Université de Montréal

Calibration, Rectification et Stéréoscopie

par

Sébastien Roy

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures

en vue de l'obtention du grade de

Philosophiae Doctor (Ph.D.)

en informatique

juin, 1999

Université de Montréal

Faculté des études supérieures

Cette thèse intitulée:

Calibration, Rectification et Stéréoscopie

présentée par

Sébastien Roy

a été évaluée par un jury composé des personnes suivantes:

Jean Meunier
(Professeur)

Pierre Poulin
(Professeur)

Neil Stewart
(Professeur)

Janusz Konrad
(Professeur)

Anthony F. J. Moffat
(Professeur)

Thèse acceptée le _____

À mes parents

et à mon frère,

RÉSUMÉ

Cette thèse s'intéresse à trois aspects de la vision par ordinateur, tous reliés à la reconstruction 3D à partir d'images: la calibration de caméra, la rectification d'images stéréoscopiques et la mise en correspondance stéréoscopique.

La calibration de caméra est un problème de grande importance en vision. Il s'agit de calculer le déplacement entre deux caméras, ainsi que les distortions internes de chaque caméra, en n'utilisant que les images provenant de ces caméras. Cette thèse présente une nouvelle méthode de calibration qui ne dépend pas de la disponibilité de points de calibration mis en correspondance.

Une fois la calibration de caméra connue, il devient possible d'établir une relation entre deux images par des méthodes stéréoscopiques de mise en correspondance. Le fait que la plupart de ces méthodes ne mettent en correspondance que les lignes horizontales implique qu'une étape de rectification des images est requise pour aligner la géométrie épipolaire à l'horizontale. Nous présentons ici une nouvelle méthode de rectification, que nous appelons cylindrique, qui permet de rectifier des mouvements arbitraires de caméra tout en conservant constante la taille des images rectifiées.

La troisième partie de cette thèse propose un nouvel algorithme de mise en correspondance stéréoscopique. Cette méthode supporte directement un nombre arbitraire d'images et ne requiert pas de rectification. Basée sur le calcul du flot maximal dans un réseau, la méthode proposée est la première à permettre de résoudre efficacement et optimalement la mise en correspondance avec contrainte de lissage sans qu'il soit fait appel aux contraintes de la géométrie épipolaire. Enfin nous terminons par l'extension de cette méthode à l'estimation des champs aléatoires de Markov, permettant ainsi d'étendre son application à des domaines autres que la stéréoscopie.

ABSTRACT

This thesis concentrates on three aspects of computer vision, all related to 3D reconstruction from multiple images of a scene: camera calibration, rectification of stereoscopic images, and stereoscopic matching.

Camera calibration is a crucial problem in computer vision. It consists in computing the relative displacement between two cameras, as well as internal camera distortions, by using only images taken by these cameras. This thesis will present a new calibration method that does not depend on the availability of established corresponding points.

Once the calibration is obtained, it becomes possible to establish a full correspondence between two images using stereo analysis. The fact that most of these methods establish correspondence between horizontal epipolar lines implies that a prior rectification step is needed when the camera displacement is not horizontal.

We present here a new rectification method, known as cylindrical, that can handle arbitrary camera displacements while preserving constant the size of the resulting rectified images.

The third part of this thesis proposes a new stereoscopic algorithm for establishing correspondence. It handles directly two or more arbitrary views simultaneously and does not require prior rectification of the images. Based on the computation of maximum flow in a graph, this method is the first to solve efficiently and optimally the correspondence problem with a smoothing constraint without relying on the epipolar constraint. Also described is the extension of this method to Markov random fields, thus broadening its application to other fields than stereoscopy.

TABLE DES MATIÈRES

LISTE DES FIGURES

REMERCIEMENTS

Je souhaite remercier tous ceux qui m'ont apporté le support si nécessaire à la création de cette thèse. En particulier, je désire remercier le NEC Research Institute pour son hospitalité durant les trois ans que j'y ai passés comme interne. Ses chercheurs forment une équipe hors du commun dont l'entourage fut pour moi une source constante de motivation et de découverte. En particulier, j'aimerais remercier Ingemar Cox pour m'avoir supervisé et encouragé. De même, je remercie Kevin Lang, Sandiway Phong et Majd Sakr.

À l'Université de Montréal, je dois remercier mon directeur Jean Meunier pour son support et ses encouragements constants ainsi que les autres membres du Laboratoire de Vision et de Modélisation Géométrique, de même que ceux du Laboratoire d'Infographie.

Chapitre 1

INTRODUCTION

La recherche en vision par ordinateur a pour but d'analyser l'environnement visible à la manière de la vision humaine, mais en y ajoutant une plus-value de précision et d'interaction. Une de ses tâches les plus importantes consiste donc à estimer et à interpréter au mieux possible l'environnement tridimensionnel des objets du monde.

La présente thèse s'intéresse à la vision tridimensionnelle, et en particulier au problème de la reconstruction de modèles tridimensionnels à partir de deux ou de plusieurs images d'une même scène, saisies à partir de points de vue différents. Cette reconstruction se décompose en trois étapes distinctes : la *Calibration*, la *Rectification*, et la *Mise en correspondance*, comme l'illustre la figure 1.1. Cette thèse, qui entend présenter des résultats nouveaux reliés à chacune de ces trois étapes, sera conséquemment divisée en trois parties.

Calibration

La calibration est un domaine primordial de la vision par ordinateur en ce qu'elle vise à déterminer quantitativement le processus de formation des images. Son importance ne saurait être surestimée, car elle est une composante essentielle d'une multitude d'algorithmes utilisant l'information provenant de caméras. Lorsqu'un algorithme de vision par ordinateur traite des images provenant de caméras, il requiert presque toujours, en plus des images, de l'information sur les caméras elles-mêmes pour pouvoir procéder à son analyse. Ainsi, les paramètres associés aux caméras sont déterminés à partir du choix du modèle de caméra. Ce modèle est lui-même choisi en fonction de

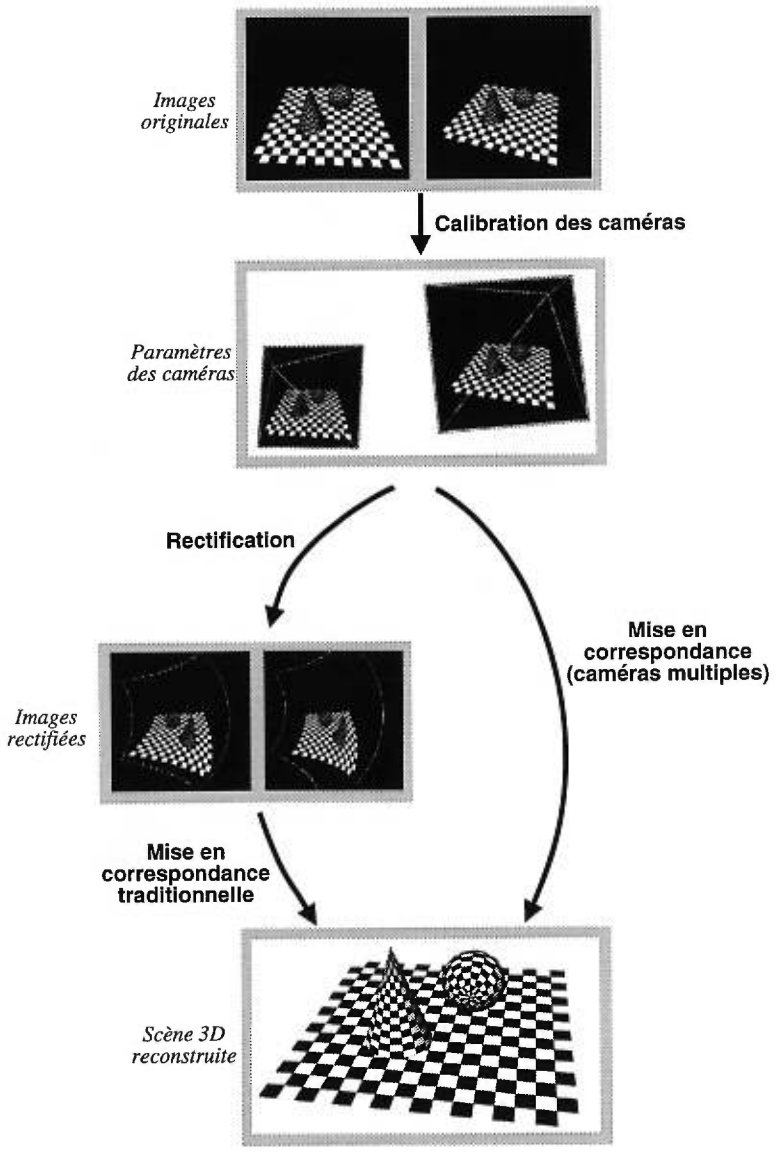

Figure 1.1. Reconstruction 3D à partir d'images. Les paramètres des caméras sont obtenus par une calibration utilisant les images de la scène. L'étape de rectification n'est requise que pour les algorithmes de type "recherche épipolaire" sur deux images.

la tâche à accomplir et du niveau escompté de détail. Par exemple, les algorithmes de reconnaissance d'objets ne requièrent presque aucune information sur la caméra, puisqu'ils tendent à une reconnaissance qui soit indépendante du point de vue et des déformations dues à la caméra. Un modèle très simple peut alors être utilisé. Par contre, les photos satellites utilisées pour construire des cartes routières exigent une information très détaillée non seulement sur la position et l'orientation de la caméra, mais aussi sur les déformations causées par la lentille.

Dans le contexte d'un modèle donné de caméra, la calibration se propose d'évaluer les paramètres des caméras à partir de leurs images et aussi parfois à l'aide d'objets de calibration ou d'une intervention manuelle. La grande variété de modèles et de problèmes à résoudre implique naturellement une grande variété dans les méthodes de calibration. Certaines requièrent une intervention manuelle (comme en photogrammétrie), ou d'objets de calibration (comme pour les appareils à rayons-X ou les scanners tomographiques), alors que d'autres procèdent automatiquement (comme pour un robot envoyé sur la planète Mars).

Quand on saisit plusieurs images à l'aide d'une ou plusieurs caméras, on s'attend à ce que certains paramètres demeurent constants (paramètres internes tels que la distorsion de l'objectif) alors que d'autres peuvent varier (paramètres externes tels que l'orientationa de la caméra). La calibration des paramètres internes s'effectue une fois pour toutes; elle peut se faire *en laboratoire* et s'accompagner d'une forte intervention manuelle. Par contre, la calibration des paramètres externes, effectuée *sur le terrain* pour chaque image, doit souvent être complètement automatisée. C'est celle-ci qui pose le plus de difficultés; elle constituera le sujet de la partie *Calibration* de cette thèse.

Pour le formuler avec plus de précision, le problème consiste à déterminer la position et l'orientation relatives des caméras à partir des images seulement, et ce, sans intervention manuelle pour guider l'algorithme. On y présentera une méthode originale de calibration basée sur les statistiques de l'image, et non sur la disponibilité

de points de correspondance établis par l'usager.

Stéréoscopie

Les deux parties *Rectification* et *Mise en correspondance*, qui suivent la *Calibration*, s'inscrivent dans le contexte de la reconstruction stéréoscopique. La stéréoscopie est un des problèmes fondamentaux de la vision par ordinateur. Elle s'intéresse au calcul de la profondeur à partir de deux ou de plusieurs images d'une même scène, prises sous des angles de vue différents. En établissant que deux points d'images différentes représentent en fait la projection d'un même point de la scène, on peut calculer par triangulation la position tridimensionnelle exacte de ce point dans le monde et ainsi connaître sa profondeur par rapport à un observateur associé à une des caméras. On qualifie la stéréoscopie de méthode *passive* de reconstruction puisqu'elle ne cherche pas activement les profondeurs dans le monde, contrairement à ce que font la chauve-souris ou le sonar, par exemple.

Le problème de la mise en correspondance est un problème fondamental de vision par ordinateur. Essentiel pour la navigation autonome, pour la détection d'obstacles, et pour la reconstruction de modèles tridimensionnels réalistes, il a fait l'objet de recherches intensives. En particulier, l'infographie, qui nécessitait des modèles tridimensionnels de plus en plus réalistes, a renouvelé l'intérêt pour la mise en correspondance. Le réalisme qu'on a pu atteindre à partir de photos réelles laisse entrevoir de grandes possibilités pour l'avenir.

Rectification

Puisque bon nombre d'algorithmes de mise en correspondance assument que les caméras sont positionnées en parfait alignement horizontal, la rectification d'images devient nécessaire pour permettre leur utilisation lorsque les caméras, une fois calibrées, s'avèrent alignées autrement qu'à l'horizontale. C'est alors que la rectification

d'image rend possible l'utilisation de caméras aux configurations arbitraires.

Cette thèse présente un nouvel algorithme de rectification de caméra, qui garantit une taille d'image rectifiée constante, donc indépendante de la géométrie des caméras.

Il devient aussi possible d'utiliser le même algorithme de mise en correspondance pour n'importe quelle géométrie de caméras. À cause de sa grande généralité, cet algorithme peut être aussi utilisé pour créer une vue panoramique sous forme d'une mosaïque d'images, à partir de déplacements arbitraires d'une caméra.

Mise en correspondance

La présente thèse apporte une nouvelle méthode de mise en correspondance basée sur le calcul du flot maximum dans les réseaux. Cet algorithme permet de reconstruire efficacement et optimalement la profondeur sous forme d'une surface, à partir de deux ou plusieurs images prises de points de vue différents.

L'algorithme présenté dans cette thèse est à notre connaissance le premier à s'appliquer indépendamment de la contrainte épipolaire et donc à permettre la mise en correspondance simultanée de plus de deux images.

Notons que la rectification des images n'est pas requise par notre méthode, puisque la correspondance n'est pas directement établie le long des lignes épipolaires mais bien dans toute l'image simultanément. La rectification n'est nécessaire que si l'algorithme de mise en correspondance est *classique*, c'est-à-dire basé sur la correspondance de lignes horizontales.

Organisation de la thèse

L'essentiel de la contribution de cette thèse est subdivisé en trois volets, constitués des chapitres 4, 6 et 8, qui sont formés chacun d'un article publié dans le cadre d'une conférence ou ayant été soumis à un journal scientifique [69–71]. Ceux-ci sont entrecoupés des chapitres 2, 3, 5, 7 ayant pour but d'introduire les concepts généraux,

une revue de la littérature, et la suggestion de nouvelles applications et de perspectives nouvelles en vue de recherches futures.

Plus précisément, les principes de base de la stéréoscopie, comme le modèle de caméra et la géométrie épipolaire, seront décrits au chapitre 2. Après une introduction aux concepts de calibration de caméra au chapitre 3, la nouvelle méthode de calibration *Motion Without Structure* sera présentée au chapitre 4.

La rectification est introduite au chapitre 5, alors que la nouvelle méthode de rectification cylindrique *Cylindrical Rectification to Minimize Epipolar Distorsion* est présentée au chapitre 6.

Finalement, les notions plus avancées de stéréoscopie à images multiples, présentées au chapitre 7, serviront d'introduction à notre étude sur la mise en correspondance par calcul de flot maximum *Stereo Without Epipolar Line : A Maximum Flow Formulation* qui constitue le chapitre 8.

Finalement, le chapitre 9 propose un essai de synthèse sur les apports de notre thèse dans le champ de la vision par ordinateur, ainsi qu'une réflexion sur les prolongements possibles et sur les travaux futurs qui pourraient en découler.

Chapitre 2

CAMÉRAS ET GÉOMÉTRIE ÉPIPOLAIRE

Les applications de la vision utilisant des images provenant de caméras requièrent un modèle de caméra. Selon le niveau de réalisme demandé, différents modèles peuvent être utilisés, allant du plus simple au plus complexe. Dans la vaste majorité des cas, comme dans cette thèse, le modèle très simple de caméra *sténopé* (ou *pinhole*) sera utilisé.

Ce chapitre présente un tour d'horizon des concepts de base de la stéréoscopie; on introduira d'abord le modèle de caméra, puis la géométrie épipolaire.

2.1 Modèle de caméra

Le modèle de caméra sténopé est illustré à la Figure 2.1. On associe à la caméra son propre système de coordonnées. La caméra *regarde* dans la direction de l'axe z positif, à partir de son centre optique. L'image créée sur le plan de projection est alignée avec les axes x et y. La distance focale f entre le centre optique et le plan de projection est fixée à $f = 1$. Un point représenté par (x, y, z) dans le système de la caméra se projette directement au point image $(x/z, y/z)$.

Plus généralement, un point tridimensionnel du monde $\mathbf{p}$ est projeté sur le plan de projection de la caméra pour former le point image $\mathbf{p}'$. Cette projection est représentée par la relation

$$\mathbf{p}' = \mathbf{T} \cdot \mathbf{p} \tag{2.1}$$

où $\mathbf{T}$ est une matrice 3×4 qui représente la transformation du système de coordonnées du monde vers celui de la caméra. Les points $\mathbf{p}'$ et $\mathbf{p}$ sont respectivement représentés dans des espaces projectifs de deux et trois dimensions. Des coordonnées projectives,

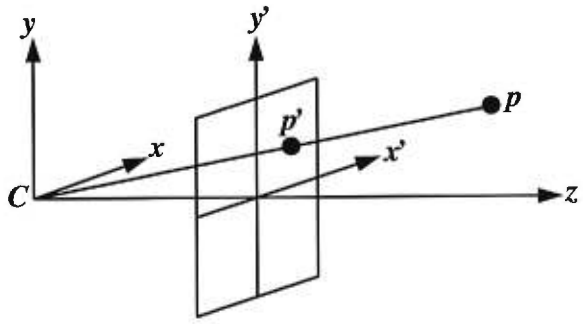

Figure 2.1. Modèle de caméra sténopé. Le centre optique (C) est situé à l'origine. L'axe optique est l'axe z. Les axes x', y' de l'image sont parallèles aux axes x et y. Le point 3D p se projette dans l'image sur le point $\mathbf{p}'$.

aussi appelées *homogènes*, sont utilisées pour représenter ces points. L'équation 2.1 devient

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{T} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{2.2}$$

La coordonnée image $\mathbf{p}'$ non projective est obtenue à partir de la représentation projective en divisant par la dernière composante w, pour obtenir

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix}.$$

2.1.1 Coordonnées projectives

Comme le démontre l'équation 2.2, la caméra applique une projection des points du monde tridimensionnel vers un monde à deux dimensions, le plan image (ou plan de projection).

On représente généralement un point 3D par ses coordonnées projectives, ou homogènes, par un vecteur à 4 composantes dans l'espace projectif à trois dimensions.

Un vecteur d'un espace projectif présente la particularité d'être considéré équivalent à lui-même multiplié par un scalaire quelconque. On a donc la relation d'équivalence

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} wu \\ wv \\ w \end{bmatrix} \quad \forall w \neq 0.$$

Dans le cas d'un point du monde 3D, on utilise l'espace projectif 3D (à quatre composantes) pour simplifier et unifier l'application des translations. C'est pour cette raison qu'en général on représente le point (x, y, z) par ses coordonnées homogènes

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

ce qui permet d'appliquer simultanément une transformation affine 3D $\mathbf{A}$ (rotation, changement d'échelle, cisaillement) et une translation (t_x, t_y, t_z) en une seule opération

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{W} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad , \text{ où } \mathbf{W} = \begin{bmatrix} & & & t_x \\ & \mathbf{A} & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

plutôt que deux opérations

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}.$$

2.1.2 Projection

La représentation projective peut aussi être directement utilisée pour le passage du monde 3D vers le monde 2D de l'image. Un point projectif 2D a trois coordonnées, et

tous ses multiples par un scalaire sont équivalents. Le passage d'un point 3D projectif vers un point 2D projectif résulte donc d'une simple multiplication par une matrice de projection **J**, c'est-à-dire

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{J} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{avec} \quad \mathbf{J} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.4}$$

Puisque qu'un point projectif 2D n'est effectivement projeté sur le plan image que lorsque sa dernière composante est 1, on doit donc le *normaliser* pour obtenir le point image (x', y')

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \frac{1}{w} \begin{bmatrix} u \\ v \\ w \end{bmatrix}.$$

On a donc exprimé par une relation linéaire projective la relation non linéaire euclidienne

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \end{bmatrix}.$$

2.1.3 Modèle détaillé de caméra

La matrice de transformation d'une caméra (**T** dans l'équation 2.2) peut être décomposée de différentes façons, en fonction du niveau de détail souhaité du modèle de caméra.

Le modèle général

La forme la plus générale de **T** est la composition d'une matrice de projection **J** (voir équation 2.4) et d'une matrice de passage **W** du système de coordonnées de référence vers celui de la caméra, où l'axe optique est l'axe z et où l'image est formée selon les

axes x et y. On a

$$\mathbf{T} = \mathbf{J} \cdot \mathbf{W} \tag{2.5}$$

où $\mathbf{W}$ est une matrice 4×4 inversible. On peut noter que cette relation définit $\mathbf{T}$ comme la matrice $\mathbf{W}$ de laquelle on élimine la dernière rangée. Pour cette raison, $\mathbf{W}$ suppose toujours la forme

$$\mathbf{W} = \begin{bmatrix} (\text{matrice } 3 \times 4) \\ 0 \quad 0 \quad 0 \quad 1 \end{bmatrix}.$$

Paramètres internes et externes

Il est possible de décomposer la matrice de passage d'une caméra ($\mathbf{W}$ dans l'équation 2.5) de façon à enrichir le modèle de caméra. Cette décomposition représente les paramètres d'une caméra, classés en deux types: paramètres *internes* et *externes*. On a

$$\mathbf{W} = \mathbf{W}^{int} \cdot \mathbf{W}^{ext} \tag{2.6}$$

où les matrices $\mathbf{W}^{int}$ et $\mathbf{W}^{ext}$ représentent respectivement les paramètres internes et externes de la caméra.

Paramètres internes

Les paramètres internes sont ceux qui ne dépendent pas de la position tridimentionnelle de la caméra. Ils caractérisent l'image projetée. Il s'agit par exemple de la distance focale, du ratio de l'image, du centre de l'image, et de l'obliquité. La forme que prend $\mathbf{W}^{int}$ est celle d'une transformation affine 2D ($\mathbf{A}_{2\times2}$) et d'une translation 2D ($\mathbf{t}_{2\times1}$) dans l'image, donc dans le plan $x - y$ de la caméra

$$\mathbf{W}^{int} = \begin{bmatrix} \mathbf{A}_{2\times2} & \mathbf{t}_{2\times1} & 0 \\ 0 \quad 0 & 1 & 0 \\ 0 \quad 0 & 0 & 1 \end{bmatrix}.$$

En général, les paramètres internes peuvent être évalués expérimentalement; on suppose qu'ils restent fixes tout au long d'une séquence d'images. Notons qu'une caméra vidéo munie d'un *zoom* peut modifier sa distance focale et fait donc exception à la règle.

Paramètres externes

Les paramètres externes décrivent la situation de la caméra dans l'espace; ils se composent de la position du centre optique et de l'orientation de la caméra par rapport à l'origine du monde tridimensionnel dans lequel elle se situe. La forme que prend $\mathbf{W}^{ext}$ est celle d'une rotation 3D ($\mathbf{A}_{3\times3}$) et d'une translation 3D ($\mathbf{t}_{3\times1}$)

$$\mathbf{W}^{ext} = \begin{bmatrix} & \mathbf{A}_{3\times3} & & \mathbf{t}_{3\times1} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.7}$$

Les paramètres externes, position et orientation de la caméra, sont généralement variables et leur détermination automatique est d'une importance primordiale en vision par ordinateur.

2.2 Segments épipolaires

Soit deux caméras dont les positions dans le monde 3D sont définies par les matrices de passage $\mathbf{W}_a$ et $\mathbf{W}_b$ (voir équation 2.5). Notons que la notion de *deux* caméras en des positions différentes prenant simultanément une image de la scène, est effectivement équivalente à *une seule* caméra prenant une première image d'une scène fixe, puis une seconde, après s'être déplacée. La différence, s'il en est une, réside dans le fait que le déplacement d'une seule caméra implique généralement que les paramètres internes de la caméra restent fixes alors que ceux de deux caméras simultanées peuvent être complètement différentes. Dans ce qui suit, on renverra au modèle le plus général (deux caméras simultanées) sauf mention explicite du contraire.

Selon les équations 2.1 et 2.5, un point $\mathbf{p}_w$ du monde 3D ($_w$ pour *world*) sera projeté par les caméras A et B en points images $\mathbf{p}'_a$ et $\mathbf{p}'_b$, respectivement, selon les relations

$$\mathbf{p}'_a = \mathbf{J} \cdot \mathbf{W}_a \cdot \mathbf{p}_w \tag{2.8}$$

$$\mathbf{p}'_b = \mathbf{J} \cdot \mathbf{W}_b \cdot \mathbf{p}_w. \tag{2.9}$$

Pour un point image donné

$$\mathbf{p}'_a = \begin{bmatrix} x' & y' & 1 \end{bmatrix}^T$$

l'ensemble des points $\mathbf{p}_a(d)$, du monde de la caméra A, qui s'y projettent est défini comme

$$\mathbf{p}_a(d) = \begin{bmatrix} x' & y' & 1 & d \end{bmatrix}^T$$

où d est la disparité, toujours positive et reliée à la profondeur z par la relation

$$d = \frac{1}{z}.$$

L'ensemble des points du monde $\mathbf{p}_w(d)$ qui se projettent en $\mathbf{p}'_a$ est donc

$$\mathbf{p}_w(d) = \mathbf{W}_a^{-1} \cdot \mathbf{p}_a(d).$$

Si on projette cet ensemble $\mathbf{p}_w(d)$ sur l'image de la seconde caméra, on obtient un ensemble de points images $\mathbf{p}'_b(d)$ défini par

$$\begin{aligned} \mathbf{p}'_b(d) &= \mathbf{J} \cdot \mathbf{W}_b \cdot \mathbf{p}_w(d) \\ &= \mathbf{J} \cdot \mathbf{W}_b \cdot \mathbf{W}_a^{-1} \cdot \mathbf{p}_a(d) \\ &= \mathbf{J} \cdot \mathbf{W}_{ba} \cdot \mathbf{p}_a(d) \end{aligned} \tag{2.10}$$

où $\mathbf{W}_{ba}$ est la *matrice de passage* de la caméra A vers la caméra B telle que

$$\mathbf{W}_{ba} = \mathbf{W}_b \cdot \mathbf{W}_a^{-1} \tag{2.11}$$

et inversement $\mathbf{W}_{ab}$ est celle de B vers A telle que

$$\mathbf{W}_{ab} = \mathbf{W}_a \cdot \mathbf{W}_b^{-1}. \tag{2.12}$$

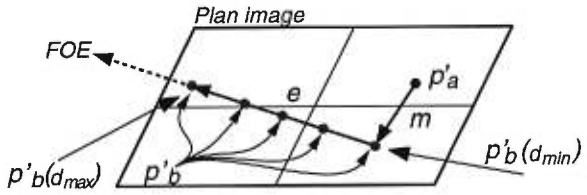

Figure 2.2. Segment épipolaire. Le point $\mathbf{p}'_b$ correspondant au point $\mathbf{p}'_a$ se situe entre $\mathbf{p}'_b(d_{min})$ et $\mathbf{p}'_b(d_{max})$ (sur le vecteur $\mathbf{e}$). Le vecteur $\mathbf{m}$ représente la composante rotationnelle du déplacement de $\mathbf{p}$. La droite contenant $\mathbf{e}$ passe toujours par le point d'expansion (FOE).

Le déplacement apparent du point $\mathbf{p}'_a$ vers le point $\mathbf{p}'_b$, induit par un déplacement de caméra, possède deux composantes distinctes, les vecteurs $\mathbf{m}$ et $\mathbf{e}$, illustrés à la Figure 2.2. On a

$$\mathbf{m} = \mathbf{p}'_b(d_{min}) - \mathbf{p}'_a \tag{2.13}$$

$$\mathbf{e} = \mathbf{p}'_b(d_{max}) - \mathbf{p}'_b(d_{min}) \tag{2.14}$$

où d_{min} et d_{max} désignent respectivement la disparité minimum et maximum, ou inversement la profondeur maximum et minimum, et sont positives. Ainsi, le déplacement d'un point $\mathbf{p}'_a$ dans l'image de la première caméra s'effectue toujours le long d'une droite, appelée *droite épipolaire*. De plus, celui-ci est restreint, le long de cette droite, à un segment qui représente les valeurs physiquement réalisables de la profondeur, c'est-à-dire $0 \leq d_{min} \leq d \leq d_{max}$, puisqu'un point ne peut être plus loin que l'infini ($d = \frac{1}{\infty} = 0$) ou plus près que le devant de la caméra ($d = d_{max}$). On a donc

$$\mathbf{p}'_b = \mathbf{p}'_a + \mathbf{m} + k\ \mathbf{e} \qquad 0 \leq k \leq 1. \tag{2.15}$$

Naturellement, la mise en correspondance consiste simplement à rechercher dans l'intervalle $[0, 1]$ une valeur de k telle que les points images $\mathbf{p}'_a$ et $\mathbf{p}'_b$ possèdent la plus grande *similarité*.

2.3 *Matrice fondamentale et matrice essentielle*

À partir du modèle de caméra élaboré précédemment, il est possible de dériver une relation linéaire simple pour la mise en correspondance de points de deux images. On désigne ici cette relation par les termes *matrice fondamentale* et aussi *matrice essentielle.*

Certaines notations utilisées dans cette section doivent être établies. Soient les vecteurs $\mathbf{a}$ et $\mathbf{b}$. On dénote par $[\mathbf{a}]_\times$ la matrice *produit vectoriel* 3×3 telle que

$$[\mathbf{a}]_\times \cdot \mathbf{b} = \mathbf{a} \times \mathbf{b} \ , \quad \forall \mathbf{b}$$

Pour $\mathbf{a} = (a_x, a_y, a_z)$, elle se définit comme

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}.$$

Soit une matrice 4×4 $\mathbf{W}$. On définit $[\mathbf{W}]_R$ comme une sous-matrice 3×3 issue des trois premières lignes et trois premières colonnes de $\mathbf{W}$. On définit $[\mathbf{W}]_t$ comme la sous-matrice 3×1 de $\mathbf{W}$ issue des trois premières lignes et de la quatrième colonne de $\mathbf{W}$. Ainsi $\mathbf{W}$ prend typiquement la forme

$$\begin{bmatrix} [\mathbf{W}]_R & [\mathbf{W}]_t \\ 0 \ 0 \ 0 & 1 \end{bmatrix}.$$

Comme le présente la Figure 2.3, un point $\mathbf{p}'_a$ de l'image de la caméra A et son homologue $\mathbf{p}'_b$ de la caméra B sont toujours coplanaires avec les centres optiques des caméras, $\mathbf{CA}$ et $\mathbf{CB}$. Si on exprime les points $\mathbf{p}'_a$, $\mathbf{p}'_b$, et $\mathbf{CB}_b$ dans un système de coordonnées commun (celui de la caméra A), ils forment alors trois vecteurs par rapport au point $\mathbf{CA}_a$ que l'on identifiera par $\mathbf{v}_0$, $\mathbf{v}_1$ et $\mathbf{v}_2$. Étant donné ces trois vecteurs coplanaires, on peut poser la relation suivante:

$$\mathbf{v}_0 \cdot (\mathbf{v}_1 \times \mathbf{v}_2) = 0$$

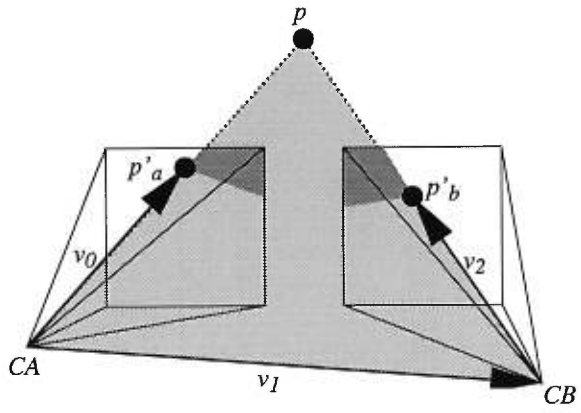

Figure 2.3. Géométrie épipolaire. Les vecteurs $\mathbf{v}_0$, $\mathbf{v}_1$ et $\mathbf{v}_2$, issus des points $\mathbf{p}'_a$, $\mathbf{p}'_b$ et des centres optiques $\mathbf{CA}$ and $\mathbf{CB}$, sont tous coplanaires.

ou sous forme matricielle

$$\mathbf{v}_0 \cdot [\mathbf{v}_1]_\times \cdot \mathbf{v}_2 = 0. \tag{2.16}$$

Un centre optique a toujours comme coordonnées $(0, 0, 0)$ dans le système de coordonnées de sa propre caméra, soit

$$\mathbf{CA}_a = \mathbf{CB}_b = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

mais il peut aussi s'exprimer dans le système de l'autre caméra par une relation similaire à l'équation 2.10

$$\begin{aligned} \mathbf{CA}_b &= \mathbf{J} \cdot \mathbf{W}_{ba} \cdot [\mathbf{CA}_a; 1] = [\mathbf{W}_{ba}]_t \\ \mathbf{CB}_a &= \mathbf{J} \cdot \mathbf{W}_{ab} \cdot [\mathbf{CB}_b; 1] = [\mathbf{W}_{ab}]_t \end{aligned} \tag{2.17}$$

où $\mathbf{W}_{ba}$ et $\mathbf{W}_{ab}$ sont définis selon les équations 2.11 et 2.12. La notation $[\,\cdot\,; 1]$ désigne un vecteur auquel on ajoute l'élément 1.

Ainsi, les trois vecteurs $\mathbf{v}_0$, $\mathbf{v}_1$ et $\mathbf{v}_2$ se définissent comme

$$\begin{aligned} \mathbf{v}_0 &= \mathbf{p}'_a - \mathbf{CA}_a = \mathbf{p}'_a \\ \mathbf{v}_1 &= \mathbf{CB}_a - \mathbf{CA}_a = [\mathbf{W}_{ab}]_t \end{aligned}$$

$$\begin{aligned}\mathbf{v}_2 &= \mathbf{J}\cdot\mathbf{W}_{ab}\cdot([\mathbf{p}'_b;1]-[\mathbf{CB}_b;1])\\ &= \mathbf{J}\cdot\mathbf{W}_{ab}\cdot[\mathbf{p}'_b;0]\\ &= [\mathbf{W}_{ab}]_R\cdot\mathbf{p}'_b.\end{aligned}$$

Nous pouvons maintenant développer l'équation 2.16 pour obtenir

$$\begin{array}{ccccc}\mathbf{v}_0 & \cdot & [\mathbf{v}_1]_\times & \cdot & \mathbf{v}_2=0\\ \mathbf{p}'_a & \cdot & [[\mathbf{W}_{ab}]_t]_\times & \cdot & ([\mathbf{W}_{ab}]_R\cdot\mathbf{p}'_b)=0\\ \mathbf{p}'_a & \cdot & ([[\mathbf{W}_{ab}]_t]_\times\cdot[\mathbf{W}_{ab}]_R) & \cdot & \mathbf{p}'_b=0\\ \mathbf{p}'_a & \cdot & \mathbf{F} & \cdot & \mathbf{p}'_b=0.\end{array} \tag{2.18}$$

On voit donc que la relation entre les caméras A et B peut se ramener à une relation linéaire simple

$$\mathbf{p}'_a\cdot\mathbf{F}\cdot\mathbf{p}'_b=0$$

où $\mathbf{F}$ est la *matrice fondamentale*, une matrice 3×3 qui résume l'orientation relative des caméras, définie à l'équation 2.18 comme

$$\mathbf{F}=[[\mathbf{W}_{ab}]_t]_\times\cdot[\mathbf{W}_{ab}]_R$$

avec $\mathbf{W}_{ab}=\mathbf{W}_a\cdot\mathbf{W}_b^{-1}$. On détermine $\mathbf{F}$ à partir de la solution d'un système d'équations linéaires issu de paires $(\mathbf{p}'_a,\mathbf{p}'_b)$ de points mis en correspondance.

Si on connaît les paramètres internes des caméras (voir équation 2.6) $\mathbf{W}_a^{int}$ et $\mathbf{W}_b^{int}$ tels que

$$\begin{aligned}\mathbf{p}'_a &= \mathbf{J}\cdot\mathbf{W}_a\cdot\mathbf{p}_w\\ &= \mathbf{J}\cdot\mathbf{W}_a^{int}\cdot\mathbf{W}_a^{ext}\cdot\mathbf{p}_w\end{aligned}$$

on peut définir les points *normalisés* $\mathbf{p}''_a$ et $\mathbf{p}''_b$ comme

$$\begin{aligned}\mathbf{p}''_a &= \mathbf{W}_a^{int^{-1}}\cdot\mathbf{p}'_a=\mathbf{J}\cdot\mathbf{W}_a^{ext}\cdot\mathbf{p}_w\\ \mathbf{p}''_b &= \mathbf{W}_b^{int^{-1}}\cdot\mathbf{p}'_b=\mathbf{J}\cdot\mathbf{W}_b^{ext}\cdot\mathbf{p}_w\end{aligned}$$

et les substituer dans l'équation de la matrice fondamentale pour obtenir la matrice *essentielle* E

$$\begin{array}{ccccc} \mathbf{p}''_a & \cdot & ([[\mathbf{W}^{ext}_{ab}]_t]_\times \cdot [\mathbf{W}^{ext}_{ab}]_R) & \cdot & \mathbf{p}''_b = 0 \\ \mathbf{p}''_a & \cdot & ([\mathbf{t}_{ab}]_\times \cdot \mathbf{R}_{ab}) & \cdot & \mathbf{p}''_b = 0 \\ \mathbf{p}''_a & \cdot & \mathbf{E} & \cdot & \mathbf{p}''_b = 0 \end{array} \tag{2.19}$$

où $\mathbf{W}^{ext}_{ab} = \mathbf{W}^{ext}_a \cdot \mathbf{W}^{ext^{-1}}_b$ et en assumant que les paramètres externes suivent la forme de l'équation 2.7, c'est-à-dire qu'ils se composent d'une rotation et d'une translation. La translation relative entre les centres des caméras est $\mathbf{t}_{ab}$ alors que $\mathbf{R}_{ab}$ représente la rotation relative entre les orientations des caméras. Dans le cas simple où la caméra A serait aussi la référence du monde, la matrice $\mathbf{W}_a$ est l'identité; $\mathbf{t}_{ab}$ devient alors la position de la caméra B, et $\mathbf{R}_{ab}$ son orientation.

La forme très simple de la matrice $\mathbf{E}$ permet de retrouver directement la position et l'orientation des caméras. Par contre, si les paramètres internes ne sont pas connus, seule la matrice fondamentale $\mathbf{F}$ est disponible et il n'est pas possible d'en extraire à la fois les paramètres internes, la position et la rotation des caméras.

2.4 La détermination de la géométrie de caméra

Le degré de détail choisi pour le modèle de caméra a un impact important sur la détermination des paramètres de ce modèle. Dans le cas d'un modèle peu détaillé, où on tente de retrouver directement la matrice $\mathbf{W}_{ba}$ (équation 2.11), il y a douze paramètres, soient les éléments de $\mathbf{W}_{ba}$. Puisque cette matrice est définie à un facteur d'échelle près, seulement onze de ces paramètres doivent être évalués, le douzième pouvant être arbitrairement fixé à la valeur 1, à condition qu'il ne soit pas nul. Il est possible de résoudre un système linéaire d'équations pour trouver ces paramètres. Par contre, ceux-ci possèdent des interdépendances non linéaires, ce qui implique qu'on pourrait évaluer moins de paramètres, à la condition d'utiliser une méthode non linéaire.

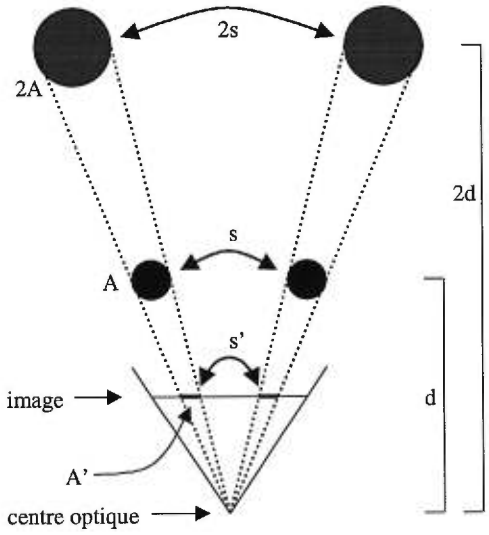

Figure 2.4. Ambiguïté de profondeur. Un objet de taille A, à une profondeur d, projette la même image A' qu'un objet de taille $2A$ à une profondeur $2d$. De même, la projection s' d'un déplacement s, à une profondeur d, est identique à celui d'un déplacement $2s$ à une profondeur $2d$.

Si la matrice $\mathbf{W}_{ba}$ est séparée en paramètres internes et externes, et que les paramètres internes sont connus, il faut alors évaluer une rotation et une translation. La rotation 3D se compose d'un axe de rotation (deux paramètres) et de l'angle de rotation (un paramètre). La translation 3D ne contient en fait que deux paramètres susceptibles d'être évalués. En effet, il est impossible de récupérer la magnitude de la translation à partir d'images projetées seulement. C'est l'ambiguïté de profondeur (*depth ambiguity*), illustrée à la Figure 2.4, qui montre qu'un changement d'échelle n'entraîne aucun changement de l'image projetée. Ainsi, le nombre total de paramètres à évaluer pour la rotation et la translation se monte à cinq.

Il est aussi possible d'évaluer la matrice fondamentale (voir section 2.3) qui est un *condensé* de la matrice $\mathbf{W}_{ba}$ (ou de son inverse $\mathbf{W}_{ab}$). Elle présente huit inconnues puisque c'est une matrice 3×3 moins une inconnue en raison de l'invariabilité aux changements d'échelles, plutôt que les onze inconnues de $\mathbf{W}_{ab}$. Par contre, ces

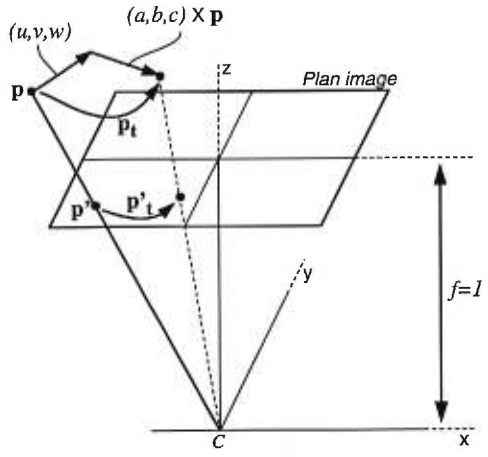

Figure 2.5. Modèle du mouvement de caméra. Le vecteur $\omega = (a, b, c)$ représente l'axe de rotation et la grandeur de la rotation. Le vecteur $\mathbf{t} = (u, v, w)$ représente la vitesse de translation. La vitesse $\mathbf{p}_t$, une fois projetée, devient $\mathbf{p}'_t$.

paramètres sont inextricablement liés et ne permettent pas de retrouver directement $\mathbf{W}_{ba}$.

2.5 Le mouvement de caméra considéré comme vitesse

Il est possible d'élaborer une variante du modèle du mouvement de caméra, plus adaptée au contexte des méthodes différentielles (voir [37]). Plutôt que de considérer le déplacement de la caméra, on parlera de vitesse de la caméra. Cette vitesse s'exprime comme la dérivée temporelle de la position de la caméra. Soit un point $\mathbf{p} = (x, y, z)$ et sa projection $\mathbf{p}' = (x', y', 1)$ (voir section 2.1). Comme le montre la Figure 2.5, le mouvement du point $\mathbf{p}$ est décrit par sa vitesse translationnelle $\mathbf{t} = (u, v, w)$ et sa vitesse rotationnelle $\omega = (a, b, c)$. Le vecteur ω représente l'axe de rotation alors que sa norme $\|\omega\|$ représente la grandeur de la rotation autour de cet axe.

La vitesse du point $\mathbf{p}$ s'exprime alors par sa dérivée dans le temps[1]

$$\mathbf{p}_t = -\mathbf{t} - \omega \times \mathbf{p} \tag{2.20}$$

qui, une fois projetée sur l'image, donne lieu au champ de vitesse (*motion field*)

$$\begin{aligned} \mathbf{p}'_t &= \frac{d\mathbf{p}'}{dt} = \frac{d}{dt}\left(\frac{\mathbf{p}}{\mathbf{p}\cdot\hat{\mathbf{z}}}\right) = \frac{\mathbf{p}_t(\mathbf{p}\cdot\hat{\mathbf{z}}) - (\mathbf{p}_t\cdot\hat{\mathbf{z}})\mathbf{p}}{(\mathbf{p}\cdot\hat{\mathbf{z}})^2} \\ &= \frac{\hat{\mathbf{z}}\times(\mathbf{p}_t\times\mathbf{p}')}{\mathbf{p}\cdot\hat{\mathbf{z}}} \\ &= -\hat{\mathbf{z}}\times\left[\mathbf{p}'\times\left(\mathbf{p}'\times\omega - \frac{\mathbf{t}}{\mathbf{p}\cdot\hat{\mathbf{z}}}\right)\right] \end{aligned} \tag{2.21}$$

où $\hat{\mathbf{z}}$ est le vecteur unitaire dirigé selon l'axe z, qui établit la relation entre le mouvement d'un point et le mouvement de sa projection dans l'image. Au chapitre qui suit, cette relation sera utilisée en conjonction avec la dérivée totale de l'intensité de l'image (équation 3.1, section 3.2.1) pour évaluer $\mathbf{t}$ et ω.

[1] Les signes "–" dans l'équation 2.20 indiquent que les points de l'image se déplacent toujours dans le sens inverse du mouvement de la caméra.

Chapitre 3

INTRODUCTION À LA CALIBRATION DE CAMÉRA

Ce chapitre s'intéresse au problème de la détermination du mouvement d'une caméra dans un environnement à partir de deux images prises à des instants différents. Notons que ce problème équivaut à celui de trouver la position relative entre deux caméras fixes, ou encore à celui d'identifier le mouvement d'un seul objet rigide qui passe devant une caméra fixe. Ainsi, conformément à ce qui a été exposé au chapitre 2, on cherche seulement à déterminer les paramètres externes des caméras, c'est-à-dire la position et orientation des caméras dans le monde, plutôt que les paramètres internes, qui sont assumés connus.

La détermination du mouvement de caméra est une première étape essentielle pour évaluer la structure des objets de la scène. Alors que la plupart des méthodes proposent d'estimer simultanément le mouvement de la caméra et la structure de la scène, notre approche ne permet d'obtenir que le mouvement, sans la structure. Cette apparente lacune est en fait compensée par une robustesse et une précision plus grandes. Une fois le mouvement de caméra connu, la structure peut être facilement récupérée par une analyse stéréoscopique conventionnelle (voir [20, 68]).

On peut classer grossièrement les différentes approches par le type d'information tirée des images qu'elles utilisent. Certaines utilisent les gradients d'intensité spatiaux ou temporels (voir section 3.2.1); certaines autres utilisent plutôt des points saillants (*feature points*) mis en correspondance (voir section 3.2.2). Le choix de l'information utilisée a un très grand impact sur les performances d'une méthode, au niveau de la tolérance au bruit et aux textures, de la grandeur du mouvement permis, etc. Ce choix de l'information à utiliser est ce qui distingue l'approche proposée dans cette thèse au chapitre suivant. Cette nouvelle méthode d'estimation du mouvement de

caméra, introduite à la section 3.3, sera développée à partir de l'information la plus simple qui puisse être extraite des images : l'intensité des pixels eux-mêmes.

Nous tentons de nous attaquer à la classe des problèmes dits de *structure et mouvement* à partir de deux images, mais sous un angle différent, le *mouvement sans structure*, impliquant que la structure n'est ni requise ni estimée par cette approche. Un ensemble de critères sera défini pour établir clairement quels types d'information et quels types de traitements peuvent être inscrits dans cette nouvelle classe.

Nous prévoyons que le fait de nous écarter de la structure entraînera un gain de robustesse et de précision. En particulier, le problème d'estimation du mouvement sera posé dans le contexte de mouvements arbitrairement grands, d'images présentant des textures complexes et corrompues par le bruit.

Les approches *structure et mouvement* tirent l'information des images sous la forme de gradients d'intensité ou de points de correspondance. Or, les gradients sont très sensibles au bruit et sont inutilisables dans les cas de grands déplacements. Semblablement, les points de correspondance sont sensibles au bruit et sont très peu fiables en présence de textures complexes. Nous choisissons donc de ne pas utiliser de gradients d'intensité ou de points de correspondance.

Pour satisfaire à cette restriction, notre approche utilisera plutôt des mesures statistiques sur les intensités des pixels. Lorsqu'elles sont effectuées le long de droites épipolaires correspondantes, ces mesures quantifient ce qu'on appelle l'*alignement épipolaire* et peuvent permettre d'estimer le mouvement de la caméra.

Pour une géométrie épipolaire bien alignée (c'est-à-dire correspondant au mouvement réel de la caméra), la mesure est élaborée de façon à être invariante à la profondeur, c'est-à-dire à la structure de la scène. Par exemple, la similarité entre les histogrammes des intensités le long des droites épipolaires constitue une telle mesure, si on suppose (1) que la contrainte d'intensité constante (*constant brightness constraint*) est respectée et (2) que les réflexions spéculaires et les occlusions ne sont pas significatives.

De plus, il sera démontré que la différence entre deux histogrammes diminue régulièrement avec le degré d'*alignement* de la géométrie épipolaire associée au mouvement de caméra. Cette propriété est dépendante de la corrélation spatiale présente dans les images, et son applicabilité sera démontrée pour une vaste gamme d'images.

Pour permettre de mieux comprendre l'impact de la corrélation spatiale, un modèle de texture simple a été développé. La texture de l'image est représentée globalement par la distribution des intensités au voisinage d'un point.

Une attention particulière sera accordée à la susceptibilité de la nouvelle approche aux cas où la contrainte d'intensité constante est inapplicable et où les images sont mal conditionnées (textures particulières, images non stationnaires, etc.), ce qui rend inutilisables les modèles probabilistes.

À partir de ces observations, un nouvel algorithme pour l'estimation de la rotation et de la translation de caméra sera développé. Il sera formalisé sous forme d'une recherche d'un minimum d'erreur correspondant à l'alignement maximal dans un espace à cinq dimensions, trois pour la rotation et deux pour la translation, en conformité avec la section 2.4.

3.1 Contraintes et hypothèses

Les différents modèles utilisés pour l'analyse du mouvement de caméra se basent sur un ensemble d'hypothèses et de contraintes qui ont souvent un grand impact sur les performances ou la généralité des solutions.

- Hypothèse d'objets rigides

  Il est assumé que les objets formant la scène sont rigides, c'est-à-dire qu'ils ne subissent aucune déformation. Les déplacements des points de l'image sont donc uniquement dus au déplacement de l'objet, ce qui rend possible l'évaluation de ce déplacement.

- Hypothèse d'un seul mouvement global

  On assume généralement que la caméra se déplace autour d'une scène immobile ou que la caméra est immobile en face d'un seul objet qui se déplace. La présence de mouvements multiples (par exemple plusieurs automobiles à une intersection) complique énormément le problème puisqu'ils rendent inconsistantes les équations du mouvement. Il faudrait alors procéder à une étape préliminaire de séparation des différents mouvements (*motion segmentation*), tâche très difficile à réaliser (voir [10, 41]).

- Hypothèse de paramètres internes de caméra constants

  Les paramètres internes de la caméra sont la distance focale, le ratio horizontal/vertical, le centre de l'image, et parfois aussi le cisaillement (*shearing*) et la distorsion radiale. Ces paramètres sont définis par l'ajustement de la caméra, et on suppose qu'ils sont connus et ne varient pas pour différents angles de vue. Cette contrainte est respectée si une seule caméra se déplace dans la scène, mais elle n'est pas toujours respectée lorsque plusieurs caméras différentes sont utilisées simultanément pour obtenir la séquence.

- Contrainte d'intensité constante

  Cette contrainte exige que l'intensité observée d'un point du monde soit conservée lorsqu'il se déplace, ou, de façon équivalente, que la variation d'intensité d'un point image lors du déplacement de la caméra soit causée uniquement par ce déplacement. Cette contrainte est rarement respectée en pratique. Elle reste cependant valide si les variations d'intensité reliées au mouvement de caméra sont beaucoup plus grandes que les variations dues à d'autres facteurs, comme les conditions d'éclairage ou les réflexions spéculaires. Une discussion détaillée de cette contrainte est donnée dans l'appendice A de Horn et Weldon [37].

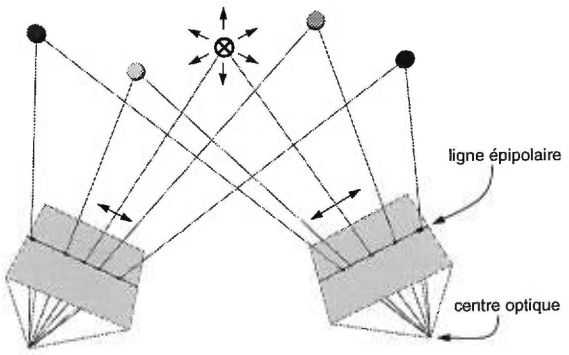

Figure 3.1. Contrainte d'ordre. L'ordre des points le long de droites épipolaires est conservé d'une image à l'autre. La position du point ⊗, par exemple, est limitée par la projection de ses voisins immédiats.

- Hypothèse d'unicité (ou d'opacité)

  Les objets sont presque toujours assumés opaques. Ceci garantit qu'un point d'une image ne peut correspondre à plus d'un point dans l'autre image. L'ambiguïté qui engendre les correspondances multiples est donc éliminée. Il est toujours possible qu'un point ne correspond à aucun autre point s'il subit une occlusion.

- Hypothèse de conservation de l'ordre (*Ordering constraint*)

  Selon cette hypothèse, l'ordre des points est conservé le long des droites épipolaires correspondantes entre deux images, sauf peut-être sur les contours des objets où l'on observe des discontinuités de profondeur. On assume essentiellement que les objets sont réguliers et ne se portent pas occlusion entre eux (voir figure 3.1). Une description détaillée de cette contrainte est donnée dans [68].

3.2 Revue des méthodes existantes

Un grand nombre de méthodes ont été proposées pour estimer le mouvement de caméra à partir d'une séquence de deux images, assumant un déplacement rigide de

la scène. On peut classer ces méthodes selon le type d'information qu'elles utilisent.

3.2.1 Méthodes utilisant les gradients d'intensité

Ces méthodes basent l'estimation du mouvement sur les dérivées de l'intensité des images. D'une part, la relation entre le mouvement de la caméra et la vitesse de déplacement des points dans l'image est donnée à l'équation 2.21. D'autre part, la contrainte d'intensité constante (voir section 3.1) permet de relier cette vitesse $\mathbf{p}_t$ aux variations d'intensité de l'image I. En effet, lorsque cette contrainte est observée, on a

$$I(x, y, t) - I(x + \delta x, y + \delta y, t + \delta t) = 0$$

pour un point (x, y) de l'image I au temps t qui se déplace de $(\delta x, \delta y)$ dans un temps δt. L'expansion en série de Taylor pour x, y, t est

$$I(x, y, t) - \left[I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + \text{t.o.s.} \right] = 0$$

où t.o.s. représente les termes d'ordre supérieur qui sont considérés négligeables. Après une simplification et division par δt, la limite lorsque δt tend vers 0 donne la relation simple (voir [38])

$$\mathbf{I}_s \cdot \mathbf{p}_t = -\mathbf{I}_t \tag{3.1}$$

où $\mathbf{I}_s$ est le vecteur $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, 0)$ représentant le gradient spatial, et où $\mathbf{I}_t$ représente le gradient temporel $\frac{\partial I}{\partial t}$.

Seule, l'équation 3.1 permet de récupérer seulement la composante normale de la vitesse $\mathbf{p}_t$, orientée dans le sens gradient spatial, pour constituer le *flux normal*. C'est une manifestation de *l'effet d'ouverture* (*aperture problem*). L'estimation de la composante manquante de la vitesse donne lieu au calcul du *flux optique*. Celui-ci est difficile à calculer (voir [4]) à cause du manque de contraintes. On contraint donc artificiellement le problème, par exemple en exigeant que la vitesse varie partout en douceur, ce qui a l'inconvénient de diminuer la fiabilité des résultats.

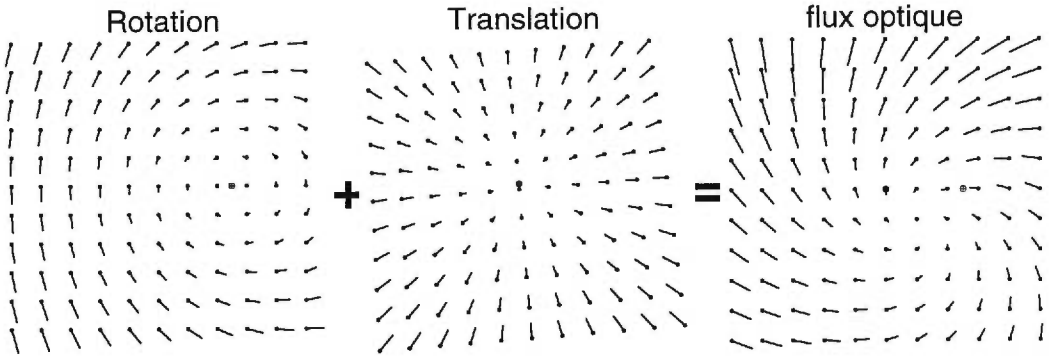

Figure 3.2. Exemple de flux optique décomposé en une somme de composantes rotationnelle et translationnelle.

De plus, le calcul des dérivées de I est un problème délicat (voir [37]). En particulier, les dérivées temporelles sont erronées lorsque les images sont trop espacées parce qu'elles ne satisfont plus le théorème d'échantillonnage (*sampling theorem*).

Flux optique

Plusieurs méthodes d'estimation du mouvement de caméra dépendent de la disponibilité du flux optique ([36, 40, 43]). Elles tentent essentiellement de calculer $\mathbf{t}$ et ω de l'équation 2.21 en assumant $\mathbf{p}_t$ (le flux optique) connu. Leur précision est profondément influencée par la précision et la densité du flux optique. Plusieurs problèmes inhérents au calcul du flux optique (effet d'ouverture, vitesse limitée, etc.) semblent suggérer que les erreurs ne peuvent pas être diminuées jusqu'à un niveau négligeable (voir [4]).

Dans le cas d'un mouvement rigide, comme celui d'une caméra devant une scène fixe, le flux optique peut être factorisé en deux composantes: les flux rotationnel et translationnel (voir figure 3.2). Une fois effectuée, cette factorisation permet une évaluation triviale du mouvement de caméra.

Sélection de références

L'important article de Horn et Schunk [38] a introduit la première méthode permettant de calculer le flux optique à partir des gradients de l'image.

Une comparaison très détaillée des algorithmes les plus populaires du calcul du flux optique est donnée dans Barron *et al.* [4].

Une analyse de la pertinence de la contrainte d'intensité constante pour le calcul du flux optique est donnée dans Bimbo *et al.* [8]. Une contrainte "étendue" y est proposée pour tenir compte de différents modèles d'illumination.

Dans Heeger et Jepson [36], une méthode basée sur les sous-espaces (*subspace method*) utilise le flux optique pour estimer le mouvement de la caméra. Le problème est décomposé en deux sous-problèmes. Tout d'abord, la translation est évaluée directement à partir du flux optique, sans dépendre de la rotation ou de la structure de la scène. Ensuite, cet estimé est utilisé conjointement au flux optique pour estimer la rotation. L'estimation de la translation peut se faire par une recherche dans le sous-espace des translations possibles ou encore en approximant le problème par un système linéaire (voir [43]). Les résultats montrent que pour une caméra dont l'angle de vue est de 60°, et une erreur de 10% sur l'évaluation du flux optique, l'erreur sur la direction de la translation est de 5°. Pour un angle de vue de 20°, l'erreur passe à 22°. Il apparaît donc que la précision de la translation obtenue dépend crucialement de l'effet de distorsion perspective de la caméra. En effet, on sait qu'un grand angle de vue augmente l'effet de perspective alors qu'un faible angle de vue réduit fortement cet effet.

Dans le cas où un système de caméra peut suivre un point particulier de la scène (*fixation*), l'article Raviv et Herman [65] propose une méthode pour évaluer la direction du mouvement basée sur les *lignes de flux constant* et en particulier les *lignes de flux nul.* En effet, lorsque la caméra suit un point, le flux optique en ce point sera toujours nul. Ce fait est utilisé pour donner de la simplicité et de la robustesse à la

méthode.

Flux normal

Les méthodes qui utilisent directement le flux normal sont généralement considérées comme des *méthodes directes.* Elles sont *directes* parce qu'elles ne requièrent pas d'étape intermédiaire de calcul du flux optique. Elles procèdent à la substitution de l'équation 2.21 dans l'équation 3.1 pour obtenir

$$\mathbf{v} \cdot \omega + \frac{\mathbf{s} \cdot \mathbf{t}}{p \cdot \hat{\mathbf{z}}} = -I_t \tag{3.2}$$

où

$$\mathbf{s} = (I_s \times \hat{\mathbf{z}}) \times \mathbf{p} \qquad \text{et} \qquad \mathbf{v} = -\mathbf{s} \times \mathbf{p}$$

proviennent uniquement de l'image et sont indépendantes du mouvement de la caméra. Cette équation ne contient plus $\mathbf{p}_t$ et ne dépend donc plus du flux optique.

Il a souvent été suggéré que la suppression du flux optique augmentait la précision et la robustesse de l'estimation du mouvement de caméra. En fait, l'équation 3.2 reste difficile à solutionner. Ainsi, certaines de ces méthodes ([33]) doivent se résoudre à utiliser les mêmes contraintes artificielles que le calcul du flux optique et souffrent donc de problèmes similaires (voir section 3.2.1). Certaines autres ([1, 37, 78]) ne traitent que les cas particuliers, dans lesquels sont connues d'avance soit la rotation, soit la translation, soit la profondeur.

Sélection de références

Dans Horn et Weldon [37], une méthode utilisant le flux normal est proposée pour estimer la rotation avec une translation connue, ou encore la translation avec une rotation connue.

Dans Sinclair *et al.* [78], l'estimation de la translation est obtenue à partir du flux normal, assumant une rotation connue. On y investigue la relation entre l'erreur sur

l'estimé de la rotation et la précision de la translation obtenue. Cet article est basé sur Horn et Weldon [37].

Dans Aloimonos et Duric [1], la translation est estimée à partir du flux normal en utilisant un système de vote. Comme Sinclair *et al.* [78], il généralise le cas de la translation pure de Horn et Weldon [37], mais en assumant une rotation d'amplitude limitée (*bounded rotation*).

Dans Fermuller [25], le flux normal est utilisé sans aucune hypothèse particulière. La solution apparaît sous la forme d'une recherche de l'espace des paramètres du mouvement de caméra (rotation et translation) utilisant seulement le signe des vecteurs composant le flux normal pour contraindre et éliminer les solutions impossibles. Selon cette approche, on réussit à obtenir une estimation très robuste du mouvement de caméra.

La relation entre le mouvement de caméra et le flux optique est décrite dans Gurvits *et al.* [32], qui propose une nouvelle méthode pour calculer la rotation et la translation dans un monde bidimensionnel en utilisant les dérivées des premier et second ordres. On y assume que l'extension à un monde tridimensionnel est simple...

Une méthode itérative utilisant les dérivées des intensités pour l'estimation du mouvement de caméra est proposée dans Hanna [33]. En plus des équations du mouvement, une contrainte supplémentaire est ajoutée en vue de simplifier le problème. Il s'agit de l'hypothèse selon laquelle les objets sont localement planaires, ou encore que les objets sont localement à une profondeur constante. Une approche par pyramide multi-résolution est proposée pour permettre les déplacements plus grands.

3.2.2 Méthodes utilisant les points de correspondance

Ces méthodes basent leur estimation du mouvement sur l'utilisation de points saillants (*feature points*) des deux images mises en correspondance. L'article Huang et Netravali [39] présente une revue très complète du problème de *structure et mouvement* à partir de points de correspondance.

Comme il a été mentionné à la section 2.4, il suffit de cinq paires de points correspondants bien choisis entre deux images pour construire un système d'équations non linéaires pouvant être résolu par des méthodes itératives.

Si l'on dispose de huit points de correspondance, on peut obtenir un système linéaire ([54, 84]), à condition de connaître, ou d'estimer à l'avance, les paramètres internes des caméras. En effet, la matrice essentielle, telle que décrite à la section 2.3, présente un système linéaire homogène à neuf inconnues (voir équation 2.19) qui requiert huit paires de points pour obtenir une solution unique. Une fois la matrice $\mathbf{E}$ connue, il est possible d'en extraire directement la rotation $\mathbf{R}$ et la translation $\mathbf{t}$ (voir [24, 84]). Si plus de huit paires de points sont disponibles, on peut utiliser une méthode *moindre carrés* et augmenter la robustesse. La précision de cette approche est difficile à évaluer; elle dépend du choix du nombre de points, de leur position et du niveau d'erreur de la mise en correspondance (voir [35, 39])

Si les paramètres internes (distance focale, ratio, centre de l'image, cisaillement) sont inconnus, alors seule la matrice fondamentale (voir équation 2.18) peut être évaluée, mais elle incorpore, sans qu'on puisse isoler chacun d'eux, la rotation et la translation de la caméra ainsi que ces paramètres internes (voir aussi [24]).

Il est important de mentionner que toutes ces méthodes peuvent presque toujours obtenir facilement la solution exacte lorsque les positions des points mis en correspondance sont exactes. Mais en pratique, il s'avère extrêmement difficile d'évaluer ces positions avec exactitude et de mettre en correspondance les points sans erreur, ce qui affecte fortement la performance.

Sélection de références

Dans Tomasi et Shi [83], on utilise l'angle entre les points de correspondance plutôt que les déplacements. Ceci confère une forme d'invariance à la rotation. Une incertitude de ± 1 pixel sur la position des points de correspondance cause généralement une erreur d'environ 9 degrés sur la direction de la translation.

Une méthode utilisant les filtres de Kalman est proposée dans Chandrashekhar *et al.* [13]. Cette méthode requiert un grand nombre d'images (au moins 15) pour que le filtre converge. Le mouvement est assumé constant, tant en rotation qu'en translation. Une variation du mouvement est "absorbée" par le filtre de Kalman en tant que bruit. Le degré de variation de mouvement permis peut donc être contrôlé en modifiant la tolérance au bruit.

Extraction et suivi des points saillants

L'extraction et le suivi de points saillants n'étant pas des notions essentielles dans la présente thèse, nous nous contenterons ici d'une brève description. Une description plus complète est donnée dans Cox [18].

Avant la mise en correspondance, il est nécessaire de trouver des points saillants (*feature points*) dans les deux images. L'extraction automatique de ces points présente un défi de taille. En effet, la densité et la qualité des points sont intimement liées à la complexité de l'image, ce qui a pour conséquence de rendre les résultats imprévisibles.

Il est difficile de trouver un nombre de points idéal. Avec trop peu de points, la précision du mouvement de caméra sera réduite. Avec trop de points, les ambiguïtés lors de la mise en correspondance entraînent aussi une baisse de la précision.

Le cas des occlusions est lui aussi difficile. En effet, un point visible dans une seule image va presque toujours être mal apparié et augmentera l'erreur.

Sélection de références

Une représentation par ondelette de Gabor (*Gabor wavelet*) est utilisée pour l'extraction de points saillants dans Chandrashekhar *et al.* [13].

Dans Mousavi et Schalkoff [57], une architecture basée sur les réseaux de neurones est proposée pour solutionner le problème de mise en correspondance. Plutôt que d'utiliser des points saillants, cette méthode met en correspondance des contours

extraits des images.

3.2.3 Autres méthodes

On aura remarqué que toutes les méthodes décrites jusqu'ici utilisent soit les gradients d'intensité, soit les points saillants mis en correspondance.

La méthode appelée *plan+parallaxe* appartient aux deux catégories (voir [49, 72]). Cette méthode a pour étape préliminaire la localisation et la mise en correspondance des points appartenant tous à une même surface plane de la scène. Le champ de mouvement associé à ces points permet d'éliminer la composante rotationelle du flux optique. En fait, on élimine du même coup une partie du champ translationel, ce qui implique que le champ résiduel représente la profondeur (parallaxe) par rapport au plan de départ.

La sélection de points saillants appartenant à un même surface plane d'une scène et leur mise en correspondance constituent un problème tellement difficile qu'il requiert presque toujours une assistance manuelle dans le repérage et le suivi des points (voir [49]).

La nouvelle méthode que nous proposons au chapitre suivant fait partie des *autres méthodes* puisqu'elle n'utilise ni les points saillants ni les gradients d'intensité.

3.2.4 Analyse de deux images versus plusieurs images

Plusieurs raisons sont citées dans Franzen [26] pour justifier l'usage de plus de deux images pour l'analyse du mouvement de la caméra:

- Augmenter la robustesse de la solution.
- Permettre de récupérer la structure et le mouvement avec moins de points saillants à extraire et à mettre en correspondance.

- Permettre d'estimer des dérivées d'ordre supérieur et d'augmenter la précision des dérivées temporelles.

Tous ces bénéfices sont subordonnés à une condition : que le mouvement de caméra varie en douceur d'une image à l'autre (voir [13, 61]). S'il arrive que la caméra présente des déplacements complètement irréguliers le long de la séquence, alors la majorité des gains par rapport à l'analyse de deux images sont perdus.

3.3 Une nouvelle approche pour l'estimation du mouvement

Le chapitre suivant propose une nouvelle approche *mouvement sans structure* pour évaluer le mouvement de caméra à partir de deux images sans utilisation ni estimation de la structure de la scène. Son originalité réside entre autres dans le type d'information utilisée. Pour éliminer la dépendance par rapport aux étapes intermédiaires de calcul de flux optique ou de mise en correspondance, il est proposé d'utiliser directement les intensités des pixels sans aucun traitement a priori. Cette information est utilisée dans une *mesure d'alignement* de la caméra. Comme cette mesure est conçue pour présenter un minimum lorsque l'alignement est atteint, il suffit d'établir une recherche dans l'espace à cinq dimensions des paramètres du mouvement de caméra (rotation et translation) pour trouver cet alignement et par le fait même le mouvement de caméra.

Nous avons publié deux articles [22, 23] liés à cette approche, préalablement à l'article [69] qui constitue notre chapitre 4.

Dans [22], la mesure proposée est la comparaison des histogrammes d'intensité de paires de droites épipolaires correspondantes. Cette mesure offre la propriété d'être invariante à la profondeur, c'est-à-dire invariante à la structure de la scène. En effet, puisque tous les points d'une droite épipolaire doivent se retrouver quelque part sur la droite épipolaire correspondante dans l'autre image, il est évident que l'histogramme de chacune de ces deux droites sont identiques si les droites correspondent au vrai

mouvement de la caméra. Ceci est possible parce que l'histogramme ne tient nullement compte de la position des points mais seulement de leurs intensités. C'est de cette propriété que découle le concept de *mouvement sans structure*. En pratique, on doit aussi assumer que les occlusions sont négligeables et que les points conservent la même intensité lorsqu'ils se déplacent. Cet article démontre la faisabilité et le potentiel de l'approche *mouvement sans structure*, en montrant que malgré l'utilisation d'une mesure d'alignement très simple, des résultats robustes et précis peuvent être obtenus pour une vaste gamme d'images.

Dans [23], une mesure plus puissante mais toujours apparentée aux histogrammes est proposée. De plus, un modèle probabiliste décrivant le comportement des intensités au voisinage d'un point est utilisé pour démontrer certaines propriétés importantes:

- Pour une rotation connue et un voisinage bien conditionné, la mesure d'alignement ne possède qu'un seul minimum qui correspond à la translation de la caméra.

- La propriété précédente s'applique symétriquement au cas de la recherche de la rotation, assumant une translation connue.

Par *voisinage bien conditionné*, on entend que le comportement des intensités au voisinage d'un point présente une certaine régularité. Ceci a pour effet d'éliminer les textures extrêmes, c'est-à-dire une corrélation trop grande, comme celle des surfaces unies ou une corrélation trop faible, comme celle des surfaces de très faible contraste (rapport signal/bruit très faible). C'est la corrélation présente dans les images qui confère à la mesure d'alignement sa force et son utilité.

Les sections qui suivent reprennent dans les grandes lignes le contenu des deux articles et introduisent par la même occasion celui du chapitre suivant.

3.3.1 Modélisation

L'approche proposée procède par une recherche du minimum d'une mesure *d'alignement* dans l'espace à cinq dimensions des paramètres du mouvement de la caméra. Le succès (ou l'échec) d'une telle recherche dépend de la régularité de la fonction à minimiser et du nombre de minimum locaux. Comme il a été mentionné précédemment, un des résultats importants a été de démontrer que si les images respectent certains critères au niveau de leur texture, alors il n'y a qu'un seul minimum local, qui est en fait le minimum global. Il n'y a donc pas de danger de rester coincé dans un minimum local. Ce résultat est valide pour l'estimation de la rotation en connaissant la translation, ou symétriquement pour l'estimation de la translation en connaissant la rotation.

L'élément central de cette démonstration réside dans la modélisation du comportement des intensités au voisinage d'un point. En effet, il est bien connu que les intensités de pixels voisins ont tendance à être similaires et que cette similarité tend à diminuer avec la distance entre ces pixels. Nous proposons donc de mesurer globalement la distribution de la différence d'intensité entre des paires de pixels en fonction de leur distance δ. En pratique, on observe généralement que la moyenne de ces distributions est toujours zéro et que leurs variances augmentent avec la distance. Un exemple de courbes des variances en fonction de la distance est donné à la figure 3.3, pour quelques images de la base de données JISCT [9].

3.3.2 Contraintes et hypothèses

En plus des hypothèses et contraintes couramment utilisées en analyse du mouvement de caméra (voir section 3.1), la nouvelle méthode fait usage de contraintes reliées à la texture.

En particulier, la variance de la distribution des différences d'intensité doit augmenter de façon monotone en fonction de la distance entre les points. Ainsi, les

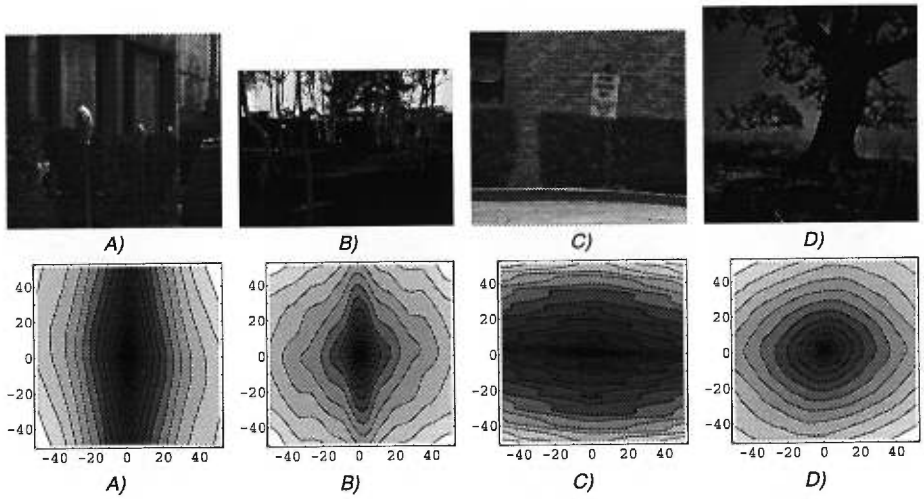

Figure 3.3. Variance au voisinage d'un point pour quatre images de la base de données JISCT. A) parking meter, B) birch, C) shrub, D) tree et les variances $\sigma^2(\delta)$ correspondantes. Les distances le long des axes sont en pixels. Les régions sombres dénotent des variances faibles.

textures trop régulières, par exemple un plancher de tuiles, présentent une variance qui oscille (donc non monotone) et sont reconnues comme étant difficiles. Dans ces cas, on ne peut garantir un seul minimum local, et la solution obtenue sera probablement erronée.

3.3.3 Expérimentation et Résultats

Plusieurs expériences ont été tentées sur des images naturelles et les courbes montrant l'erreur en fonction des paramètres du mouvement de caméra présentent toutes un minimum correspondant à la solution attendue, autant pour la rotation que pour la translation.

Même si notre méthode exige présentement une estimation soit de la rotation soit de la translation, elle démontre cependant une bonne tolérance aux erreurs d'estimation (jusqu'à $\pm 10°$).

Une séquence calibrée prise à partir d'un hélicoptère a permis d'établir que la

précision de la rotation est d'environ $\pm 1°$, même lorsque l'estimé de la translation est erroné de 5°.

Sélection de références

Nous ajoutons, pour compléter cette revue, une sélection d'autres articles pertinents à cette nouvelle approche de l'estimation du mouvement de caméra.

Dans [40, 80] une méthode est proposée pour évaluer le mouvement de caméra. Elle utilise le flux optique, mais calcule la translation en cherchant le point d'expansion par la minimisation d'une *fonction de mérite* sur l'espace de tous les points d'expansion possibles. Dans Hummel *et al.* [40], la recherche se restreint aux points d'expansion visibles dans l'image ou situés à l'infini. Plutôt que de calculer cette fonction à chaque point de cet espace, six points sont sélectionnés pour définir une surface d'erreur quadratique. La précision de l'estimation de la translation est d'environ 10°.

Dans Geman et Manbeck [28], une image est modélisée par un champ aléatoire de Markov (*Markov random field*). Ce modèle a comme avantage de modéliser explicitement le bruit. Pour une image donnée, les probabilités optimales (paramètres du modèle) sont évaluées par programmation dynamique. La robustesse de cette méthode est très bonne et démontre la puissance de modélisation des champs aléatoires de Markov. Il est envisagé que cette approche serait utile pour modéliser les textures plus efficacement.

Dans Lavalle *et al.* [51], une bonne introduction à l'approche Bayesienne est fournie avec un exemple d'application à la segmentation de texture. La modélisation des textures proposée est similaire à celle utilisée dans notre méthode.

Chapitre 4

MOTION WITHOUT STRUCTURE

Cet article [69] a été publié comme l'indique la référence bibliographique

> Sébastien Roy et Ingemar J. Cox, Motion Without Structure, dans *International Conference on Pattern Recognition (ICPR'96)*, Vienne, Autriche, Août 1996, vol. 1, pages 728-734.

Gagnant du prix du meilleur article étudiant, cet article est présenté ici dans sa version originale.

Abstract

We propose a new paradigm, motion without structure, for determining the ego-motion between two frames. It is best suited for cases where reliable feature point correspondence is difficult, or for cases where the expected camera motion is large. The problem is posed as a five-dimensional search over the space of possible motions during which the structural information present in the two views is neither implicitly or explicitly used or estimated.

To accomplish this search, a cost function is devised that measures the relative likelihood of each hypothesized motion. This cost function is invariant to the structure present in the scene. An analysis of the global scene statistics present in an image, together with the geometry of epipolar misalignment, suggests a measure based on the sum of squared differences between pixels in the first image and their corresponding epipolar line segments in the second image.

The measure relies on a simple statistical characteristic of neighboring image intensity levels. Specifically, that the variance of intensity differences between two

arbitrary points in an image is a monotonically increasing symmetrical function of the distance between the two points. This assumption is almost always true, though the size of the neighborhood over which the monotonic dependency holds varies from image to image. This range determines the maximum permissible motion between two frames, which can be quite large.

Experiments with both outdoor scenes and an indoor calibrated sequence achieve very good accuracy (less then 1 pixel image displacement error) and robustness to noise.

4.1 Introduction

Much work has been done on trying to recover camera motion (i.e. ego-motion) parameters from image pairs. In almost all cases, either optical flow or feature point correspondences are used as the initial measurements. In the first case, some inherent problems (aperture, large motions, etc.) related to optical flow computation, suggest that errors can never be lowered to a negligible level (see [4, 38, 43, 80]). Even methods using the intensity derivatives directly or normal flow (see [1, 25, 37, 58, 78, 80, 83]), suffer from high noise sensitivity. For feature-based methods, the reliable selection and tracking of meaningful feature points is generally very difficult, see [18, 50, 82, 83].

All prior methods of ego-motion implicitly or explicitly determine the structure present in the scene. For example, while feature based methods compute a motion estimate directly, the structure is implicitly available given the feature correspondences. Direct methods explicitly estimate both the ego-motion and structure, typically in an iterative fashion, refining first the motion, and then the structure estimates. Thus, good motion estimation appears to require good structure estimation (or at least point correspondence estimation). In contrast, we propose a paradigm that we call *motion without structure*. Under this paradigm, the recovery of ego-motion is independent of any structure or correspondence estimation. The benefit is that there are

only five unknown motion parameters to be estimated. As such, we expect that the approach should be both robust and accurate. The experimental results support this.

The algorithm relies on statistically modeling the image behavior in the neighborhood of a point, as discussed in Section 4.2.1. This model is then used to estimate the likelihood of an assumed camera motion. In Cox and Roy [22], we proposed using the difference between histograms computed along assumed correspondence epipolar lines as a likelihood function. This statistical measure is very effective in determining the rotational component of ego-motion, but is not always a reliable measure of the likelihood of a translational motion. Consequently, we proposed in Cox and Roy [23] a likelihood measure based on the sum of sums of squared differences between pixels in one image and their hypothesized corresponding line segments in the other image that is a reliable estimate of either the rotational or translational components of motion. This measure is detailed in Section 4.2.2.

Determining the true motion is then accomplished by searching for the maximum likelihood estimate over the space of translations and rotations. The search is straightforward since we show in Section 4.2.3 that the function to be minimized has only one minimum (which is the solution), provided the image is well behaved, i.e. the variance between neighboring intensity points increases monotonically and symetrically with the distance between the points. In previous work [23], the sub-problems of finding rotation or translation when the other component of motion is known was shown to be solvable by locating the single local minimum, which is also the global minimum. This paper extends these results and considers the full motion case when *both* rotation and translation must be simultaneously estimated. The effect of motion ambiguity (see in [60]) on the accuracy of motion estimation is also discussed.

Section 4.3 presents experimental results from a comprehensive evaluation based on real images of stereoscopic pairs and an indoor calibrated motion sequence.

4.2 Motion estimation as a 5-D search

Our goal is to determine the motion between two frames by a search over the space of possible rotations and translations. The number of parameters to be estimated are three for rotation and two for translation. Only two translational components are needed because the magnitude of the translation cannot be estimated, only its direction (due to the depth-scale ambiguity). The translation is thus assumed to have unit magnitude, and the estimation of translation reduces to determination of the direction of translation on the surface of a unit sphere.[1]

In order for such a search to be possible, a cost function is needed that evaluates the likelihood of an assumed motion. Essential characteristics of such a cost function are (1) invariance to structure in the scene, (2) a well-defined global minimum at the correct motion estimate, and (3) no local minima in the neighborhood of the correct motion.

In Section 4.2.2, we describe one such structure-invariant cost function, based on a simple statistical model of local intensity variation (see Section 4.2.1), that possesses these desired properties.

4.2.1 A statistical model of image intensities

A simple statistical model is used to represent image behavior around a point. Consider the intensity distribution in the neighborhood of a given image point $\mathbf{p}$. We are interested in the probability of differences in intensity between point $\mathbf{p} + \boldsymbol{\delta}$ and $\mathbf{p}$, conditioned on the displacement $\boldsymbol{\delta}$ between the two points.

This property is intuitively related to the correlation present in a scene. For a given image, we can evaluate the parameters of the distributions, namely $\sigma^2(\boldsymbol{\delta})$, for all possible displacements $\boldsymbol{\delta}$.

---

[1] Consequently, in the experimental section, the translational error is recorded in degrees over the unit sphere.

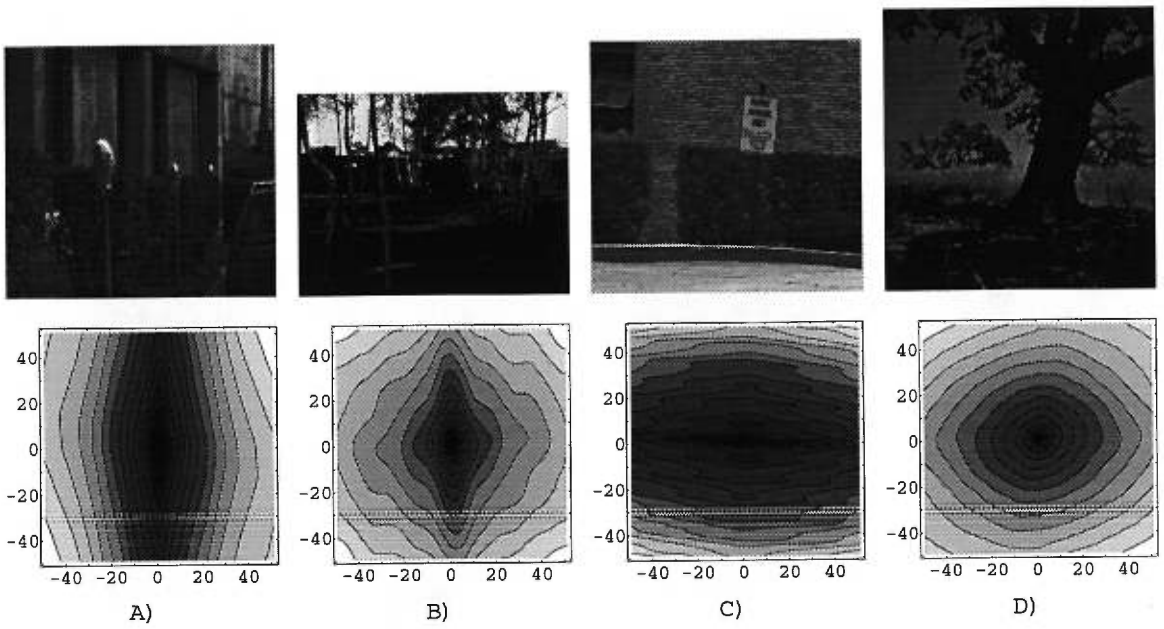

Figure 4.1. JISCT image database. The four images A) *Parking meter*, B) *Birch*, C) *Shrub*, D) *Tree* are shown on top of their variance functions $\sigma^2(\boldsymbol{\delta})$. Distances along the axis are in pixels. Darker points have smaller variance.

Example of these variance functions are shown in Figure 4.1 for a neighborhood of 50 pixels. The mean of the distributions is not shown here since it is always very close to 0. The variance functions increase approximately monotonically with distance, with a single minimum centered at $\boldsymbol{\delta} = (0, 0)$. This property is exploited to derive the likelihood measure in Section 4.2.2. Note that while the relationship between variance and distance is monotonically increasing, it is not always symmetrical, indicating that intensities are more correlated in certain directions. It is straightforward to find a mapping between two monotonically increasing functions to restore symmetry. This mapping will be applied to correct pixel value differences in the cost function.

Our experimental observations indicate that most natural images are usually well-behaved. We define a *well-behaved* image as one that possesses a monotonically increasing variance function. Only images that contain repetitive textures or those

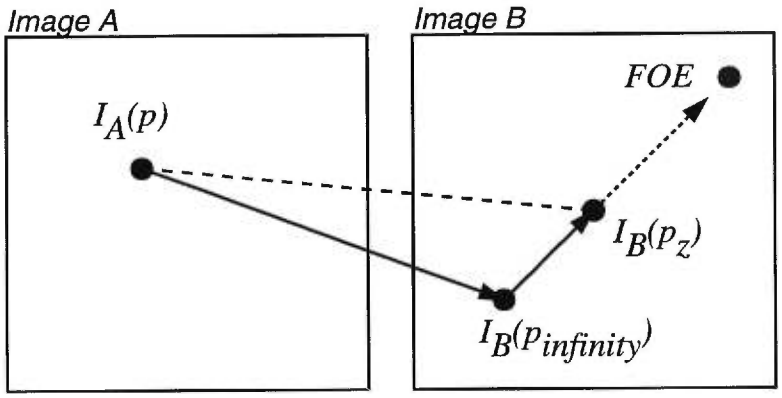

Figure 4.2. Basic geometry for known rotation. For a given $I_A(\mathbf{p})$, its unknown corresponding point $I_B(\mathbf{p}_z)$ is on the line joining $I_B(\mathbf{p}_\infty)$ and the FOE.

that are highly non-stationary, generally present badly-behaved (i.e. non-monotonic) variance functions. By examining how well-behaved the variance function is, it should be possible to measure how accurate the method is expected to perform.

4.2.2 A Depth-invariant cost function

We wish to evaluate the likelihood of a motion, composed of a rotational and a translational component, to be the true motion of the camera. As shown in Figure 4.2, for a given point $I_A(\mathbf{p})$ in image A and a camera motion, we can compute the matching point $I_B(\mathbf{p}_\infty)$ (the *zero-disparity* point) in image B that corresponds to infinite depth, as well as the *focus of expansion* (FOE). The point $I_B(\mathbf{p}_\infty)$ is related to the rotational component of the motion while the FOE is related to the translational component.

Since we do not know the real depth z of point $I_A(\mathbf{p})$, we can only assume that the actual corresponding point $I_B(\mathbf{p}_z)$ is somewhere in the neighborhood of point $I_B(\mathbf{p}_\infty)$. In fact, it is always located on the line joining the true $I_B(\mathbf{p}_\infty)$ and the true focus of expansion.

For a given camera motion, a line segment, u, of length r_{max} is selected starting at the zero-disparity point $I_B(\mathbf{p}_\infty)$ and oriented toward the FOE. The value of r_{max}

is chosen to reflect the maximum disparity expected. After selecting a number of sample intensity values u_i along the segment u, we define the error measure e_u as

$$e_u = \sum_{i=1}^{n}(u_i - I_B(\mathbf{p}_z))^2 = \sum_{i=1}^{n}(u_i - I_A(\mathbf{p}))^2 \tag{4.1}$$

which will be a minimum when the segment u contains $I_B(\mathbf{p}_z)$. Equation 4.1 can assume that $I_B(\mathbf{p}_z) = I_A(\mathbf{p})$ since these points correspond and therefore should have the same intensity value. To get a global estimate of the likelihood of a motion, we select a number of points $I_A(\mathbf{p}_i)$ and compute the sum

$$S = \sum e_{q_i}$$

of the individual line segment errors e_{q_i} corresponding to each of these points.

The next section will show how this cost function satisfies the requirement enumerated in Section 4.2. It is expected that for well-behaved images, this cost function will exhibit a single minimum at the true camera motion and that a simple search based on gradient descent will be sufficient to find it.

4.2.3 Convergence and smoothness properties

In order to successfully search over the motion space, the cost function must have a well-defined global minimum and few, if any, local minima. Section 4.2.3 shows that for a known rotation, the translational search space features only a single local minimum which is also the global minimum, assuming monotonic and symmetrical image intensity variances. The converse is also demonstrated, that is searching for rotation with known translation.

The preceding discussion assumed that either the translation or rotation was already known. In practice, both must be estimated. We do not have a proof of convergence for this situation and have proceeded with an experimental investigation to determine the utility of the cost function under these circumstances.

A second condition for successful search, is that the region of convergence should be large, to allow easy selection of an initial search point. This region (and the general smoothness of the function) should be derivable from the local image intensity statistics. Qualitatively, it is clear that large and frequent intensity variations do not allow a wide region of convergence (because of ambiguities) while low frequency variations allow for much larger motions.

Existence of a single minimum

In this section we show that for well-behaved images, a single minimum of the error measure e_u of Equation 4.1 is observed when a segment u contains $I_B(\mathbf{p}_z)$ and joins the true zero-disparity point and the true FOE. Since by definition a well-behaved variance function always features a global minimum at $(0,0)$, this condition is enough to ensure that the likelihood function possesses a unique minimum. This is demonstrated next.

Consider a segment u in the neighborhood of $\mathbf{p}_z$, starting at $\mathbf{p}_\infty$, and containing n sample intensities as depicted in Figure 4.3A. Then we can assume that each sample behaves like a random variable u_i with distribution

$$f(u_i) = G_{[I_A(\mathbf{p});\sigma^2(\mathbf{d}_{u_i})]}(u_i)$$

where $G_{[\mu;\sigma^2]}$ is an arbitrary probability distribution and $\mathbf{d}_{u_i}$ is the distance (x,y) from sample u_i to position $\mathbf{p}_z$, the unknown location of the corresponding point to $I_A(\mathbf{p})$. From Equation 4.1, the error measure e_u is a random variable defined as

$$e_u = \sum_{i=1}^{n}(u_i - I_A(\mathbf{p}))^2$$

with an expectation value defined as

$$E(e_u) = E(\sum_{i=1}^{n}(u_i - I_A(\mathbf{p}))^2) = \sum_{i=1}^{n}\sigma^2(\mathbf{d}_{u_i}).$$

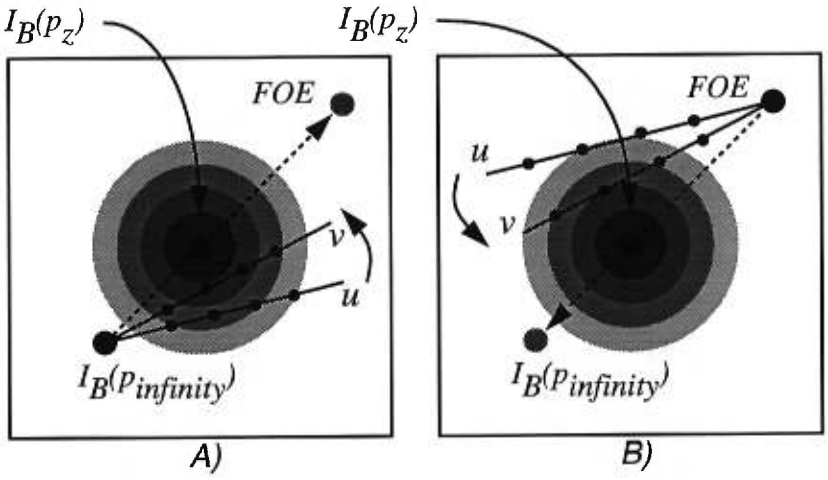

Figure 4.3. Error function for two segments u and v. When v is closer to $\mathbf{p}_z$ then u, its expectation is smaller for a well behaved variance function. A) Unknown translation. B) Unknown rotation.

Suppose we now take a second segment v starting also at $\mathbf{p}_\infty$, but closer to the point $\mathbf{p}_z$. A set of samples v_i is chosen with the same sampling[1] as segment u. The error measure e_v is defined as the random variable

$$e_v = \sum_{i=1}^{n} (v_i - I_A(\mathbf{p}))^2$$

which has an expected value

$$E(e_v) = \sum_{i=1}^{n} \sigma^2(\mathbf{d}_{v_i})$$

where $\mathbf{d}_{v_i}$ is the distance (x, y) from sample v_i to position $\mathbf{p}_z$. We now wish to show that the expectation of e_v is always smaller then $E(e_u)$. First, it is straightforward to see that

$$\|\mathbf{d}_{v_i}\| < \|\mathbf{d}_{u_i}\| \quad , \quad \forall i$$

[1] The case of different sampling and different lengths of u and v can also be handled in a more elaborate proof.

since v is a rotated version of u toward $\mathbf{p}_z$, except for the special pathological case where $\mathbf{p}_z = \mathbf{p}_\infty$. Second, the variance function $\sigma^2(\mathbf{d})$ is assumed to be monotonically and symetrically increasing with $\|\mathbf{d}\|$ from $\mathbf{p}_z$. From these two observations, we can immediately conclude that

$$\sigma^2(\mathbf{d}_{v_i}) < \sigma^2(\mathbf{d}_{u_i}) \quad , \quad \forall i.$$

It then follows that

$$E(e_v) = \sum_{i=1}^{n} \sigma^2(\mathbf{d}_{v_i}) < \sum_{i=1}^{n} \sigma^2(\mathbf{d}_{u_i}) = E(e_u)$$

which shows that as we get closer to the segment containing $I_B(\mathbf{p}_z)$, the expected error value gets smaller until it reaches a minimum when the candidate FOE corresponds to the true FOE. As long as the variance function is monotonic and symmetrical, this minimum is guaranteed to exist and is unique. Since this is true for any epipolar line segment, it is also true for the sum of these segments in global cost function. The same procedure is applied for rotation estimation, just by exchanging the role of the FOE and the zero-disparity point (see Figure 4.3B).

4.3 Experiments and results

Results of the motion without structure method are shown here for different kinds of real images pairs and for a calibrated motion sequence. The image pairs are taken from the SRI JISCT stereo database which provide partial ground truth since the motion between frames is a horizontal translation. However, we do not exploit this knowledge during the estimation procedure, and only use it to compare qualitatively the estimated and expected motions.

Most of the motions estimated here have a small forward (or backward) component. Our experiments show that large forward translation is much easier to estimate then lateral (i.e. sideway) motion. This is caused by the infamous rotation-translation ambiguity stating that a lateral translation (i.e. little or no forward component)

combined with a small camera field of view is hardy distinguishable from a rotation. Inversely, forward translation is not much affected by this ambiguity and therefore is easier to estimate.

4.3.1 Searching the solution space

A direct search of the motion space is performed by approximating the gradient and following steepest descent. The algorithm usually needs around 60 to 100 iterations to converge to the solution. Much improvement could be made to this search method, since no emphasis has yet been put on speed.

In all experiments conducted, we took care to select realistic initial estimates, i.e. as far as possible from the solution while taking into account the convergence constraint derived from the image texture. It is important to note that in most practical situations of motion tracking, the motion parameters from the previous frame can be used as an initial estimate for the next frame, taking advantage of the fact that motion tends to be similar and thus allowing faster convergence.

For all the experiments presented, only about 4% of the points of the images are arbitrarily selected for likelihood estimation. The typical running time is between 30 seconds and 10 minutes on a 150 MHz Silicon Graphics workstation. The execution time can be reduced by selecting a smaller number of points, at the expense of less accuracy in the motion estimate.

4.3.2 JISCT image pairs

The **Pentagon** image pair has a very well-behaved local intensity statistic. The image pair is very well aligned so that the motion between frames is purely due to horizontal translation. However, the magnitude of the translation is small, on average less then 2 pixels, which would usually make accurate estimation of the translation difficult.

The results are illustrated in Figure 4.4. The initial translation was 35° from the

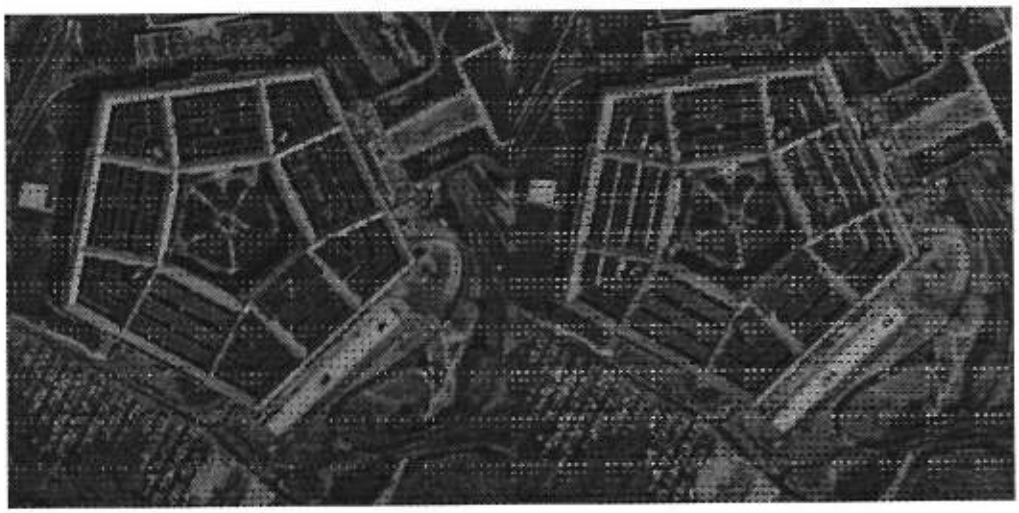

Figure 4.4. The *Pentagon* image pair. The solution is superimposed over the images as a grid of selected points with their corresponding epipolar segments. The epipolar line segments are approximately horizontal, indicating good alignment.

correct translation on the unit sphere, while the initial rotation was set to 10° around an arbitrary axis. The rotation obtained is 0.17°, corresponding to a maximum of 0.4 pixels error anywhere in the image. The true rotation is 0°. The translation obtained is $(-0.994, -0.102, 0.035)$, which correspond to a 6° error. While at first sight this appears large, we note that this is well within the accuracy of other two-frame algorithms [40, 80] and that, within the image, this error correspond to a maximum displacement of 0.3 pixel. The expected translation is $(-1, 0, 0)$.

The results for the **Tree** image pair , which also exhibits a pure horizontal translation, are illustrated in Figure 4.5.

The initial motion estimate is a translation oriented 35° from the horizontal on the unit sphere and the initial rotation estimate is 5°, corresponding to an image displacement of up to 12 pixels. The estimated translation is $(0.996, -0.0765, 0.0485)$, which is 5.4° from the true horizontal motion of $(-1, 0, 0)$. The estimated rotation is 0.4° which is believed to accurately reflect a slight vergence effect of the camera that

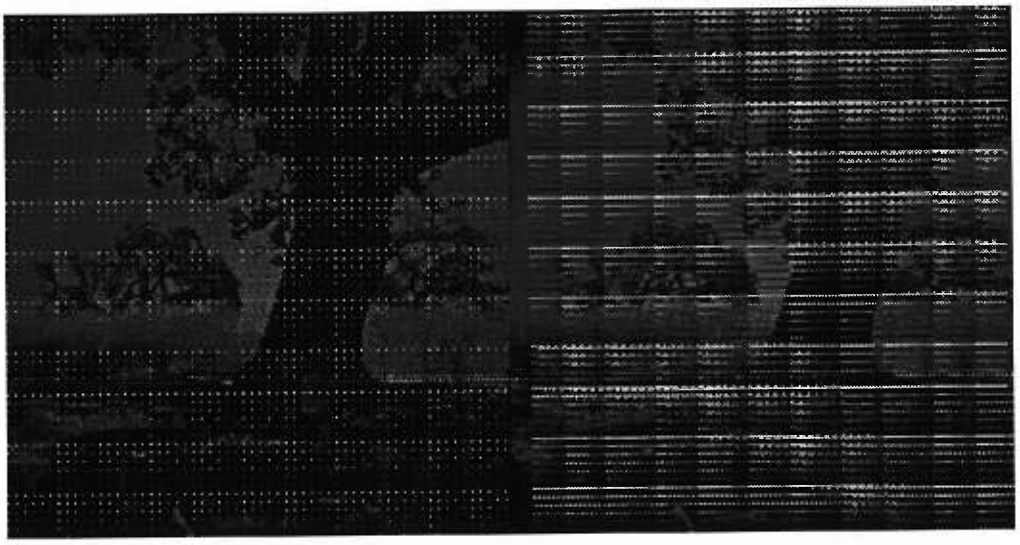

Figure 4.5. The *Tree*. The estimated motion is superimposed over the images as a grid of points and their corresponding epipolar segments. The motion is approximately horizontal.

can be manually observed.

The third example, the **Shrub**, also features only a horizontal translation. However, this type of imagery is usually difficult to analyze because of the ambiguous textures presented by the brick wall and the bushes. The results are illustrated in Figure 4.6. The initial motion estimate has a translation at 35° from the horizontal and a rotation of 5°. The estimated translation is $(-0.9992, 0.0369, 0.00836)$ which corresponds to a 2.2° error, for an expected translation of $(-1, 0, 0)$. The estimated rotation is 0.1° which corresponds to an image displacement of maximum 0.2 pixels.

4.3.3 The PUMA sequence

The motion without structure algorithm was tested on a Puma calibrated motion sequence, courtesy of the University of Massachusetts and shown in Figure 4.7. The rotation between each frame is approximately 4° around an axis parallel to the optical axis of the camera and located at $(0.909, 0.416, -0.005)$ feet from the optical center

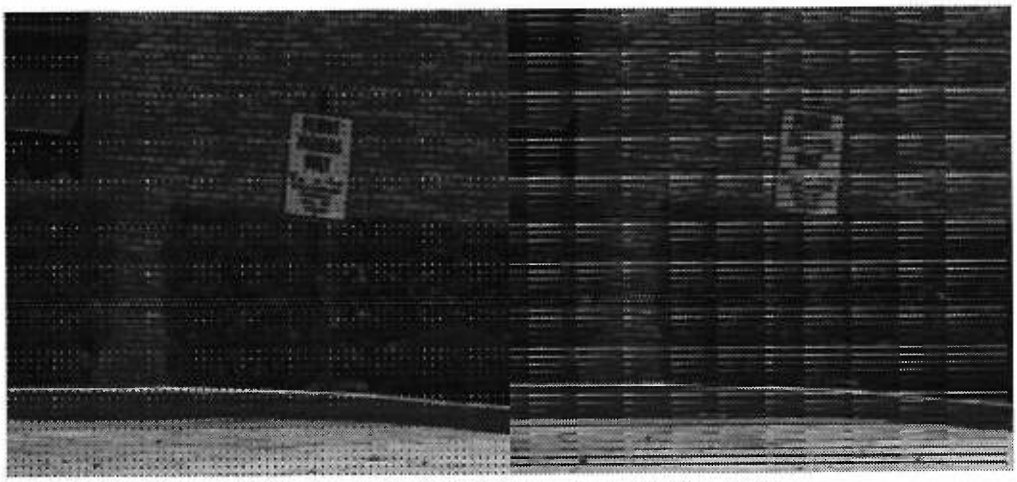

Figure 4.6. The *Shrub*. The recovered motion (approximately horizontal translation), superimposed on the right image.

Figure 4.7. The *Puma* image sequence, frames 1,4,7,10,13. The camera is at the end of a Puma robot arm rotating around its elbow in increments of 4°.

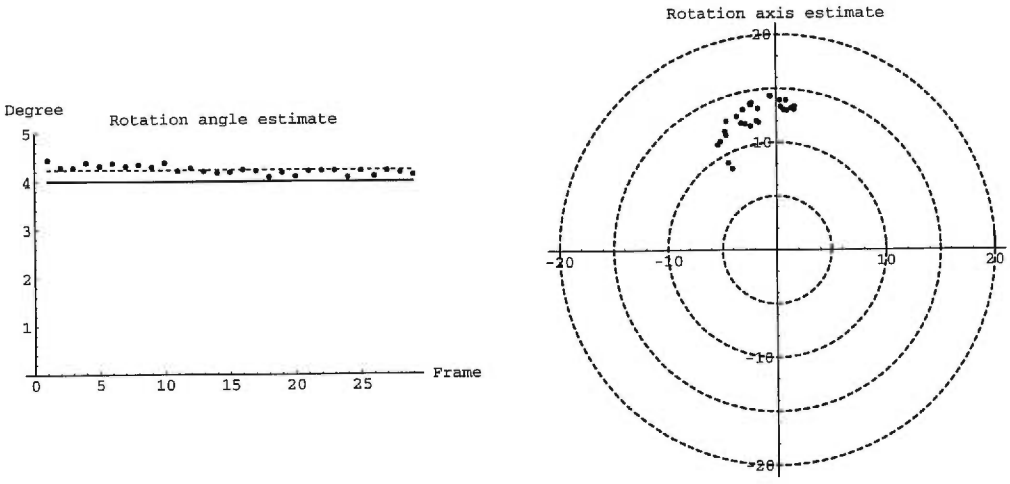

Figure 4.8. *Puma* **sequence. On the left, the magnitude of rotation is shown along with the average angle (dashed line) and true calibration angle (solid line). On the right, the axis of rotation on a** *flattened* **unit sphere, shown with** $(0°, 0°)$ **as the true axis of rotation.**

of the camera.

We performed the motion analysis using only two successive frames at a time. The initial estimates for the motion are always at least 5° off around an arbitrary axis for rotation, and at least 35° off for direction of translation. The rotation angle and axis estimates are shown in Figure 4.8. The rotation axis is estimated with an average of 13° error, while the rotation angle is estimated with an accuracy of 0.2°, which corresponds to a maximum image displacement of around 0.5 pixels. The results for translation are illustrated in Figure 4.9. When compared with calibration data, it appears that the estimated translations (thick line) are accurate and well within the calibration accuracy.

Since the calibration information is only available for the first 15 frames, the missing information was extrapolated whenever possible without affecting the reliability of the calibration. The fact that this motion analysis method does not require any *a priori* information such as feature point correspondence while providing excellent

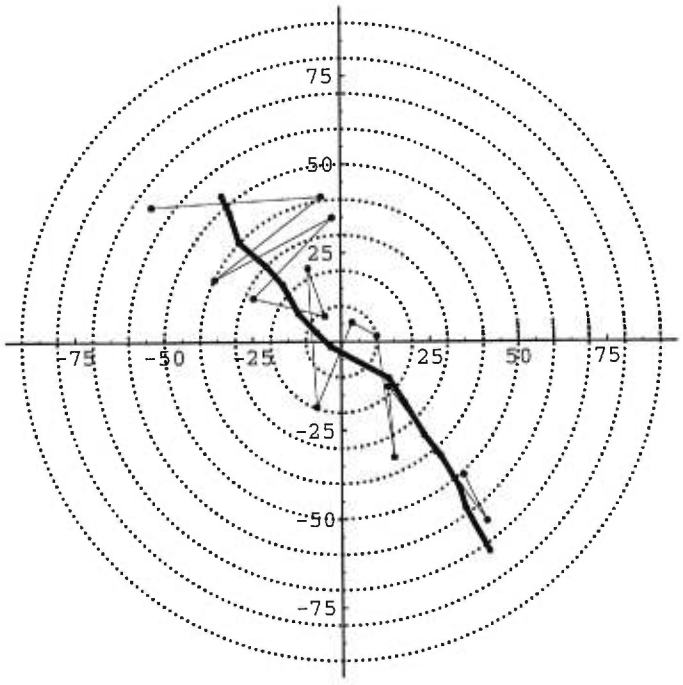

Figure 4.9. *Puma* **sequence. Estimated (thick line) and calibrated (thin line) translations shown on the** *flattened* **unit sphere.**

accuracy confirms the usefulness and convenience of the "motion without structure" approach.

4.3.4 Noise sensitivity

The evaluation function for any hypothesized motion does not rely on image gradients, and consists of accumulating large amount of intensity difference information. We therefore expect this measure to be very insensitive to noise.

As a simple test for noise sensitivity, we degraded the first two images of the *Puma* sequence using uniform noise in the range ± 10 up to ± 100, which corresponds to standard deviations ranging from 5.7 to 57.7 (see Figure 4.10). We computed the motion between the two frames 17 times at selected noise levels and observed the distribution of rotation angles recovered. In Figure 4.11, these angles are shown

Figure 4.10. Image degraded by uniform noise. "s" is the standard deviation of the noise.

along with ellipses whose heights are the standard deviations of rotation angles at particular noise levels. These standard deviations range from 0.01 to 0.1 degree. The relationship between the image noise level and the observed rotation angle error is approximately linear, implying that image noise has to double to result in doubling the error on the estimated rotation angle.

These results clearly indicate that the algorithm is very resistant to noise[2].

4.4 Conclusion

We presented a new paradigm to find the full motion between two frames. We refer to the approach as "motion without structure" because it does not require or compute any information related to the structure of the scene. The motion analysis problem is posed as a search in the space of possible motions and a likelihood measure is developed that evaluates hypothesized motion based on the sum of sum of squared differences between points in one image and their corresponding epipolar segments in the other.

This likelihood function was shown to exhibit exactly one local minimum for

[2] This is for uncorrelated noise (e.g. low contrast). For correlated noise (e.g. a single camera with a dirty lens), the effect on accuracy is likely to be larger.

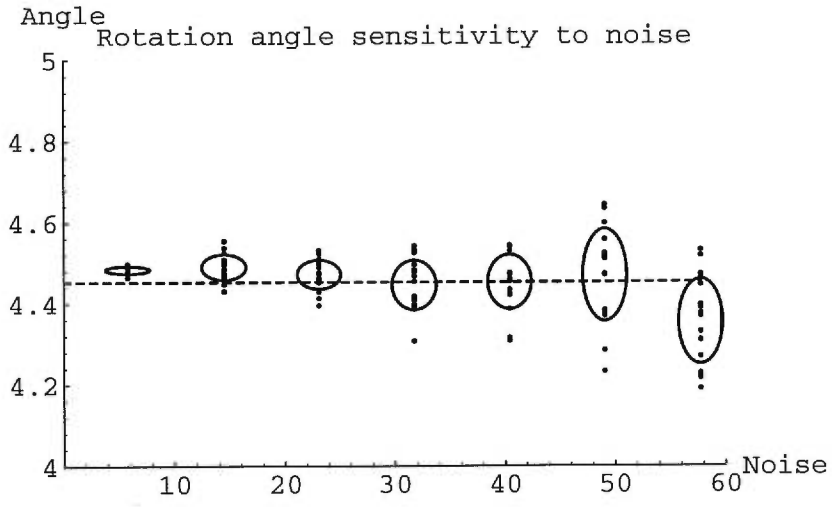

Figure 4.11. Rotation angles obtained for different pixel noise levels. The height of an ellipse gives the standard deviation of angles for a particular noise level.

the cases of either known rotation or known translation, provided the images are well-behaved, i.e. that the variance of intensity difference between two points is a monotonically increasing function of their distance apart. In the full motion case, a unique local minimum also exists, but may be subject to the well known ambiguity between rotational and translational motion.

Experimental results suggest that the method is applicable to a wide range of images while achieving very good accuracy and presenting strong robustness to noise. Large frame-to-frame motions can be handled and are only limited by the characteristics of the local intensity variation present in the image.

We believe that the paradigm of motion without structure can provide a robust and accurate algorithm to estimate the ego-motion between two frames. Moreover, we hope that it will prove superior to feature-based and direct or indirect methods of motion-and-structure estimation since neither optical flow, intensity derivatives or feature correspondence are needed.

Chapitre 5

INTRODUCTION À LA RECTIFICATION

Ce chapitre présente une introduction au problème de la rectification d'images stéréoscopiques.

La stéréoscopie conventionnelle assume généralement que la géométrie des caméras est horizontale, c'est-à-dire que les axes optiques sont parallèles et que les centres optiques sont déplacés parallèlement aux lignes horizontales des images, comme l'illustre la Figure 5.1. Cette façon de présenter les choses simplifie la mise en correspondance des images, puisqu'elle suppose que les lignes épipolaires correspondent aux lignes horizontales de pixels des images.

Toutefois, il est très rare en pratique que les caméras soient parfaitement alignées selon ce modèle. Les lignes épipolaires ne sont pas horizontales et une étape de *rectification* devra être effectuée pour transformer les images de façon à rendre les lignes épipolaires parallèles et horizontales. Cette situation est illustrée à la Figure 5.2. Pour que les droites épipolaires soient parallèles, il faut que les plans de projections des différentes caméras soient parallèles entre eux. Or, la rectification a pour but de transformer directement les images des caméras en les *reprojetant* de façon à rendre parallèles les droites épipolaires correspondantes entre les deux images.

5.1 Rectification plane

Une solution à la rectification d'image a été proposée par Ayache et Hansen [2] et Faugeras [24]. Celle-ci utilise une transformation linéaire projective d'application simple et rapide. Nous présentons dans les paragraphes qui suivent une version simplifiée de Ayache et Hansen [2] et Faugeras [24], servant d'introduction à notre méthode ex-

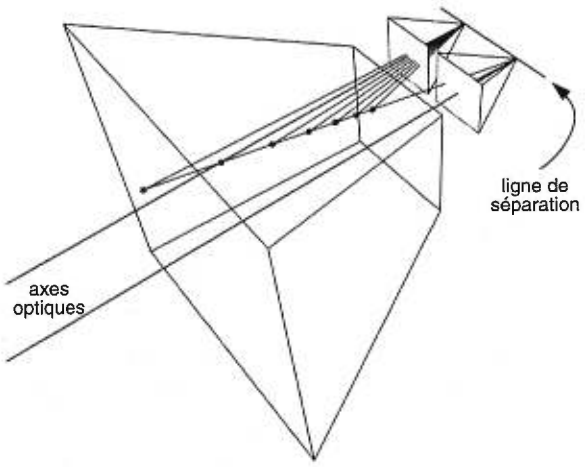

Figure 5.1. Géométrie de caméra horizontale. Les axes optiques sont parallèles. La ligne de séparation est parallèle à l'axe horizontal des images.

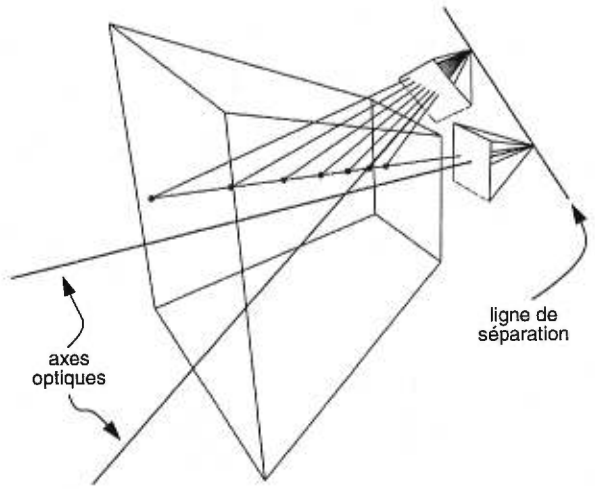

Figure 5.2. Géométrie de caméra arbitraire. Les axes optiques ne sont ni parallèles entre eux, ni perpendiculaires à la ligne de séparation.

posée au chapitre suivant.

Il est possible, par une simple transformation linéaire projective, de *reprojeter* une image sur un plan arbitraire. Pour rectifier ces images, il suffit de choisir un nouveau plan de projection commun entre les deux caméras, et ainsi garantir que les droites épipolaires soient parallèles, une fois reprojetées.

Comme l'a illustré la figure 2.3, les droites épipolaires sont issues de l'intersection des plans de projection des caméras et d'un *plan épipolaire*, formé des deux centres optiques des caméras et d'un point choisi dans une image. Dans chaque image, le *point d'expansion* (*focus of expansion*), point d'intersection commun des droites épipolaires, est la projection du centre optique d'une caméra dans l'image de l'autre caméra.

Pour que les droites épipolaires soient parallèles, il faut que le point d'expansion soit à l'infini. De plus, pour que ces droites soient horizontales, le point d'expansion (**foe**) doit être dans la direction horizontale, c'est-à-dire en coordonnées homogènes

$$\mathbf{foe} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T.$$

Supposons que les matrices de passage de deux caméras A et B sont respectivement $\mathbf{W}_A$ et $\mathbf{W}_B$, comme dans les équations 2.8 et 2.9. La matrice de passage de B vers A est donc $\mathbf{W}_{AB}$ comme à l'équation 2.12. Le centre optique **CB** de la caméra B est à l'origine du système de coordonnées de la caméra B, donc

$$\mathbf{CB}_b = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

et se projette dans le système de la caméra A au point $\mathbf{CB}_a$, représentant aussi le point d'expansion $\mathbf{foe}_a$, obtenu par la relation de l'équation 2.17

$$\mathbf{foe}_a = \mathbf{CB}_a = \mathbf{J} \cdot \mathbf{W}_{AB} \cdot [\mathbf{CB}_b; 1] = [\mathbf{W}_{AB}]_t$$

où l'opérateur $[\cdot]_t$ a été introduit à la section 2.3.

La rectification d'image est une transformation $\mathbf{R}$ de l'espace projectif, appliquée aux points de l'image A pour former l'image rectifiée A^*. Ainsi, on transforme un

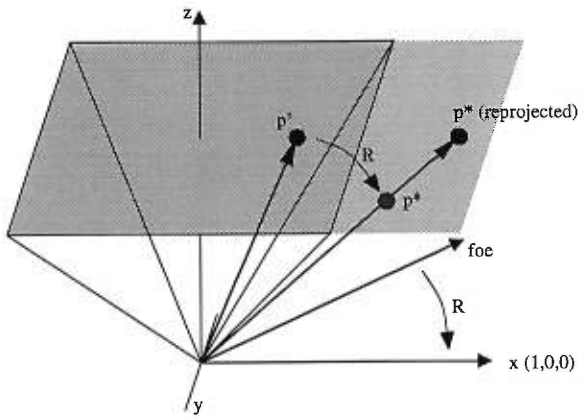

Figure 5.3. Rectification plane. La rotation $\mathbf{R}$ qui transforme le $\mathbf{foe}$ vers l'axe $\mathbf{x}$ est appliquée à un point image $\mathbf{p}'$ pour donner un point rectifié $\mathbf{p}^*$ qui doit ensuite être reprojeté sur le plan image.

point image $\mathbf{p}'_a$ en un point rectifié $\mathbf{p}^*_a$ selon la relation

$$\mathbf{p}^*_a = \mathbf{R} \cdot \mathbf{p}'_a.$$

Le point d'expansion $\mathbf{foe}_a$ doit être transformé vers l'infini horizontal, c'est-à-dire

$$\mathbf{R} \cdot \mathbf{foe}_a = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

La transformation la plus simple qui satisfait à cette contrainte est une rotation de l'espace projectif. On peut donc définir la transformation $\mathbf{R}$ comme une rotation qui transforme le vecteur $\mathbf{foe}_a$ vers $(1, 0, 0)$. Cette rotation, appliquée aux points à rectifier comme l'illustre la figure 5.3, se définit

$$\mathbf{R} = R_{aligne}(\mathbf{foe}_a, \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T, \phi) \tag{5.1}$$

où $R_{aligne}(\mathbf{u}, \mathbf{v}, \phi)$ est une matrice de rotation qui aligne le vecteur $\mathbf{u}$ sur le vecteur $\mathbf{v}$, c'est-à-dire

$$R_{aligne}(\mathbf{u}, \mathbf{v}, \phi) \cdot \mathbf{u} = \mathbf{v}$$

et dont la définition est

$$R_{aligne}(\mathbf{u}, \mathbf{v}, \phi) = R_{axe}(\phi, \mathbf{v}) \cdot R_{axe}(Angle(\mathbf{u}, \mathbf{v}), \mathbf{u} \times \mathbf{v})$$

avec $R_{axe}(\theta, \mathbf{a})$ défini comme la matrice de rotation d'un angle θ autour de l'axe $\mathbf{a}$ et $Angle(\cdot)$ comme l'angle entre deux vecteurs. Plus d'une matrice $\mathbf{R}$ peuvent éffectuer la rotation de $\mathbf{u}$ vers $\mathbf{v}$. Ce degré de liberté est représenté par l'angle ϕ à l'équation 5.1. Ce paramètre n'a donc aucun effet sur l'orientation des lignes épipolaires. Il affecte plutôt le degré de distorsion de l'image rectifiée.

5.2 Conclusion

La rectification plane, décrite précédemment, présente plusieurs difficultés rédhibitoires. Puisque le *point d'expansion* est porté à l'infini lors de la reprojection, il est évident que si ce point est à l'intérieur de l'image, l'image rectifiée sera dans ce cas de dimension infinie puisqu'elle contient ce point. Cette situation se présente lorsque le mouvement latéral d'une caméra est faible comparativement à son mouvement vers l'avant. Plus formellement, ceci revient à dire que le centre optique d'une caméra se trouve reprojeté à l'intérieur de l'image de l'autre caméra. La rectification plane ne peut être appliquée dans ces cas.

La nouvelle méthode de *rectification cylindrique*, que nous présentons au chapitre suivant, résout ce problème en remplaçant le plan commun de reprojection par un cylindre dont l'axe est parallèle à l'axe reliant les deux centres optiques.

On peut finalement résumer l'essentiel des méthodes de rectification plane:

- Une transformation linéaire projective est appliquée à chaque pixel de l'image originale. Les coordonnées ainsi obtenues sont celles du pixel *rectifié*.

- La transformation peut être obtenue de plusieurs façons, mais toujours à partir de la géométrie des caméras. Le critère fondamental est que le nouveau plan de projection soit parallèle à l'axe reliant les deux centres optiques.

- Presque toutes les méthodes calculent une seule transformation par image, ce qui correspond à reprojeter sur un plan commun aux caméras.

- La transformation préserve les lignes droites de l'image, qu'elles soient ou non épipolaires.

- La longueur d'un segment (épipolaire ou non) n'est pas préservée, ce qui implique que les droites épipolaires subissent une certaine quantité de distorsion, créant ainsi des problèmes de perte d'information.

- La taille de l'image rectifiée dépend de la géométrie des caméras. Cette propriété constitue un grave défaut, puisque pour un grand nombre d'orientations de caméra, l'image rectifiée est de taille infinie. Ce problème provient du fait que le point d'intersection des droites épipolaires (le point d'expansion) est toujours reprojeté à l'infini, excluant du coup toutes les géométries de caméras où ce point d'expansion serait visible dans une des images.

La nouvelle méthode de *rectification cylindrique* que nous proposons en contrepartie possède les caractéristiques suivantes:

- Elle utilise une transformation linéaire projective appliquée aux pixels de l'image. Cette transformation varie selon la ligne épipolaire rectifiée, et doit donc être recalculée pour chacune de ces lignes. On a donc un effet équivalent à la reprojection sur un cylindre plutôt que sur un plan.

- Elle fonctionne pour toutes les géométries de caméras, sans aucune exception.

- La taille des images rectifiées n'est pas fonction de la géométrie des caméras et est donc connue d'avance.

- La distorsion des droites épipolaires rectifiées est toujours nulle. Par contre, les droites non épipolaires deviennent des courbes, une fois rectifiées.

- Cette méthode peut être utilisée pour créer des vues panoramiques (mosaïques) d'une scène à partir d'une séquence vidéo où la caméra traverse la scène.

Chapitre 6

CYLINDRICAL RECTIFICATION TO MINIMIZE EPIPOLAR DISTORTION

Cet article [71] a été publié comme l'indique la référence bibliographique

> Sébastien Roy, Jean Meunier et Ingemar J. Cox, Cylindrical Rectification to Minimize Epipolar Distortion, dans *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, San Juan, Puerto Rico, Juin 1997, pages 393-399.

Cet article, aussi accepté pour publication dans le journal scientifique *IEEE Transactions on Pattern Analysis and Machine Intelligence*, est présenté ici dans sa version originale.

Abstract

We propose a new rectification method for aligning epipolar lines of a pair of stereo images taken under any camera geometry. It effectively remaps both images onto the surface of a cylinder instead of a plane, which is used in common rectification methods. For a large set of camera motions, remapping to a plane has the drawback of creating rectified images that are potentially infinitely large and presents a loss of pixel information along epipolar lines. In contrast, cylindrical rectification guarantees that the rectified images are bounded for all possible camera motions and minimizes the loss of pixel information along the epipolar line. The processes (eg. stereo matching, etc.) subsequently applied to the rectified images are thus more accurate and general since they can accommodate any camera geometry.

6.1 Introduction

Rectification is a necessary step of stereoscopic analysis. The process extracts epipolar lines and realigns them horizontally into a new *rectified* image. This allows subsequent stereoscopic analysis algorithms to easily take advantage of the *epipolar constraint* and reduce the search space to one dimension along the horizontal rows of the rectified images.

For different camera motions, the set of matching epipolar lines varies considerably and extracting those lines for the purpose of depth estimation can be quite difficult. The difficulty does not reside in the equations themselves; for a given point, it is straightforward to locate the epipolar line containing that point. The problem is to find a set of epipolar lines that will cover the whole image and introduce a minimum of distortion, for arbitrary camera motions. Since subsequent stereo matching occurs along epipolar lines, it is important that no pixel information is lost along these lines in order to efficiently and accurately recover depth.

Fig. 6.1 depicts the rectification process. A scene S is observed by two cameras to create images I_1 and I_2. In order to align the epipolar lines of this stereo pair, some image transformation must be applied. The most common of such transformations, proposed by Ayache and Hansen [2] and referred to as *planar rectification*, is a remapping of the original images onto a single plane that is parallel to the line joining the two cameras optical centers (see Fig. 6.1, images P_1 and P_2). This is accomplished by using a linear transformation in projective space applied to each image pixel.

The new rectification method presented in this paper, referred to as *cylindrical rectification*, proposes a transformation that remaps the images onto the surface of a cylinder whose principal axis goes through both cameras optical centers (see Fig. 6.1, images C_1 and C_2). The actual images related to Fig. 6.1 are shown in Fig. 6.2.

The line joining the optical centers of the cameras (see Fig. 6.1) defines the focus of expansion (**foe**). All epipolar lines intersect the focus of expansion. The rectification

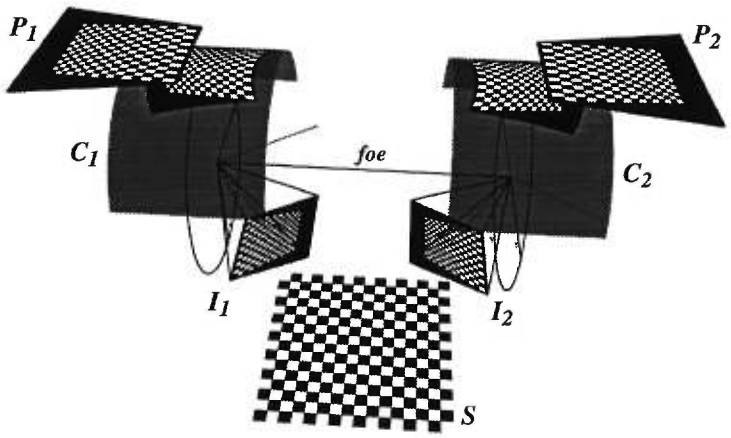

Figure 6.1. Rectification. Stereo images (I_1, I_2) of scene S shown with planar rectification (P_1, P_2) and cylindrical rectification (C_1, C_2)

process applied to an epipolar line always makes that line *parallel* to the foe. This allows the creation of a rectified image where the epipolar lines do not intersect and can be placed as separate rows. Obviously, both plane and cylinder remappings satisfy the alignment requirement with the foe.

Planar rectification, while being simple and efficient, suffers from a major drawback: it fails for some camera motions, as demonstrated in Sec. 6.2. As the forward motion component becomes more significant, the image distortion induced by the transformation becomes progressively worse until the image is unbounded. The image distortion induces a loss of pixel information that can only be partly compensated for by making the rectified image size larger[1]. Consequently, this method is useful only for motions with a small forward component, thus lowering the risk of unbounded rectified images. One benefit of planar rectification is that it preserves straight lines, which is an important consideration if stereo matching is to be performed on edges or lines.

[1] See Sec. 6.3.6 for a detailed discussion.

On the other hand, cylindrical rectification is guaranteed to provide a bounded rectified image and to significantly reduce pixel distortion, for all possible camera motions. This transformation also preserves epipolar line *length*. For example, an epipolar line 100 pixels long will always be rectified to a line 100 pixels long. This ensures a minimal loss of pixel information when resampling the epipolar lines from the original images. However, arbitrary straight lines are no longer preserved, though this may only be a concern for edge based stereo.

Planar rectification uses a single linear transformation matrix applied to the image, making it quite efficient. Cylindrical rectification uses one such linear transformation matrix for *each* epipolar line. In many cases, these matrices can be precomputed so that an equivalent level of performance can be achieved.

Although it is assumed throughout this paper that internal camera parameters are known, cylindrical rectification works as well with unknown internal parameters, as is the case when only the *fundamental matrix* (described in [55]) is available (see Sec. 6.3.5).

Many variants of the planar rectification scheme have been proposed [2, 24, 47]. A detailed description based on the *essential matrix* is given in Hartley and Gupta [34]. In Courtney *et al.* [17], a hardware implementation is proposed. In Papadimitriou and Dennis [62], the camera motion is restricted to a *vergent stereo* geometry to simplify computations. It also presents a faster way to compute the transformation by approximating it with a non-projective linear transformation. This eliminates the risk of unbounded images at the expense of potentially severe distortion. In Robert *et al.* [66], a measure of image distortion is introduced to evaluate the performance of the rectification method. This strictly geometric measure, based on edge orientations, does not address the problem of pixel information loss induced by interpolation (see Sec. 6.3.6).

Sec. 6.2 describes planar rectification in more detail. The cylindrical rectification method is then presented in Sec. 6.3. It describes the transformation matrix whose

three components are explicitly detailed in Sec. 6.3.3, 6.3.2, and 6.3.1. Sec. 6.3.4 discusses the practical aspects of finding the set of corresponding epipolar lines in both images to rectify. It is demonstrated in Sec. 6.3.5 that it is possible to use uncalibrated as well as calibrated cameras. A measure of image distortion is introduced in Sec. 6.3.6 and used to show how both rectification methods behave for different camera geometries. Examples of rectification for different camera geometries are presented in Sec. 6.4.

6.2 *Linear transformation in projective space*

In this section we show how rectification methods based on a single linear transformation in projective space [2, 24, 47] fail for some camera geometries.

As stated earlier, the goal of rectification is to apply a transformation to an image in order to make the epipolar lines parallel to the focus of expansion. The result is a set of images where each row represents one epipolar line and can be used directly for the purpose of stereo matching (see Fig. 6.2).

In projective space, an image point is expressed using homogenous coordinates as $\mathbf{p} = (h\,p_x, h\,p_y, h)^T$ where h is a scale factor. Thus we can assume these points are projected to $\mathbf{p} = (p_x, p_y, 1)^T$.

The linear projective transformation $\mathbf{F}$ is used to transform an image point $\mathbf{u}$ into a new point $\mathbf{v}$ with the relation

$$\mathbf{v} = \mathbf{F} \cdot \mathbf{u} = \begin{bmatrix} F_0 & F_1 & F_2 \\ F_3 & F_4 & F_5 \\ F_6 & F_7 & F_8 \end{bmatrix} \cdot \mathbf{u} \tag{6.1}$$

where

$$\mathbf{v} = (v_x, v_y, v_h)^T \quad \mathbf{u} = (u_x, u_y, u_h)^T \quad u_h \neq 0 \ .$$

The fact that $u_h \neq 0$ simply implies that the original image has a finite size. Enforcing

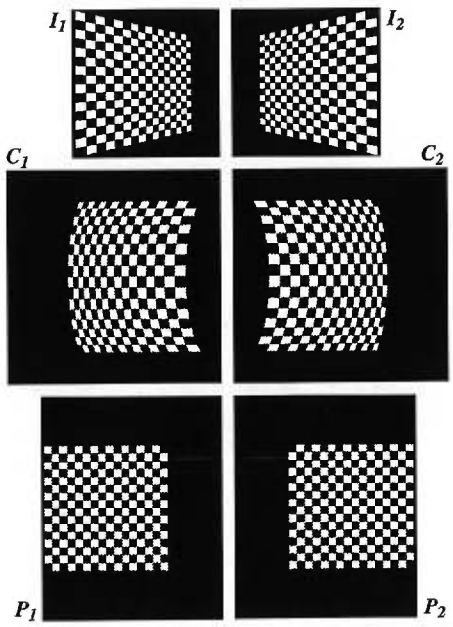

Figure 6.2. Images from Fig. 6.1. Original images (I_1, I_2) are shown with cylindrical rectification (C_1, C_2) and planar rectification (P_1, P_2).

that the reprojected point is *not* at infinity implies that v_h must be non-zero, that is

$$v_h = u_x F_6 + u_y F_7 + u_h F_8 \neq 0. \tag{6.2}$$

Since u_x, u_y are arbitrary, Eq. 6.2 has only one possible solution $(F_6, F_7, F_8) = (0, 0, 1)$ since only u_h can guarantee v_h to be non-zero and $\mathbf{F}$ to be homogeneous. Therefore, the transformation $\mathbf{F}$ must have the form

$$\mathbf{F} = \begin{bmatrix} F_0 & F_1 & F_2 \\ F_3 & F_4 & F_5 \\ 0 & 0 & 1 \end{bmatrix}$$

which corresponds to a camera displacement with *no* forward (or backward) component.

In practice, the rectified image is unbounded only when the **foe** is *inside* the image. Therefore, any camera motion with a large forward component (making the **foe**

visible) *cannot* be rectified with this method. Moreover, as soon as the forward component is large enough, the image points are mapped so far apart that the rectification becomes unusable due to severe distortion.

In the next section, we described how *cylindrical rectification* can alleviate these problems by making a different use of linear transformations in projective space.

6.3 Cylindrical rectification

The goal of cylindrical rectification is to apply a transformation of an original image to remap on the surface of a carefully selected cylinder instead of a plane. By using the line joining the camera's optical centers as the cylinder axis (Fig. 6.1), all straight lines on the cylinder surface are necessarily parallel to the cylinder axis and focus of expansion, making them suitable to be used as epipolar lines.

The transformation from image to cylinder, illustrated in Fig. 6.3, is performed in three stages. First, a rotation is applied to a selected epipolar line (step $\mathbf{R}_{foe}$). This rotation is in the epipolar plane and makes the epipolar line parallel to the foe. Then, a change of coordinate system is applied (step $\mathbf{T}_{foe}$) to the rotated epipolar line from the image system to the cylinder system (with **foe** as principal axis). Finally, (step $\mathbf{S}_{foe}$), this line is *normalized* or *reprojected* onto the surface of a cylinder of unit radius. Since the line is already parallel to the cylinder, it is simply scaled along the direction perpendicular to the axis until it lies at unit distance from the axis. A particular epipolar line is referenced by its angle θ around the cylinder axis, while a particular pixel on the epipolar line is referenced by its angle and position along the cylinder axis (see Fig. 6.3).

Even if the surface of the cylinder is infinite, it can be shown that the image on that surface is always bounded. Since the transformation aligns an epipolar line with the axis of the cylinder, it is possible to remap a pixel to infinity only if its epipolar line is originally infinite. Since the original image is finite, all the visible parts of the

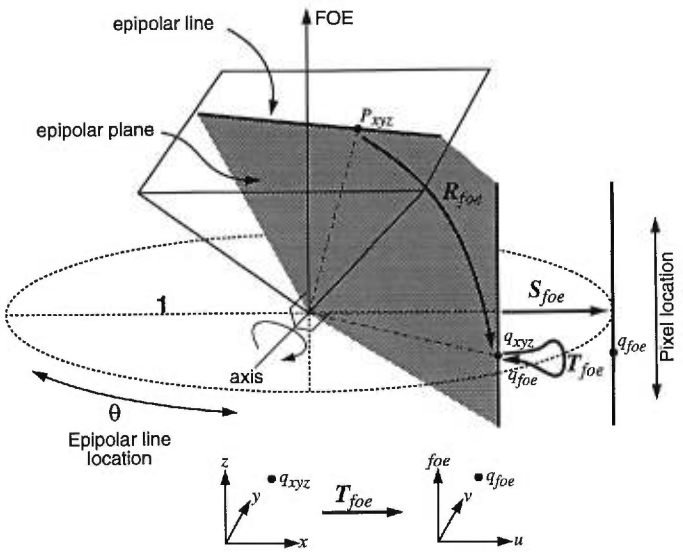

Figure 6.3. The basic steps of the cylindrical rectification method. First ($\mathbf{R}_{foe}$), an epipolar line is rotated in the epipolar plane until it is parallel to the foe. Second ($\mathbf{T}_{foe}$), a change of coordinate system is applied. Third ($\mathbf{S}_{foe}$), a projection onto the surface of the unit cylinder is applied.

epipolar lines are also of finite length and therefore the rectified image cannot extend to infinity.

The rectification process transforms an image point $\mathbf{p}_{xyz}$ into a new point $\mathbf{q}_{foe}$ which is expressed in the coordinate system **foe** of the cylinder. The transformation matrix $\mathbf{L}_{foe}$ is defined so that the epipolar line containing $\mathbf{p}_{xyz}$ will become parallel to the cylinder axis, the **foe**. Since all possible epipolar lines will be parallel to the foe, they will also be parallel to one another and thus form the desired *parallel aligned epipolar geometry.*

We have the linear rectification relations between $\mathbf{q}_{foe}$ and $\mathbf{p}_{xyz}$ stated as

$$\begin{aligned} \mathbf{q}_{foe} &= \mathbf{L}_{foe}\mathbf{p}_{xyz} \\ &= (\mathbf{S}_{foe}\mathbf{T}_{foe}\mathbf{R}_{foe})\mathbf{p}_{xyz} \end{aligned} \tag{6.3}$$

and inversely

$$\begin{aligned} \mathbf{p}_{xyz} &= \mathbf{L}_{foe}^{-1}\mathbf{q}_{foe} \\ &= (\mathbf{R}_{foe}^{T}\mathbf{T}_{foe}^{T}\mathbf{S}_{foe}^{-1})\mathbf{q}_{foe} \end{aligned} \tag{6.4}$$

where

$$\mathbf{S}_{foe} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{k} & 0 \\ 0 & 0 & \frac{1}{k} \end{bmatrix} \quad \text{and} \quad \mathbf{T}_{foe} = \begin{bmatrix} \mathbf{foe} \\ \mathbf{u} \\ \mathbf{v} \end{bmatrix}.$$

These relations are completely invertible (except for the special case $\mathbf{p}_{xyz} = \mathbf{foe}$, which is quite easily handled). The matrix $\mathbf{R}_{foe}$ represents the rotation of the image point in projective space. The matrix $\mathbf{T}_{foe}$ represents the change from the camera coordinate system to the cylinder system. The matrix $\mathbf{S}_{foe}$ represents the projective scaling used to project a rectified point onto the surface of the unit cylinder.

The next three subsections will describe how to compute the coordinate transformation $\mathbf{T}_{foe}$, the rotation $\mathbf{R}_{foe}$, and the scaling $\mathbf{S}_{foe}$.

6.3.1 Determining transformation $\mathbf{T}$

The matrix $\mathbf{T}_{foe}$ is the coordinate transformation matrix from system $(\mathbf{x};\mathbf{y};\mathbf{z})$ to system $(\mathbf{foe};\mathbf{u};\mathbf{v})$ such that

$$\begin{aligned} \mathbf{q}_{foe} &= \mathbf{T}_{foe}\mathbf{q}_{xyz} \\ \mathbf{q}_{xyz} &= \mathbf{T}_{foe}^{T}\mathbf{q}_{foe} \end{aligned} \tag{6.5}$$

and is uniquely determined by the position and motion of the cameras (see Fig. 6.3).

Any camera has a position **pos** and a rotation of ϕ degrees around the axis **axis** relative to the world coordinate system. A homogeneous world point $\mathbf{p}_w$ is expressed in the system of camera a (with $\mathbf{pos}_a$, $\mathbf{axis}_a$, and ϕ_a) as

$$\mathbf{p}_a = \mathbf{R}_{aw}\mathbf{p}_w$$

where $\mathbf{R}_{aw}$ is the 4×4 homogeneous coordinate transformation matrix obtained as

$$\begin{aligned}\mathbf{R}_{aw} &= \begin{bmatrix} \text{rot}(\mathbf{axis}_a, -\phi_a) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} I & -\mathbf{pos}_a \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{r}_{aw} & -\mathbf{r}_{aw} \cdot \mathbf{pos}_a \\ \mathbf{0} & 1 \end{bmatrix}\end{aligned}$$

where

$$\mathbf{r}_{aw} = \text{rot}(\mathbf{axis}_a, -\phi_a)$$

and $\text{rot}(\mathbf{a}, \theta)$ is a 3×3 rotation matrix of angle θ around axis $\mathbf{a}$. The corresponding matrix $\mathbf{R}_{bw}$ for camera b with $\mathbf{pos}_b$, $\mathbf{axis}_b$, and ϕ_b is defined in a similar way.

The direct coordinate transformation matrices for camera a and b such that

$$\begin{aligned}\mathbf{p}_a &= \mathbf{R}_{ab}\mathbf{p}_b \\ \mathbf{p}_b &= \mathbf{R}_{ba}\mathbf{p}_a\end{aligned}$$

are defined as

$$\begin{aligned}\mathbf{R}_{ab} &= \mathbf{R}_{aw} \cdot \mathbf{R}_{bw}^{-1} = \begin{bmatrix} \mathbf{r}_{ab} & \mathbf{foe}_a \\ \mathbf{0} & 1 \end{bmatrix} \\ \mathbf{R}_{ba} &= \mathbf{R}_{bw} \cdot \mathbf{R}_{aw}^{-1} = \begin{bmatrix} \mathbf{r}_{ba} & \mathbf{foe}_b \\ \mathbf{0} & 1 \end{bmatrix}\end{aligned}$$

where

$$\begin{aligned}\mathbf{r}_{ab} &= \mathbf{r}_{aw}\mathbf{r}_{bw}^T \\ \mathbf{r}_{ba} &= \mathbf{r}_{bw}\mathbf{r}_{aw}^T \\ \mathbf{foe}_a &= \mathbf{r}_{aw} \cdot (\mathbf{pos}_b - \mathbf{pos}_a) \\ \mathbf{foe}_b &= \mathbf{r}_{bw} \cdot (\mathbf{pos}_a - \mathbf{pos}_b)\end{aligned}$$

from which we can derive the matrix $\mathbf{T}_{foe;a}$ for rectifying the image of camera a as

$$\mathbf{T}_{foe;a} = \begin{bmatrix} n(\mathbf{foe}_a) \\ n(\mathbf{z} \times \mathbf{foe}_a) \\ n(\mathbf{foe}_a \times (\mathbf{z} \times \mathbf{foe}_a)) \end{bmatrix} \tag{6.6}$$

where $n(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|$ is a normalizing function. The corresponding matrix $\mathbf{T}_{foe;b}$ for rectifying the image of camera b can be derived similarly or more simply by the relation

$$\mathbf{T}_{foe;b} = -\mathbf{T}_{foe;a} \cdot \mathbf{r}_{ab}. \tag{6.7}$$

For the case where $\mathbf{foe}_a = \mathbf{z}$, the last two rows of $\mathbf{T}_{foe;a}$ can be any two orthonormal vectors perpendicular to $\mathbf{z}$.

6.3.2 Determining rotation $\mathbf{R}$

The epipolar line containing a point $\mathbf{p}_{xyz}$ will be rotated around the origin (the camera's optical center) and along the epipolar plane until it becomes parallel to the **foe**. The epipolar plane containing $\mathbf{p}_{xyz}$ also contains the **foe** (by definition) and the origin. The normal to that plane is

$$\mathbf{axis} = \mathbf{foe} \times \mathbf{p}_{xyz} \tag{6.8}$$

and will be the axis of rotation (see Fig. 6.3), thus ensuring that $\mathbf{p}_{xyz}$ remains in the epipolar plane. In the case $\mathbf{p}_{xyz} = \mathbf{foe}$, the axis can be any vector normal to the **foe** vector.

The angle of rotation needed can be computed by using the fact that the normal $\mathbf{z} = (0, 0, 1)^T$ to the image plane has to be rotated until it is perpendicular to the **foe**. This is because the new epipolar line has to be parallel to the **foe**. The rotation angle is the angle between the normal $\mathbf{z} = (0, 0, 1)^T$ projected on the epipolar plane (perpendicular to the rotation axis) and the plane normal to the **foe** also containing

the origin. By projecting the point $\mathbf{p}_{xyz}$ onto that plane, we can directly compute the angle. We have $\mathbf{z}'$, the normal $\mathbf{z}$ projected on the epipolar plane defined as

$$\mathbf{z}' = \mathbf{axis} \times (\mathbf{z} \times \mathbf{axis}) = \begin{bmatrix} -\mathbf{axis}_x\mathbf{axis}_z \\ -\mathbf{axis}_y\mathbf{axis}_z \\ \mathbf{axis}_x^2 + \mathbf{axis}_y^2 \end{bmatrix}$$

and $\mathbf{p}'$, the projected $\mathbf{p}_{xyz}$ on the plane normal to the **foe**, defined as

$$\mathbf{p}' = \mathbf{T}_{foe}^T \mathbf{B} \mathbf{T}_{foe} \mathbf{p}_{xyz} \tag{6.9}$$

where $\mathbf{T}_{foe}$ was previously defined in Eq. 6.6 and $\mathbf{B}$ is defined as

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation matrix $\mathbf{R}_{foe}$ rotates the vector $\mathbf{z}'$ onto the vector $\mathbf{p}'$ around the axis of Eq. 6.8 and is defined as

$$\mathbf{R}_{foe} = \mathbf{rot}_{\mathbf{p}',\mathbf{z}'} \tag{6.10}$$

where $\mathbf{rot}_{\mathbf{a},\mathbf{b}}$ rotates vector $\mathbf{b}$ onto vector $\mathbf{a}$ such that

$$\mathbf{rot}_{\mathbf{a},\mathbf{b}} = \begin{bmatrix} n(\mathbf{a}) \\ n(\mathbf{a} \times \mathbf{b}) \\ n((\mathbf{a} \times \mathbf{b}) \times \mathbf{a}) \end{bmatrix}^T \begin{bmatrix} n(\mathbf{b}) \\ n(\mathbf{a} \times \mathbf{b}) \\ n((\mathbf{a} \times \mathbf{b}) \times \mathbf{b}) \end{bmatrix}.$$

If the point $\mathbf{q}_{foe}$ is available instead of point $\mathbf{p}_{xyz}$, (as would be the case for the inverse transformation of Eq. 6.4) we can still compute $\mathbf{R}_{foe}$ from Eq. 6.10 by substituting $\mathbf{q}_{xyz}$ for $\mathbf{p}_{xyz}$ in Eq. 6.8 and 6.9 where $\mathbf{q}_{xyz}$ is derived from $\mathbf{q}_{foe}$ using Eq. 6.5. Notice that because $\mathbf{p}_{xyz}$ and $\mathbf{q}_{xyz}$ are in the same epipolar plane, the rotation axis will be the same. Also, the angle of rotation will also be the same since their projection onto the plane normal to the **foe** is the same (modulo a scale factor).

6.3.3 Determining the scaling **S**

The matrix $\mathbf{S}_{foe}$ is used to project the epipolar line from the unit image plane (i.e. located at $z = 1$) onto the cylinder of unit radius. As shown in Eq. 6.3 and 6.4, $\mathbf{S}_{foe}$ has one scalar parameter k. This parameter can be computed for a known point $\mathbf{p}_{xyz}$ (Eq. 6.3) by enforcing unit radius and solving the resulting equation

$$\begin{aligned} \|\mathbf{B}\,\mathbf{q}_{foe}\| &= 1 \\ \left\| \mathbf{B} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{k} & 0 \\ 0 & 0 & \frac{1}{k} \end{bmatrix} \mathbf{T}_{foe}\mathbf{R}_{foe}\mathbf{p}_{xyz} \right\| &= 1 \end{aligned} \tag{6.11}$$

which yields the solution

$$k = \|\mathbf{B}\,\mathbf{T}_{foe}\mathbf{R}_{foe}\mathbf{p}_{xyz}\|\,.$$

For the case of a known point $\mathbf{q}_{foe}$ (Eq. 6.4), enforcing that the epipolar lines all have their z coordinates equal to 1 gives the equation

$$\begin{aligned} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \mathbf{p}_{xyz} &= 1 \\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \left(\mathbf{R}_{foe}^T \mathbf{T}_{foe}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & k \end{bmatrix} \mathbf{q}_{foe} \right) &= 1 \end{aligned}$$

which can be simplified to

$$(\mathbf{T}_{foe}\mathbf{c}_3) \cdot (\mathbf{A}\,\mathbf{q}_{foe}) + k(\mathbf{T}_{foe}\mathbf{c}_3) \cdot (\mathbf{B}\,\mathbf{q}_{foe}) = 1$$

where $\mathbf{c}_3$ is the third column of rotation matrix $\mathbf{R}_{foe}$ and $\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The solution is then

$$k = \frac{1 - (\mathbf{T}_{foe}\mathbf{c}_3) \cdot (\mathbf{A}\,\mathbf{q}_{foe})}{(\mathbf{T}_{foe}\mathbf{c}_3) \cdot (\mathbf{B}\,\mathbf{q}_{foe})}.$$

It should be noted that the denominator can never be zero because of Eq. 6.11 and the fact that $\mathbf{T}_{foe}\mathbf{c}_3$ can never be zero, nor non-zero and orthogonal to $\mathbf{B}\,\mathbf{q}_{foe}$.

6.3.4 Common angle interval

In general, a rectified image does not span the whole cylinder. The common angle interval is the interval that yields all common epipolar lines between two views. In order to control the number of epipolar lines extracted, it is important to determine this interval for each image.

Notice that the rectification process implicitly guarantees that a pair of corresponding epipolar lines have the *same* angle on their respective cylinder, and therefore the same row in the rectified images. The concern here is to determine the angle interval of epipolar lines effectively present in both images.

It can be shown that if a rectified image does *not* span the whole cylinder, then the extremum angles are given by two corners of the image. Based on this fact, it is sufficient to compute the angle of the four corners and one point between each pair of adjacent corners. By observing the ordering of these angles and taking into account the periodicity of angle measurements, it is possible to determine the angle interval for one image.

Given the angle intervals computed for each image separately, their intersection is the common angle interval sought. The subsequent stereo matching process has only to consider epipolar lines in that interval. Notice that if the field of view of the cameras do not overlap, this intersection will be empty.

6.3.5 The case of uncalibrated cameras

Until now, it was always assumed that the cameras where calibrated, i.e. their internal parameters are known. The parameters are the principal point (optical axis), focal length and aspect ratio. More generally, we can represent all these parameters by a 3×3 upper triangular matrix. In this section, we assume that only the *fundamental matrix* is available. This matrix effectively combines the internal parameters with the camera motion (external parameters) in a single matrix.

The *fundamental matrix* $\mathbf{F}$ defines the epipolar relation between points $\mathbf{p}_a$ and $\mathbf{p}_b$ of the images as

$$\mathbf{p}_b^T \cdot \mathbf{F} \cdot \mathbf{p}_a = 0. \tag{6.12}$$

It is straightforward to extract the focus of expansion for each image by noticing that all points of one image must satisfy Eq. 6.12 when the point selected in the other image is its **foe**. More precisely, the relations for $\mathbf{foe}_a$ and $\mathbf{foe}_b$ are

$$\begin{aligned} \mathbf{p}_b^T \cdot \mathbf{F} \cdot \mathbf{foe}_a &= 0 \quad \forall \mathbf{p}_b \\ \mathbf{foe}_b^T \cdot \mathbf{F} \cdot \mathbf{p}_a &= 0 \quad \forall \mathbf{p}_a \end{aligned}$$

which yield the homogeneous linear equation systems

$$\mathbf{F} \cdot \mathbf{foe}_a = 0 \tag{6.13}$$

$$\mathbf{F}^T \cdot \mathbf{foe}_b = 0 \tag{6.14}$$

which are easily solved.

At this point, it remains to show how to derive the constituents of matrix $\mathbf{L}_{foe}$ of Eq. 6.3 from the *fundamental matrix* $\mathbf{F}$. These are the matrices $\mathbf{S}_{foe}$, $\mathbf{R}_{foe}$, and $\mathbf{T}_{foe}$.

The transformation $\mathbf{T}_{foe;a}$ can be directly obtained from Eq. 6.6, using $\mathbf{foe}_a$ obtained in Eq. 6.13. Symmetrically (using Eq. 6.14) we obtain

$$T_{foe;b} = \begin{bmatrix} n(\mathbf{foe}_b) \\ n(\mathbf{z} \times \mathbf{foe}_b) \\ n(\mathbf{foe}_b \times (\mathbf{z} \times \mathbf{foe}_b)) \end{bmatrix}.$$

The rotation matrix $\mathbf{R}_{foe}$ is computed from the **foe** (which is readily available from the *fundamental matrix* $\mathbf{F}$) and the transform matrix $\mathbf{T}_{foe}$, exactly as described in Sec. 6.3.2.

Since the scaling matrix $\mathbf{S}_{foe}$ is directly computed from the value of rotation matrix $\mathbf{R}_{foe}$ and transform $\mathbf{T}_{foe}$, it is computed exactly as described in Sec. 6.3.3.

The rectification method is applicable regardless of the availability of the internal camera parameters. However, without these parameters, it is impossible to determine the minimum and maximum disparity interval which is of great utility in stereo matching. In this paper, all the results where obtained with known internal parameters.

6.3.6 Epipolar distortion and image size

The distortion induced by the rectification process in conjunction with the resampling of the original image can create a loss of pixel information, i.e. pixels in the original image are not accounted for and the information they carry is simply discarded during resampling. We measure this loss along epipolar lines, since it is along these lines that a subsequent stereo process will be carried out. To establish a measure of pixel information loss, we consider rectified epipolar line segments of a length of one pixel and compute the length L of the original line segment that is remapped to it. For a given length L, we define the loss as

$$\text{loss}(L) = \begin{cases} 0 & L \leq 1 \\ 1 - 1/L & L > 1 \end{cases} .$$

A shrinking of original pixels (i.e. $L > 1$) creates pixel information loss while a stretching (i.e. $L < 1$) simply reduces the density of the rectified image. For a whole image, the measure is the expected loss over all rectified epipolar lines, broken down into individual one pixel segments.

The fundamental property of cylindrical rectification is the conservation of the

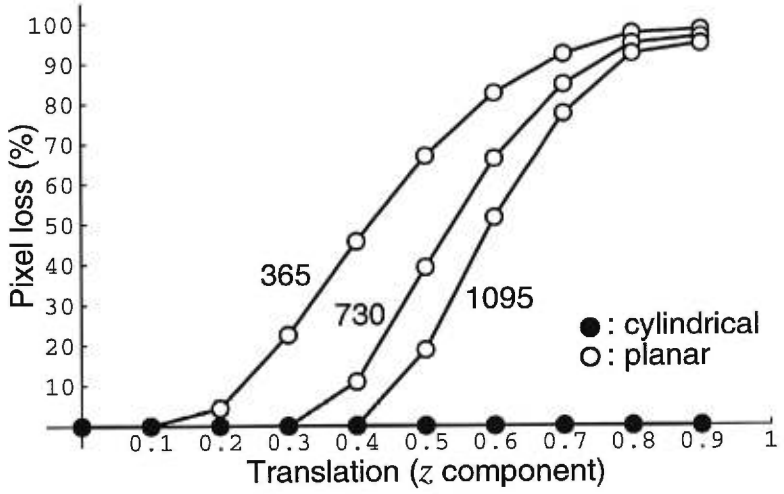

Figure 6.4. Pixel loss as a function of camera translation $T = (1, 0, z)$. Rectified image width is 365, 730 and 1095 pixels for an original width of 256 pixels.

length of epipolar lines. Since pixels do not stretch or shrink on these lines, no pixel information is lost during resampling. For planar rectification, the length of epipolar lines is not preserved. This implies that some pixel loss will occur if the rectified image size is not large enough. In Fig. 6.4, three different rectified image widths (365, 730, 1095 pixels) were used with both methods, for a range of camera translations $T = (1, 0, z)$ with a z component in the range $z \in [0, 1]$. Cylindrical rectification shows no loss for any camera motion and any rectified image width[2]. However, planar rectification induces a pixel loss that depends on the camera geometry. To compensate for such a loss, the rectified images have to be enlarged, sometimes to the point where they become useless for subsequent stereo processing. For a z component equal to 1 (i.e. $T = (1, 0, 1)$), all pixels are lost, regardless of image size.

[2] The minimum image width that guarantees no pixel loss is equal to $\sqrt{w^2 + h^2}$ for an original image of size (w, h)

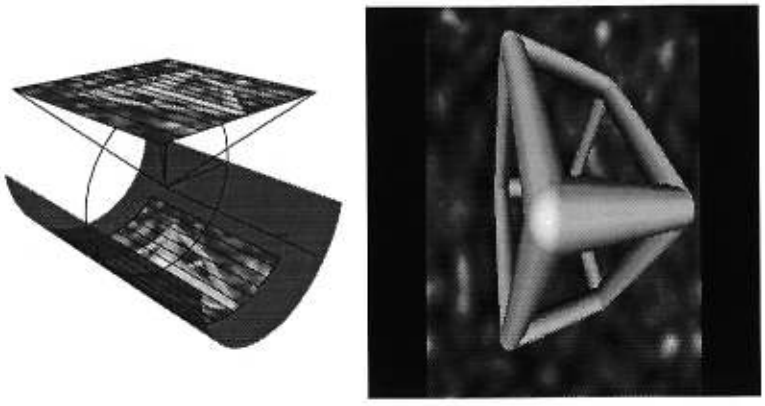

Figure 6.5. Image cube rectified. Horizontal camera motion (foe $= (1, 0, 0)$). A row represents an individual epipolar line.

6.4 Experiments and results

Some examples of rectification applied to different camera geometries are illustrated in this section. Fig. 6.5 presents an image plane and the rectification cylinder with the reprojected image, for a horizontal camera motion. In this case, the epipolar lines are already aligned. The rows represent different angles around the cylinder, from 0° to 360°. The image always appears twice since every cylinder point is *projective* across the cylinder axis. The number of rows determines the number of epipolar lines that are extracted from the image.

Fig. 6.6 depicts a camera geometry with forward motion. The original and rectified images are shown in Fig. 6.7 (planar rectification cannot be used in this case). Notice how the rectified displacement of the sphere and cone is purely horizontal, as expected.

Fig. 6.8 depicts a typical camera geometry, suitable for planar rectification, with rectified images shown in Fig. 6.9. While the cylindrical rectification (images C_1, C_2 in Fig. 6.9) introduces little distortion, planar rectification (images P_1, P_2) significantly distorts the images, which are also larger to compensate for pixel information loss.

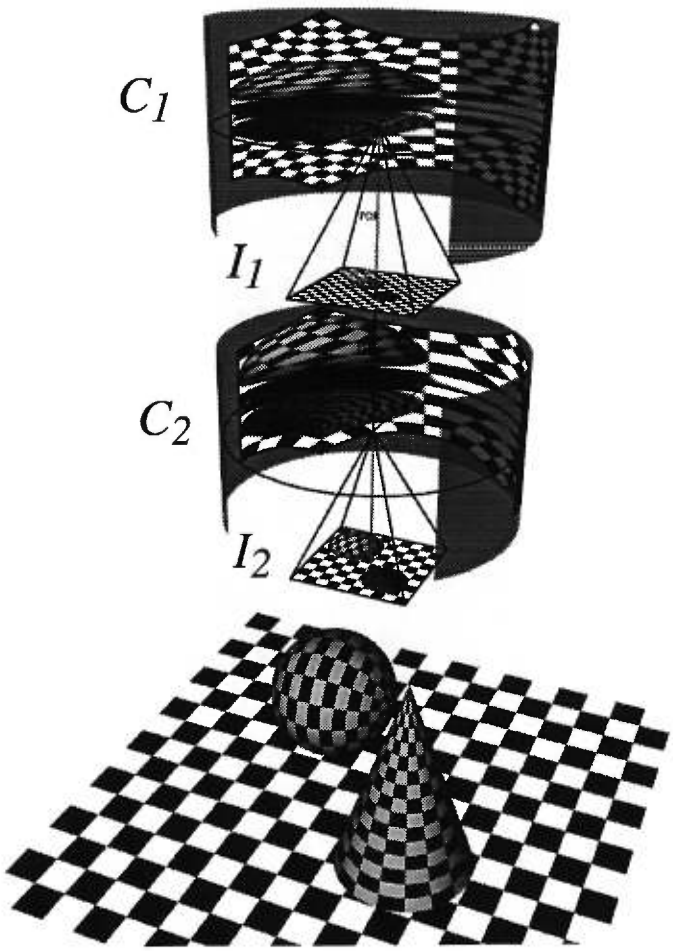

Figure 6.6. Forward motion. A sphere and cone are observed from two cameras displaced along their optical axis. The original images I_1, I_2 are remapped onto the cylinder as C_1, C_2.

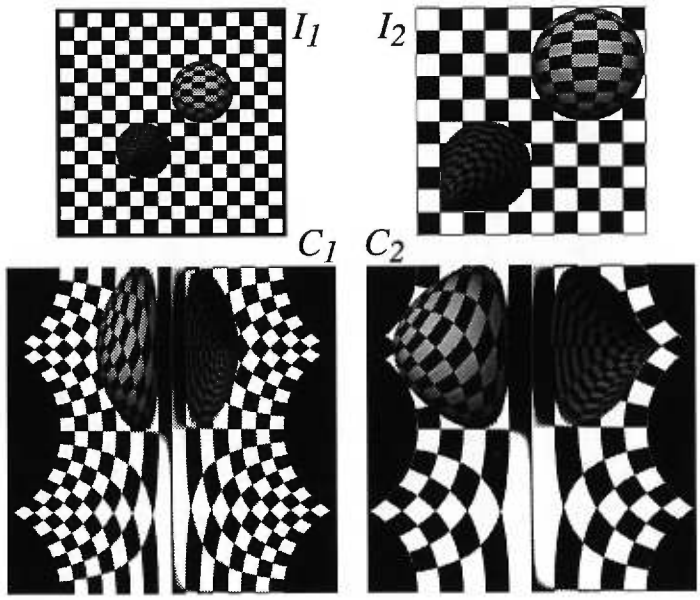

Figure 6.7. Rectification of forward camera motion. The images I_1, I_2 are shown with their cylindrical rectification C_1, C_2. The rectified image displacements are all horizontal.

Examples where the **foe** is inside the image are obtained when the forward component of the motion is large enough with respect to the focal length (as in Fig. 6.7). It is important to note that planar rectification always yields an unbounded image (i.e. infinite size) for these cases and thus cannot be applied.

The execution time for both methods is very similar. For many camera geometries, the slight advantage of planar rectification relating to the number of matrix computations is overcome by the extra burden of resampling larger rectified images to reduce pixel loss.

6.5 Conclusion

We presented a new method, called *cylindrical rectification*, for rectifying stereoscopic images under arbitrary camera geometry. It effectively remaps the images onto the

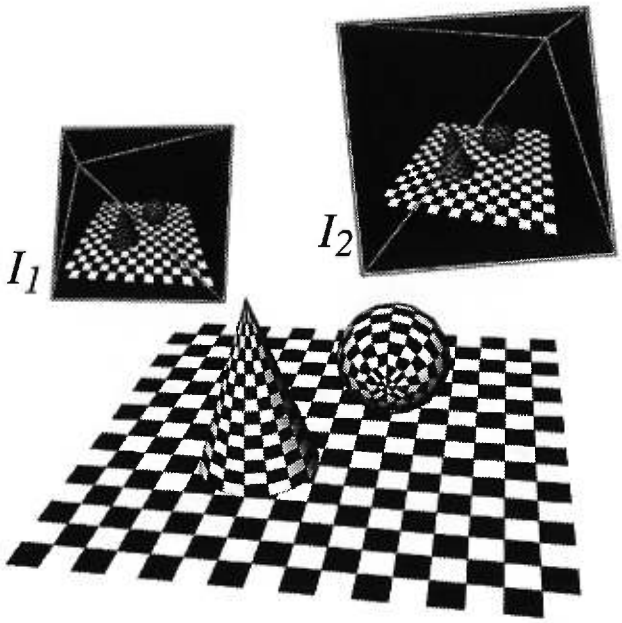

Figure 6.8. Camera geometry suitable for planar rectification. I_1, I_2 **are the original images.**

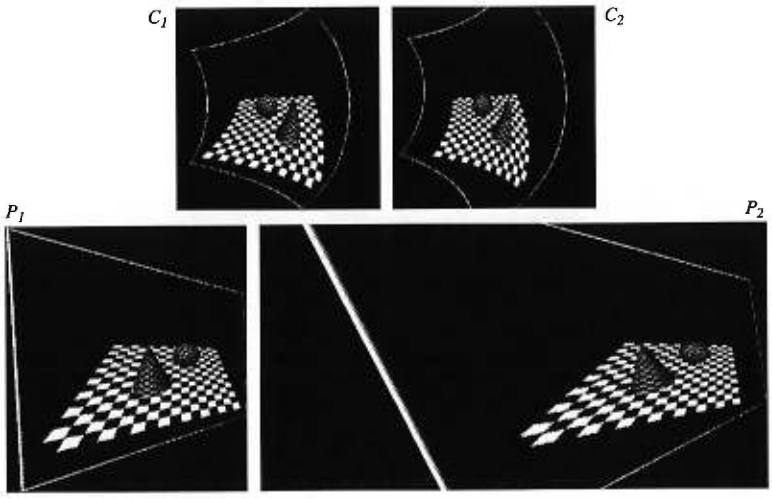

Figure 6.9. Rectified images. Cylindrical rectification (C_1, C_2) **and planar rectification** (P_1, P_2)

surface of a unit cylinder whose axis goes through both cameras optical centers. It applies a transformation in projective space to each image point. A single linear transformation is required per epipolar line to rectify. While it does not preserve arbitrary straight lines, it preserves epipolar line lengths, thus ensuring minimal loss of pixel information. As a consequence of allowing arbitrary camera motions, the rectified images are always bounded, with a size independent of camera motion.

The approach has been implemented and used successfully in the context of stereo matching [21], ego-motion estimation [69] and three-dimensional reconstruction, and has proved to provide added flexibility and accuracy at no significant cost in performance.

Chapitre 7

LE PROBLÈME DE LA MISE EN CORRESPONDANCE

Ce chapitre présente une introduction à l'analyse stéréoscopique, c'est-à-dire au problème de la mise en correspondance avec géométrie de caméra connue. Il sert d'entrée en matière à la nouvelle méthode de mise en correspondance par flot maximum présentée au chapitre 8.

Les concepts de base de la stéréoscopie sont issus de la géométrie épipolaire, telle que décrite au Chapitre 2. Traditionnellement, l'analyse stéréoscopique s'est surtout intéressée à reproduire la vision stéréoscopique humaine (voir Marr et Poggio [56]) et sa capacité de perception de la profondeur. Ainsi, la plupart des algorithmes ont été développés en assumant deux caméras séparées horizontalement comme les yeux humains. De même, la convergence des axes optiques, ou la fixation des yeux sur un point fixe, est aussi une géométrie qui a reçu beaucoup d'attention.

Plus récemment, l'élargissement des applications de la stéréoscopie a rendu nécessaire l'adaptation de ces algorithmes traditionnels aux géométries arbitraires de caméras. La rectification d'images, présentée aux chapitres 5 et 6, a rendu possible cette adaptation. Néanmoins, la mise en correspondance reste toujours limitée à l'utilisation de deux caméras.

Les besoins de l'infographie ont ajouté une nouvelle dimension au problème de la stéréoscopie, celui de la reconstruction d'une scène à partir de vues multiples. Cette nouvelle application a forcé une reformulation plus générale du problème de la mise en correspondance. Une telle formulation, incluant notamment le concept de *volume de reconstruction*, est utilisée ci-après au chapitre 8.

Le reste de ce chapitre propose une introduction aux différents algorithmes stéréoscopiques, en mettant en relief leurs caractéristiques principales. Ainsi, on classe

ces algorithmes selon les caractéristiques suivantes:

- Choix des primitives à mettre en correspondance
- Hypothèses sur la nature de la scène (objets solides, opaques, mats, etc.)
- Fonction de coût de correspondance à minimiser
- Méthode utilisée pour la minimisation
- Contraintes sur le nombre et la géométrie des caméras
- Volume de reconstruction

Chacune de ces caractéristiques fera l'objet d'une des sections qui suivent.

7.1 Choix des primitives à mettre en correspondance

Le choix du type de primitive est déterminant dans le processus de mise en correspondance. En effet, les différentes primitives ont un *contenu informationnel* très différent. Par exemple, l'intensité d'un pixel est la primitive la plus simple qui soit. Elle offre la plus grande densité possible, mais contient peu d'information utilisable pour la mise en correspondance. Inversement, les contours sont des primitives clairsemées qui contiennent beaucoup d'information (longueur, orientation, intensité de chaque côté, etc.).

Le processus de formation des images stéréoscopiques introduit des variations d'intensité non reliées à la profondeur qui peuvent affecter les primitives. Le bruit, les variations géométriques liées à la perspective, les occlusions et la spécularité des surfaces sont autant de facteurs qui détériorent le contenu informationnel et donc l'utilité des primitives. Alors que les primitives clairsemées peuvent être rendues relativement invariantes à ces facteurs, il est impossible de faire de même pour les

primitives denses. Ce dilemme a fait émerger deux tendances opposées, les approches par points saillants (ou *feature-based*) et par régions (ou *area-based*), qui utilisent respectivement des points saillants, peu denses mais robustes, ou des régions de pixels, denses mais peu robustes. En général, plus une primitive est dense, moins elle contient d'information utile. Dans la majorité des applications, un champ de profondeur dense est requis et la primitive utilisée est l'intensité du pixel. Le manque d'information de cette primitive est compensé en partie par l'application d'une contrainte de lissage.

Notons qu'une primitive à la fois dense et au contenu informationnel élevé incorpore forcément de l'information provenant de ses voisins. Ceci équivaut à imposer implicitement une contrainte de lissage. Le degré de lissage est directement relié à la taille de ce voisinage, dont la détermination optimale constitue un problème difficile. Un exemple d'une telle primitive serait l'ensemble des pixels voisins à une distance d'au plus w. La fonction de coût utilisée peut être la corrélation des ensembles de pixels voisins, et la technique de minimisation la recherche directe (i.e. sans contrainte de lissage), comme présenté dans le chapitre 7 de Shirai [76]. Malheureusement, une seule largeur w ne convient généralement pas à toute l'image. Ainsi, plusieurs méthodes [11, 45] proposent de varier la taille du voisinage w localement en choisissant une fenêtre plus large dans les zones plus lisses et une plus petite là où il y a des discontinuités de profondeur.

Il existe un équilibre délicat entre le contenu informationnel des primitives et l'utilisation d'une contrainte de lissage. Malgré tout, le désir d'obtenir un champ de profondeur dense impose l'utilisation des primitives à faible contenu informationnel et donc d'une forte contrainte de lissage.

7.2 Hypothèses sur la nature de la scène

Un certain nombre d'hypothèses sur la nature de la scène sont toujours utilisées, même implicitement, lors de la mise en correspondance. Les principales, décrites dans cette

section, supposent des objets solides, opaques et mats ainsi que des sources lumineuses fixes (voir aussi la section 3.1).

Les objets de la scène étant assumés solides, ils ne se déforment pas d'une vue à l'autre. Cette contrainte est automatiquement respectée lorsque les vues sont prises simultanément à partir de plusieurs caméras, plutôt qu'espacées dans le temps (c'est-à-dire une caméra qui se déplace). Cette hypothèse garantit qu'un déplacement observé entre deux vues n'est causé que par l'effet du déplacement de caméra lui-même.

Les objets sont aussi assumés opaques. On peut ainsi garantir que l'intensité d'un pixel d'une image ne provient que d'un seul objet, ce qui simplifie la mise en correspondance. Si cette hypothèse ne s'applique pas, la transparence doit être modélisée et de ce fait un pixel donné peut posséder des disparités multiples, ce qui cause une grande difficulté. Une telle modélisation est présentée dans Shizawa [77].

De plus, les objets sont assumés parfaitement mats, ou *Lambertiens*. Ceci implique qu'un point sur un objet se projette toujours avec la même intensité, peut importe l'angle de vue. Deux pixels mis en correspondance ont donc la même intensité. Cette propriété est essentielle à la majorité des algorithmes stéréo. Puisque les objets sont rarement parfaitement mats, des efforts ont été consacrés à éliminer cette contrainte. Par exemple, dans Bhat et Nayar [7], les réflexions spéculaires sont modélisées pour permettre d'améliorer le choix des angles de vue des caméras.

Les sources lumineuses sont assumées fixes. Cette hypothèse contribue elle aussi à garantir l'intensité identique des pixels correspondants.

7.3 Fonction de coût de correspondance à minimiser

Le processus de mise en correspondance est généralement exprimé sous forme d'une minimisation d'une fonction de coût, le *coût de correspondance*, sur les intervalles de disparité permis. Le choix de la fonction de coût de correspondance est élaboré en fonction des hypothèses et contraintes utilisées. Ces fonctions vont du plus simple,

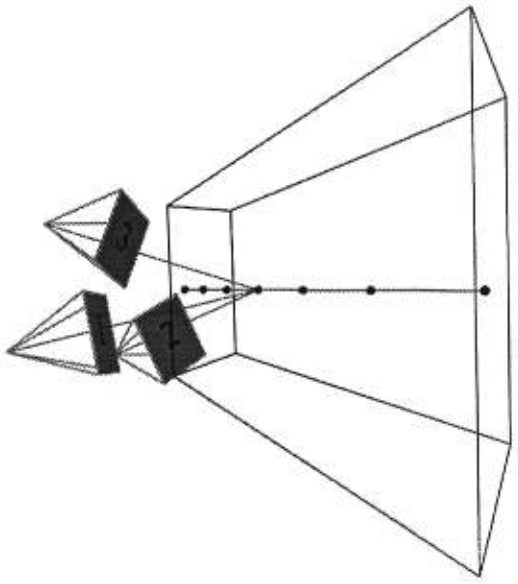

Figure 7.1. Reprojection d'un point 3D. Un point tridimensionnel peut être reprojeté par plusieurs caméras à géométrie arbitraire.

assumant la préservation des intensités entre les images, au plus complexe, en tenant compte des effets de réflexion spéculaire, de variations d'éclairage, d'occlusions, etc...

Un concept de base est essentiel à l'élaboration d'une fonction de coût : il est possible de reprojeter un point 3D donné sur le plan image de n'importe quelle caméra, comme l'illustre la figure 7.1. C'est de cette façon qu'une correspondance, à laquelle est associé un point 3D, est reprojetée sur chaque image disponible pour constituer un ensemble de primitives (i.e. l'intensité du pixel) qui sera ensuite mis à contribution dans le calcul de la fonction de coût et donc de la pertinence de cette correspondance. La possibilité de reprojeter sur un nombre quelconque d'images sera essentielle lors de la généralisation vers la stéréoscopie à caméras multiples.

Si on se réfère au chapitre 2, il est très facile de reprojeter un point image d'une caméra vers une autre, pour une profondeur donnée.

Soit $\mathbf{p}'_a$ un point de l'image de la caméra A auquel on a associé une disparité d

(ou $\frac{1}{z}$ pour une profondeur z). Ce point se reprojette au point $\mathbf{p}'_b(d)$ dans l'image de la caméra B selon l'équation 2.10 qui se généralise au point $\mathbf{p}'_i(d)$ dans l'image d'une caméra i, choisie parmi l'ensemble des caméras $\mathcal{V} = \{A, B, C, D, \ldots\}$, pour donner un ensemble $\mathcal{P}(\mathbf{p}'_a, d)$ de points reprojetés

$$\begin{aligned} \mathcal{P}(\mathbf{p}'_a, d) &= \{\mathbf{p}'_i(d) : i \in \mathcal{V}\} \\ &= \{\mathbf{J} \cdot \mathbf{W}_i \cdot \mathbf{W}_A^{-1} \cdot [\mathbf{p}'_a; d] : i \in \mathcal{V}\} \end{aligned}$$

où $\mathbf{W}_A$ est la matrice de passage de la caméra A et où $\mathbf{W}_i$ est celle de la caméra i, avec $i \in \mathcal{V}$.

La fonction de coût $F(\mathbf{p}'_a, d)$ d'une correspondance peut être définie comme

$$F(\mathbf{p}'_a, d) = f(\ \mathcal{P}(\mathbf{p}'_a, d)\ )$$

où $f(\cdot)$ est une fonction positive de l'ensemble des points reprojetés. Un exemple très simple d'une telle fonction consisterait à calculer la variance des intensités des points reprojettés, c'est-à-dire

$$f(\ \mathcal{P}(\mathbf{p}'_a, d)\ ) = \text{variance}\,\{I_i(\ \mathbf{p}'_i(d)\ ) : i \in \mathcal{V}\}$$

où I_i est l'image fournie par la caméra i. Cette fonction ne tient pas compte de nombreux facteurs, comme les occlusions ou les réflexions spéculaires. Malgré tout, elle sera utilisée avec succès par notre nouvelle méthode de mise en correspondance, ce qui tend à démontrer que même les fonctions de coût les plus simples sont utiles, si on les utilise en conjonction avec des contraintes fortes, comme la contrainte de lissage.

7.4 Méthode utilisée pour minimiser la fonction de coût

Cette section décrit les méthodes les plus populaires utilisées pour la mise en correspondance. On peut classer grossièrement les techniques de mise en correspondance par la localité des calculs effectués, comme l'illustre la figure 7.2.

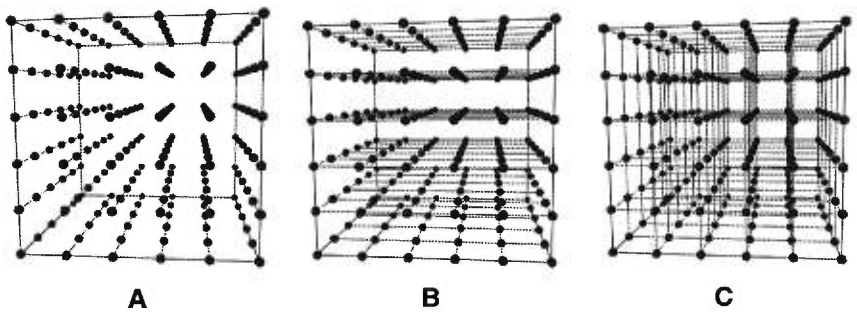

Figure 7.2. Différentes approches de mise en correspondance. Les points dans les volumes de reconstruction représentent les correspondances possibles. Les liens représentent la dépendance. Recherche directe (A), épipolaire (B), et globale (C)

7.4.1 Recherche directes

L'approche la plus simple, la recherche directe, utilise une fonction de coût de correspondance qui est indépendante de la disparité des pixels voisins, comme l'illustre la figure 7.3. La minimisation procède donc indépendamment en chaque pixel, comme l'illustre la figure 7.4, et peut être fortement parallélisée, il va de soi. Le problème de la recherche directe est qu'il est très difficile d'imposer une contrainte de lissage sur les disparités. Les formulations qui imposent ces contraintes sont généralement affectées par un lissage exagéré des discontinuités de profondeur qui se doivent d'être préservées.

La plupart des algorithmes stéréoscopiques très rapides ou cablés (*hardware*) sont du type direct, comme dans Raffo [64] et Kanade *et al.* [46]. Par ailleurs, Boykov *et al.* [11] propose d'utiliser la *disparité potentielle* des pixels voisins, plutôt que la vraie disparité, qui n'est pas disponible. Cette disparité potentielle est calculée pour chaque pixel indépendamment et donne pour chacun une sélection de disparités considérées plausibles. La disparité finale d'un pixel est celle qui est aussi plausible chez le plus grand nombre de voisins de ce pixel.

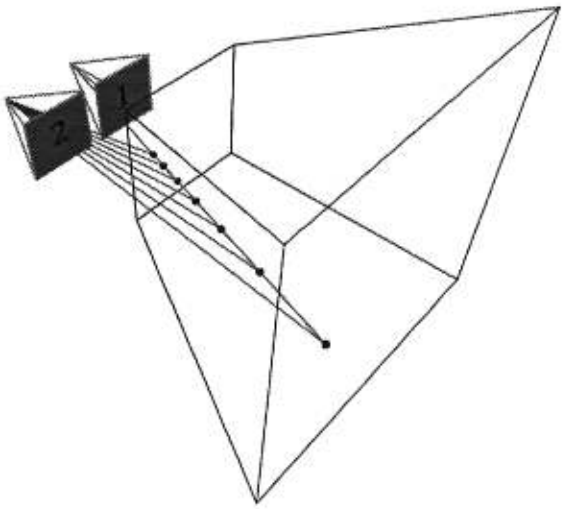

Figure 7.3. Géométrie stéréoscopique traditionnelle. Recherche directe. Le déplacement relatif des deux caméras est horizontal et le volume de reconstruction correspond au volume de vision de la caméra 1.

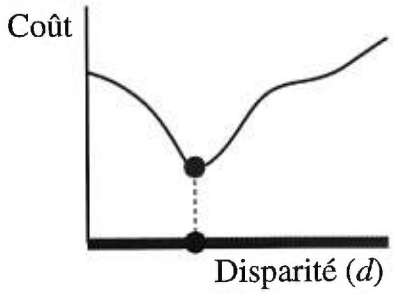

Figure 7.4. Recherche directe. Pour chaque pixel, la mise en correspondance s'établit par la minimisation d'une fonction de coût sur l'intervalle de disparité d.

7.4.2 Recherche globale

Pour remédier aux problèmes de la recherche directe, la fonction de coût doit incorporer un certain degré de dépendance entre les pixels voisins. En ce sens, la recherche globale offre un maximum de flexibilité et de puissance en permettant n'importe quelle forme de dépendance. La minimisation procède alors globalement et devient extrêmement difficile à résoudre, la plupart du temps. Un exemple de recherche globale serait les méthodes de recuit simulé (*simulated annealing*), typiquement utilisées pour solutionner les champs aléatoires de Markov, comme dans [5, 14, 52, 73]. Le recuit simulé, qui converge théoriquement vers la solution optimale, est en pratique trop lent pour être vraiment utilisable. Les méthodes utilisant la diffusion, comme dans Shah [75], leur sont aussi très apparentées.

D'autres méthodes à caractère global utilisent les réseaux de neurones [57] et la programmation génétique [48]. Ces méthodes n'ont pas réussi jusqu'à présent à démontrer une supériorité mesurable sur les autres méthodes.

Un autre exemple est celui du système de particules orientées présenté par Fua [27]. Par itérations successives, les particules sont alignées pour éventuellement représenter des surfaces. Les particules sont initialement disposées selon des reconstructions stéréo imprécises obtenues au préalable.

Certains, comme Scheuing et Niemann [74], suggèrent l'utilisation du flux optique pour calculer la profondeur. Bien que le calcul de la disparité à partir du flux optique soit presque trivial, l'obtention du flux optique lui-même est considérée comme plus difficile que la mise en correspondance stéréoscopique, parce que la contrainte épipolaire ne peut plus être utilisée.

La méthode de flot maximum présentée dans cette thèse au chapitre 8 est aussi une méthode globale, mais contrairement à toutes les autres méthodes, elle se calcule efficacement et obtient toujours la solution optimale. Une discussion de l'efficacité du calcul du flot maximum est présentée dans Goldberg et Rao [30] et Goldberg [29].

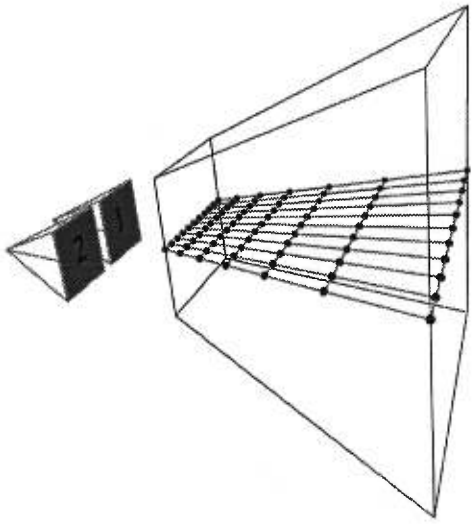

Figure 7.5. Recherche épipolaire. La mise en correspondance s'éffectue par programmation dynamique sur une ligne épipolaire complète à la fois.

7.4.3 Recherche épipolaire

Pour palier aux difficultés associées à la minimisation globale, une variante très efficace, la recherche épipolaire a été utilisée par [21, 53, 59]. Ici, la fonction de coût ne possède de dépendance que dans une seule direction, le long des lignes épipolaires (voir la figure 7.5). Comme l'illustre la figure 3.1, le long d'une ligne épipolaire la contrainte de lissage se transforme et devient une contrainte d'ordre. En effet, un point qui respecte la contrainte d'ordre ne peut se déplacer très loin de ses voisins et sa profondeur doit donc leur être similaire, ce qui constitue en pratique une manifestation de la contrainte de lissage. La propriété extraordinaire de la contrainte d'ordre est qu'elle permet d'utiliser la programmation dynamique pour la minimisation, méthode très efficace et optimale. Malheureusement, le problème des méthodes basées sur la programmation dynamique est que leurs solutions possèdent des variations exagérées (ou *streaking*) entre les lignes épipolaires, puisque aucun lissage n'est imposé entre ces lignes. Certaines solutions approximatives ont été tentées, avec des

résultats mitigés, par Ohta et Kanade [59] avec l'utilisation des segments verticaux dans l'images, par Belhumeur [6] avec le réajustement aléatoire de lignes épipolaires (ou *iterative stochastic dynamic programming*), ou par Cox *et al.* [21] avec une seconde étape de mise en correspondance qui raffine les correspondances originales.

7.4.4 Domaine des fréquences

Une quatrième approche de la mise en correspondance, qui n'est pas incluse dans l'illustration de la figure 7.2, est celle de la transformation du problème dans le domaine des fréquences. Ainsi, comme présenté dans Smith *et al.* [79] et Yeshurun *et al.* [86], on peut considérer que la seconde image d'une paire d'images stéréoscopiques mises côte à côte est en fait un écho de la première image, avec des variations de phases reliées à la disparité entre les images. Ainsi, on peut utiliser l'analyse de Fourier pour récupérer ces variations. Une forte limitation de cette approche vient de ce que seule la distribution des disparités est obtenue et qu'aucune localisation n'est possible.

Pareillement, Chen et Bovic [15] associe la disparité à un changement de phase et propose de mesurer ces différences en utilisant une banque de filtres de Gabor de différentes fréquences et résolutions.

7.5 Contraintes sur le nombre et la géométrie des caméras

Tous les algorithmes stéréoscopiques possèdent un modèle du nombre de caméras et de leurs positions relatives. Les algorithmes supportent deux, trois, ou même quatre caméras et plus avec des déplacements relatifs allant du simple déplacement horizontal au déplacement complètement arbitraire. Les sections suivantes détaillent ces grandes familles d'algorithmes.

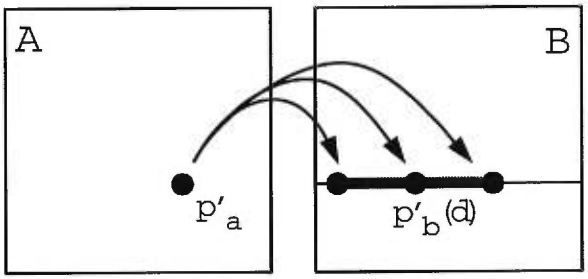

Figure 7.6. Stéréoscopie traditionnelle. Les images A et B sont issues d'un déplacement latéral des caméras. Un point $\mathbf{p}'_a = (x', y')$ se projette au point $\mathbf{p}'_b(d) = (x' + d, y')$.

7.5.1 Stéréoscopie traditionnelle

Traditionnellement, la stéréoscopie s'inspire du modèle biologique et utilise deux caméras déplacées horizontalement. Ce modèle, illustré à la figure 7.3, est mathématiquement très simple. En assumant un déplacement de caméra horizontal b, le déplacement relatif d'un point $\mathbf{p}'_a$ de la caméra A vers $\mathbf{p}'_b$ de la caméra B est dérivé des équations 2.13, 2.14 et 2.15 pour donner[1]

$$\mathbf{p}'_b = \mathbf{p}'_a + \mathbf{m} + k\ \mathbf{e} \qquad 0 \leq k \leq 1$$

avec

$$\begin{aligned} \mathbf{m} = \quad & \mathbf{p}'_b(d_{min}) - \mathbf{p}'_a \quad && = (0,0,0) \\ \mathbf{e} = \quad & \mathbf{p}'_b(d_{max}) - \mathbf{p}'_b(d_{min}) && = (b,0,0). \end{aligned}$$

La mise en correspondance, illustrée à la figure 7.6, s'effectue directement sur les lignes horizontales de pixels des images. L'exemple classique est donné par Marr et Poggio [56]. De même, Ohta et Kanade [59] et Cox et al. [21] constituent aussi des exemples de stéréoscopie traditionnelle, mais se distinguent néanmoins de l'aspect *humain* par leur approche plus *computationnelle.*

[1] On suppose aussi une distance focale $f = 1$ et un intervalle de disparité $[d_{min}, d_{max}] = [0,1]$ correspondant à l'intervalle de profondeur $[\infty, 1]$.

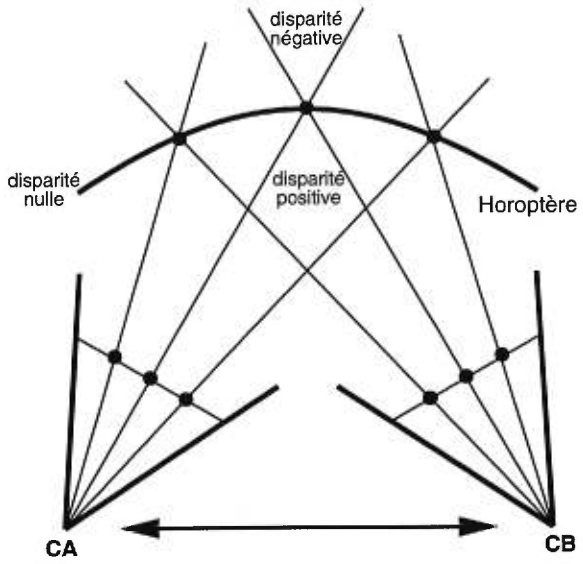

Figure 7.7. Géométrie de caméra convergente. L'horoptère est la zone ou le déplacement apparent est nul.

7.5.2 Géométrie de caméra convergente

La géométrie convergente est directement inspirée du modèle humain. Les caméras sont séparées horizontalement et tournent autour d'un axe de façon à pouvoir *fixer* un objet particulier de la scène, c'est-à-dire placer un objet dans les images de façon à ne pouvoir observer aucun déplacement apparent de cet objet entre les deux images, comme l'illustre la Figure 7.7. L'avantage de cette représentation est qu'une zone de disparité nulle, dite *horoptère*, est positionnée au milieu de la scène et les objets qui sont situés en avant et en arrière de cette zone présentent respectivement des disparités positives et négatives. Cette géométrie maximise l'utilité d'un intervalle de disparité fixe en déplaçant la zone de fixation plutôt qu'en variant cet intervalle de disparité. Ceci constitue un avantage certain pour les systèmes biologiques qui ne disposent que d'un nombre restreint de détecteurs de disparité, n'offrant en fait qu'un intervalle de disparité fixe. En utilisant la fixation, un système biologique peut

donc estimer un très large éventail de profondeurs, tout en utilisant un intervalle de disparité qui ne représente qu'une fraction de l'étendue réelle des profondeurs pouvant être reconstruites.

Plusieurs algorithmes traditionnels supportent la convergence en permettant les disparités négatives [21, 59]. Ils ne tiennent pas compte de la géométrie épipolaire qui établit des lignes épipolaires non horizontales, ce qui introduit une source importante d'erreur. Si l'on traite la convergence comme un déplacement arbitraire, il est possible de rectifier les images pour compenser cet effet.

7.5.3 Géométrie de caméra arbitraire

Dans le cas où deux caméras présentent un déplacement relatif arbitraire, on peut procéder à une étape de rectification, présentée aux chapitres 5 et 6, pour obtenir un alignement horizontal des droites épipolaires. Les images rectifiées peuvent être utilisées par un algorithme traditionnel, tel que décrit à la section 7.5.1, qui requiert l'alignement horizontal des droites épipolaires. Notons que la nature *artificielle* de l'alignement rectifié peut introduire des disparités négatives. Les algorithmes doivent être en mesure de mettre en correspondance sur ces intervalles et ne peuvent être restreints aux seules disparités positives.

7.5.4 Caméras multiples

Lorsque plus de deux caméras sont utilisées pour la mise en correspondance, l'information supplémentaire disponible pour améliorer la qualité de la solution pose certains problèmes. En effet, la géométrie épipolaire dont dépendent la plupart des algorithmes stéréoscopiques n'est définie que pour deux caméras. La contrainte d'ordre, si importante, n'est respectée que pour deux caméras à la fois, ce qui rend impossible l'utilisation de la recherche épipolaire.

Une approche simple est de choisir deux caméras comme *références*, comme dans

Cox [19]. On rectifie les images de ces deux caméras pour procéder à la mise en correspondance traditionnelle. L'information provenant de l'ensemble des caméras est utilisée dans la fonction de coût, comme nous l'avons présenté à la section 7.3. En effet, une paire de points mis en correspondance entre les deux caméras de référence correspond à un point 3D qui peut être reprojeté dans les images des autres caméras, pour ainsi donner une valeur de pixel supplémentaire qui peut être utilisée pour juger de la qualité de la paire de points de référence.

Le problème avec cette méthode est que le choix des deux caméras de référence a un impact important sur la solution, à cause de la géométrie épipolaire liée à ces caméras. Certains proposent de prendre tour à tour comme référence toutes les paires de caméras de la scène [44]. Bien que ceci améliore sensiblement la solution, cette approche n'est pas très efficace ou élégante.

Une approche intéressante est celle du *plan+parallaxe* [49]. On identifie dans la scène une surface plane dite de *référence*, visible de l'ensemble des caméras. La profondeur est ensuite exprimée par rapport à cette surface plutôt qu'avec le volume de visualisation d'une caméra. Ce cadre de référence global est imposé aux résultats des mises en correspondance qui sont effectuées entre deux caméras à la fois.

Cette idée de choisir un cadre de référence indépendant des caméras est essentiel lorsqu'il y a un grand nombre de caméras. Un façon simple de constituer un tel cadre est de créer une caméra supplémentaire *virtuelle*, qui ne contribue pas à la fonction de coût puisqu'elle ne fournit pas d'image mais qui définit par son volume de vision l'espace où s'effectuera la reconstruction. L'approche de flot maximum utilise une telle *caméra virtuelle* pour définir son volume de reconstruction.

7.6 Volume de reconstruction

Le concept de caméra *virtuelle* est lié à celui du *volume de reconstruction*, c'est-à-dire l'espace tridimensionnel qui contient tous les points reconstruits. Les figures 7.8 et

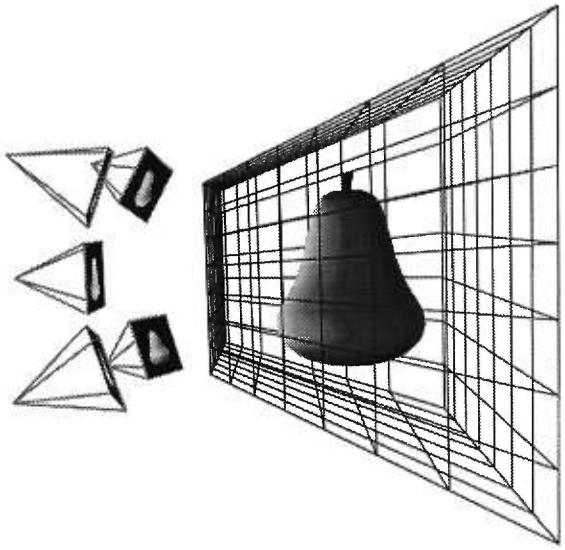

Figure 7.8. Volume de reconstruction. Le volume correspond au volume de vision d'une des caméras.

7.9 illustrent des volumes de reconstruction associés respectivement au volume de vision de caméras réelle et virtuelle.

La projection d'une caméra virtuelle n'a bas besoin d'être perspective comme à la figure 7.8. Elle peut aussi être orthographique, ce qui explique la forme cubique du volume de la figure 7.9. En fait, on verra que ce volume de reconstruction n'a pas besoin de correspondre au volume de vision d'une caméra, qu'elle soit virtuelle ou non. Il peut être d'une forme quelconque, mais doit respecter certains critères pour assurer qu'il est toujours possible de récupérer une surface tridimensionnelle (plus précisement, une *depth map*) valide:

- Le volume de reconstruction doit posséder une surface *devant* et une surface *derrière* disjointes.

- Il doit exister une bijection entre ces deux surfaces (i.e. chaque point de *devant* correspond à un point de *derrière*, et vice versa).

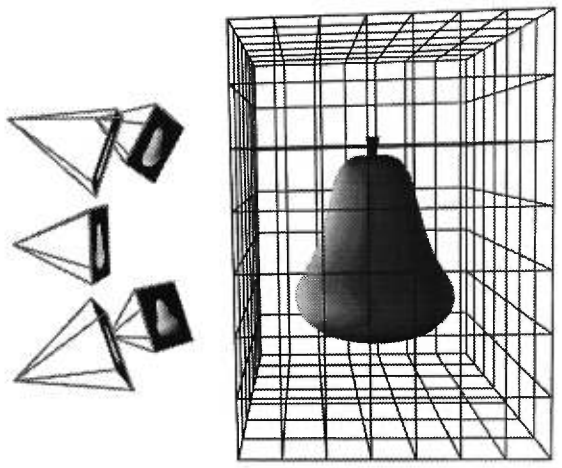

Figure 7.9. Volume de reconstruction arbitraire. Le volume, associé à une caméra *virtuelle*, peut être tout volume qui possède une surface frontale et arrière.

- Il existe aussi une association entre tout point du volume et une paire de points *devant* et *derrière*.

- La surface reconstruite sépare le volume de reconstruction en deux parties, une contenant le *devant* et l'autre le *derrière*.

Ainsi, si on définit respectivement le *devant* et le *derrière* comme les surfaces de profondeurs minimum et maximum de la scène, la mise en correspondance consiste alors à trouver une surface qui coupe le volume en deux tout en séparant le *devant* du *derrière*. Puisque le volume de reconstruction est défini dans l'espace projectif 3D, il est possible de représenter sans difficultés des volumes qui vont à l'infini.

En stéréoscopie traditionnelle à deux caméras, le volume de reconstruction est implicitement défini comme le volume de vision de la première caméra. Le *devant* et le *derrière* correspondent respectivement aux disparités maximum et minimum, ou de façon équivalente, aux profondeurs minimum et maximum.

En procédant à la mise en correspondance indépendamment sur chaque pixel pour

la recherche directe, ou sur chaque ligne épipolaire pour la recherche épipolaire, les méthodes traditionnelles calculent la surface de reconstruction en la subdivisant en morceaux individuellement reconstruits qui, une fois assemblés, composent la surface complète.

Jusqu'à tout récemment, trouver efficacement une surface globalement optimale semblait impossible (voir [52]). Seule une subdivision du problème en sous-problèmes indépendants, lignes épipolaires ou pixels individuels, pouvait permettre une solution efficace mais pas globalement optimale. La nouvelle méthode présentée au chapitre suivant possède cette propriété d'être globalement optimale et efficace à la fois.

Chapitre 8

STEREO WITHOUT EPIPOLAR LINES : A MAXIMUM-FLOW FORMULATION

Cet article [70] a été publié comme l'indique la référence bibliographique

Sébastien Roy et Ingemar J. Cox, A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem, *International Conference on Computer Vision (ICCV'98)*, Bombay, Indes, Janvier 1998, pages 492-499

Il est présenté ici dans sa version étendue qui a été accepté pour publication dans le journal scientifique *International Journal on Computer Vision.*

Abstract

This paper describes a new algorithm for solving the stereo correspondence problem by transforming it into a maximum-flow problem in a graph. This transformation effectively removes explicit use of epipolar geometry, thus allowing direct use of multiple cameras with arbitrary geometries. The maximum-flow, solved both efficiently and globally, yields a minimum-cut that corresponds to a disparity surface for the whole image at once. This global and efficient approach to stereo analysis allows the reconstruction to proceed in an arbitrary volume of space and provides a more accurate and coherent depth map than the traditional stereo algorithms. In particular, smoothness is applied uniformly instead of only along epipolar lines, while the global optimality of the depth surface is guaranteed. Results show improved depth estimation as well as better handling of depth discontinuities. While the worst case running time is $O(s^{1.5}d^{1.5}log(sd))$, the observed average running time is $O(s^{1.2}\ d^{1.3})$ for an image size of s pixels and depth resolution d.

8.1 Introduction

It is well known that depth-related displacements in stereo pairs always occur along lines associated with the camera motion, the epipolar lines. These lines reduce the stereo correspondence problem to one dimension and the ordering constraint allows dynamic programming to be applied [3, 21, 24, 59]. However, it is clear that this reduction to 1-d is an oversimplification of the problem, primarily required to enforce smoothness constraints in a computationally efficient way. The solutions obtained on consecutive epipolar lines can vary significantly and create artifacts across epipolar lines, especially affecting object boundaries that are perpendicular to the epipolar lines (e.g. vertical object boundary with horizontal epipolar lines).

In this paper, we address the full 2-d matching problem, eliminating the need for explicit epipolar lines and replacing the traditional ordering constraint with the more general *local coherence* constraint. To perform the global 2-d optimization, we cast the stereo correspondence problem as a maximum-flow problem in a graph and show how the associated minimum-cut can be interpreted as a disparity surface. While the theoretical worst case computational complexity is significantly higher for maximum-flow than dynamic programming, in practice, the average case performance is similar. We also show how this new paradigm can support both binocular and n-camera stereo configurations, as well as arbitrary 3-d reconstruction volumes.

There have been several earlier attempts to relate the solutions of consecutive epipolar lines matched with dynamic programming. In Ohta and Kanade [59], dynamic programming is used to first match epipolar lines and then iteratively improve the solutions obtained by using vertical edges as reference. In Cox *et al.* [21], a probabilistic approach is used to relate the individual matchings obtained by dynamic programming to improve the depth map quality. First, it proposes to improve a given epipolar line matching by using the previous line solution to improve its own solution. However, this introduces a non-desirable vertical asymmetry. A second ap-

proach is to iteratively improve each epipolar line solutions with its neighboring lines solution. While this *local* approach is not globally optimal, it provides an efficient way to introduce smoothness constraints across epipolar lines. In Belhumeur [6], a Bayesian approach to the stereo correspondence problem is described. The resulting optimization problem can be solved efficiently by using dynamic programming along epipolar lines, resulting in the same problem as [21, 59] of relating the independent solutions. It proposes a heuristic method called *iterated stochastic dynamic programming* that uses previously computed adjacent epipolar line solutions to iteratively improve randomly selected solutions. This approach is not globally optimal and furthermore introduces a large amount of smoothness that tends to blur depth discontinuities.

The concept of using maximum-flow appeared in Greig *et al.* [31] in the context of binary Markov Random Fields, where the each pixel of a binary image is given one of two labels. The maximum-flow formulation for more than two labels and a linear discontinuity cost was presented by Roy and Cox [70] in the context of stereoscopic correspondence. Recently, Ishikawa and Geiger [42] presented a similar method as Roy and Cox [70], but expressed in the context of Markov Random Fields and applied to image segmentation. Also, Boykov *et al.* [12] presented a Markov Random Field formulation with non-linear discontinuity costs that give rise to a minimum multi-way cut problem. They present an approximate method based on efficient maximum-flow steps applied to binary sub-problems.

Some multiple-cameras algorithms have been presented (see [19, 24, 46, 47]). In Cox [19], a pair of camera is used as a *reference* or base pair. Other cameras provide extra information to enrich the matching cost function of the reference camera pair. The matching then proceeds using dynamic programming as in Cox *et al.* [21]. In Kang *et al.* [47] and in Kanade *et al* [46], a multiple-camera real-time stereo system is presented. They use a single *reference* camera to perform the matching. All the other cameras provide the information pertinent to each possible depth of points in

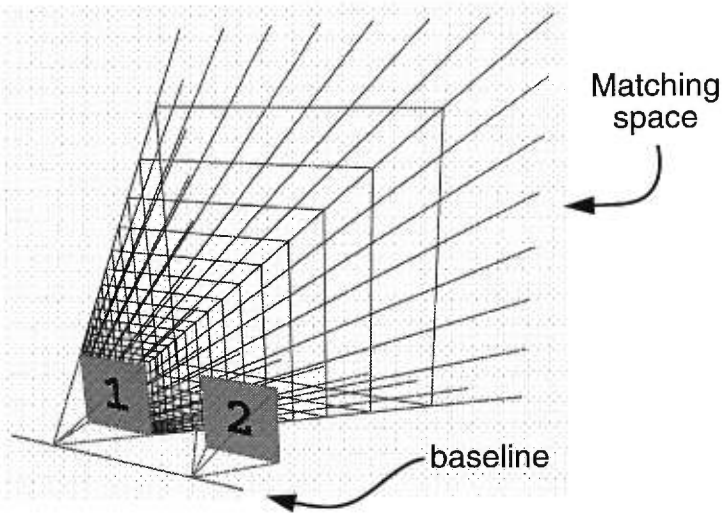

Figure 8.1. Standard stereo framework. Two horizontally separated cameras with parallel optical axes. The stereo matching volume is the viewing volume of camera 1.

the reference image. The depth is computed independently for each pixel, making it impossible to enforce a smoothness constraints between pixels. Instead, the images are low-pass filtered before the matching process. While this achieves some level of smoothness in the solution, it has the undesirable side effect of blurring the depth discontinuities.

Section 8.2 describes a general stereo framework to be used with multiple images from arbitrary viewpoints and arbitrary reconstruction volumes. It also describes a simple stereo matching cost function that supports those multiple images. In Section 8.3, the stereo problem is extended from matching single epipolar lines to solving for a full disparity map, making use of the *local coherence* constraint. In Section 8.4, the stereo matching problem is formulated as a maximum-flow problem. Details of the maximum-flow algorithm and performance issues are presented in Section 8.4.3. Experiments on both classic two-image and multiple-image stereo sequence are presented and discussed in Section 8.5.

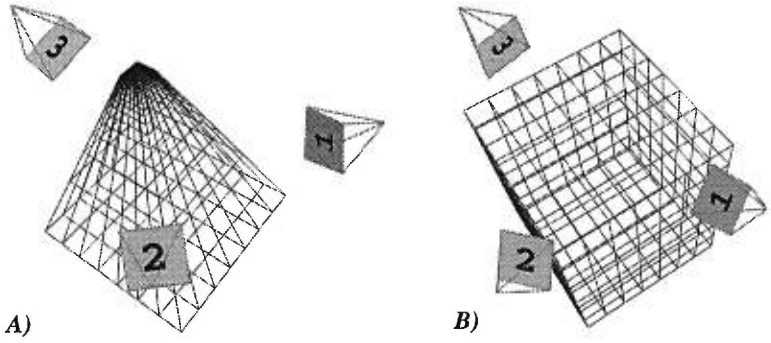

Figure 8.2. General stereo framework. Three cameras at arbitrary positions and orientations in 3D space, around two types of matching spaces, (A) with uniform disparity steps and (B) with uniform depth steps.

8.2 The Stereo Framework

This section describes a general stereo framework. It consists of two distinct parts. First, a volume of the 3D world is selected to constrain where the stereo matching actually occurs. Any resulting reconstructed surface must lie inside that volume. Second, each 3D world point inside the matching volume is projected onto the set of images to provide pixel intensity values. This information is then used to derive the matching cost necessary to perform stereo analysis. Even though it is performed inside a 3D volume of space, our algorithm always recovers a depth surface that cuts this volume in two parts, and not an arbitrary 3D shape inside the volume.

8.2.1 The Stereo matching space

The volume of 3D space that contains every possible depth surface is referred to as the *matching space* and has been used before in stereo (see Yang and Yuille [85] and Marr and Poggio [56]). This volume is discretized and searched by the stereo algorithm for an optimal depth surface. It is characterized by *front* and *back* regions that must be disjoint. By definition, a valid stereo depth surface always separates the

front and *back* of the matching space, and is therefore defined as a function of the *front* (or *back*). This definition of *valid* was chosen to enforce a dense reconstruction of disparity.

For standard stereo, the matching space is a truncated pyramid corresponding to the viewing volume of a camera (as in Figure 8.1). The front and back are simply the near and far planes of the viewing pyramid. Obviously, any valid surface (separating the front and near planes) will yield exactly one disparity value for every pixel of the selected camera.

In order to be solved using this stereo algorithm, there is no other restriction placed on the matching space other than to possess a front and a back. This implies that arbitrary chunks of the world can be analyzed and the recovered surfaces can be fully or partially closed, depending on the dimensionality and relationship of the front and back regions. For the purpose of this paper, we selected a partition of space that only allows open surfaces with uniform quantization of either disparity or depth, as depicted in Figure 8.2.

The matching space is defined as a projective 3D volume (to allow pyramids as well as cubes) formed by three axes a, b, and d containing respectively a'_{size}, b'_{size}, and d'_{size} quantized steps, that is

$$\begin{bmatrix} a' \\ b' \\ d' \\ 1 \end{bmatrix} \quad \text{with} \quad \begin{array}{rcll} a' & \in \mathbb{N} & , & 0 \leq a' < a'_{size} \\ b' & \in \mathbb{N} & , & 0 \leq b' < b'_{size} \\ d' & \in \mathbb{N} & , & 0 \leq d' < d'_{size} \end{array}$$

where a' and b' intuitively correspond to a pixel coordinate inside a viewing volume such as in Figure 8.1 while d' corresponds to the disparity or depth of that pixel.

A point (a', b', d') is expressed in the 3D world as an homogeneous point $\mathbf{p}_w$ defined

as

$$\mathbf{p}_w = \mathbf{Q} \begin{bmatrix} a' \\ b' \\ d' \\ 1 \end{bmatrix} \tag{8.1}$$

where $\mathbf{Q}$ is a 4×4 matrix that allows for changing the shape and position of the matching space in the world.

In particular, the matching space is made identical to the viewing volume of a camera (see Figure 8.2A) by defining $\mathbf{Q}$ as

$$\mathbf{Q} = \mathbf{W}^{-1} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{x_{size}}{a'_{size}-1} & & & 0 \\ & \frac{y_{size}}{b'_{size}-1} & & 0 \\ & & \frac{d_{max}-d_{min}}{d'_{size}-1} & d_{min} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where x_{size} and y_{size} represent the image size, d_{min} and d_{max} are the allowed disparity interval, and where $\mathbf{W}$ is the 4×4 viewing transformation matrix of the camera. Notice that d' is moved to the fourth row, making it represent disparity rather than depth, as would be the case for standard stereo with uniformly quantized disparities.

Similarly, if a uniform quantization of depth is desired (see Figure 8.2B) , the last row of $\mathbf{Q}$ should be $[0, 0, 0, 1]$, as in this definition

$$\mathbf{Q} = \begin{bmatrix} \frac{a_{max}-a_{min}}{a'_{size}-1} & & & a_{min} \\ & \frac{b_{max}-b_{min}}{b'_{size}-1} & & b_{min} \\ & & \frac{d_{max}-d_{min}}{d'_{size}-1} & d_{min} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the intervals $[a_{min}, a_{max}]$, $[b_{min}, b_{max}]$, and $[d_{min}, d_{max}]$ represent the span of the matching space position in the world. Notice that in this case, the matching space is defined independently of the camera geometries.

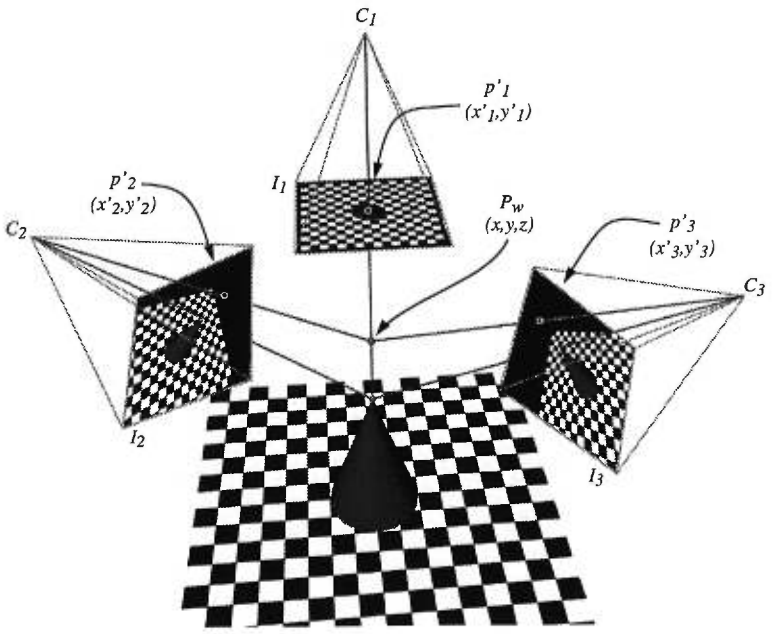

Figure 8.3. Multiple-camera stereo setup. You can back-project any world point $\mathbf{p}_w$ to each inspection camera (C_1, C_2, C_3), obtaining the set of image points (p'_1, p'_2, p'_3).

8.2.2 Pixel intensity values

In this section, we present a general framework to handle stereo in the context of multiple images taken under arbitrary camera geometries. It naturally extends the traditional two-image, single-baseline framework for stereo. In this context, the cameras do not need to be fully calibrated. Each camera i must simply provide a single transformation matrix W_i from the world coordinate system to the camera image space. This allows usage of *uncalibrated* cameras, but in that case the disparities obtained by stereo matching do not have a known relation to the real depth in the scene.

A set of n *inspection* cameras $C_1, \ldots, C_n$ provides n images $I_1, \ldots, I_n$ of a scene, as depicted in Figure 8.3 (with $n = 3$). A *cube* (not shown in Figure 8.3) provides the matching volume where we wish to compute the depth surface. Inside the matching

volume, a cube point (a', b', d') can be transformed to the homogeneous image point $\mathbf{p}_i$ in the image of camera i by the relation

$$\begin{aligned} \mathbf{p}_i &= \mathbf{J}\,\mathbf{W}_i\,\mathbf{p}_w \\ &= \mathbf{J}\,\mathbf{W}_i\,\mathbf{Q}\begin{bmatrix} a' & b' & d' & 1 \end{bmatrix} \end{aligned}$$

where $\mathbf{W}_i$ is a 4×4 matrix describing the camera geometry, $\mathbf{Q}$ is from Equation 8.1, and internal parameters and $\mathbf{J}$ is a simple 3×4 projection matrix

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

From a transformed and projected point $\mathbf{p}_i$, the corresponding image coordinates $\mathbf{p'}_i$ are obtained from the relation

$$\mathbf{p'}_i = H(\mathbf{p}_i)$$

where H is a homogenizing function

$$H\left(\begin{bmatrix} x \\ y \\ h \end{bmatrix}\right) = \begin{bmatrix} x/h \\ y/h \end{bmatrix}.$$

The pixel intensity vector $\mathbf{v}_{(a',b',d')}$ associated to each cube point (a', b', d') is defined as

$$\mathbf{v}_{(a',b',d')} = \left\{ I_i\left(H\left(\mathbf{J}\,\mathbf{W}_i\,\mathbf{Q}\begin{bmatrix} a' & b' & d' & 1 \end{bmatrix}^T\right)\right) \;, \forall i \in [1, \ldots, n] \right\} \tag{8.2}$$

where $I_i([\ x' \quad y'\ ]^T)$ is the intensity of pixel $[\ x' \quad y'\ ]^T$ in image i. This vector contains all the pixel intensity information from the inspection cameras for a particular value of (a', b', d').

8.2.3 Matching Cost

In order to perform stereo matching, a *matching cost* function is required. Ideally, it is minimum for a likely match and large for an unlikely one. Deriving a matching cost that represents well the stereo problem is not a trivial task. Deriving one that can also be globally minimized in polynomial time is even more difficult. Until now, dynamic programming provided an efficient way to minimize cost functions that enforce smoothness, which are generally viewed as very appropriate for the stereo problem. However, as a side effect of this method, the cost function had to be *weakened* by enforcing smoothness along a line instead of a surface. In this paper, the maximum-flow minimization method removes this limitation and therefore solves better suited cost functions than previously possible. There is however a new restriction on the cost function: the smoothness term must be linear, rather than arbitrary for the dynamic programming approach. This, as experiments will show, is not a major problem and does not significantly *weaken* the cost function. This new cost function is described next.

If we assume that surfaces are lambertian (i.e. their intensity is independent of the viewing direction) then the pixel intensity values, components of $\mathbf{v}_{(a',b',d')}$, should be identical when (a', b', d') is on the surface of an object and thus a valid match. Then, we can naturally define the matching cost $cost(a', b', d')$ as the L_2-norm of the pixel intensity vector $\mathbf{v}_{(a',b',d')}$, that is

$$cost(a', b', d') = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}_{(a',b',d')_i} - \overline{\mathbf{v}_{(a',b',d')}})^2. \tag{8.3}$$

where $\overline{\mathbf{v}_{(a',b',d')}}$ is the mean of the components of $\mathbf{v}_{(a',b',d')}$.

8.3 Recovering a full disparity map

Typically, stereo matching is performed independently along epipolar lines, to allow an efficient algorithm to be used, such as dynamic programming. A natural extension

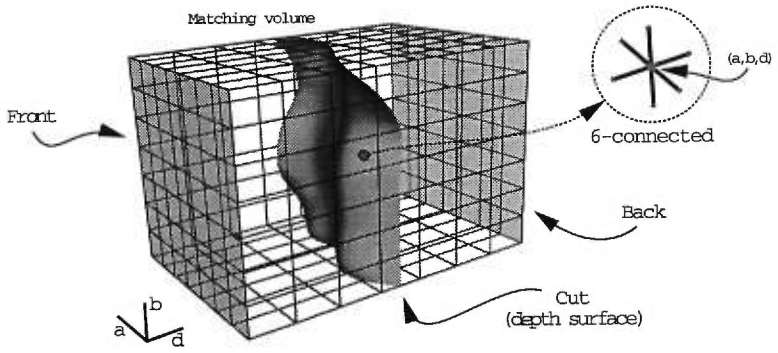

Figure 8.4. Matching whole images. In the matching volume, the `Front` and `Back` correspond respectively to the minimum and maximum disparities. The depth surface cuts the matching volume in two parts, isolating `Front` and `Back`.

to matching a single pair of epipolar lines at a time would be to extend it to the whole image at once, as depicted in Figure 8.4, by matching all pairs of epipolar lines simultaneously. The matching volume is quantized in three dimensions with two axes (a, b) representing image pixels and an axis d for the disparity associated with each pixel (a, b). The depth surface contains all the computed disparities of the base image. The goal of this construction is to take advantage of one very important property of disparity fields, *local coherence*, which suggests that disparities tend to be locally very similar in all directions, including *across* epipolar lines.

Dynamic programming cannot be used anymore to globally establish correspondence since there is no two-dimensional ordering that can be used in a way similar to the use of the one-dimensional ordering along individual epipolar lines.

Many solutions for global disparity surface matching have been proposed [6, 19, 59]. Typically, these algorithms propose an approach in which a solution is iteratively improved by using the previous matching obtained for neighboring epipolar lines. While this can sometimes work in practice, these solutions are not very efficient and not optimal with regard to their inability to find the global minimum of the cost function they are minimizing.

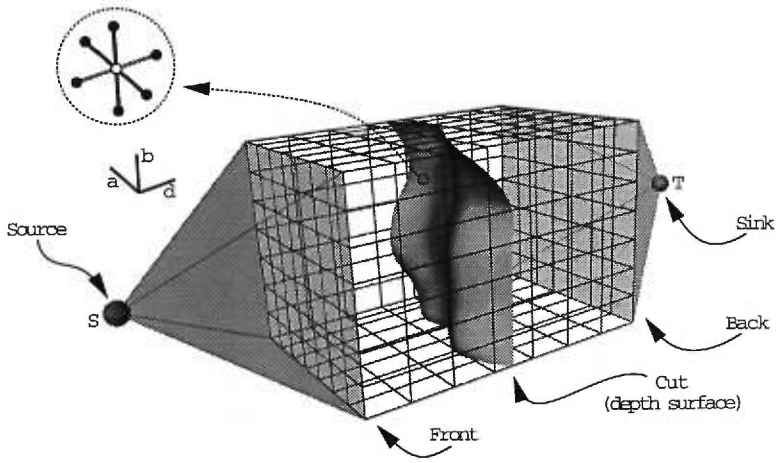

Figure 8.5. Image Matching as a Maximum Flow problem.

8.4 Stereo matching as a Maximum Flow problem

We propose to solve globally for the disparity surface by adding a source and a sink to the formulation of Figure 8.4, and treat it as a flow problem in a graph, as depicted in Figure 8.5. The graph depicts a flow network where each arc has a stated flow capacity, and each node acts as a junction. The flow entering a node is always equal to the flow leaving a node, thereby enforcing a *flow conservation* property. The maximum-flow problem we wish to solve is concerned about finding the largest flow that can leave the source and reach the sink through the graph, without exceeding the capacities of the arcs [16]. According to the *Max-flow min-cut theorem* [16], the set of edges that are saturated by the maximum flow through the graph represents a *minimum-cut* of the graph. By connecting the source and sink respectively to the *front* and *back* of the matching volume , as in Figure 8.5, a cut separating the source and sink effectively represents a disparity surface. Moreover, a minimum-cut will represent the minimum cost disparity surface sought.

Consider the graph $G = (V, E)$ forming a 3-D mesh as in Figure 8.5. The vertex set V is defined as

$$V = V^* \cup \{s, t\}$$

where s is the source, t is the sink, and V^* is the 3D mesh

$$V^* = \{(a', b', d') : 0 \leq a' < a'_{size},\ 0 \leq b' < b'_{size},\ 0 \leq d' < d'_{size} + 1\}$$

where (a'_{size}, b'_{size}) is the base image size and d'_{size} is the depth or disparity solution space. This space is made larger by one node to provide a dummy node required for an appropriate graph formulation.

Internally the mesh is six-connected. There are two disjoint sets of vertices, V_{front} and V_{back}, that represent the *front* and *back* of the graph, such that the source s is connected to each node of V_{front}, while each node of V_{back} is connected to the sink t. We define them as

$$\begin{aligned} V_{front} &= \{(a', b', 0) : 0 \leq a' < a'_{size}, 0 \leq b' < b'_{size}\} \\ V_{back} &= \{(a', b', d'_{size}) : 0 \leq a' < a'_{size}, 0 \leq b' < b'_{size}\}. \end{aligned}$$

The edges of the graph are defined as

$$E = E_{label} \cup E_{penalty} \cup E_{in} \cup E_{out}$$

with

$$\begin{aligned} E_{in} &= \{(s, u) : u \in V_{front}\} \\ E_{out} &= \{(u, t) : u \in V_{back}\} \\ E_{label} &= \{(u, v) \in V^* \times V^* : |u - v| = (0, 0, 1)\} \\ E_{penalty} &= \{(u, v) \in V^* \times V^* : \|u - v\| = 1 \text{ and } u_{d'} = v_{d'}\} \end{aligned}$$

where $u_{d'}$ and $v_{d'}$ are the d' components of the nodes u and v. With respect to Figure 8.5, the edge set E_{in} is the section connecting the source and the front, while E_{out} is the section connecting the back and the sink. The set E_{label} expresses the pixel matching costs and contains all edges parallel to the d axis. The set $E_{penalty}$ expresses the smoothness constraint and contains all edges inside the (a, b) planes.

After the minimum cut is obtained, cut edges belonging to $E_{penalty}$ will be discarded while those from E_{label} will represent the obtained disparity. The different role of *label* and *penalty* edges will be further described in Sections 8.4.1 and 8.4.2.

We define the edge capacities in the graph in a straightforward way. The connections to the source or the sink have infinite capacities. Each vertex (a', b', d') in the graph corresponds to a potential match that assigns disparity d' to pixel (a', b'), so we can use Equation 8.3 to derive its matching cost. This cost is directly used as the capacity of the label edge ($\in E_{label}$) associated to this vertex. To express smoothness, a constant capacity is given to penalty edges ($\in E_{penalty}$). The edge capacity $c(u, v)$ from node u to v is thus defined as

$$c(u,v) = \begin{cases} 0 & \text{if } (u,v) \notin E \\ \infty & \text{if } (u,v) \in E_{in} \text{ or } (u,v) \in E_{out} \\ cost(u) & \text{if } (u,v) \in E_{label} \text{ and } u_{d'} < v_{d'} \\ cost(v) & \text{if } (u,v) \in E_{label} \text{ and } u_{d'} > v_{d'} \\ K & \text{if } (u,v) \in E_{penalty} \end{cases} \tag{8.4}$$

where K is a smoothness factor.

The minimum-cut C_{min} obtained by computing the maximum-flow over G contains a set of edges of minimum total capacity that isolates the source and the sink. Being a minimum-cut, C_{min} is defined as

$$\begin{aligned} & \arg\min_{C} \left(\sum_{(u,v) \in C} c(u,v) \right) \\ = \; & \arg\min_{C} \left(\sum_{(u,v) \in C_{label}} c(u,v) + \sum_{(u,v) \in C_{penalty}} c(u,v) \right) \end{aligned}$$

where $C_{label} = C \cap E_{label}$ and $C_{penalty} = C \cap E_{penalty}$. We define the labelling functions $L_{(a',b')}$ as the smallest d' component of the nodes of an edge in C_{label}. Such an edge is of the form

$$(\ (a', b', L_{(a',b')}), (a', b', L_{(a',b')} + 1)\ ) \text{ or } (\ (a', b', L_{(a',b')} + 1), (a', b', L_{(a',b')})\ ).$$

Notice that these two forms have the same capacity: $cost(a', b', L_{(a',b')})$. By replacing the edge capacities according to Equation 8.4 we have

$$\begin{aligned} & \min_{C} \left(\sum_{(u,v)\in C_{label}} cost(u) + \sum_{(u,v)\in C_{penalty}} K \right) \\ = & \min_{C} \left(\sum_{((a',b',L_{(a',b')}),v)\in C_{label}} cost(a', b', L_{(a',b')}) + \sum_{((a',b',L_{(a',b')}),v)\in C_{penalty}} K \right) \\ = & \sum_{\forall (a',b')} cost(a', b', L^*_{(a',b')}) + \frac{K}{2} \sum_{\forall (a',b'),\forall (i',j')\in \mathcal{N}_{(a',b')}} |L^*_{(a',b')} - L^*_{(i',j')}| \end{aligned} \tag{8.5}$$

where $\mathcal{N}_{(a',b')}$ is a neighborhood of (a', b') and $L^*_{(a',b')}$ is the labelling function associated to the minimum-cut C_{min}. This transformation is possible since a minimum-cut has the property that for all (a', b') there exists exactly one disparity $L_{(a',b')}$ such that the edge $((a', b', L_{(a',b')}), (a', b', L_{(a',b')} + 1))$ belongs to C_{label}. This property is discussed in Section 8.4.2.

This cost function corresponds to finding a disparity surface that globally minimizes a *pixel matching cost* term and a *smoothness* term that assigns a linear penalty to a jump in disparity between neighboring pixels. The tradeoff between these terms is determined by the factor K.

8.4.1 Expressing smoothness through edge capacity

From the partition of E in two sets of edges, the set of *penalty* edges $E_{penalty}$ is used to control the level of smoothness of the disparity surface (second term of Equation 8.5). As depicted in Figure 8.6, *penalty* edges consists of all edges not oriented along the disparity axis d. As shown in Equation 8.4, the matching cost defines the capacity of a label edge, while penalty edges are given the constant value K which also corresponds intuitively to an *occlusion* cost. In Figure 8.6, the darker edges connecting the black vertices are penalty edges and the lighter edges are label edges. A higher occlusion cost (i.e. larger K) increases the smoothness of recovered surfaces while, inversely, a lower occlusion cost facilitates depth discontinuities.

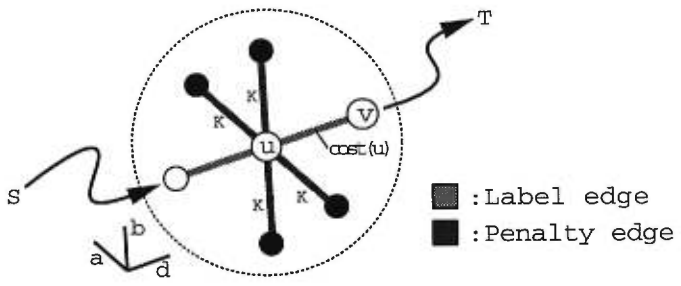

Figure 8.6. Expressing smoothness through edge capacity.

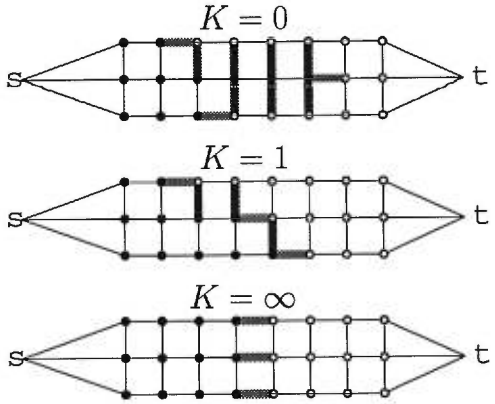

Figure 8.7. Example cuts for different smoothness values. $K = 0$, maximal discontinuity. $K = 1$, intermediate smoothness. $K = \infty$, infinite smoothness.

The effect of the smoothness parameter K is illustrated by a 2-D example problem with a simple cost function, as shown in Figure 8.7. The minimum-cut of this simple graph is computed for different smoothness values (0, 1, and ∞) and displayed in Figure 8.7 as thick edges. Notice that the label edges, which determine the solution, are horizontal. These extreme values of the smoothness parameter K have intuitive consequences. Setting $K = 0$, each row of the graph is independently given a disparity, therefore achieving maximal discontinuity in the disparity surface. When $K = \infty$, the resulting disparity surface is flat (maximally smooth) and features a single disparity value for the whole image. For $K = 1$, a balance is reached between the matching cost and the smoothness required.

8.4.2 From a cut to a disparity surface

The *max-flow min-cut* theorem states that once the maximum flow is found, a minimum-cut C_{min} separates the source and the sink in such a way that the sum of edge capacities of C_{min} is minimized. This cut is therefore the globally optimal way to separate the source and the sink for our particular cost function. To derive our cost function of Equation 8.5, we used an important property of our graph formulation, namely that the minimum cut is guaranteed to provide exactly one depth estimate for each image point, or more simply that the cut does not *fold* on itself. This property can be guaranteed in various ways.

As proposed by Boykov *et al.* [12], a large constant can be added to the likelihoods of equation 8.3. The capacity function (equation 8.4) becomes

$$c(u,v) = \begin{cases} \ldots \\ cost(u) + B & \text{if } (u,v) \in E_{label} \text{ and } u_{d'} < v_{d'} \\ cost(v) + B & \text{if } (u,v) \in E_{label} \text{ and } u_{d'} > v_{d'} \end{cases}$$

where B is a value larger than the sum of all the penalty edges of the graph. The minimum-cut of the new graph can not *fold* on itself and the associated energy value is given by equation 8.5 with a constant added.

Also, Ishikawa and Geiger [42] suggested to assign an infinite capacity to label edges returning from the sink toward the source. The capacity function is modified from equation 8.4 to become

$$c(u,v) = \begin{cases} \ldots \\ cost(u) & \text{if } (u,v) \in E_{label} \text{ and } u_{d'} < v_{d'} \\ B & \text{if } (u,v) \in E_{label} \text{ and } u_{d'} > v_{d'} \end{cases}$$

where B is infinity [42]. In fact, it is sufficient and more practical to define B as a value larger than the sum of all the penalty edges of the graph.

Notice that the two previous solutions [12, 42] do not make any assumption about

the capacities of penalty edges. This adds flexibility to the choice of these capacities and might prove to be useful in the future.

The full disparity surface can now be constructed easily from the minimum cut C of graph G as follows. For each point (a', b'), the disparity is $L_{(a',b')}$ since the edge $(a', b', L_{(a',b')}) - (a', b', L_{(a',b')} + 1)$ belongs to C, as stated in equation 8.5.

8.4.3 Solving the Maximum Flow problem

There is an abundant literature on algorithms to solve the maximum-flow problem [16, 30]. For this paper, we implemented a well known algorithm, *preflow-push lift-to-front* (see [16]). Currently, the best maximum-flow algorithm is presented in Goldberg and Rao [30] and is particularly well suited for sparse graphs like the ones built for stereo matching.

The number of vertices v in the graph is equal to the number of image pixels multiplied by the depth resolution. For an image of total size $s = ab$ pixels, i.e. of dimension $a \times b$, and a depth resolution of d steps, we have $v = sd$. Since the graph is a three-dimensional mesh where each vertex is six-connected[1], the number of edges e is $e \approx 6sd$.

This implies that the preflow-push algorithm used, with a running time

$$O(ve \log(v^2/e))$$

yields for our problem

$$O(s^2 d^2 \log(sd)).$$

The algorithm with the currently best bound [30] runs in

$$O(e^{\frac{3}{2}} \log(v^2/e) \log(U))$$

[1] The nodes on the side of the graph are in fact less than 6-connected.

where U is the largest edge capacity, in our case a constant since pixels have finite values, yielding a running time of

$$O(s^{1.5}d^{1.5}\log(sd)\log(U)).$$

However, we did not use this algorithm in practice since this performance improvement is for the worst case only, and not for the average case. No significant improvement in the average case is expected over the preflow-push relabel algorithm we used.

The dynamic programming approach on separate epipolar lines proposed by Cox *et al.* [21] requires a total running time of $\Theta(sd)$, which might seem much better than the maximum-flow algorithm. However, the topology of the graph, the positions of the source and sink, and the structure of edge capacities all tend to make the problem easier to solve, making the average running time much better than the worst case analysis would suggest. Figure 8.8 shows the typical performance as a function of total image size s (in pixels) and depth resolution d. The average running time is $O(s^{1.2}\ d^{1.3})$, which is almost linear with respect to image size s (in pixels) and compares favorably with the dynamic programming approach. The typical running time for 256×256 images is anywhere between 1 and 30 minutes, on a 160 MHz Pentium computer, depending on the depth resolution used. While this is considerably slower than Cox *et al.* [21], which was originally built for speed, our algorithm was not optimized for speed. Performance improvement is expected in the future.

8.5 Experiments and results

In this section, results of binocular and n-camera stereoscopic matching from maximum-flow are presented and compared with two other algorithms, both based on dynamic programming. The requirement to support multiple images is not readily handled by the vast majority of stereo algorithms, making many comparisons unpractical.

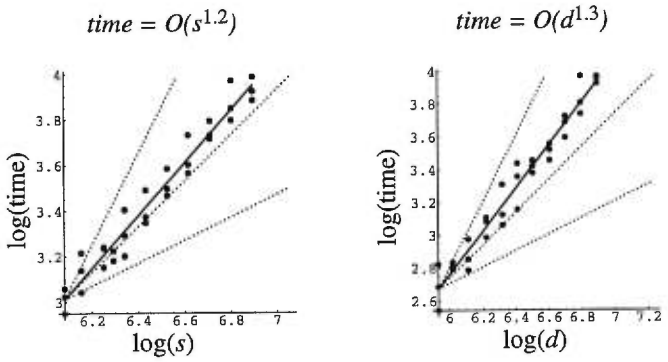

Figure 8.8. (left) Performance as a function of image size s in pixels, for fixed depth resolution. (right) Performance as a function of depth resolution d for a fixed size s. Three dotted lines show performance levels of $O(\sqrt{s})$, $O(s)$, and $O(s^2)$.

First, the algorithm referred to as standard stereo uses line-by-line dynamic programming on n-camera with variable depth resolutions. It differs from the maximum-flow algorithm only in the way it computes the disparity surface. They are otherwise identical and their results use the same disparity scale and are not equalized. By equalization, we refer to a solution-dependent transformation, usually non-linear, applied to the solution in order to improve the contrast of the displayed results. The most common such equalization is *histogram equalization.* Often, this transformation makes fair comparison of results very difficult, if at all possible.

Second, the algorithm referred to as MLMH+V is the efficient dynamic programming implementation from Cox *et al.* [21] (for the binocular version) and from Cox [19] (for the n-camera version). It differs from the previous algorithm in that performs an iterative optimization of its disparity solution to enforce smoothness across disparity lines. It should be noted that the results from this algorithm use a different disparity scale (gray levels) than maximum-flow or standard stereo and are equalized to improve their contrast.

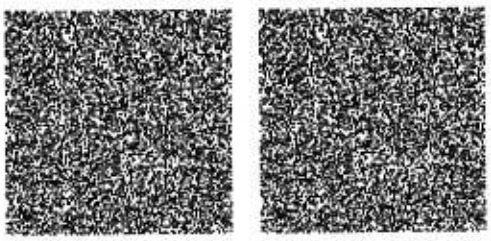

Figure 8.9. A random dot stereogram (displayed for cross-eyed stereo viewing)

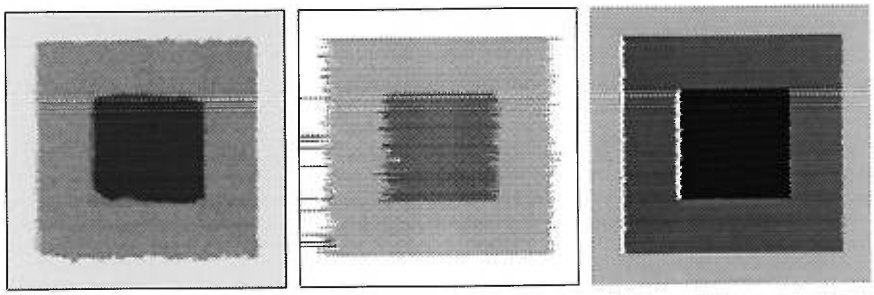

Figure 8.10. Disparity map for random dot stereogram.

Random dot stereogram

To demonstrate the symmetry in the disparity map achieved by maximum-flow, we applied it on a random-dot stereogram (see Figure 8.9) with disparities set at 0, 4 and 8 pixels. The resulting disparity maps, shown in Figure 8.10, differ mostly around depth discontinuities. maximum-flow features similar boundaries in all directions while standard stereo yields very different boundary shapes, due to the fact that solutions are computed horizontally and no information is shared vertically.

Granite

Figure 8.11 presents the camera and scene setup for a synthetic sequence of 5 views of a smooth textured surface. The camera images, displayed in Figure 8.12, are put in correspondence over the matching space shown as the 3-D mesh of Figure 8.11.

Results for different number of images and different smoothness values are shown

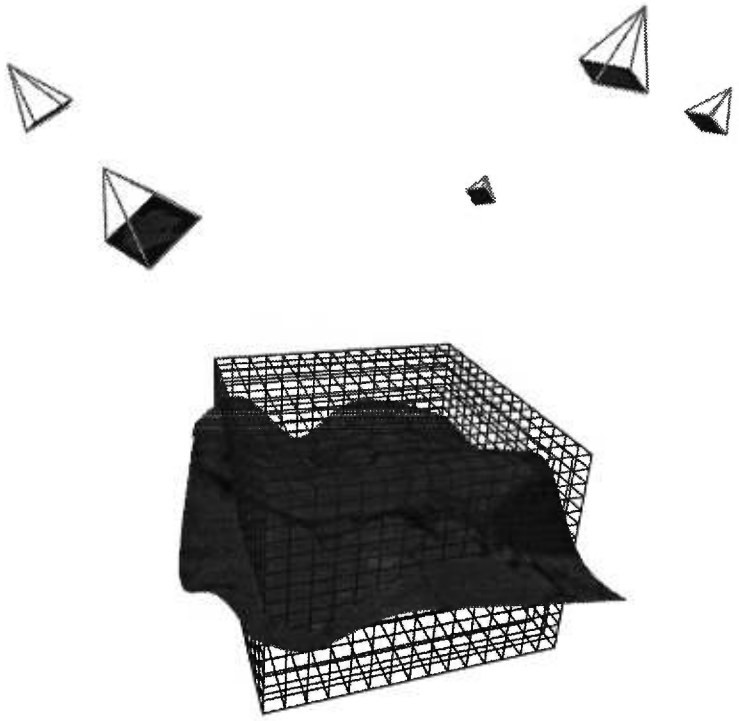

Figure 8.11. The Granite scene and camera setup. The mesh represents the matching volume.

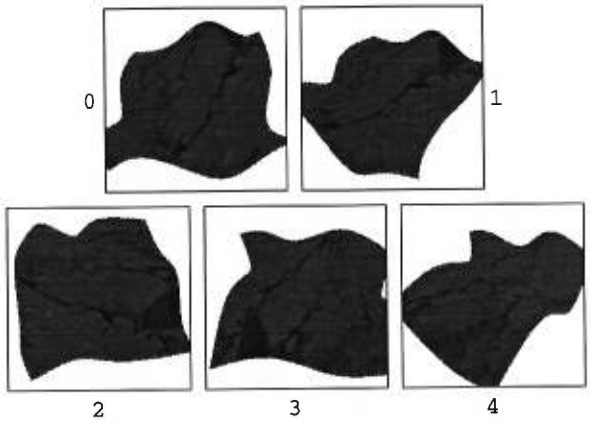

Figure 8.12. The Granite camera images (256x256).

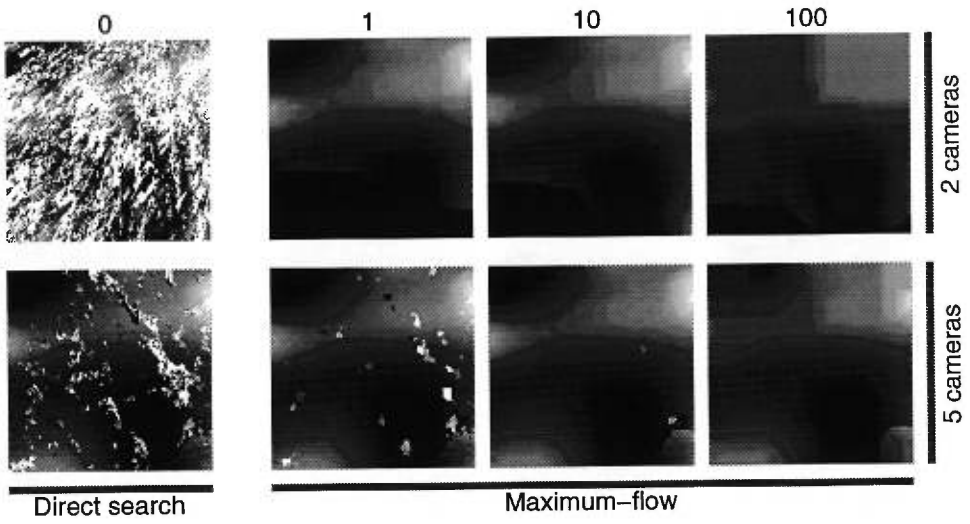

Figure 8.13. The Granite results. Results shown for smoothness factors 0 to 100, for 2 and 5 cameras.

in Figures 8.13 and 8.14. The case $K = 0$ corresponds to using direct search to solve for depth and yields a noisy depth map. Figure 8.14 presents the accuracy of the depth map as a function of smoothness value, for 2 and 5 cameras. Not surprisingly, these curves suggest that better depth map accuracy is achieved with more images used for matching. Also, enforcing some degree of smoothness, even a small amount, is always better than none at all ($K = 0$). Finally, the accuracy degrades slowly as the smoothness is increased to large levels. This implies that the maximum-flow method is very tolerant of bad estimation of the smoothness parameter.

Shrub

Figure 8.15 shows a pair of the Shrub image sequence (courtesy of T. Kanade and T. Nakahara of CMU). The results in Figure 8.16 show how maximum-flow tends to extract sharp and precise depth discontinuities, while standard stereo and MLMH+V produce many artifacts along vertical depth discontinuities. Two levels of depth resolutions are shown (32 and 128 steps) with different level of smoothness. It is notable

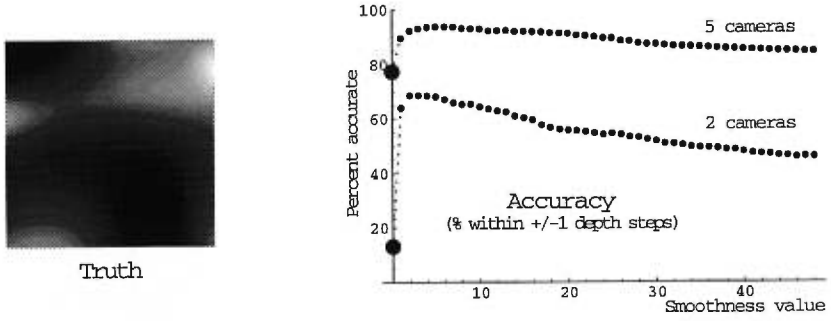

Figure 8.14. The Granite results.

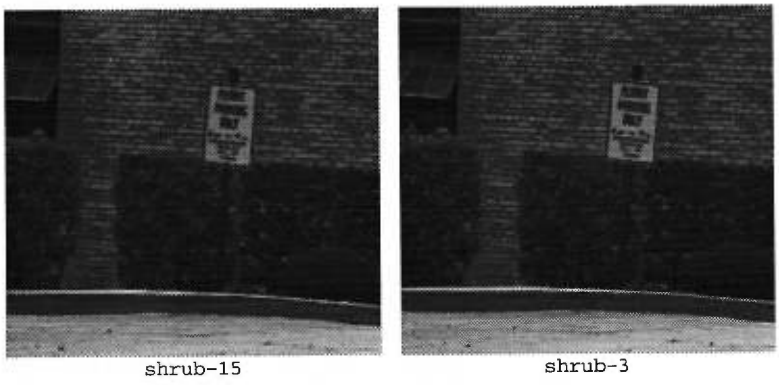

Figure 8.15. The Shrub stereo pair.

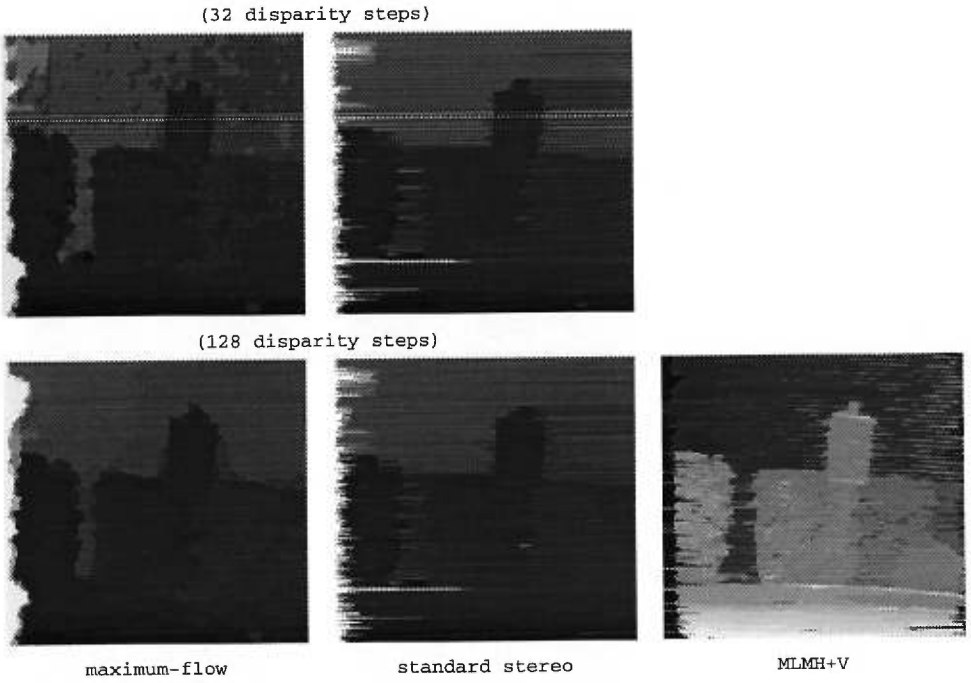

Figure 8.16. Disparity maps for the Shrub **a two precision level (32 and 128 disparity steps). On the left, the** maximum-flow **results. In the middle and right, results for** standard stereo **and** MLMH+V **respectively**

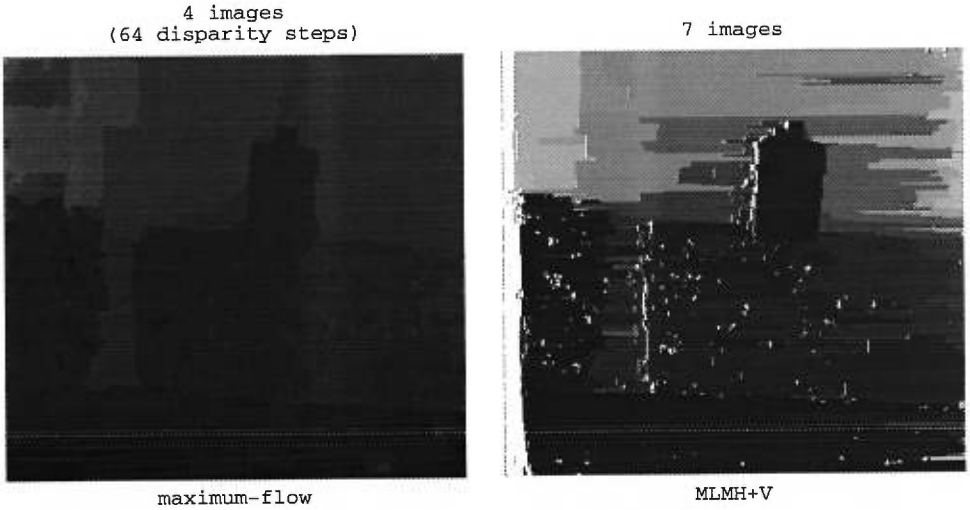

Figure 8.17. Disparity maps for images 4 and 7 of the Shrub sequence. Both sequences span the same total horizontal displacement and should yield similar results. White points on the right denote detected occlusions.

that even at high smoothness levels, maximum-flow does not produce spurious horizontal links across the gap between the two larger shrubs. The results of multiple-camera analysis are shown in Figure 8.17. All the images of this sequence share a common horizontal baseline. Even though the algorithms use different number of images (4 and 7), the total spanned camera displacement is the same and therefore provides about the same depth discrimination. Some image normalization is performed for MLMH+V prior to matching. None was used for the other two algorithms.

Pentagon

The stereo pair Pentagon is shown in Figure 8.18. The matching results are presented in Figure 8.19. This stereo pair presents a challenge since the camera motion is not exactly horizontal and contains some rotation, creating image motions that violate the epipolar constraint. Fortunately, algorithms like MLMH+V resist these misalignments better since they allow negative disparities as well as positive. This explains how the

Figure 8.18. The Pentagon stereo pair.

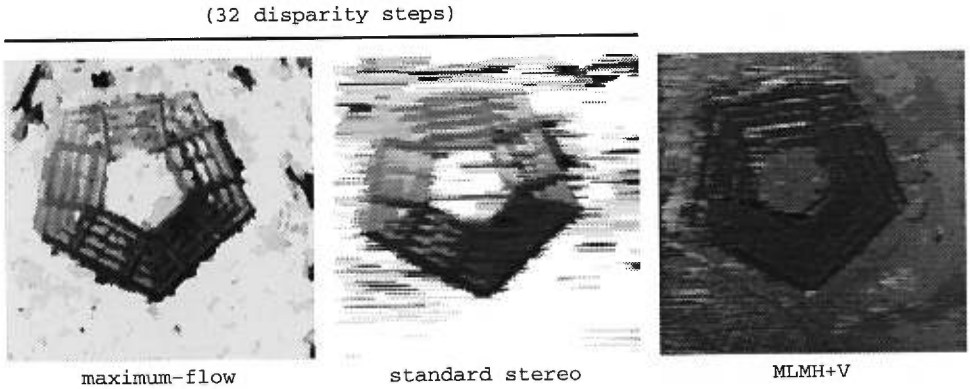

Figure 8.19. Disparity maps for the Pentagon stereo pair.

Figure 8.20. The Park meter stereo pair.

Figure 8.21. Disparity maps for the Park meter sequence. Results are shown for 2 image sequence.

highway structures at the top left are well recovered for MLMH+V while the other algorithms produced some noticeable spurious mismatches. As predicted, maximum-flow does produce a more symmetric result, with less spurious horizontal streaks.

Park meter

The image sequence Park meter shown in Figure 8.20 was analyzed for different numbers of images. The results of the binocular case are presented in Figure 8.21. Here a number of vertical objects show the difficulties that standard stereo and MLMH+V have to relate horizontal epipolar line solutions. No horizontal streaks are present in

Figure 8.22. Disparity maps for the Park meter. Results are shown for 4 image sequence. The matching volume is 256x240x64.

the results obtained by maximum-flow. Using 4 images (horizontally displaced along a single baseline), the results shown in Figure 8.22 improve significantly from those of Figure 8.21. No results were available for MLMH+V.

Roof

The image sequence Roof (courtesy of T. Kanade and E. Kawamura of CMU) is shown in Figure 8.23. It contains 13 images featuring either horizontal or vertical translations. The results for maximum-flow and MLMH+V are presented in Figure 8.24. The disparity map obtained by maximum-flow is very detailed. In particular, the structure of the roof is well reconstructed. Figure 8.25 presents a 3-D reconstruction of the Roof sequence based on the maximum-flow depth map. It demonstrates that fine details can be very effectively recovered.

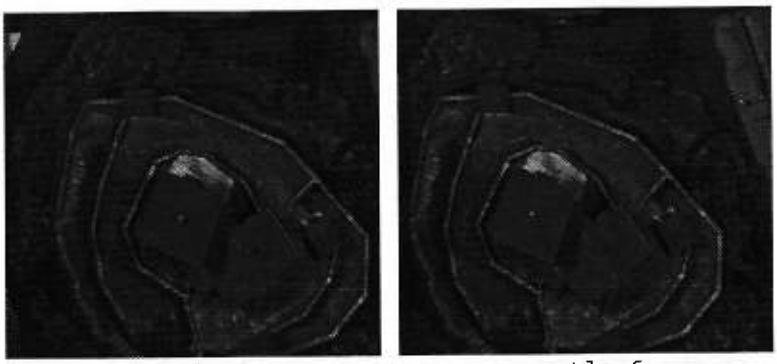

Figure 8.23. Two horizontally separated images from the sequence Roof.

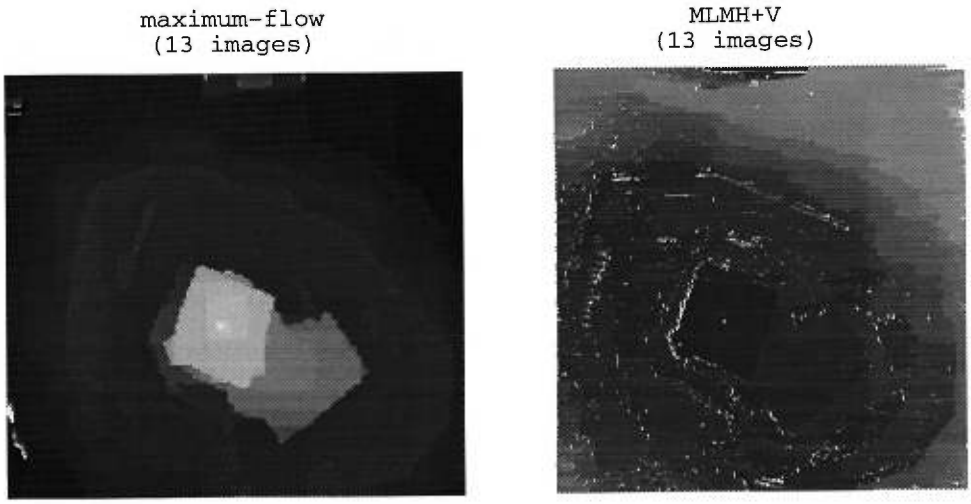

Figure 8.24. Disparity maps for the Roof sequence. Results are shown for 13 images. White points on the right denote detected occlusions. The maximum-flow matching volume is 256x240x64.

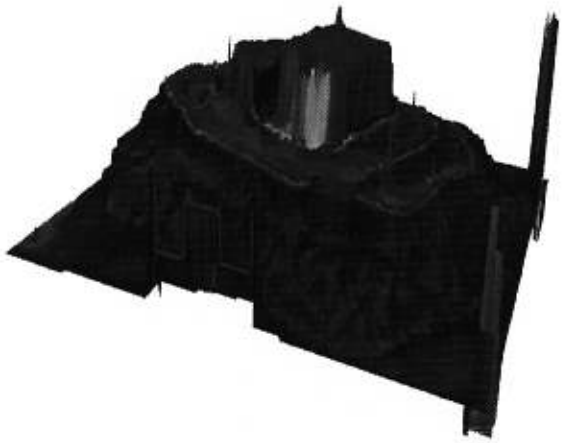

Figure 8.25. Reconstructed 3D surface model for the Roof sequence. The depth map of maximum-flow disparity map is used.

Castle

The sequence Castle from CMU is shown in Figure 8.26 and contains 11 images with various combinations of horizontal, vertical and forward camera motion. The 11 images were used to create the disparity map shown on the right for the image shown on the left. A high level of detail and very few spurious matches are present.

It is important to note that this sequence represents a challenge since the actual disparity range, that is, the difference in disparity between the closest and the farthest object, is only 2.7 pixels. Performed at a depth resolution of 96 steps, this implies that the disparity precision achieved is 0.03 pixels.

8.5.1 Level of Smoothness

In this section, we wish to illustrate how the level of smoothness, represented by the parameter K of Section 8.4.1, affects the quality of the disparity maps. Figure 8.27 illustrates this for four levels of smoothness, namely $K = 0, 1, 10, 100$. For $K = 0$, the capacity of smoothness edges is zero and therefore each pixel is given a disparity

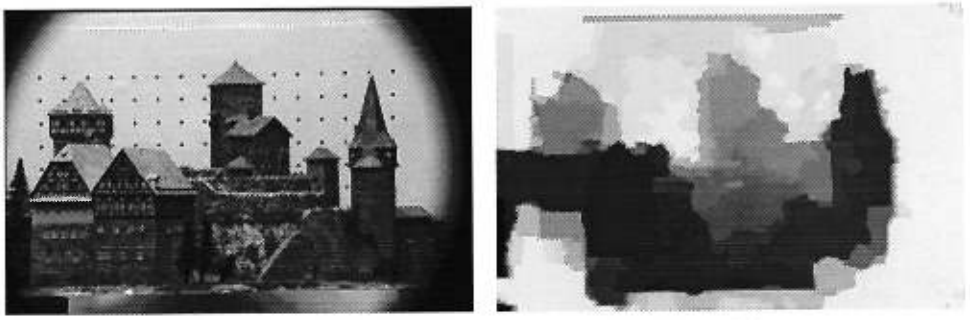

Figure 8.26. The Castle **image stereo sequence. On the left, one of the 11 images. On the right, the resulting** maximum-flow **disparity map.**

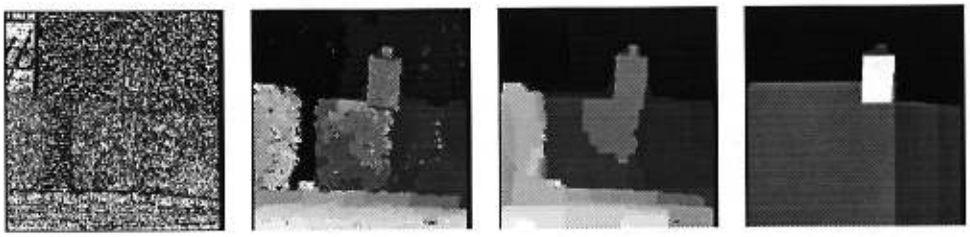

Figure 8.27. Disparity maps for the Shrub **sequence for 4 smoothness levels. On the left, $K = 0$ enforce no smoothness. For $K = 1$, $K = 10$, and $K = 100$, progressively more smoothness is applied, resulting in graceful degradation of depth map.**

independently of its neighbors. It is essentially equivalent to using direct search with correlation over a single pixel window (on the left of Figure 8.27).

As expected, lowering the smoothness capacities favors depth discontinuities and therefore creates sharper object edges, at the expense of surface smoothness.

It is observed that large depth discontinuities tend to stay sharp as the level of smoothness increases. This is probably due to the fact that the smoothness is expressed in all directions instead of only along epipolar lines. This result differs strongly from most other methods where a high level of smoothness induces blurred or missing depth discontinuities.

8.6 Conclusion

We have presented a new algorithm for establishing n-camera stereo correspondence, based on a reformulation of the stereo matching problem to finding the maximum-flow in a graph. It is able to solve optimally for the full disparity surface in a single step, therefore avoiding the usual disparity inconsistencies across neighboring epipolar lines. The *ordering* constraint, required for dynamic programming, is replaced with a more general *local coherence* property that applies in all directions instead of along epipolar lines. The new stereo problem formulation supports multiple arbitrary cameras in a natural way and can estimate depth for an arbitrary virtual camera. For any desired level of smoothness, depth discontinuities are well preserved since smoothness is applied in all directions instead of only along epipolar lines.

We believe that this paper established clearly that a simple cost function, such as the one we used, can yield very high quality solutions when minimized globally and efficiently. These solutions easily rival and generally surpass much more sophisticated cost functions that are impossible to globally minimize because of their complexity.

As for future research, there are many avenues open to improve the maximum-flow formulation proposed in this paper. In particular, a multi-resolution approach as

well as local smoothness variations could be directly embedded in the graph, further improving performance and depth map quality.

Acknowledgment

I would like to thank Ingemar Cox, Jean Meunier and Neil Stewart for their suggestions and comments. I am grateful to Satish Rao and Andrew Goldberg for helpful discussions regarding the computation of maximum-flow in graphs.

Chapitre 9

DISCUSSION ET CONCLUSION

Ce chapitre situe dans le domaine de la vision par ordinateur les nouvelles méthodes présentées dans cette thèse. Il élaborera sur leurs applications, leurs extensions, et sur les travaux futurs qui pourraient éventuellement s'y rattacher.

Calibration de caméra

Cette thèse a présenté un nouveau paradigme pour le calcul du mouvement de caméra entre deux prises de vue. Nous appellons cette approche *Mouvement sans structure* (*Motion without structure*) car elle ne requière ni ne calcule d'information relative à la structure de la scène. L'analyse du mouvement de caméra est posée sous la forme d'une optimisation d'une fonction de vraisemblance dans l'espace des mouvements possibles. Cette fonction évalue un mouvement hypothétique par la somme des différences mises au carré entre les points d'une image et leurs segments épipolaires correspondant dans l'autre image. Il a été démontré que cette fonction possède un seul minimum global pour les cas où soit la rotation ou bien la translation est connue, à condition que les images respectent notre critère d'uniformité. Ce critère, généralement respecté en pratique, impose que la variance des différences d'intensité entre deux points d'une image augmente régulièrement avec la distance entre ces points.

Les résultats expérimentaux suggèrent que la méthode est applicable à un grand nombre d'images, tout en maintenant une bonne précision et une grande robustesse au bruit. Même les grands déplacements de caméra sont possibles; ils ne sont limités que par les caractéristiques statistiques de l'image, établies par notre critère de régularité.

Nous croyons que ce nouveau paradigme *Mouvement sans structure* peut être

utilisé avec succès pour estimer le déplacement de caméra à partir d'images. De plus, nous espérons qu'il se montrera supérieur aux autres méthodes, comme celles qui utilisent les points saillants ou comme les méthodes dites directes ou indirectes de *mouvement et structure*, parce qu'il ne requière pas de flux optique, de dérivées d'intensité des images, ou de mise en correspondance de points saillants.

Certains développements futurs de notre méthode sont possible et demandent à être considérés.

Puisque la garantie de convergence de notre méthode dépend des statistiques des images, il serait important de raffiner le critère de régularité pour tenir compte des images présentant des situations particulières, comme des textures répétitives, de façon à élargir l'applicabilité de la méthode.

En second lieu, il serait important de généraliser la garantie de convergence pour le cas d'une recherche simultanée de la rotation et de la translation, plutôt qu'une recherche de la rotation suivie d'une seconde pour la translation. Bien que cet espace de recherche soit plus vaste, la solution serait probablement préférable à celle de la méthode originale.

Rectification

La vaste majorité des algorithmes de mise en correspondance stéréoscopique assument une géométrie de caméra simple, celle d'un déplacement horizontal, sans rotation, avec les axes optiques parallèles. La raison principale qui motive ce choix est que cette géométrie très simple ne requiert qu'un traitement mathématique minimum, et s'apparente au modèle de la vision humaine.

La rectification d'image stéréoscopique a été introduite pour permettre à ces algorithmes traditionnels d'être utilisés pour des géométries de caméra différentes de l'horizontale, en reprojetant les images de façon à créer artificiellement cette géométrie très simple.

Notre méthode, présentée au chapitre 6, permet d'effectuer cette rectification pour

des géométries de caméra arbitraires, ce qui représente un gain important par rapport à la rectification plane qui ne peut rectifier qu'un sous-ensemble des géométries possibles. Cette nouvelle transformation équivaut à reprojeter les images sur un cylindre dont l'axe passe par les centres optiques des deux caméras. Bien qu'elle ne préserve pas les lignes droites quelconques, elle préserve la longueur des lignes épipolaires et n'introduit donc aucune distorsion des images le long de ces droites, contrairement à la rectification plane qui introduit un distorsion entraînant une perte d'information pouvant affecter la mise en correspondance. De plus, seule la rectification cylindrique construit des images dont la taille est indépendante de la géométrie des caméras, ce qui accroît sa flexibilité.

Néanmoins, il est prévisible que dans un futur proche, tous les algorithmes stéréoscopiques soient appelés à intégrer directement la géométrie des caméras, à la manière de notre algorithme *flot maximum*, décrit au chapitre 8. De tels algorithmes peuvent du coup utiliser plus de deux images et reconstruire dans des volumes arbitraires, ce qui introduit une flexibilité considérable. D'ici là, la rectification d'image restera la solution simple et efficace à la généralisation des algorithmes stéréoscopiques traditionnels.

Il est possible d'utiliser la méthode de rectification cylindrique pour générer des mosaïques à partir d'une séquence vidéo. Les principaux travaux reliés aux mosaïques sont Szeliski [81], Peleg *et al.* [63] et Rousso *et al.* [67]. Notons que les travaux originaux de Peleg *et al.* [63] ne permettent pas les mouvements arbitraires de caméra. Ils ont introduit par la suite et de façon indépendante dans Rousso *et al.* [67] ce qu'ils nomment la *pipe reprojection*, qui correspond essentiellement à notre rectification cylindrique.

Stéréoscopie

Nous avons présenté au chapitre 8 une nouvelle méthode de mise en correspondance stéréoscopique, basée sur une reformulation du problème en celui du calcul du flot

maximal dans un graphe. Elle présente une divergence radicale par rapport aux algorithmes stéréoscopiques traditionnels, qui utilisent la recherche directe ou épipolaire, car elle ne requiert pas de rectification des images en gérant directement la géométrie des caméras. Cette propriété lui confère, entre autres, la possibilité d'utiliser un nombre quelconque d'images de points de vue arbitraires. Elle représente, dans un sens large, une généralisation des méthodes de recherche épipolaire (par exemple la programmation dynamique) vers une recherche globale, qu'elle peut résoudre efficacement et optimalement. Ainsi, elle utilise une contrainte de lissage plus naturelle, en ce sens que celle-ci s'applique dans toutes les directions dans l'image, et pas seulement dans le sens des droites épipolaires.

Dans le futur, plusieurs voies s'offrent en vue de l'amélioration de la formulation par flot maximum. En particulier, il sera possible d'incorporer au graphe une composante multi-résolution ainsi que des variations locales de lissage, ce qui permettra d'augmenter significativement les performances et la précision des résultats.

Il subsiste un élément commun avec les autres algorithmes, celui de la reconstruction du *champs de profondeur* qui ne peut pas représenter complètement une scène tridimensionnelle mais plutôt une vision $2\frac{1}{2}$D de celle-ci, associée au volume de reconstruction choisi. L'évolution naturelle se porte vers la reconstruction globale d'une scène par la mise en commun de plusieurs reconstructions $2\frac{1}{2}$D de façon automatique et transparente, comme l'a tenté Kanade *et al.* [44]. Malheureusement, la composition de reconstructions $2\frac{1}{2}$D partielles n'est pas simple; elle requiert presque toujours une intervention manuelle et donne des résultats imprécis. Dans une reconstruction globale, le phénomène des occlusions prend une ampleur qui semble insurmontable. La nouvelle génération d'algorithmes devra donc modéliser explicitement les occlusions et peut-être même les utiliser au même titre que les correspondances visibles, puisque dans le cas de caméras nombreuses et dispersées, les occlusions sont plus courantes que les correspondances. Cela représente un défi de taille, puisque l'hypothèse de base en stéréoscopie, qui assume que toutes les images présentent la même scène sous des

points de vue légèrement différents, devra disparaître en raison de la variété accrue des points de vue nécessaires à la reconstruction complète d'une scène observée.

À titre de développement futur, il est à prévoir que notre méthode de mise en correspondance par flot maximum puisse être utilisée dans le contexte de l'infographie, pour la reconstruction de modèles 3D réalistes à partir d'images réelles. En effet, la sélection manuelle des points de correspondance, méthode privilégiée en infographie, ne permet de définir que des surfaces polygonales, qui sont souvent grossières.

On pourrait associer une *tolérance* aux polygones reconstruits, c'est-à-dire une région de l'espace tridimensionnel, associée à chaque polygone, qui contient la vraie surface dans le voisinage de ce polygone. Cette région doit présenter un *devant* et un *derrière*, de façon à définir une zone de reconstruction qui est transformée en graphe de flot maximum. Ainsi, on pourrait estimer pour chaque polygone la surface optimale contenue dans cette zone, et de ce fait obtenir un modèle beaucoup plus détaillé sans faire appel à une intervention manuelle supplémentaire. Ceci constituera un excellent exemple de collaboration entre la vision par ordinateur et l'infographie, répondant ainsi à la demande sans cesse croissante pour des modèles 3D de plus en plus réalistes.

Champs aléatoires de Markov

La plupart des nombreuses applications des champs de Markov ont jusqu'ici été impraticables à cause de la nature exponentielle du problème. L'extension de notre méthode de flot maximum pour résoudre certaines classes de champs aléatoires de Markov nous paraît revêtir une importance certaine. En effet, la classe des problèmes d'étiquetage possèdant un ordre unidimensionel des étiquettes, comme la stéréoscopie, la restauration d'images et plusieurs autres, peuvent maintenant être résolus efficacement et optimalement. On entrevoit dans l'avenir que plusieurs applications des champs aléatoires de Markov feront de nouveau surface, grâce à l'efficacité que leur confère notre nouvelle méthode.

RÉFÉRENCES

[1] Y. Aloimonos et Z. Duric. Estimating the heading direction using normal flow. *Int. J. Computer Vision*, 13(1):33–56, 1994.

[2] N. Ayache et C. Hansen. Rectification of images for binocular and trinocular stereovision. Dans *Proc. of Int. Conf. on Pattern Recognition*, pages 11–16, Washington, D.C., 1988.

[3] H. H. Baker. *Depth from Edge and Intensity Based Stereo.* PhD thesis, University of Illinois at Urbana-Champaign, 1981.

[4] J. L. Barron, D. J. Fleet, et S. S. Beauchemin. Performance of optical flow techniques. *Int. J. Computer Vision*, 2(1):43–77, 1994.

[5] P. N. Belhumeur. A binocular stereo algorithm for reconstructing sloping, creased, and broken contours in the presence of half-occlusion. Dans *Proc. Int. Conference on Computer Vision*, pages 431–438, 1993.

[6] P. N. Belhumeur. A Bayesian approach to binocular stereopsis. *Int. J. Computer Vision*, 19(3):237–260, 1996.

[7] D. N. Bhat et S. K. Nayar. Stereo in the presence of specular reflection. Dans *Proc. 5th Int. Conference on Computer Vision*, pages 1086–1092, Cambridge, 1995.

[8] A. Del Bimbo, P. Nesi, et J. L. C. Sanz. Analysis of optical flow constraints. Rapport technique, Faculty of Engineering, University of Florence, Firenze, Italy, 1993.

[9] R. C. Bolles, H. H. Baker, et M. J. Hannah. The JISCT stereo evaluation. Dans *Proc. of DARPA Image Understanding Workshop*, pages 263–274, 1993.

[10] P. Bouthemy et E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. J. Computer Vision*, 2(10):157–182, 1993.

[11] Y. Boykov, O. Veksler, et R. Zabih. Disparity component matching for visual correspondence. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[12] Y. Boykov, O. Veksler, et R. Zabih. Markov random fields with efficient approximations. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, juin 1998.

[13] S. Chandrashekhar et R. Chellappa. Passive navigation in a partially known environment. Dans *Proc. IEEE Workshop on Visual Motion*, pages 2–7, Princeton, NJ, 1991.

[14] C. Chang et S. Chatterjee. Multiresolution stereo - a bayesian approach. Dans *Proc. of Int. Conf. on Pattern Recognition*, pages 908–912, Atlantic City, New Jersey, USA, juin 1990.

[15] T. Y. Chen et A. C. Bovic. Stereo disparity from multiscale processing of local image phase. Dans *ISCV*, pages 188–193, 1995.

[16] T. H. Cormen, C. E. Leiserson, et R. L. Rivest. *Introduction to Algorithms.* McGraw-Hill, New York, 1990.

[17] P. Courtney, N. A. Thacker, et C. R. Brown. A hardware architecture for image rectification and ground plane obstacle detection. Dans *Proc. of Int. Conf. on Pattern Recognition*, pages 23–26, The Hague, Netherlands, 1992.

[18] I. J. Cox. A review of statistical data association techniques for motion correspondence. *Int. J. Computer Vision*, 10(1):53–66, 1993.

[19] I. J. Cox. A maximum likelihood N-camera stereo algorithm. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–739, 1994.

[20] I. J. Cox, S. Hingorani, B. M. Maggs, et S. B. Rao. Stereo without disparity gradient smoothing: a Bayesian sensor fusion solution. Dans D. Hogg et R. Boyle, editeurs, *British Machine Vision Conference*, pages 337–346. Springer-Verlag, 1992.

[21] I. J. Cox, S. Hingorani, B. M. Maggs, et S. B. Rao. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.

[22] I. J. Cox et S. Roy. Direct estimation of rotation from two frames via epipolar search. Dans *6th Int. conf. on Computer Analysis of Images and Patterns*, 1995.

[23] I. J. Cox et S. Roy. Statistical modelling of epipolar misalignment. Dans *International Workshop on Stereoscopic and Three-Dimensional Imaging*, 1995.

[24] O. Faugeras. *Three-dimentional computer vision.* MIT Press, Cambridge, 1993.

[25] C. Fermuller. Global 3-d motion estimation. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 415–421, New York, N.Y., 1993.

[26] Wolfgang Otto Franzen. Structure and motion from uniform 3d acceleration. Dans *Proc. IEEE Workshop on Visual Motion*, pages 14–20, Princeton, NJ, 1991.

[27] P. Fua. Reconstructing complex surfaces from multiple stereo views. Dans *Proc. Int. Conference on Computer Vision*, pages 1078–1085, 1995.

[28] Stuart Geman et Kevin Manbeck. Experiments in syntactic recognition. *Reports in Pattern Analysis*, 158, 1993.

[29] A. V. Goldberg. Recent developments in maximum flow algorithms. Dans *Algorithm Theory - SWAT 98 - Lecture Notes in Computer Science 1432, Proceedings of the 6th Skandinavian Workshop on Algorithm Theory*, pages 1–10. Springer-Verlag, 1998.

[30] A. V. Goldberg et S. B. Rao. Length functions for flow computations. Rapport Technique 97-055, NEC Research Institute, Princeton NJ, 1997.

[31] D. M. Greig, B. T. Porteous, et A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc.*, 51(2):271–279, 1989.

[32] Leonid Gurvits et Barak A. Pearlmutter. Relating egomotion and image evolution. communication, 1995.

[33] K. J. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. Dans *Proc. IEEE Workshop on Visual Motion*, pages 156–162, Princeton, NJ, 1991.

[34] R. Hartley et R. Gupta. Computing matched-epipolar projections. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 549–555, New York, N.Y., 1993.

[35] R. I. Hartley. In defence of the 8-point algorithm. Dans *Proc. 5th Int. Conference on Computer Vision*, pages 1064–1070, Cambridge, 1995.

[36] D. J. Heeger et A. D. Jepson. Subspace methods for recovering rigid motion. to appear in ijcv, 1990.

[37] B. K. P. Horn et E. J. Weldon, Jr. Direct methods for recovering motion. *Int. J. Computer Vision*, 2:51–76, 1988.

[38] K. Horn et B. Schunck. Determining optical flow. *Artificial intelligence*, 17:185–203, 1981.

[39] T. S. Huang et A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, 1994.

[40] R. Hummel et V. Sundareswaran. Motion parameter estimation from global flow field data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(5):459–476, 1993.

[41] Michal Irani, Benny Rousso, et Shmuel Peleg. Computing occluding and transparent motions. *Int. J. Computer Vision*, 12(1):5–16, 1994.

[42] H. Ishikawa et D. Geiger. Segmentation by grouping junctions. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, juin 1998.

[43] A. D. Jepson et D. J. Heeger. A fast subspace algorithm for recovering rigid motion. Dans *Proc. IEEE Workshop on Visual Motion*, pages 124–131, Princeton, NJ, 1991.

[44] T. Kanade, P. J. Narayanan, et P. W. Rander. Virtualized reality: Concepts and early results. Dans *IEEE Workshop on the Representation of Visual Scenes (in conjunction with ICCV)*, 1995.

[45] T. Kanade et M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(9):920–932, 1994.

[46] T. Kanade, A. Yoshida, K. Oda, H. Kano, et M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, 1996.

[47] S. B. Kang, J. A. Webb, C. L. Zitnick, et T. Kanade. An active multibaseline stereo system with real-time image acquisition. Rapport Technique CMU-CS-94-167, School of Computer Science, Carnegie Mellon University, 1994.

[48] Y.-S. Kim, K.-P. Han, E.-J. Lee, et Y.-H. Ha. Robust 3-d depth estimation using genetic algorithm in stereo image pairs. Dans *Proc. of IEEE Asia Pacific Conf. on Circuits and Systems*, pages 357–360, Seoul, Korea, novembre 1996.

[49] R. Kumar, P. Anandan, et K. Hanna. Shape recovery from multiple views: a parallax based approach. Dans *ARPA Image Understanding Workshop*, Monterey, CA, 1994.

[50] R. Kumar et A. R. Hanson. Robust estimation of camera location and orientation from noisy data having outliers. Dans *Proc. Workshop on Interpretation of 3D Scenes*, pages 52–60, Austin, TX, USA, 1989.

[51] S. M. LaValle et S. A. Hutchinson. A bayesian segmentation methodology for parametric image models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(2):211–217, 1995.

[52] S. Li. *Markov random field modeling in computer vision.* Springer-Verlag, 1995.

[53] Ze-Nian Li. Stereo correspondence based on line matching in hough space using dynamic programming. *IEEE Trans. Systems Man and Cybernetics*, 24(1):144–152, 1994.

[54] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[55] Q.-T. Luong et O. D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *Int. J. Computer Vision*, 17:43–75, 1996.

[56] D. Marr et T. Poggio. A theory of human stereopsis. *Proceedings of the Royal Society*, B 204:301–328, 1979.

[57] M. S. Mousavi et R. J. Schalkoff. Ann implementation of stereo vision using a multi-layer feedback architecture. *IEEE Trans. Systems Man and Cybernetics*, 24(8):1220–1238, 1994.

[58] S. Negahdaripour et B. K. P. Horn. Direct passive navigation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.

[59] Y. Ohta et T. Kanade. Stereo by intra- and inter-scanline using dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[60] J. Oliensis. Rigorous bounds for two-frame structure from motion. Rapport Technique 95-155, NEC Research Institute, Princeton, NJ, 1993.

[61] J. Oliensis. A linear solution for multiframe structure from motion: Constant translation direction. Rapport Technique 95-006-3-0215-1, NEC Research Institute, Princeton NJ, 1995.

[62] D. V. Papadimitriou et T. J. Dennis. Epipolar line estimation and rectification for stereo image pairs. *IEEE Trans. Image Processing*, 5(4):672–676, 1996.

[63] S. Peleg et J. Herman. Panoramic mosaics by manifold projection. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 338–343, 1997.

[64] L. Raffo. Adaptive resistive network for stereo depth estimation. *Electronics Letters*, 31(22):1909–1910, octobre 1995.

[65] Daniel Raviv et Martin Herman. A unified approach to camera fixation and vision-based road following. *IEEE Trans. Systems Man and Cybernetics*, 24(8):1125–1141, 1994.

[66] L. Robert, M. Buffa, et M. Hébert. Weakly-calibrated stereo perception for rover navigation. Dans *Proc. 5th Int. Conference on Computer Vision*, pages 46–51, Cambridge, 1995.

[67] B. Rousso, S. Pelel, I. Finci, et A. Rav-Acha. Universal mosaicing using pipe projection. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 945–952, 1998.

[68] S. Roy. Analyse d'images stéréoscopiques basée sur la détermination du flux optique. Mémoire de maîtrise, Université de Montréal, Décembre 1992.

[69] S. Roy et I. J. Cox. Motion without structure. Dans *Proc. of Int. Conf. on Pattern Recognition*, volume 1, pages 728–734, Vienna, Austria, 1996.

[70] S. Roy et I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. Dans *Proc. Int. Conference on Computer Vision*, pages 492–499, Bombay, India, 1998.

[71] S. Roy, J. Meunier, et I. J. Cox. Cylindrical rectification to minimize epipolar distortion. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 393–399, San Juan, Puerto Rico, 1997.

[72] H. S. Sawhney. Simplifying motion and structure analysis using planar parallax and image warping. Dans *12th Int. Conference on Pattern Recognition*, pages 403–408, Jerusalem, 1994.

[73] D. Scharstein et R. Szeliski. Stereo matching with non-linear diffusion. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–350, 1996.

[74] A. Scheuing et H. Niemann. Computing depth from stereo images by using optical flow. *Pattern recognition letters*, 4:205–212, 1986.

[75] J. Shah. A nonlinear diffusion model for discontinuous disparity and half-occlusions in stereo. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 34–40, 1993.

[76] Y. Shirai. *Three-Dimensional Computer Vision.* Springer-Verlag, Berlin, 1987.

[77] M. Shizawa. Direct estimation of multiple disparities for transparent multiple surfaces in binocular stereo. Dans *Proc. Int. Conference on Computer Vision*, pages 447–454, 1993.

[78] D. Sinclair, A. Blake, et D. Murray. Robust estimation of egomotion from normal flow. *Int. J. Computer Vision*, 13(1):57–69, 1994.

[79] P. W. Smith et N. Nandhakumar. An improved power cepstrum based stereo correspondence method for textured scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(3), 1996.

[80] V. Sundareswaran. Egomotion from global flow field data. Dans *Proc. IEEE Workshop on Visual Motion*, pages 140–145, Princeton, NJ, 1991.

[81] R. Szeliski. Image mosaicing for tele-reality applications. Dans *IEEE Workshop on Applications of Computer Vision*, pages 44–53, 1994.

[82] C. Tomasi. Pictures and trails: a new framework for the computation of shape and motion from perspective image sequences. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 913–918, 1994.

[83] C. Tomasi et J. Shi. Direction of heading from image deformations. Dans *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 422–427, New York, N.Y., 1993.

[84] R. Y. Tsai et T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objetcs with curved surfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:13–27, 1984.

[85] Y. Yang et A. L Yuille. Multilevel enhancement and detection of stereo disparity surfaces. *Artificial Intelligence*, 78:121–145, 1995.

[86] Y. Yeshurun et E. L. Schwartz. Cepstral filtering on a columnar image architecture : a fast algorithm for binocular stereo segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):759–767, 1989.