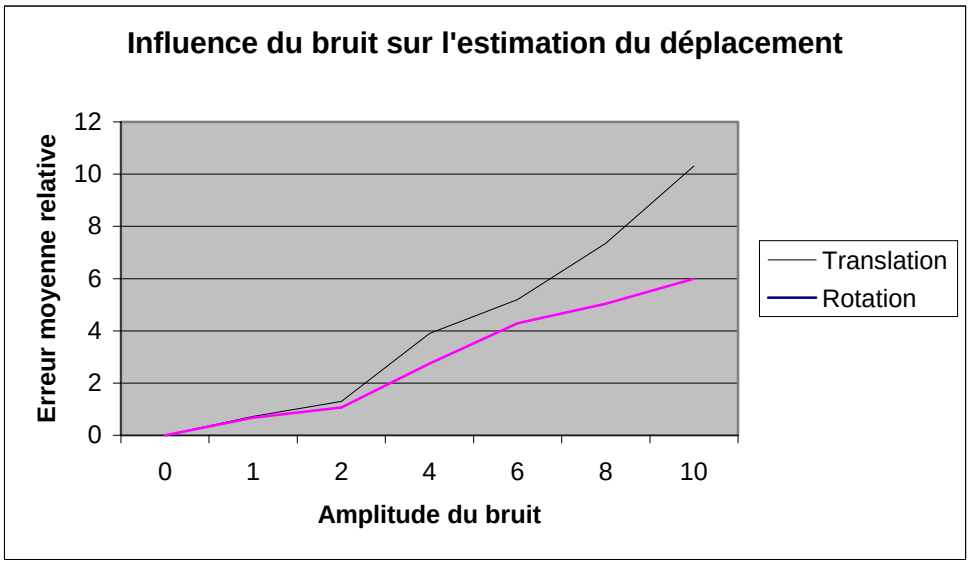


Figure 81. Influence du bruit sur les paramètres de déplacement

Heuristique sur les degrés de liberté du robot

Il est courant d'avoir connaissance des degrés de liberté du véhicule portant le capteur et ainsi, d'inférer une connaissance *a priori* sur le déplacement possible. Si l'on réduit, par exemple, les déplacements du robot à une seule translation suivant l'axe *z (*déplacement vers l'avant ou vers l'arrière) et à une seule rotation autour de l'axe *x* (pivot sur lui-même), on simplifie considérablement la matrice **H**. Il est suffisant, dans cet exemple précis, d'observer un seul plan dans l'environnement pour en déduire le mouvement. Le tableau ci-dessous présente un exemple réalisé dans ces conditions ; le bruit est de ±1 mm sur les points 3-D.

	Paramètres du mouvement		
	Solution estimée	Solution simulée	Erreur abs / rel
t_z (mm)	99.2529	100	0.7471 / 0.75%
θ (rad)	-0.3970	-0.4	-0.0030 / 0.76%

Table 17. Déplacement contraint et un seul plan

Ces bons résultats ne doivent pas nous faire perdre de vue deux inconvénients majeurs : la très forte sensibilité au bruit des mesures (pour un bruit de ±5 mm, on obtient parfois des estimations très éloignées des valeurs du déplacement réels – parfois de l'ordre de 50% d'erreur !) et l'impossibilité d'estimer les paramètres pour certains déplacements du plan si celui-ci est d'orientation particulière (un déplacement le long de la normale au plan, par exemple, ou perpendiculairement à celle-ci). En pratique, on considérera toujours un nombre maximal de plans pour éviter ces inconvénients.

4.4 Conclusions

Ce chapitre avait pour objectif d'introduire les algorithmes basiques de la navigation en évaluant leur compatibilité avec la vision en lumière structurée. Dans cette conclusion, nous essaierons d'établir les réels avantages de cette dernière par rapport à la stéréovision, les points où elle peut rivaliser avec elle et ceux où elle reste encore en deçà. Commençons par la partie traitant de la détection des obstacles. En s'affranchissant de la photométrie de la scène (les niveaux de gris) et en analysant uniquement l'information géométrique particulière qui nait de la projection d'une grille sur les objets, nous avons montré que la lumière structurée permettait l'élaboration d'un algorithme simple et rapide de détection des obstacles, fiable et capable de prendre en compte une grande diversité d'obstacles de formes différentes. En l'état, il est difficile de reconnaître les objets perçus : parce que l'on perd justement les niveaux de gris, les couleurs et textures intrinsèques de l'objet, et, dans une moindre mesure, parce que, sans traitement supplémentaire, on ne dispose pas des arêtes réelles des objets, des *contours* au sens de la vision classique. Nous montrons que la détection des obstacles peut en outre être complétée par un algorithme de construction d'une carte de l'espace libre. A partir de deux critères, l'un basé sur l'élévation et l'autre sur la conservation du point d'impact le plus proche du véhicule pour chaque droite projetée, on a vu qu'il était possible d'avoir une information sur la localisation des obstacles dans le champ de vision. Cette information, très dépendante du processus de traitement des images, peut s'avérer imprécise et doit donc être manipulée avec précaution.

L'analyse du mouvement sur des images en lumière structurée nous a conduit à une première étude sur le mouvement apparent et la détection du mouvement. La perte des points 3-D d'une image sur l'autre et quelques caractéristiques propres au mouvement d'un capteur en lumière structurée sont expliqués. Nous montrons qu'il est quand même possible de détecter le mouvement à partir du calcul (très simple puisque basé sur la seule mise en correspondance des éléments du motif) du champ de mouvement apparent. La seconde étude consiste à estimer le déplacement du robot dans un environnement statique à partir de la mise en correspondance de primitives. La perte des points 3-D nous oblige à choisir une primitive autre que le point : nous travaillons sur le plan et l'équation du plan. Nous décrivons comment estimer les équations des plans, comment mettre en correspondance les plans d'une image à l'autre et comment en tirer les paramètres du mouvement du robot. Les résultats, obtenus sur des simulations, sont satisfaisants. La méthode reste perfectible en plusieurs points : on peut utiliser un autre algorithme que les moindres carrés (une méthode itérative, par exemple) garantissant une meilleure précision et une plus grande stabilité numérique pour la résolution du système d'équations ; on peut imaginer aussi une méthode permettant de rejeter les mouvements aberrants que l'on pourrait interpréter comme des obstacles mobiles.

Conclusions et Perspectives

Pour les robots, comme pour les véhicules maritimes ou aériens, la navigation est la technique du déplacement, de la détermination de la position et de la trajectoire. Le concept est très général et laisse la place à de nombreux classements et subdivisions ; en robotique, on classe souvent la navigation selon le degré d'autonomie qu'elle confère au véhicule : *pilotée* ou téléguidée quand c'est un opérateur humain qui la dirige complètement à partir des informations perçues directement ou au travers d'un ou plusieurs capteurs, *orientée* quand les buts sont définis par l'opérateur mais que les moyens d'y parvenir sont calculés par la machine, *autonome* quand le robot dirige seul les opérations. Il semble évident que c'est pour cette dernière classe qu'il a fallu conduire nos recherches.

Le domaine est vaste, c'est pourquoi nous nous en sommes tenus aux fonctionnalités de base, aux outils fondamentaux permettant d'ouvrir le champ d'application de la vision en lumière structurée et codée à la navigation d'un véhicule intelligent. En ce sens, il nous a semblé important d'aborder les problèmes de décodage, d'auto-calibration, de détection d'obstacles et d'analyse du mouvement, principalement. Le décodage permet la mise en correspondance de l'image capturée avec l'image projetée, indispensable à la triangulation des lignes de vue. L'auto-calibration permet la mesure tri-dimensionnelle de l'environnement sans intervention de l'opérateur et d'adapter les paramètres de prise de vue aux conditions d'illumination, aux stratégies d'observation et d'exploration. La détection d'obstacles est le premier mécanisme qui vient à l'esprit quand on traite de navigation ; elle est incontournable. L'analyse du mouvement, enfin, permet de positionner relativement le véhicule, de détecter les obstacles mobiles, et, comme nous l'avons évoqué dans le chapitre qui lui est consacré, produit une information utile pour la construction itérative de cartes de l'environnement.

Si l'on met de côté le décodage, trop spécifique pour en tirer une conclusion de cet ordre, on peut dire que la détection d'obstacles fut antérieurement implémentée mais à partir de lumières structurées de type mono-dimensionnelles ; qu'on trouve, dans les travaux qui nous ont précédés, des techniques de calibration adaptées à la lumière structurée, mais que l'auto-calibration n'y est que rarement, voire pas du tout, discutée ; enfin, c'est, à notre connaissance, la première fois que l'analyse du mouvement est abordée pour un capteur de vision en lumière structurée. Les algorithmes proposés ont, pour la plupart, été implémentés et ont montrés leur efficience. On peut avancer que la vision en lumière structurée et codée peut être choisie comme organe perceptif pour la navigation d'un robot mobile, que les présupposés y sont réunis et les outils fondamentaux applicables. La projection d'un motif structurant permet de combiner les avantages des lumières structurées mono-dimensionnelles (simplicité des traitements, mise en correspondance) et de la stéréovision (auto-calibration, analyse du mouvement).

Bien sûr, les réponses apportées dans ce mémoire n'ont pas la prétention d'être optimales ; elles doivent plutôt être vues comme une première approche,

perfectible aussi bien dans sa conception que dans ses méthodes de résolution. Nous essayons, dans la suite de cette conclusion, de dégager les problèmes qu'il reste à résoudre et les points pour lesquels nos travaux sont, ou vont être, approfondis.

On remarque que la méthode de traitement des images décrite dans le deuxième chapitre offre de très bons résultats sur des images simples, mais que sa qualité décroit avec la complexité de celles-ci. Nous envisageons donc de diviser l'image originale en imagettes représentant une seule surface, d'effectuer les traitements sur ces imagettes et de les juxtaposer pour obtenir l'image segmentée. La difficulté provient du processus de division / concaténation : comment diviser convenablement l'image originale pour que les imagettes soient sémantiquement cohérentes ? En profitant de la texture artificielle produite par la projection de la grille sur la surface des objets, nous avons pensé segmenter l'image originale en régions homogènes au sens de la texture (puisqu'on aura une texture de taille et d'orientation différente pour chaque surface). L'idée est d'utiliser ces régions comme masques sur l'image d'origine et d'effectuer les traitements proposés au deuxième chapitre à l'intérieur de ces régions. En mettant bout à bout chaque région, on espère obtenir la segmentation de l'ensemble de l'image. On espère aussi, par ce biais, pouvoir obtenir une estimation de la localisation des contours réels des objets (arêtes, séparations de surfaces).

Pour la reconstruction sans calibration, la voie dans laquelle nous nous sommes engagés semble, en théorie, assez complètement explorée. Nous avons déterminé les techniques de reconstruction projective permises par la vision en lumière structurée et celles-ci ont déjà été étudiées par ailleurs. On pourrait compléter la méthode en imaginant d'autres contraintes Euclidiennes ou encore étendre ces contraintes au modèle projectif (la complexité accrue des contraintes, pour cette dernière extension, ne garantissant pas que la précision obtenue soit d'autant meilleure). Ce qui nous parait le plus important pour l'heure, c'est d'automatiser la génération des contraintes en sortie de traitement des images et d'insérer, dans le processus de reconstruction, les tests de validité du modèle affine, de colinéarité et de coplanarité.

La méthode de détection d'obstacles que nous avons proposée a l'avantage d'être simple, rapide et générique. On a vu qu'elle permettait de détecter la quasi-totalité des obstacles de l'environnement, même les objets courbes. Cependant, ces derniers sont détectés de manière plus grossière et donnent une carte de l'espace libre plus approximative. Améliorer les résultats sur ce point revient à améliorer le processus de traitement des images : nous avons déjà évoqué plus haut nos projets en ce domaine. On pourrait également généraliser la transformée de Hough à des primitives autre que des droites, au risque de buter sur une charge de calcul rédhibitoire.

Nous avons montré qu'il était possible d'analyser le mouvement malgré les problèmes engendrés par la projection du motif. Il semble qu'on ne puisse pas aller beaucoup plus loin en ce qui concerne le calcul du champ du mouvement apparent et la détection du mouvement. Pour l'estimation du déplacement du robot, en revanche, il est possible, à notre sens, d'améliorer l'algorithme à la fois dans sa précision et dans ses limites d'application. On pourrait utiliser d'autres méthodes de résolution numérique pour rendre l'estimation plus robuste au bruit et plus précise ; les méthodes existent, il suffit de les appliquer. On sait qu'il existe une multiplicité de solutions à l'estimation du mouvement, *a fortiori*, si l'on observe une scène mono-planaire, bi-planaire ou tri-planaire. Une voie à explorer pourrait être la régularisation de ce problème mal-posé, c'est-à-dire, la réduction de l'espace des solutions à un singleton, en adjoignant des contraintes au système d'équations. Pour reprendre le vocabulaire des problèmes inverses, la question est de savoir quelle fonctionnelle régularisante choisir, quelle information *a priori* ajouter pour faire le pendant à la fidélité aux données représentées par l'équation aux moindres carrés. Ce problème reste ouvert et n'est pas forcément solvable.

D'une manière générale, nous pouvons corroborer les avantages et les limites déjà connus de la vision en lumière structurée. A savoir que la projection d'une lumière indépendante de la luminosité ambiante impose des contraintes à cette dernière. Si l'écart entre l'intensité de la lumière structurée et l'intensité de la lumière ambiante est trop faible ou si l'intensité de la première est plus faible que celle de la seconde, les images capturées seront difficilement, ou ne seront pas du tout, traitables. En ce sens, les sources laser ou les vidéo-projecteurs ont un avantage sur les projecteurs de lumière traditionnels (de type diapositive), mais ne résolvent pas entièrement ce problème. C'est une des raisons pour lesquelles la lumière structurée est fréquemment utilisée dans des milieux contrôlés, en métrologie ou en anthropométrie, et assez peu dans des milieux "ouverts". On peut toutefois relativiser ce problème, souvent mis en exergue, en posant cette question : n'a-t-on pas un problème similaire, que l'on pourrait qualifier de dual, en stéréovision quand il s'agit d'observer une scène peu ou pas éclairée ?
Un point important, quand on projette une lumière structurée et codée, est d'adapter la forme du motif à la précision souhaitée (nombre et diamètre des points ou nombre et épaisseur des lignes) et son codage au type de surfaces à mesurer (albedo, absorption, couleur, discontinuités, etc.) Ces inconvénients ne doivent pas nous faire oublier les indéniables avantages de la lumière structurée ; avant tout, sa capacité à résoudre le problème de la mise en correspondance (qui est en fait converti, dans le cas de la lumière structurée et codée, en problème de décodage). Mieux, elle permet de lever les ambiguïtés et de se soustraire en grande partie aux problèmes des occlusions. Au surplus, elle permet la mesure de surfaces non texturées ou donnant des images comportant peu de points d'intérêts (au sens de la vision classique) : coins, arêtes, intersections, etc.

Références Bibliographiques

[ALO90] J.Y. Aloimonos, D.P. Tsakiris, "Tracking in a complex visual environment", *Proceedings of the European Conference on Computer Vision*, pp. 249-258, 1990.

[ALT81] M.D. Altschuler, B.R. Altschuler, J. Taboada, "Laser electro-optic system for rapid three-dimensional (3-D) topographic mapping of surfaces", *Optical Engineering*, vol.20, n°6, pp. 953-961, 1981.

[ARN72] D. Arnush, "Underwater light beam propagation in the small angle scattering approximation", *Journal of Optical Society of America*, vol.62, pp. 1109-1111, 1972.

[AYA89] N. Ayache, "Vision stéréoscopique et perception multisensorielle – Application à la robotique mobile", Collection Science Informatique, InterEditions, 1989.

[BAT98] J. Batlle, E. Mouaddib, J. Salvi, "Recent progress in coded structured light as a technique to solve the correspondence problem: A survey", *Pattern Recognition*, vol.31, n°7, pp. 963-982, 1998.

[BEN96] R. Benosman, T. Manière, J. Devars, "Multidirectional stereovision sensor, calibration and scenes reconstruction", *Proceedings of the 13th International Conference on Pattern Recognition*, pp. 161-165, 1996.

[BEN97] R. Benosman, T. Manière, J. Devars, "Panoramic sensor calibration using computational projective geometry", *IEEE Proceedings of the International Conference on Robotics and Automation*,), pp. 1353-1358, Albuquerque (New Mexico, USA), 1997.

[BOL81] R.C. Bolles, J.H. Kremers, R.A. Cain, "A simple sensor to gather three-dimensional data", *Rapport Technique*, n°249, SRI International, Juillet 1981.

[BOR91] F. Bortolozzi, "Vision trinoculaire : une solution géométrique", *Thèse de Doctorat*, Université de Technologie de Compiègne, 1991.

[BOU93] B. Boufama, R. Mohr, F. Veillon, "Euclidean constraints for uncalibrated reconstruction", *Proceedings of the 4th International Conference on Computer Vision* (ICCV'93), Berlin (Allemagne), pp. 466-470, Mai 1993.

[BOY87] K.L. Boyer, A.C. Kak, "Color-encoded structured light for rapid active ranging", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.9, n°1, pp.14-28, 1987.

[BUF93] M. Buffa, O.D. Faugeras, Z. Zhang, "A Stereovision-based navigation system for a mobile robot", *Rapport de Recherche*, n°1895, INRIA, 1993.

[BUN01] R. Bunschoten, B. Kröse, "Range estimation from a pair of omnidirectional images", *IEEE Proceedings of the International Conference on Robotics and Automation*, pp. 1174-1179, Seoul (Corée), 2001.

[CAR85] B.Carrihill, R. Hummel, "Experiments with the intensity ratio depth sensor", *Computer Vision, Graphics, Image Processing*, n°32, pp. 337-358, 1985.

[CAS98] D. Caspi, N. Kiryati, J. Shamir, "Range imaging with adaptive color structured light", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.20, n°8, pp.470-480, Mai 1998.

[CAU99] C. Cauchois, E. Brassart, C. Drocourt, P. Vasseur, "Calibration of the omnidirectional vision sensor: SYCLOP", *IEEE Proceedings of the International Conference on Robotics and Automation*, pp. 1287-1292, Detroit (Michigan, USA), 1999.

[CAU00] C. Cauchois, E. Brassart, L. Delahoche, T. Delhommelle, "Reconstruction with the calibrated SYCLOP sensor", *IEEE Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 1493-1498, Takamatsu (Japon), 2000.

[CHA95] G. Chazan, N. Kiryati, "Pyramidal intensity ratio depth sensor", *Technical Report*, n°121, Center of Communication and Information Technologies, Department of Electrical Engineering, Haifa (Israel), Octobre 1995.

[CHA97A] M.J. Chantier, J. Clark, M. Umasuthan, "Calibration and operation of an underwater laser triangulation sensor: the varying baseline problem", *Optical Engineering*, vol.36, n°9, pp. 2604-2611, Septembre 1997.

[CHA97B] F. Chavand, E. Colle, Y. Chekhar, E.C. N'zi, "3-D measurements using a video camera and a range finder", *IEEE Transactions on Instrumentation and Measurement*, vol.46, n°6, Décembre 1997.

[CRO90] J.L. Crowley, P. Stelmaszyk, "Measurement and integration of 3D structures by tracking edge lines", *Proceedings of the European Conference on Computer Vision*, pp. 269-280, 1990.

[DUB90] B. Dubuisson, *Diagnostic et reconnaissance des formes*, Editions Hermes, 1990.

[EFE96] W. Efenberger, V. Graefe, "Distance-invariant object recognition in natural scenes", *Proceedings of the International Conference on Intelligent Robots and Systems* (IROS'96), pp. 1433-1439, 1996.

[FAU86] O.D. Faugeras, G. Toscani, “The calibration problem for stereo”, *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 15-20, 1986.

[FAU88] J.M. Faure, Y. Yang, "Calibrating a 3D visual sensor: a factory-oriented approach", *Proceedings of the International Conference on Robot Vision and Sensory Controls*, pp. 21-29, Février 1988.

[FAU92A] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?", *Proceedings of the 2nd European Conference on Computer Vision*, Santa Maria Ligure (Italie), pp. 563-578, 1992.

[FAU92B] O. Faugeras, Q.-T. Luong, S.J. Maybank, "Camera self-calibration: theory and experiments", *Proceedings of the European Conference on Computer Vision*, 1992.

[FAU93] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*, MIT Press, 1993.

[FOX88] J.S. Fox, "Structured light imaging in turbid water", *Underwater Imaging, Proceedins of the SPIE International Society for Optical Engineering*, vol.980, pp. 66-71, 1988.

[FOR00] J. Forest, J. Salvi, J. Batlle, "Image ranging system for underwater applications", *IFAC Conference on Manoeuvring and Control of Marine Craft* (MCMC' 2000), Aalborg (Danemark), Août 2000.

[GAR92] P. Garnesson, G. Giraudon, "L'approximation polygonale, bilans et perspectives", *Rapport de Recherche*, n°1621, INRIA, 1992.

[GAR93] J. Garding, "Direct estimation of shape from texture", *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol.15, pp. 1202-1208, 1993.

[GAR95] D.F. Garcia, M. Garcia, F. Obeso, V. Fernandez, "Linear-vision techniques applied to inspection production in iron and steel industry", *Revue d'Automatique et de Productique Appliquées*, vol.8, n°2-3, pp. 147-156, 1995.

[GIB50] J. Gibson, *The perception of the visual world*, Houghton Mifflin, Boston, 1950.

[GOL69] M.J.E. Golay, "Hexagonal parallel pattern transformations", *IEEE Transactions on Computers*, vol.18, n° 8, pp. 733-740, 1969.

[GRA53] H.G. Grassman, "Zur Theorie der Farbenmischung", *Poggendorf Annalen der Physik und Chemie*, 89, pp. 69-84, 1853. Traduction Anglaise, "On the theory of compound colors", *Philosophical Magazine S.4*, vol.7, n°45, pp. 254-264, 1854.

[GRI92] P.M. Griffin, L.S. Narasimhan, S.R. Yee, "Generation of uniquely encoded light patterns for range data acquisition", *Pattern Recognition*, vol.25, n°6, pp. 609-616, 1992.

[GRO93] P. Gros, "Outils géométriques pour la modélisation et la reconnaissance d'objets polyédriques", *Thèse de Doctorat*, Institut National Polytechnique de Grenoble, Juillet 1993.

[GUI31] J. Guild, "The colorimetric properties of the spectrum", *Philosophical Transactions of the Royal Society of London*, A230, pp. 149-187, 1931.

[HAL82] E.L. Hall, J.B.K. Tio, A. McPherson, F.A. Sadjadi, "Measuring Curved Surfaces for Robot Vision", *Computer Journal*, vol. December, pp. 42-54, 1982.

[HAR92] R. Hartley, R. Gupta, T. Chang, "Stereo from uncalibrated cameras", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Urbana-Champaign, Illinois (USA), pp. 761-764, 1992.

[HAR94] R. Hartley, "Euclidean reconstruction from uncalibrated views", *Lecture Notes in Computer Science*, vol. 825, pp. 237-256, 1994.

[HAR95] R. Hartley, "In defence of the 8-point algorithm", *Proceedings if the 5th International Conference on Computer Vision*, pp. 1064-1070, 1995.

[HAR97] R.I. Hartley, "Kruppa's Equations Derived from the Fundamental Matrix", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, n°2, pp. 133-135, 1997.

[HAS96] H. Hashimoto et al., "Detection of obstacle from monocular vision based on histogram matching method", *Proceedings of the IECON*, vol.2, pp. 1047-1051, 1196.

[HEL60] H. von Helmholtz, "Handbuch der Physiologischen Optik", Band II, Sektion 20, 1860.

[HEN97] L. Henriksen, E. Krotkov, "Natural terrain hazard detection with a laser rangefinder", *IEEE Proceedings of the International Conference on Robotics and Automation*, Albuquerque, New Mexico (USA), Avril 1997.

[HEY96] A. Heyden, K. Aström, "Euclidean reconstruction from constant intrinsic parameters", *Proceedings of the International Conference on Pattern Recognition*, 1996.

[HOR81] B.K.P. Horn, B.G. Schunck, "Determining optical flow", *Artificial Intelligence*, pp. 185-203, 1981.

[HOR89] B.K.P. Horn, M.J. Brooks, *Shape from shading*, MIT Press, Cambridge, 1989.

[HOR99] E. Horn, N. Kiryati, "Towards optimal structured light patterns", *Image and Vision Computing*, vol.17, pp. 87-98, 1999.

[HOT96] M. Hötter, R. Mester, F. Müller, "Detection and description of moving objects by stochastic modelling and analysis of complex scenes", *Signal Processing: Image Communication*, vol.8, pp. 281-293, 1996.

[HOU62] P.V.C. Hough, "Methods and means for recognizing complex patterns", *U.S. Patent*, 3 069 654, 1962.

[HU86] G. Hu, A.K. Jain, G. Stockman, "Shape form light stripe texture", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 412-414, 1986.

[HU89] G. Hu, G. Stockman, “3-D surfaces solution using structured light and constraint propagation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 349-402, vol.11, n°4, 1989.

[HUN93] D.C.D. Hung, "3D scene modelling by sinusoid encoded illumination", *Image and Vision Computing*, vol.11, n°5, pp. 251-256, 1993.

[HUY99] D.Q. Huynh, R.A. Owens, P.E. Hartman, "Calibrating a structured light stripe system: a novel approach", *Internationl Journal of Computer Vision*, vol.33, n°1, pp. 73-86, 1999.

[ISH92] H. Ishiguro, M. Yamamoto, S. Tsuji, "Omni-directional stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence* , pp. 257-262, vol.14, n°2, 1992.

[JAC80] C.J. Jacobus, R.T. Chien, J.M. Selander, "Motion detection and analysis by matching graphs of intermediate level primitives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.2, n°6, pp. 495-510, 1980.

[KOE80] J.J. Koenderink, A.J. van Doorn, "Photometric invariants related to solid shape", *Optica Acta*, vol.27, n° 7, pp. 981-996, 1980.

[KOE89] J.J. Koenderink, A.J. van Doorn, "Affine Structure from Motion", *Technical Report*, Utrecht University, Utrecht (Netherlands), 1989.

[KRO86] E. Krotkov, J.P. Martin, "Range from focus", *IEEE International Conference on Robotics and Automation*, pp.1093-1098, 1986.

[KRO91] E. Krotkov, "Laser rangefinder calibration for a walking robot", *IEEE Proceedings of the International Conference on Robotics and Automation*, Avril 1991.

[LEE85] C.H. Lee, A. Rosenfeld, A., "Improved methods of estimating shape from shading using the light source coordinate system", *Artificial Intelligence*, vol.26, n°2, pp. 125-143, 1985.

[LEE90] C.H. Lee, T. Huang, "Finding point correspondences and determining motion of a rigid object from two weak perspective views", *Computer Vision, Graphics and Image Processing*, n°52, pp. 309-327, 1990.

[LEM84] J. Le Moigne, A.M. Waxman, "Projected light patterns for short range navigation of autonomous robots*", Proceedings of the International Conference on Pattern Recognition*, vol.1, pp. 203-206, 1984.

[LON81] H.C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections”, *Nature*, vol.293, pp. 133-135, Septembre 1981.

[LUO88] R.C. Luo, R.E. Mullen Jr, D.E. Wessel, "An adaptive robotic tracking system using optical flow", *IEEE Proceedings of the International Conference on Robotics and Automation*, pp. 568-573, 1988.

[LUO90] Q.-T. Luong, "La couleur en vision par ordinateur : 1. Une revue", *Rapport de Recherche*, n°1251, INRIA, Juin 1990.

[LUO93] Q.-T. Luong, R. Deriche, O. Faugeras, T. Papadopoulo, "On determining the fundamental matrix: analysis of different methods and experimental results", *Rapport de recherche*, n°1894, INRIA, 1993.

[LUO94] Q.-T. Luong, T. Viéville, "Canonical representations for the geometries of multiple projective views", *Proceedings of the 3rd European Conference on Computer Vision*, Stockholm (Sweden), Mai 1994.

[MAR63] D.W. Marquardt, "An algorithm for the estimation of nonlinear parameters", *Society for Industrial and Applied Mathematics Journal*, n°11, pp. 431-441, 1963.

[MAR93] M. Maruyama, S. Abe, "Range sensing by projecting multiple slits with random cuts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.15, n°6, pp. 647-651, 1993.

[MAT95] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, T. Litwin. "Mars Microrover Navigation: Performance Evaluation and Enhancement", *Autonomous Robots Journal, Special Issue on Autonomous Vehicle for Planetary Exploration*, vol.2, n°4, 1995.7

[MAT97] L. Matthies, T. Balch, B. Wilcox, "Fast optical hazard detection for planetary rovers using multiple spot laser triangulation", *IEEE Proceedings of the International Conference on Robotics and Automation*, Albuquerque, New Mexico (USA), Avril 1997.

[MAX55] J. C. Maxwell, "Experiments on colour as perceived by the eye, with remarks on colour-blindness", *Transactions of the Royal Society Edinburgh*, 21, pp. 275-298, 1855.

[MAX60] J. C. Maxwell, "On the Theory of compound colours, and the relations of the colours of spectrum", *Philosophical Transactions of the Royal Society of London*, 150, pp. 57-84, 1860.

[MCI95] A.M. McIvor, R.J. Valkenburg, "Calibrating a structured light system", *Industrial Research Limited Report*, n°362, Auckland, Février 1995.

[MCL76] K. McLaren, "The development of the CIE 1976 (L*a*b*) uniform colour space and colour-difference formula", *Journal of the Society of Dyers and Colourists*, 92, pp. 338-341, 1976.

[MEY90] A. Meygret, "Vision Automatique Appliquée à la Détection d'Obstacles dans un Environnement Routier", *Thèse de Doctorat*, Institut National de Recherche en Informatique et Automatique, 1990.

[MOH91] R. Mohr, L. Morin, "Relative positioning from geometric invariants", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1991.

[MOH93] R. Mohr, B. Boufama, P. Brand, "Accurate Projective Reconstruction", *Proceedings of the 2nd ESPRIT-ARPA-NSF Workshop on Invariance*, Azores, pp. 257-276, 1993.

[MOR88] H. Morita, K. Yajima, S. Sakata, "Reconstruction of surfaces of 3-D objects by M-array pattern projection method", *Proceedings of the International Conference on Computer Vision*, pp. 468-473, 1988.

[MOR93] L. Morin, "Quelques contributions des invariants projectifs à la vision par ordinateur", *Thèse de Doctorat*, Institut National Polytechnique de Grenoble, Janvier 1993.

[MOU96] E. Mouaddib, E. Brassart, "Etude de faisabilité du relevé de profils tridimensionnels très précis par vision artificielle", *Rapport de Fin de Contrat*, Saint-Gobain Vitrage, Thourotte, 1996.

[NAY94] S.K. Nayar, Y. Nakagawa, "Shape from focus", *Transactions on Pattern Analysis and Machine Intelligence*, vol.16, n°8, pp. 824-831, 1994.

[POE94] C. J. Poelman, T. Kanade, "A paraperspective factorization method for shape and motion recovery", *Proceeding of the 3rd European Conference on Computer Vision*, pp. 97-108, Stockholm (Sweden), Mai 1994.

[QUA00] L. Quan, B. Triggs, "A unification of autocalibration methods", *To appear in the Proceedings of the Asian Conference on Computer Vision*, 2000.

[POS82] J.L. Posdamer, M.D. Altschuler, "Surface measurement by space-encoded projected beam systems, *Computer Graphics Image Process*, 18, pp.1-17, 1982.

[PRO96] M. Proesmans, L. Van Gool, A. Oosterlinck, "One-shot active range acquisition", *Proceedings of the International Conference on Pattern Recognition*, Vienne (Autriche),vol.C, pp. 336-340, 1996.

[SAL97] J. Salvi, "An Approach to Coded Structured Light to Obtain Three Dimensional Information", *Thèse de Doctorat*, Université de Girona, Girona (Espagne), Décembre 1997.

[SAL98] J. Salvi, J. Batlle, E. Mouaddib, "A Robust-Coded Pattern Projection for Dynamic 3D Scene Measurement", *Pattern Recognition Letters*, 19(1998), pp. 1055-1065, 1998.

[SER82] J. Serra, *Image Analysis and Mathematical Morphology*, vol.1, Ac. Press, London, 1982.

[SHA84] M.A. Shah, R. Jain, "Detecting time varying corners", *Computer Vision, Graphics and Image Processing*, vol.28, pp. 345-355, 1984.

[SHA92] A. Shashua, "Projective Structure from two Uncalibrated Images: Structured from Motion and Recognition", *Technical Report*, A.I. Memo n°1363, MIT, september 1992.

[SHA94] L. S. Shapiro, A. Zisserman, M. Brady, "Motion from point matches using affine epipolar geometry", *Proceedings of the 3rd European Conference on Computer Vision*, Stockholm (Sweden), Mai 1994.

[SHI71] Y. Shirai, M. Suwa, "Recognition of polyhedrons with a range finder", *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 80-87, 1971.

[SIL99] J.A. Silva, A.C. Paiva, A. Restivo, A.C. Campilho, J. Laranja Pontes, "A system for 3D measurement of human body surfaces", *Workshop on Medical Imaging*, Aveiro, pp. 1-4, 1999.

[SMI93] S.M. Smith, J.M. Brady, "A scene segmenter ; visual tracking of moving vehicles", *IFAC International Workshop on Intelligent Autonomous Vehicles*, pp. 119-126, 1993.

[SOT00] J. M. Sotoca, M. Buendia, J.M. Iñesta, "A new structured light calibration method for large surface topography", *Pattern Recognition and Applications, Frontiers in Artificial Intelligence*, n°56, pp.261-270, 2000.

[STO86] G. Stockman, G. Hu, "Sensing 3-D surface patches using a projected grid", *Computer Vision and Pattern Recognition*, pp. 602-607, 1986.

[STO90] K. Storjohann, T. Zielke, H.A. Mallot, W. von Seelen, "Visual Obstacle Detection for Automatically Guided Vehicles", *Proceedings of the International Conference on Robotics and Automation* (ICRA'90), pp. 761-765, 1990.

[STRA92] O. Strauss, "Perception de l'environnement par vision en lumière structurée : segmentation des images par poursuite d'indices", *Thèse de Doctorat*, Université de Montpellier II, Janvier 1992.

[STU96] P. Sturm, B. Triggs, "A factorization based algorithm for multi-image projective structure and motion", *Proceedings of the European Conference on Computer Vision* (ECCV'96), 1996.

[TAJ90] J. Tajima, M. Iwakawa, "3-D data acquisition by rainbow range finder", *Proceedings of the International Conference on Pattern Recognition*, pp. 309-313, 1990.

[TAK96] N. Takeda, M. Watanabe, K. Onoguchi, "Moving obstacle detection using residual error of foe estimation", *Proceedings of the International Conference on Robots and System* (IROS'96), pp. 1642-1647, 1996.

[TOM91] C. Tomasi, T. Kanade, "Factoring Image Sequences into Shape and Motion", *Proceedings of the Workshop on Visual Motion*, Los Alamitos (California, USA), pp. 21-28, 1991.

[TRI96] B. Triggs, "Factorization methods for projective structure and motion", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1996.

[TRI97] B. Triggs, "Autocalibration and the absolute quadric", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1997.

[TSAI87] R.Y. Tsai, "A Versatile Camera Calibration Techniques for High-Accuracy 3D Machine Vision Metrology using Off-the-Shelf TV Cameras and Lenses", *IEEE International Journal on Robotics and Automation*, vol. RA-3, pp. 323-344, Août 1987.

[TSI93] L. Tsinas, V. Graefe, "Coupled neural networks for real-time road and obstacle recognition by intelligent road vehicles", *Proceedings of the International Joint Conference on Neural Networks*, pp. 2081-2084, Nagoya (Japon), 1993.

[VUY90] P. Vuylsteke, A. Oosterlinck, "Range image acquisition with a single binary-encoded light pattern", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.12, n°2, pp. 148-164, 1990.

[WEI93] D. Weinshall, "Model-based invariants for 3-D vision", *International Journal of Computer Vision*, 10(1), pp. 27-42, 1993.

[WEN92] J. Weng, P. Cohen, M. Herniou, "Camera calibration with distortion models and accuracy evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.14, n°10, pp. 965-980,1992.

[WILL71] P.M. Will, K.S. Pennington, "Grid Coding: A Preprocessing Technique for Robot and Machine Vision", *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 66-70, 1971.

[WIT81] A.P. Witkin, "Recovering surface shape and orientation from texture", *Journal of Artificial Intelligence*, vol.17, pp. 17-45, 1981.

[WRI28] W.D. Wright, "A re-determination of the trichromatic coefficients of the spectral colours", *Transactions of the Optical Society*, 30, pp. 141-164, 1928.

[WUS91] C. Wust, D.W. Capson, "Surface profile measurement using color fringe projection", *Machine Vision Applications*, vol.4, pp. 193-203, 1991.

[XIE95] M. Xie, "Ground plane obstacle detection from stereo pair of images without matching", *Proceedings of the Asian Conference on Computer Vision*, vol. 2, pp. 280-284, Décembre 1995.

[ZHA88] Z. Zhang, O.D. Faugeras, N. Ayache, "Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints", *IEEE Proceedings of the second Conference on Computer Vision*, Tampa (Florida, USA), pp. 177-186, Décembre 1988.

[ZHA97A] Z. Zhang, G. Xu, "A general expression of the fundamental matrix for both perspective and affine cameras", *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Nagoya (Japon), 1997.

[ZHA97B] Z. Zhang, R. Weiss, A.R. Hanson, “Obstacle detection based on qualitative and quantitative 3-D reconstruction”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19, n°1, pp.15-26, Janvier 1997.

[ZHA98] Z. Zhang, K. Isono, S. Akamatsu, "Euclidean structure from uncalibrated images using fuzzy domain knowledge: application to facial images synthesis", *Proceedings of the International Conference on Computer Vision*, Bombay (India), 1998.

Bibliographie de l'Auteur

Articles de Revue

- B. Marhic, E. Mouaddib, D. Fofi, E. Brassart, "Localisation Absolue par le Capteur Omnidirectionnel SYCLOP", *Traitement du Signal*, Octobre 2000.
- D. Fofi, J. Salvi, E. Mouaddib, "Uncalibrated Reconstruction: An Adaptation to Structured Light Vision", soumis en seconde lecture à la revue *Pattern Recognition*.

Communications Internationales

- D. Fofi, E. Mouaddib, J. Salvi, "How to Self-Calibrate a Structured Light Sensor?", *9th Symposium on Intelligent Robotic System (SIRS'2001)*, Toulouse (France), Juillet 2001.
- D. Fofi, J. Salvi, E. Mouaddib, "Uncalibrated Vision based on Structured Light", *IEEE International Conference on Robotics and Automation (ICRA'2001)*, Séoul (Corée), Mai 2001.
- D. Fofi, J. Salvi, E. Mouaddib, "Euclidean Reconstruction by means of an Uncalibrated Structured Light Sensor", *Vth Ibero-American Symposium on Pattern Recognition (SIARP'2000)*, pp. 159-170, Lisboa (Portugal), Septembre 2000.

Communications Nationales

- D. Fofi, E. Mouaddib, J. Salvi, "Décodage d'un Motif Structurant Codé par la Couleur", *Colloque du Groupe de Recherche en Traitement du Signal et des Images*, Toulouse (France), 10-13 septembre 2001.
- D. Fofi, E. Mouaddib, "La Vision en Lumière Structurée Appliquée à la Navigation d'un Robot Mobile", *Réunion du groupe de travail GT 3.2 Couleur*, GDR-PRC ISIS, 15 Juin 1999.
- D. Fofi, E. Mouaddib, "Vision en Lumière Structurée : Application à la Détection d'Obstacles", *Actes des 9es Journées des Jeunes Chercheurs en Robotique (JJCR '9)*, pp. 95-99, Clermont-Ferrand (France), Juin 1998.

Rapports et Documentations

- D. Fofi, E. Mouaddib, J. Salvi, "3-D Reconstruction from an Uncalibrated Structured Light Sensor", *Rapport de Recherche*, n°IIiA 99-14-RR, Université de Girona (Espagne), Août 1999.
- D. Fofi, E. Mouaddib, "Développement d'un Logiciel de Photo-Emaillage d'Images en Couleurs à l'aide d'un Laser", *Rapport de Fin de Contrat*, Saint-Gobain Vitrages, Thourotte (France), Février 1999.
- D. Fofi, "Détection d'Obstacles par Perception Visuelle pour un Robot Mobile*", Mémoire de DEA de Traitement des Images et du Signal*, ENSEA / Université de Cergy-Pontoise, Septembre 1997.
- D. Fofi, "Mesure des Déformations du Verre par Techniques de Vision Linéaire", *Mission Industrielle*, Saint-Gobain Vitrages, Thourotte (France), Août 1996.

Annexe A : Les modèles de caméra

Nous présentons les modèles de caméra les plus communément utilisés en vision par ordinateur. Deux modèles optiques sont d'abord décrits. Ils prennent en compte les caractéristiques physiques des objectifs et permettent de modéliser le flou ainsi que les variations de focale. Ce sont les modèles à lentille mince et épaisse. Les deux modèles qui les suivent sont des modèles géométriques ; plus simples, ce sont aussi les plus courant en vision par ordinateur : le nombre réduit de leurs paramètres et la relativement bonne précision qu'ils procurent en font des outils efficaces. Ce sont les modèles projectif (ou sténopé) et affine. Les notations et les figures suivent le travail de Gros [GRO93].

A.1 Le Modèle à Lentille Mince

Ce modèle est basé sur l'étude des lentilles convergentes et divergentes, bien connues en optique géométrique, dont sont composés les objectifs des caméras. Considérons une lentille convergente centrée en Ω, appelé centre optique ; *F* est le point focal objet et *F'* le point focal image. La distance focale *f* est donnée par L'axe (Ω*z*) est appelée axe optique du système. Plaçons un plan rétinien Π orthogonalement à l'axe optique et coupant celui-ci au point *C* avec $\left\|\overrightarrow{\Omega C}\right\| = \gamma$ (Figure 82). Le projeté *m* sur le plan rétinien d'un point *M* de la scène observée par une telle caméra se trouve à l'intersection de trois rayons lumineux :

- La droite $\langle M, \Omega \rangle$.
- La droite parallèle à (Ω*z*) coupant $\langle M, F \rangle$ sur la lentille.
- La droite passant par *F'* qui coupe le rayon parallèle à (Ω*z*) et passant par *M*.

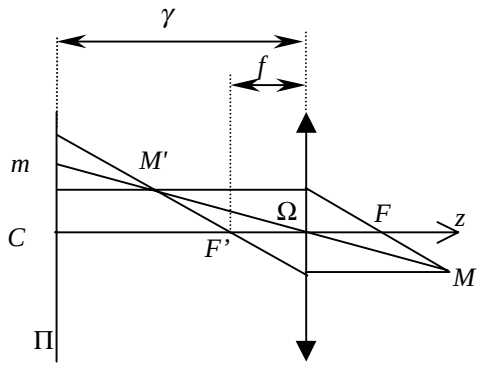


Figure 82. Le modèle à lentille mince

Dans un système de coordonnées centré en *C*, nous obtenons les vecteurs de coordonnées suivant pour *M* et *M'* :

$$\mathbf{M} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \mathbf{M}' = \begin{pmatrix} \dfrac{xf}{\gamma + f - z} \\ \dfrac{yf}{\gamma + f - z} \\ \gamma + \dfrac{(z-\gamma)f}{\gamma + f - z} \end{pmatrix} \tag{156}$$

En examinant la figure, on peut se convaincre que l'image *m* du point *M* est nette, mise au point, si $M' \in \Pi$. L'ensemble des points dont l'image est correctement mise au point forment le *plan focal* ; son équation est :

$$z = \frac{\gamma^2}{(\gamma - f)} \tag{157}$$

Si les points observés n'appartiennent pas tous à un plan orthogonal à l'axe optique, il est impossible d'assurer la netteté pour chacun d'eux. Dans ce cas, l'image de *M* est un disque centré en *m* et de rayon R_m. Pour une lentille circulaire dont le rayon est *R*, on obtient :

$$\mathbf{m} = \begin{pmatrix} \dfrac{\gamma x}{\gamma - z} \\ \dfrac{\gamma y}{\gamma - z} \\ 1 \end{pmatrix} \text{ et } R_m = R \cdot \left(1 + \frac{\gamma}{f} \cdot \frac{\gamma z - \gamma - f}{\gamma - z} \right) \tag{158}$$

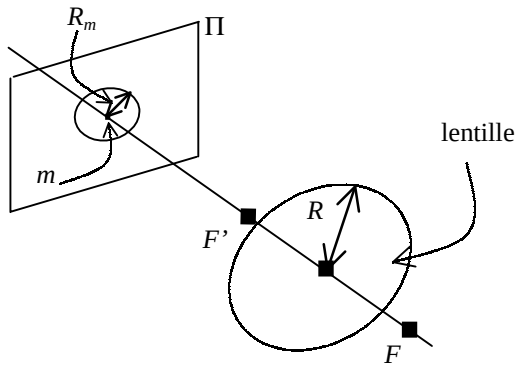


Figure 83. Le flou

L'existence de R_m explique et décrit le flou. Quand un point 3-D n'appartient pas au plan focal, son image est un disque dont les coordonnées du centre peuvent être déduites de *M* par une transformation projective linéaire :

$$\mathbf{m} = \begin{pmatrix} \frac{\gamma f}{\gamma - z} \\ \frac{\gamma y}{\gamma - z} \\ 1 \\ \end{pmatrix} \approx \begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 \\ 0 & 0 & -1 & \gamma \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{159}$$

A.2 Le Modèle à Lentille Epaisse

Le modèle à lentille épaisse est un modèle à lentille mince pour lequel l'épaisseur du système optique est prise en compte. Ceci a pour conséquence l'ajout d'un paramètre supplémentaire : l'*interstice*. Ce modèle est caractérisé par son axe optique (Ωz), ses points focaux *F* et *F'* et par deux plans orthogonaux à l'axe optique appelé *plan image principal* et *plan objet principal* séparés par l'interstice *i*. Chaque point focal est distant de *f* du plan principal auquel il correspond, avec $f = \Omega' F' = \Omega F$; c'est la distance focale (Figure 84).

On peut voir la lentille épaisse comme une lentille mince que l'on aurait scindée et dont les deux parties auraient été placées à une distance *i* l'une de l'autre. Par conséquent, l'équation du plan focal devient :

$$z = \frac{\gamma^2 + \gamma i - if}{\gamma - f} \tag{160}$$

Comme précédemment, il est possible de formuler la transformation d'un point objet *M* au point image *m* par une transformation projective :

$$\mathbf{m} = \begin{pmatrix} \frac{\gamma x}{\gamma + i - z} \\ \frac{\gamma y}{\gamma + i - z} \\ 1 \\ \end{pmatrix} \approx \begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 \\ 0 & 0 & -1 & \gamma + i \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{161}$$

Le flou étant décrit par le rayon R_m du disque formé sur le plan rétinien (pour une lentille circulaire de rayon *R*) :

$$R_m = R \cdot \left(1 + \frac{\gamma}{f} \cdot \frac{\gamma z - f - \gamma - i}{z - \gamma - i} \right) \tag{162}$$

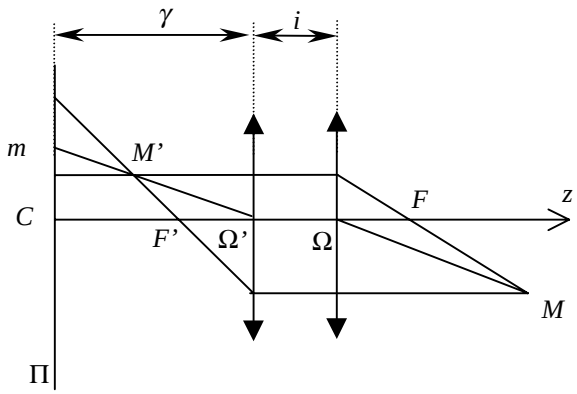


Figure 84. Le modèle à lentille épaisse

D'un point de vue strictement géométrique, ce modèle ne véhicule aucune information supplémentaire par rapport au précédent. Il permet la prise en compte d'une caractéristique réelle des objectifs, l'interstice.

A.3 Le Modèle Projectif

Ce modèle décrit uniquement les caractéristiques géométriques de la caméra. Elle y est considérée comme un système optique ponctuel et le flou n'est pas modélisé ; c'est le modèle de la boite noire, hermétique à la lumière, dont la face avant est percée d'une ouverture ponctuelle : le *sténopé*. L'hypothèse sous-entendue ici est que tous les points de la scène sont mis au point.
Une caméra est définie par un centre optique Ω et un plan rétinien Π. *C* est la projection orthogonale de Ω sur Π ; la distance Ω*C* est appelée distance focale (Figure 85). Il est important de différencier la distance focale du modèle projectif, qui correspond à la distance entre le centre optique et le plan rétinien, de la distance focale des modèles optiques qui dépend du rayon de courbure de la lentille et est indépendante du plan rétinien.

$$\mathbf{m} = \begin{pmatrix} \frac{\gamma x}{\gamma - z} \\ \frac{\gamma y}{\gamma - z} \\ 1 \\ \end{pmatrix} \approx \begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 \\ 0 & 0 & -1 & \gamma \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{163}$$

$$R_m = 0 \tag{164}$$

Comme on peut le constater, le modèle projectif est équivalent au modèle à lentille mince auquel on aurait ôté l'information de flou.

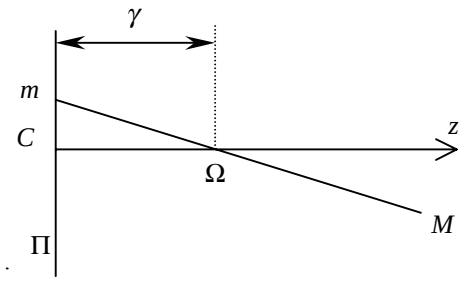


Figure 85. Le modèle sténopé

A.4 Le Modèle Affine

Dans ce modèle, la projection n'est pas effectuée à partir d'un point Ω mais le long d'un direction donnée par un vecteur $\mathbf{v} = \begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T$ (Figure 86). Ceci correspond au cas où Ω est placée à l'infini (ce qui équivaut à négliger la profondeur de la scène). Les équations deviennent :

$$\mathbf{m} = \begin{pmatrix} \frac{k(x - zv_x)}{v_Z} \\ \frac{k(y - zv_y)}{v_Z} \\ 1 \end{pmatrix} \approx \begin{bmatrix} 1 & 0 & -v_x/v_z & 0 \\ 0 & 1 & -v_y/v_z & 0 \\ 0 & 0 & 0 & 1/k \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{165}$$

$$R_m = 0 \tag{166}$$

Ce modèle ne prend en compte ni le flou, ni la profondeur de scène. En plaçant Ω à l'infini, une contrainte quant à la validité du modèle apparait : le *rapport de profondeur*, vu comme le rapport entre la longueur de la scène le long de l'axe optique l_s et la distance scène-caméra d_{sc} doit être "assez petite" (Figure 87)

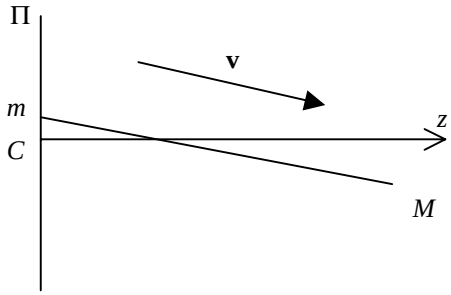


Figure 86. Le modèle affine

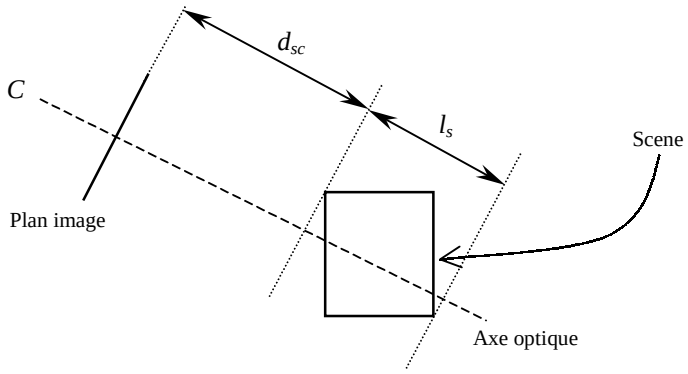


Figure 87. Le rapport de profondeur

Quand **v** est orthogonal au plan image, le modèle devient *le modèle de projection orthographique*. C'est le modèle le plus simple pour une caméra ; sa matrice de projection est :

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/k \end{bmatrix} \tag{167}$$

Le modèle orthographique peut être étendu de manière à prendre en compte la distance séparant la scène de la caméra. Les points observés sont d'abord projetés orthogonalement sur un plan parallèle au plan image et passant par leur barycentre M_0, puis projetés sur la rétine par une transformation de type sténopé. Ce nouveau modèle est appelé modèle de projection perspective faible (Figure 88).

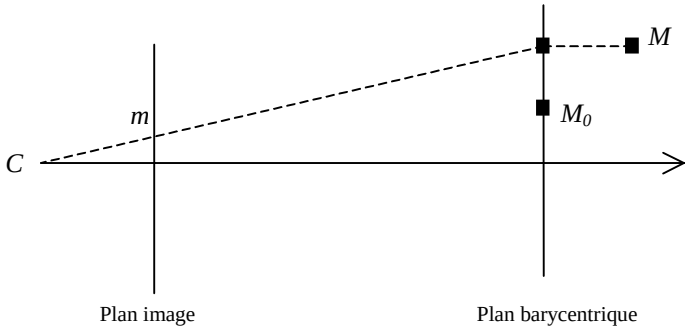


Figure 88. Modèle de projection perspective faible

Si les points sont projetés parallèlement à la droite $\langle C, M_0 \rangle$ sur le plan barycentrique, on obtient le modèle de projection para-perspective (Figure 89).

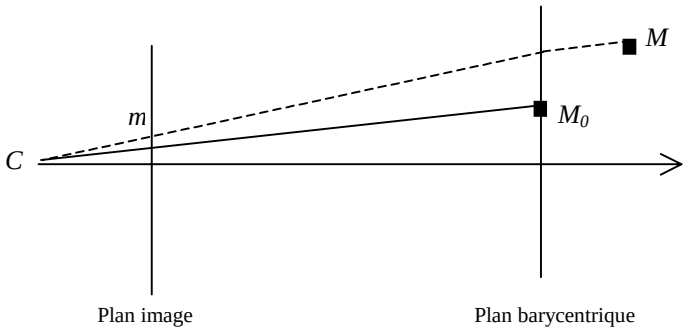


Figure 89. Modèle de projection para-perspective

Annexe B : Eléments de Géométrie

B.1 La Géométrie Projective

B.1.1 Introduction

La géométrie projective est unanimement considérée comme la géométrie étalon en vision par ordinateur. Möbius en a donné une formulation élémentaire, en adéquation[3] avec le processus de formation des images : une transformation projective transforme un point en un point et une ligne en une ligne. Cette définition, très générale, met déjà en lumière quelques caractéristiques de cette géométrie : aucune contrainte sur les angles, les distances, la position ou l'orientation n'est garantie par les transformations de cette géométrie. Autrement dit, si l'on applique une transformation projective à un carré, on obtiendra un quadrilatère quelconque (Figure 90).

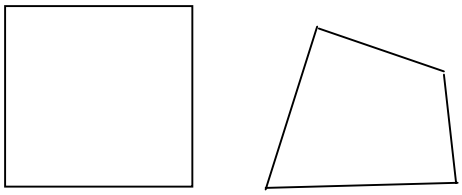


Figure 90. Une transformation projective d'un carré

Intuitivement, l'espace projectif peut être défini comme une extension de l'espace usuel (l'espace des architectes). Pour un plan, on peut ajouter une droite additionnelle, appelée *droite de l'infini* et notée Δ_∞, qui peut être vue comme la ligne d'horizon ; elle a les mêmes propriétés et la même "valance" que toutes les autres droites de l'espace. Deux droites parallèles qui, dit-on communément, ne se coupent pas, se couperont à l'infini dans l'espace projectif : leur point d'intersection appartient à Δ_∞. Mieux, toutes les droites parallèles à une même direction se couperont en un même point à l'infini ; en d'autres termes, chaque point de Δ_∞ représente une direction de l'espace. Pour illustrer de manière très simple cette propriété, l'exemple des voies de chemin de fer semblant se rejoindre à l'horizon est largement et judicieusement cité.
On peut tenir un raisonnement similaire avec les coniques. On distingue usuellement les ellipses, les paraboles et les hyperboles. En géométrie projective, la parabole est définie comme une ellipse tangente à la droite de l'infini (ce qui

[3] Si tant est que les distorsions optiques sont considérées comme négligeables.

conforte l'impression que la parabole est une ellipse infiniment étirée dans une direction) ; de la même manière, une hyperbole est définie comme une ellipse coupant la droite de l'infini (d'où l'impression qu'une hyperbole disparait pour ré-apparaitre de l'autre côté du plan). En fait, il n'existe qu'un seul type de coniques en géométrie projective puisque la droite de l'infini n'a pas de rôle particulier.

B.1.2 L'espace Projectif

Considérons le corps $\mathbf{K}^{n+1} \setminus \{0,\ldots,0\}=\mathbf{K}^{n+1*}$ (c'est-à-dire $\mathbf{K}^{n+1}$ auquel on a ôté le point de l'origine) muni de la relation d'équivalence suivante :

$$\begin{aligned} &\Re : (x_1, \ldots, x_{n+1}) \approx (x_1', \ldots, x_{n+1}') \Leftrightarrow \\ &\exists \lambda \neq 0 \,\big|\, (x_1', \ldots, x_{n+1}') = \lambda (x_1, \ldots, x_{n+1}) \end{aligned} \tag{168}$$

L'espace quotient $\mathbf{K}^{n+1} / \Re$ résultant de cette relation d'équivalence définit l'espace projectif de dimension n, noté $\wp^n$. Les deux n-uplets, représentant le même point dans l'espace projectif, sont appelés *coordonnées homogènes* de ce point. Le signe ≈ est appelé signe de *l'égalité projective* ; il définit une égalité à *un facteur d'échelle non-nul près*.
Il existe une injection de $\mathbf{K}$ dans $\wp$ définie par :

$$\begin{aligned} &\mathbf{K}^n \to \wp^n \\ &(x_1,\ldots,x_n) \mapsto (x_1,\ldots,x_n,1) \end{aligned} \tag{169}$$

Cette application n'est pas surjective : elle n'atteint pas les points de $\wp$ tels que $x_{n+1} = 0$. Ces points ne sont associés avec aucun point de K ; on les appelle les *points de l'infini*. L'hyperplan d'équation $x_{n+1} = 0$ est notamment appelé *hyperplan de l'infini* (c'est une généralisation de la droite de l'infini dont nous avons parlé plus haut).

B.1.3 Le principe de Dualité

Une autre caractéristique importante de la géométrie projective est qu'elle assure l'équivalence entre points et lignes dans $\wp^2$, points et plans dans $\wp^3$, etc. Cette caractéristique est connue sous le nom de *principe de dualité*. Dans $\wp^2$, par exemple, un point est défini par un triplet de nombres, $\mathbf{m} = (x_1 \quad x_2 \quad x_3)^T$, non tous nuls. Une droite est également définie par un triplet de nombres, $\mathbf{l} = (l_1 \quad l_2 \quad l_3)^T$ non tous nuls ; son équation est donnée par :

$$\mathbf{l}^T \cdot \mathbf{m} = \sum_{i=1}^{3} l_i \cdot x_i = 0 \qquad (170)$$

Formellement, il n'y a aucune différence entre un point et une droite dans $\wp^2$. En généralisant ce résultat à tous les hyperplans, on en déduit le principe de dualité :

> *Quelque soit le résultat projectif établi en utilisant des points et des hyperplans, un résultat symétrique est obtenu pour lequel le rôle des points et des hyperplans est interchangé : les points deviennent des hyperplans, les points appartenant à un hyperplan deviennent des hyperplans passant par un point, etc.*

Pour illustrer ce principe, le tableau ci-dessous présente quelques propriétés et leurs duales en regard.

Propriété	**Propriété duale**
▪ Une droite relie deux points.	▪ Un point relie deux droites : un point est l'intersection de deux droites.
▪ Trois points alignés sont situés sur la même droite.	▪ Trois droites sont situées sur un même point : trois droites sont concourantes.
▪ Le triangle est formé par trois points non alignés et par trois droites qui les relie deux à deux.	▪ Le triangle dual est formé par trois droites non concourantes et par trois points d'intersection de droites prises deux à deux.
▪ *Théorème de Desargue* : si deux triangles sont tels que les droites qui relient les sommets correspondants passent par un point, alors les côtés correspondants se coupent en trois points alignés.	▪ Si deux triangles sont tels que les points d'intersection des côtés correspondants sont alignés, alors les sommets correspondants sont reliés par trois droites concourantes.
▪ *Théorème de Pascal* : si un hexagone est inscrit dans une conique, les points d'intersection des côtés opposés sont alignés.	▪ Si un hexagone est circonscrit à une conique, les droites reliant les sommets opposés, les diagonales, sont concourantes.

Table 18. Le principe de dualité

B.1.4 La dépendance Linéaire

On dit des points $P_1,\ldots,P_m$ de $\wp^n$ qu'ils sont linéairement indépendants s'il existe un ensemble de scalaires, non tous nuls, et tels que :

$$\sum_{i=1}^{m} \lambda_i P_i = \left(0,0,\ldots^n,0\right)^T \qquad (171)$$

Base projective

Une base projective de $\wp^n$ est formée de $n+2$ points tels que pour chaque sous-ensemble de $n+1$ points, ces points soient linéairement indépendants. La base canonique projective est formée des points suivants :

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \; e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \; e_{n+1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \quad e_{n+2} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} \tag{172}$$

Où les n premiers points représentent n directions, e^{n+1} le point de l'origine et e^{n+2} le point unité. Tous les points de l'espace projectif peuvent être décrit par une combinaison linéaire de n'importe quel sous-ensemble de $n+1$ points de la base.

B.1.5 Les Transformations Projectives

Une transformation projective, également appelée *collinéation*, est une application linéaire en les coordonnées homogènes. Sous forme matricielle, on peut la représenter ainsi :

$$\begin{aligned} &\wp^n \rightarrow \wp^m \\ &\mathbf{X} \mapsto \mathbf{Y} = \mathbf{W} \cdot \mathbf{X} \end{aligned} \tag{173}$$

où **W** est une matrice non-singulière de dimension (m+1)×(n+1). **X**, **Y** et **W** sont définis à un facteur d'échelle près. En conséquence une collinéation de $\wp^n$ dans $\wp^m$ admet (m+1)×(n+1)-1 degrés de liberté (ou paramètres linéairement indépendants). Définir une transformation projective équivaut donc à estimer sa matrice associée, c'est-à-dire les (m+1)×(n+1)-1 paramètres indépendants qui la composent.
Les transformations projectives d'un espace dans lui-même sont appelées *homographies* ; les transformations projectives d'un espace dans un de ses sous-espaces sont appelées *projections*.

B.1.6 Un invariant Projectif : Le Birapport

Un invariant est une valeur numérique, calculée à partir d'une certaine configuration, qui reste inchangée pour une certaine classe de transformations. L'invariant projectif fondamental est le *birapport*.
Le birapport $k = \{P; A, B, C, D\}$ d'un faisceau de quatre droites convergentes est défini par (Figure 91) :

$$k = \frac{\sin(A\hat{P}C)}{\sin(B\hat{P}C)} \cdot \frac{\sin(B\hat{P}D)}{\sin(A\hat{P}D)} \tag{174}$$

Si l'on ajoute une droite coupant le faisceau en *A'*, *B'*, *C'*, *D'*, le même birapport peut être calculé par la formule suivante :

$$k = \{A, B, C, D\} = \frac{\overline{A'C'}}{\overline{B'C'}} \cdot \frac{\overline{B'D'}}{\overline{A'D'}} \tag{175}$$

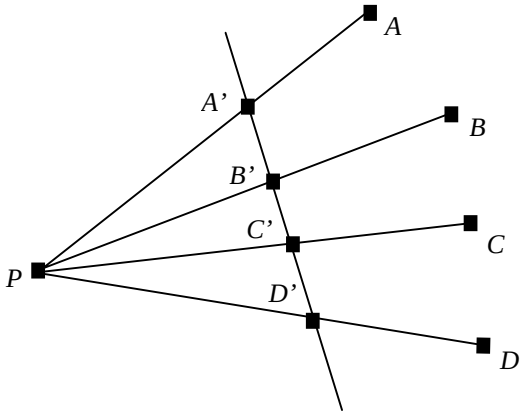


Figure 91. Le calcul du birapport

La formulation donnée en (174) est appelée forme *duale* et celle donnée en (175), forme *primale* du birapport.
Cette fonction peut être formellement étendue en utilisant les conventions suivantes :

$$\frac{a}{\infty} = 0,\ \frac{0}{0} = \frac{\infty}{\infty} = 1,\ \frac{a}{0} = \infty,\ a \cdot \infty = \infty \quad \text{avec } a \in \mathbf{K}^* \tag{176}$$

Grâce au principe de dualité, le calcul du birapport peut être généralisé : quatre points sur une droite, faisceau de quatre plans, etc.

B.2 La Géométrie Affine

B.2.1 Introduction

La géométrie affine peut être vue comme une restriction, un cas particulier de la géométrie projective. Au lieu de considérer toutes les transformations linéaires en les coordonnées homogènes, on ne garde que celles qui laissent l'hyperplan de l'infini globalement invariant. Les transformations affines sont moins nombreuses, elles admettent donc un plus grand nombre d'invariants. Néanmoins, les angles et les distances ne sont pas préservés par celles-ci : si l'on applique une transformation affine à un carré, on obtiendra cette fois un parallélogramme (Figure 92).
Les transformations affines préservent la colinéarité des points : si trois points appartiennent à la même droite, leur image appartiendra également à une même droite et le point central restera entre les deux autres points. Voici quelques autres caractéristiques des transformations affines :

- Les parallèles restent parallèles.
- Les droites concourantes restent concourantes.
- Le rapport des longueurs des segments appartenant à une même droite demeure constant.
- Le rapport des aires de deux triangles demeure constant.
- Les ellipses restent des ellipses, les paraboles des paraboles et les hyperboles des hyperboles.
- Les barycentres restent des barycentres.

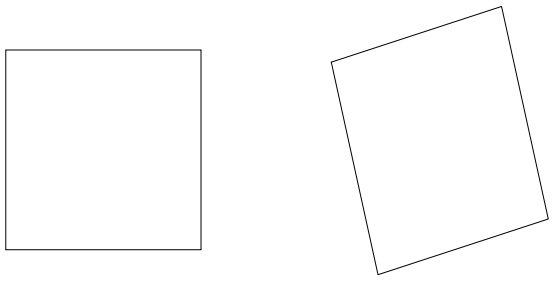


Figure 92. Une transformation affine d'un carré

B.2.2 Les Transformations Affines

Analytiquement, les transformations affines sont représentés sous forme matricielle $\mathbf{Y} = \mathbf{TX} + \mathbf{b}$, pour lesquelles le déterminant de la matrice carré **T** est non-nul. Dans le plan, cette matrice est de dimension 2×2 ; dans l'espace, elle est de dimension 3×3.

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ 1 \end{pmatrix} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1m} & b_1 \\ t_{21} & t_{22} & \cdots & t_{2m} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nm} & b_n \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ 1 \end{pmatrix} \tag{177}$$

La dernière ligne d'une matrice de transformation affine est toujours (0, 0, …, 1).

B.2.3 Un Invariant Affine : Le Rapport de Distances

Trois points alignés *A*, *B* et *C* définissent un unique point *N*, aligné avec les trois autres, appartenant à l'hyperplan de l'infini (Figure 93). Le rapport des trois premiers points est égal au birapport des quatre points :

$$\{A, B, C, N\} = \{A, B, C, \infty\} = \frac{\overline{AC}}{\overline{BC}} \cdot \frac{\overline{BN}}{\overline{AN}} = \frac{\overline{AC}}{\overline{BC}} \cdot \frac{\infty}{\infty} = \frac{\overline{AC}}{\overline{BC}} \tag{178}$$

Ce rapport est l'invariant fondamental de la géométrie affine comme le birapport l'est pour la géométrie projective.

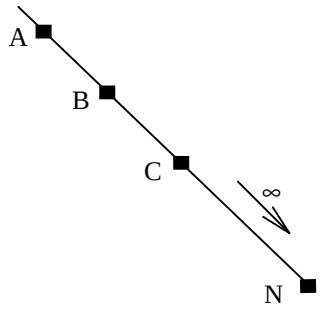


Figure 93. Rapport des distances

B.3 La Géométrie Euclidienne

B.3.1 Introduction

La géométrie Euclidienne est un cas particulier de la géométrie affine. Elle est, de ce fait, encore plus restrictive, eu égard aux nombres de transformations, que

celle-ci. Les transformations Euclidiennes laissent non seulement l'hyperplan de l'infini globalement invariant, mais également un ensemble de points, appelés *absolus*, appartenant à cet hyperplan. Dans l'espace tri-dimensionnel, ces points forment la *conique absolue* :

$$\begin{cases} x^2 + y^2 + z^2 = 0 \\ t = 0 \end{cases} \tag{179}$$

Cette conique est obtenue par l'intersection d'une quadrique avec le plan de l'infini.
Il est important de noter que ce que nous appellerons ici géométrie Euclidienne, et en conséquence, reconstruction Euclidienne, est la géométrie des *similarités* (ou *similitudes*) et non celle des *isométries*. Rappelons brièvement la différence entre ces deux classes de transformations. Les isométries sont les transformations *I* qui satisfont la condition (180) pour deux points *A* et *B* :

$$\overline{AB} = \overline{I(A)I(B)} \tag{180}$$

Elles conservent le produit scalaire, les aires, les barycentres et les angles non-orientés (les transformations qui préservent l'orientation des angles sont appelés *déplacements* ou *mouvements rigides*) ; bien sûr, les distances sont conservées par de telles transformations. Les rotations, les translations, les réflexions par rapport à un axe sont des isométries. Les similarités sont les isométries auxquelles on adjoint un *facteur d'échelle anisotrope*. Les similarités ne préservent pas les distances, mais le rapport des distances de trois points, alignés ou non.

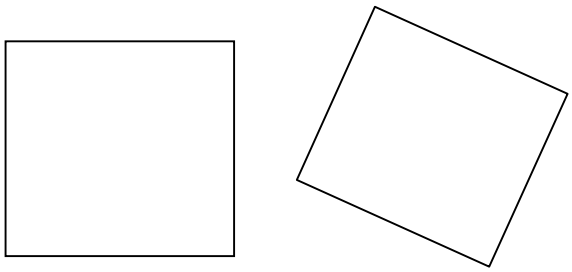


Figure 94. Une transformation Euclidienne d'un carré

B.3.2 Les Transformations Euclidiennes

Les transformations Euclidiennes (ici, les similarités) sont formulées matriciellement par :

$$\mathbf{Y} = \lambda \cdot \mathbf{R} \cdot \mathbf{X} + \mathbf{t} \tag{181}$$

où **R** est cette fois une matrice de rotation (donc une matrice orthogonale telle que $\mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}$), **t** est le vecteur translation et λ est le facteur d'échelle (en fixant λ à 1, on se restreint aux transformations isométriques).

B.4 Stratification des Géométries

Nous avons vu jusqu'ici que les trois principales géométries, la projective, l'affine et l'Euclidienne, pouvaient s'encapsuler l'une dans l'autre. Nous avons vu dans le troisième chapitre que le fait de stratifier les géométries et, subséquemment, les classes de transformations qu'elles admettent, permet d'arriver à des méthodes de reconstruction par restriction successive de l'espace considéré. De manière plus détaillée, la hiérarchie des transformations géométriques est la suivante :

$$\text{Déplacements} \subset \text{Isométries} \subset \text{Similitudes} \subset \text{Affinités} \subset \text{Collinéations}$$

En guise de rappel et de complément, la Table 19 présente les transformations admises dans l'espace Euclidien, affine, projectif et, puisqu'il a, comme on pourra s'en rendre compte par la suite, une importance particulière pour les techniques de reconstruction non-calibrée, l'espace des similitudes ; il donne aussi les principaux invariants de ces espaces. On peut se rendre compte que le nombre de transformations admises par une géométrie est inversement proportionnel au nombre d'invariants admis.

	Euclidien	Similitude	Affine	Projectif
Transformations				
Rotation	X	X	X	X
Translation	X	X	X	X
Echelle isotrope		X	X	X
Echelle anisotrope			X	X
Cisaillement			X	X
Projection perspective				X
Composition de projections				X
Invariants				
Distance	X			
Angle	X	X		
Rapport de distances	X	X		
Parallélisme	X	X	X	
Incidence	X	X	X	X
Birapport	X	X	X	X

Table 19. Géométries, transformations et invariants

Annexe C : Comparaison des birapports

Les birapports sont des grandeurs projectives, non-métriques ; lorsque l'on est amené à comparer deux birapports, pour évaluer par exemple la ressemblance entre deux configurations de points, la différence (grandeur Euclidienne) ou le rapport (grandeur affine) des deux valeurs n'a pas de signification. Une mesure, qui tienne lieu de *distance projective*, est nécessaire : la méthode des birapports aléatoires en est une [MOR93]. Elle consiste à choisir quatre points de manière uniforme dans un intervalle et de calculer leur birapport. Ces données nous permettent de calculer la fonction de densité et la fonction de répartition F des valeurs du birapport. Le décompte du nombre de birapports se situant entre deux valeurs nous fournit une mesure de comparaison. On pourrait calculer la différence $|F(x)-F(y)|$, mais on ne prendrait pas en compte le fait que le point à l'infini est un point quelconque en géométrie projective. Si l'on envisage les deux chemins permettant d'aller de x à y, celui qui reste à distance finie et celui qui passe par l'infini, la distance suivante est plus appropriée :

$$d(x,y)=\min\left(\left\lfloor F(x)-F(y)\right\rfloor, 1-\left|F(x)-F(y)\right|\right) \tag{182}$$

L'estimation de la fonction de répartition, valide pour le birapport de quatre points alignés et pour le birapport de cinq points coplanaires, donne :

$$F(x)=\begin{cases} 0 & \text{si } x=\infty \\ F_1(x)+F_3(x) & \text{si } \infty < x < 0 \\ 1/3 & \text{si } x=0 \\ 1/2+F_2(x)+F_3(x) & \text{si } 0 < x < 1 \\ 2/3 & \text{si } x=1 \\ 1+F_1(x)+F_2(x) & \text{si } 1 < x < \infty \end{cases}$$

$$F_1(x)=\frac{1}{3}\left(x(1-x)\ln\left(\frac{x-1}{x}\right)-x+\frac{1}{2}\right)$$

$$F_2(x)=\frac{1}{3}\frac{x-x\ln(x)-1}{(x-1)^2}$$

$$F_3(x)=\frac{1}{3}\frac{(1-x)\ln(1-x)+x}{x^2}$$

NAVIGATION D'UN VEHICULE INTELLIGENT A L'AIDE D'UN CAPTEUR DE VISION EN LUMIERE STRUCTUREE ET CODEE

Les travaux présentés dans ce mémoire ont pour but l'application de la vision en lumière structurée (capteur formé d'une caméra CCD et d'une source lumineuse) à la navigation de robots mobiles. Ceci nous a conduit à étudier différentes techniques et approches de la vision par ordinateur et du traitement des images. Tout d'abord, nous avons passé en revue les principaux types de codage de la lumière structurée et les principales applications qu'elle trouve en robotique, imagerie médicale et métrologie, pour en dégager une problématique quant à l'utilisation qui nous intéresse. En second lieu, nous proposons une méthode de traitement des images en lumière structurée ayant pour objectif l'extraction des segments de l'image et le décodage du motif structurant. Nous détaillons ensuite une méthode de reconstruction tri-dimensionnelle à partir du capteur non-calibré. La projection d'un motif lumineux sur l'environnement impose des contraintes sévères aux techniques d'auto-calibration. Il ressort que la reconstruction doit être effectuée en deux temps, en une seule prise de vue et une seule projection. Nous précisons la méthode de reconstruction projective utilisée pour nos expérimentations et nous donnons une méthode permettant le passage du projectif à l'Euclidien. En profitant des relations géométriques générées par la projection du motif lumineux, nous montrons qu'il est possible de trouver des contraintes Euclidiennes entre les points de la scène, indépendantes des objets de la scène. Nous proposons également une technique de détection d'obstacles quantitative, permettant d'estimer la carte de l'espace libre observé par le robot. Finalement, nous faisons une étude complète du capteur en mouvement et nous en tirons un algorithme permettant d'estimer son déplacement dans l'environnement à partir de la mise en correspondance des plans qui le compose.

MOTS-CLEFS Lumière structurée, reconstruction, détection d'obstacles, analyse du mouvement.

NAVIGATION OF AN INTELLIGENT VEHICLE BY MEANS OF A CODED STRUCTURED LIGHT SENSOR

The purpose of the work presented in this thesis is the application of structured light vision (a sensor composed by a CCD camera and a light source) to the navigation of mobile robots. This led us to study various techniques and approaches of computer vision and image processing. First of all, we reviewed the principal types of codification for structured light and its main applications in robotics, medical imagery and metrology. Besides, we propose a method of image processing for structured light with the aim to extract the segments of the image and to decode the pattern. Then, we detail a method of three-dimensional reconstruction from an uncalibrated sensor. The projection of a light pattern onto the environment imposes constraints to self-calibration methods. It arises that the reconstruction has to be carried out in two steps, with a unique image capture and a unique pattern projection. We specify the method of projective reconstruction used for our experiments and we give a method which permits to pass from a projective to a Euclidean reconstruction. By using the geometrical relations generated by the projection of the light pattern, we show that it is possible to find Euclidean constraints between the points of the scene, independent of the objects of the scene. We also propose a technique of quantitative detection of obstacles, allowing to estimate the map of free space observed by the robot. Finally, we make a complete study of the sensor in motion : it leads to an algorithm that allows to estimate the displacement of the robot from planes correspondences.

KEYWORDS Structured light, reconstruction, obstacle detection, motion analysis.