

Université de Montréal

Calibration, Rectification et Stéréoscopie

par

Sébastien Roy

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en informatique

juin, 1999

© Sébastien Roy, 1999

Université de Montréal
Faculté des études supérieures
Ce mémoire intitulé :
Calibration, Rectification et Stéréoscopie
présenté par
Sébastien Roy

a été évalué par un jury composé des personnes suivantes:

Jean Meunier
(Professeur)

Pierre Poulin
(Professeur)

Neil Stewart
(Professeur)

Janusz Konrad
(Professeur)

Anthony F. J. Moffat
(Professeur)

Mémoire accepté le _____

*À mon poisson Bubule,
et à mon hamster Bouboule,*

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	iii
Chapitre 1: Introduction	1
Chapitre 2: Caméras et géométrie épipolaire	7
2.1 Modèle de caméra	7
2.2 Segments épipolaires	12
2.3 Matrice fondamentale et matrice essentielle	15
2.4 La détermination de la géométrie de caméra	18
2.5 Le mouvement de caméra considéré comme vitesse	19
Chapitre 3: Introduction à la calibration de caméra	22
3.1 Contraintes et hypothèses	24
Chapitre 4: (Article) Motion without Structure	27
Abstract	27
4.1 Introduction	28
4.2 Motion estimation as a 5-D search	30
Chapitre 5: Introduction à la rectification	36
5.1 Rectification plane	37
5.2 Conclusion	40
Chapitre 6: (Article) Cylindrical Rectification to minimize Epipolar Distortion	42

Abstract	42
6.1 Introduction	43
6.2 Linear transformation in projective space	46
Chapitre 7: Le problème de la mise en correspondance	48
7.1 Choix des primitives à mettre en correspondance	49
7.2 Hypothèses sur la nature de la scène	50
7.3 Fonction de coût de correspondance à minimiser	51
7.4 Méthode utilisée pour minimiser la fonction de coût	53
7.5 Contraintes sur le nombre et la géométrie des caméras	57
7.6 Volume de reconstruction	61
Chapitre 8: (Article) Stereo Without Epipolar lines : A Maximum-Flow Formulation	64
Abstract	64
8.1 Introduction	65
8.2 The Stereo Framework	67
Chapitre 9: Discussion et Conclusion	73
Références	78
Annexe A: Champs aléatoires de Markov	87
A.1 Étiquetage MAP-MRF	87
Glossaire	98

LISTE DES FIGURES

1.1	Reconstruction 3D à partir d'images	2
2.1	Modèle de caméra sténopé	7
2.2	Segment épipolaire	14
2.3	Géométrie épipolaire	16
2.4	Ambiguïté de profondeur	19
2.5	Modèle du mouvement de caméra	20
3.1	Contrainte d'ordre	26
4.1	Images from the JISCT database and their variance functions	31
4.2	Basic geometry for known rotation	32
4.3	Error function for two segments u and v	34
5.1	Géométrie de caméra horizontale	36
5.2	Géométrie de caméra arbitraire	37
5.3	Rectification plane	39
6.1	Rectification	43
6.2	Images from Fig. 6.1	46
7.1	Reprojection d'un point 3D	52
7.2	Différentes approches de mise en correspondance	53
7.3	Géométrie stéréoscopique traditionnelle (recherche directe)	54
7.4	Mise en correspondance par recherche directe	54
7.5	Recherche épipolaire	56

7.6	Stéréoscopie traditionnelle	58
7.7	Géométrie de caméra convergente	59
7.8	Volume de reconstruction	61
7.9	Volume de reconstruction arbitraire	62
8.1	Standard stereo framework	67
8.2	General stereo framework	67
8.3	Multiple-camera stereo setup	70
A.1	Systèmes de voisinage d'un champ aléatoire de Markov	88
A.2	Ordre des étiquettes	92
A.3	Représentation des discontinuités	95

REMERCIEMENTS

Je souhaite remercier tous ceux qui m'ont apporté le support si nécessaire à la création de cette thèse.

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

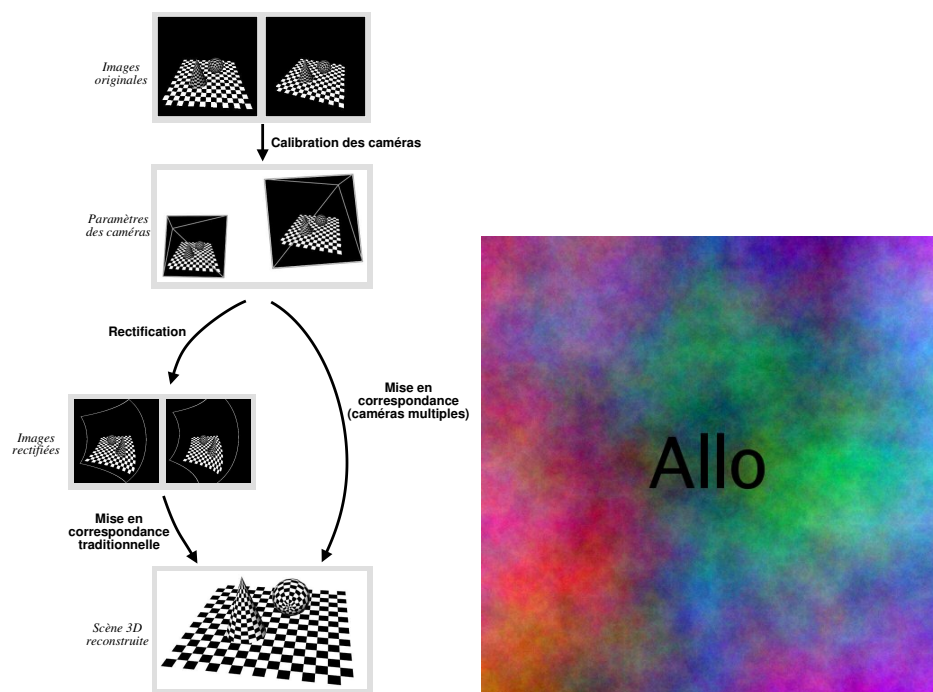


Figure 1.1. Reconstruction 3D à partir d'images. Les paramètres des caméras sont obtenus par une calibration utilisant les images de la scène. L'étape de rectification n'est requise que pour les algorithmes de type "recherche épipolaire" sur deux images.

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'orientation 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'utilisateur.

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 [56–58]. 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

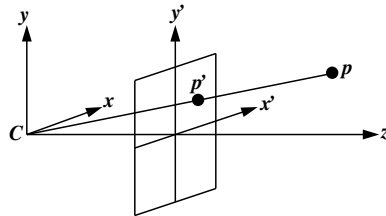


Figure 2.1. Modèle de caméra sténopé. Le centre optique (C) est situé à l’origine. L’axe optique est l’axe z . Les axes x' , y' de l’image sont parallèles aux axes x et y . Le point 3D p se projette dans l’image sur le point p' .

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

$$\mathbf{p}' = \mathbf{T} \cdot \mathbf{p} \quad (2.1)$$

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

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

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

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

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 \mathbf{J} , c'est-à-dire

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

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 (\mathbf{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 \mathbf{T} est la composition d'une matrice de projection \mathbf{J} (voir équation 2.4) et d'une matrice de passage \mathbf{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

$$\mathbf{T} = \mathbf{J} \cdot \mathbf{W} \quad (2.5)$$

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

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

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

$$\mathbf{W} = \mathbf{W}^{int} \cdot \mathbf{W}^{ext} \quad (2.6)$$

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 tridimensionnelle 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 ($\mathbf{A}_{2 \times 2}$) et d'une translation 2D ($\mathbf{t}_{2 \times 1}$) dans l'image, donc dans le plan $x - y$ de la caméra

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

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 ($\mathbf{A}_{3 \times 3}$) et d'une translation 3D ($\mathbf{t}_{3 \times 1}$)

$$\mathbf{W}^{ext} = \begin{bmatrix} \mathbf{A}_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.7)$$

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

$$\mathbf{p}'_a = \mathbf{J} \cdot \mathbf{W}_a \cdot \mathbf{p}_w \quad (2.8)$$

$$\mathbf{p}'_b = \mathbf{J} \cdot \mathbf{W}_b \cdot \mathbf{p}_w. \quad (2.9)$$

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 liée 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

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

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

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

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

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

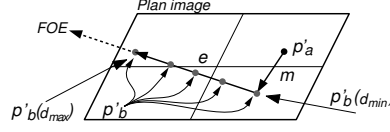


Figure 2.2. Segment épipolaire. Le point \mathbf{p}'_b correspondant au point \mathbf{p}'_a se situe entre $\mathbf{p}'_b(d_{min})$ et $\mathbf{p}'_b(d_{max})$ (sur le vecteur \mathbf{e}). Le vecteur \mathbf{m} représente la composante rotationnelle du déplacement de \mathbf{p} . La droite contenant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

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

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

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

$$\mathbf{p}'_b = \mathbf{p}'_a + \mathbf{m} + k \mathbf{e} \quad 0 \leq k \leq 1. \quad (2.15)$$

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

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

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:

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

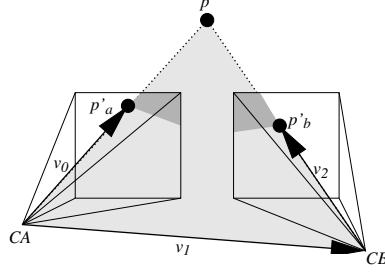


Figure 2.3. Géométrie épipolaire. Les vecteurs \mathbf{v}_0 , \mathbf{v}_1 et \mathbf{v}_2 , issus des points \mathbf{p}'_a , \mathbf{p}'_b et des centres optiques CA and CB, sont tous coplanaires.

ou sous forme matricielle

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

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

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

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

$$\begin{aligned} \mathbf{v}_0 &= \mathbf{p}'_a - \mathbf{CA}_a = \mathbf{p}'_a \\ \mathbf{v}_1 &= \mathbf{CB}_a - \mathbf{CA}_a = [\mathbf{W}_{ab}]_t \\ \mathbf{v}_2 &= \mathbf{J} \cdot \mathbf{W}_{ab} \cdot ([\mathbf{p}'_b; 1] - [\mathbf{CB}_b; 1]) \\ &= \mathbf{J} \cdot \mathbf{W}_{ab} \cdot [\mathbf{p}'_b; 0] \\ &= [\mathbf{W}_{ab}]_R \cdot \mathbf{p}'_b. \end{aligned}$$

Nous pouvons maintenant développer l'équation 2.16 pour obtenir

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

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* \mathbf{E}

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

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

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

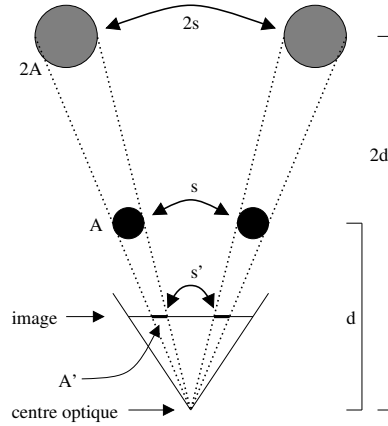


Figure 2.4. Ambiguïté de profondeur. Un objet de taille A , à une profondeur d , projette la même image A' qu'un objet de taille $2A$ à une profondeur $2d$. De même, la projection s' d'un déplacement s , à une profondeur d , est identique à celui d'un déplacement $2s$ à une profondeur $2d$.

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 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 [29]). Plutôt que de con-

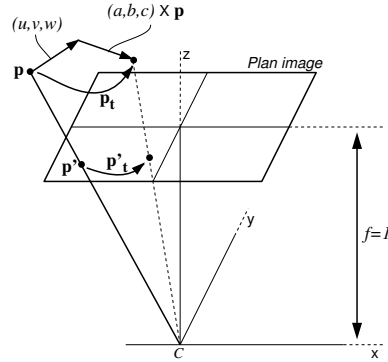


Figure 2.5. Modèle du mouvement de caméra. Le vecteur $\omega = (a, b, c)$ représente l'axe de rotation et la grandeur de la rotation. Le vecteur $\mathbf{t} = (u, v, w)$ représente la vitesse de translation. La vitesse \mathbf{p}_t , une fois projetée, devient \mathbf{p}'_t .

sidérer 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 temps¹

$$\mathbf{p}_t = -\mathbf{t} - \omega \times \mathbf{p} \quad (2.20)$$

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

$$\begin{aligned} \mathbf{p}'_t &= \frac{d\mathbf{p}'}{dt} = \frac{d}{dt} \left(\frac{\mathbf{p}}{\mathbf{p} \cdot \hat{\mathbf{z}}} \right) = \frac{\mathbf{p}_t(\mathbf{p} \cdot \hat{\mathbf{z}}) - (\mathbf{p}_t \cdot \hat{\mathbf{z}})\mathbf{p}}{(\mathbf{p} \cdot \hat{\mathbf{z}})^2} \\ &= \frac{\hat{\mathbf{z}} \times (\mathbf{p}_t \times \mathbf{p}')}{\mathbf{p} \cdot \hat{\mathbf{z}}} \end{aligned}$$

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.

$$= -\hat{\mathbf{z}} \times \left[\mathbf{p}' \times \left(\mathbf{p}' \times \omega - \frac{\mathbf{t}}{\mathbf{p} \cdot \hat{\mathbf{z}}} \right) \right] \quad (2.21)$$

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 2.1, section 2.1) pour évaluer \mathbf{t} et ω .

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 [17, 55]).

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 ??); certaines autres utilisent plutôt des points saillants (*feature points*) mis en correspondance (voir section ??). 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 ??, 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 [9, 31]).

- 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 [29].

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 \otimes , 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 corresponde 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 [55].

Chapitre 4

MOTION WITHOUT STRUCTURE

Cet article [56] 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 than 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, 30, 34, 66]). Even methods using the intensity derivatives directly or normal flow (see [1, 22, 29, 47, 64, 66, 69]), suffer from high noise sensitivity. For feature-based methods, the reliable selection and tracking of meaningful feature points is generally very difficult, see [15, 41, 68, 69].

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 [19], 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 [20] 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 symmetrically with the distance between the points. In previous work [20], 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 [49]) on the accuracy of motion estimation is also discussed.

Section ?? 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 $\sigma^2(\delta)$. 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 $\delta = (0, 0)$. 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 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.

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.

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

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

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}_∞ , 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)

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.

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

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

Suppose we now take a second segment v starting also at \mathbf{p}_∞ , but closer to the point \mathbf{p}_z . A set of samples v_i is chosen with the same sampling¹ 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

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

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

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

and symmetrically 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).

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.

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.

5.1 Rectification plane

Une solution à la rectification d'image a été proposée par Ayache et Hansen [2] et Faugeras [21]. 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 [21], servant d'introduction à notre méthode exposée 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 reprojétées.

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 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

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

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

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

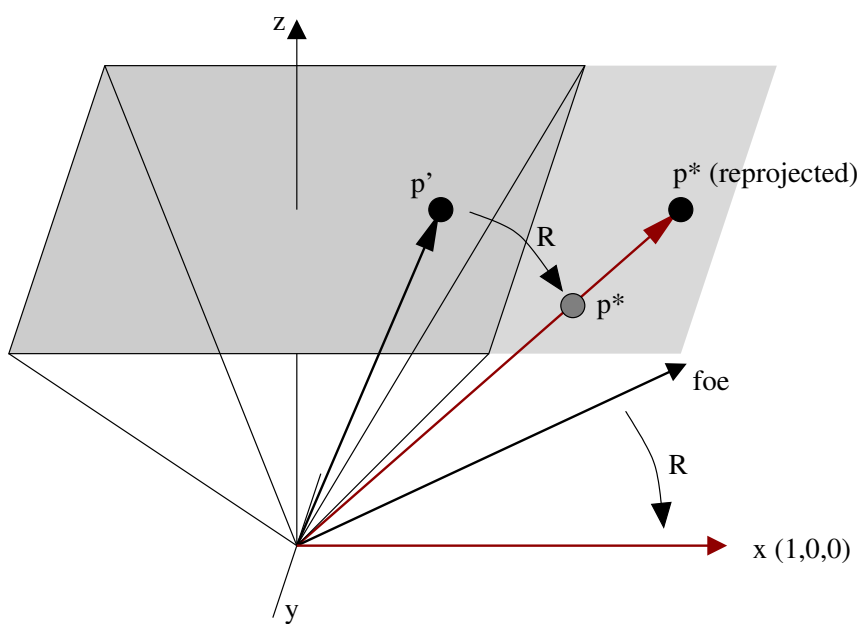


Figure 5.3. Rectification plane. La rotation R qui transforme le foe vers l'axe x est appliquée à un point image p' pour donner un point rectifié p^* qui doit ensuite être reprojété sur le plan image.

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 effectuer 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 reprojété 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 reprojecté 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 [58] 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.

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)

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 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 larger¹. 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.

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 im-

¹ See Sec. ?? for a detailed discussion.

age, 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 [44]) is available (see Sec. ??).

Many variants of the planar rectification scheme have been proposed [2, 21, 38]. A detailed description based on the *essential matrix* is given in Hartley and Gupta [28]. In Courtney *et al.* [14], a hardware implementation is proposed. In Papadimitriou and Dennis [50], 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.* [53], 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. ??).

Sec. 6.2 describes planar rectification in more detail. The cylindrical rectification method is then presented in Sec. ?. It describes the transformation matrix whose three components are explicitly detailed in Sec. ?, ?, and ?. Sec. ? discusses the practical aspects of finding the set of corresponding epipolar lines in both images to rectify. It is demonstrated in Sec. ? that it is possible to use uncalibrated as well as calibrated cameras. A measure of image distortion is introduced in Sec. ? and used to show how both rectification methods behave for different camera geometries. Examples of rectification for different camera geometries are presented in Sec. ?.

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) .

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, 21, 38] 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

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

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

$$v_h = u_x F_6 + u_y F_7 + u_h F_8 \neq 0. \quad (6.2)$$

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.

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 [45]) 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 [62]. Malheureusement, une seule largeur w ne convient généralement pas à toute l'image. Ainsi, plusieurs méthodes [10, 36] 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 [63].

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, peu 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 reprojété 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 reprojétée 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 $\mathcal{V} = \{A, B, C, D, \dots\}$, pour donner un ensemble $\mathcal{P}(\mathbf{p}'_a, d)$ de points reprojétés

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

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

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

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)

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

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

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 .

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 [52] et Kanade *et al.* [37]. Par ailleurs, Boykov *et al.* [10] 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.

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, 12, 42, 59]. 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 [61], leur sont aussi très apparentées.

D'autres méthodes à caractère global utilisent les réseaux de neurones [46] et la programmation génétique [39]. 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 [23]. 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 [60], 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 [26] et Goldberg [25].

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 [18, 43, 48]. 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

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

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 [48] 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.* [18] 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.* [65] et Yeshurun *et al.* [71], 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 Bovik [13] 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.

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 donner¹

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

avec

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

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 [45]. De même, Ohta et Kanade [48] et Cox et al. [18] 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 7.6. 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 $\mathbf{p}'_b(d) = (x' + d, y')$.

7.5.2 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 [18, 48]. 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.

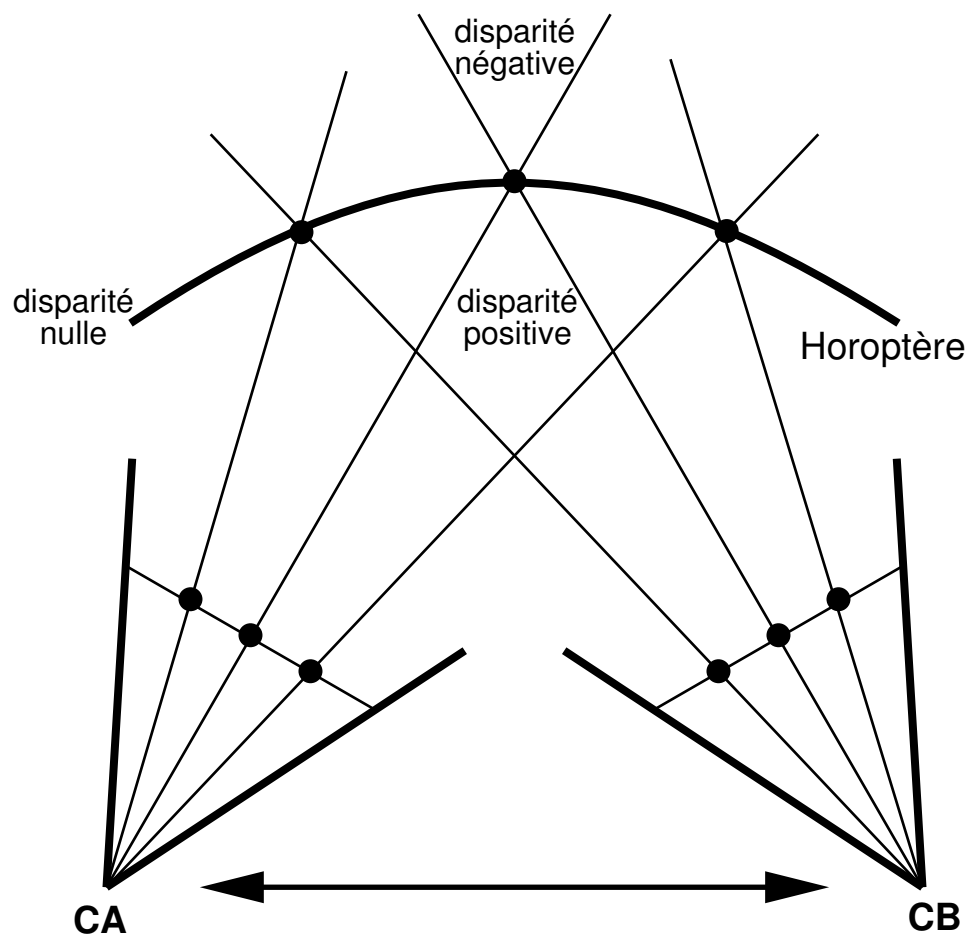


Figure 7.7. Géométrie de caméra convergente. L'horoptère est la zone où le déplacement apparent est nul.

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 [16]. 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 reprojété 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

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

caméras. Certains proposent de prendre tour à tour comme référence toutes les paires de caméras de la scène [35]. Bien que ceci améliore sensiblement la solution, cette approche n'est pas très efficace ou élégante.

Une approche intéressante est celle du *plan+parallaxe* [40]. 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. Une 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 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 pas 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

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.

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 précisément, 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).
- 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 [42]). 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 [57] 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, 18, 21, 48]. 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 [48], 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.* [18], 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 [18, 48] 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.* [27] 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 [57] in the context of stereoscopic correspondence. Recently, Ishikawa and Geiger [33] presented a similar method as Roy and Cox [57], but expressed in the context of Markov Random Fields and applied to image segmentation. Also, Boykov *et al.* [11] 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 [16, 21, 37, 38]). In Cox [16], 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.* [18]. In Kang *et al.* [38] and in Kanade *et al* [37], 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.

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.

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 ??, 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 ??, the stereo matching problem is formulated as a maximum-flow problem. Details of the maximum-flow algorithm and performance issues are presented in Section ?. Experiments on both classic two-image and multiple-image stereo sequence are presented and discussed in Section ?.

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 [70] and Marr and Poggio [45]). 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

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

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

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

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.

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) .

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.

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, \dots, C_n provides n images I_1, \dots, 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

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

where $I_i([x' \ y']^T)$ is the intensity of pixel $[x' \ y']^T$ 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

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

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

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 appelons 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 reprojétant 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 une 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 liés aux mosaïques sont Szeliski [67], Peleg *et al.* [51] et Rousso *et al.* [54]. Notons que les travaux originaux de Peleg *et al.* [51] ne permettent pas les mouvements arbitraires de caméra. Ils ont introduit par la suite et de façon indépendante dans Rousso *et al.* [54] 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.* [35]. 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 possédant un ordre unidimensionnel 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] M. Bonneville, S. Roy, J. Meunier, et A. C. Evans. Exact solution to maximum a posteriori in mrf : an application to fmri activation segmentation. Rapport Technique 98-159, NEC Research Institute, 1998.

- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] T. Y. Chen et A. C. Bovik. Stereo disparity from multiscale processing of local image phase. Dans *ISCV*, pages 188–193, 1995.
- [14] 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.
- [15] I. J. Cox. A review of statistical data association techniques for motion correspondence. *Int. J. Computer Vision*, 10(1):53–66, 1993.
- [16] 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.
- [17] 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.
- [18] 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.
 - [19] 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.
 - [20] I. J. Cox et S. Roy. Statistical modelling of epipolar misalignment. Dans *International Workshop on Stereoscopic and Three-Dimensional Imaging*, 1995.
 - [21] O. Faugeras. *Three-dimentional computer vision*. MIT Press, Cambridge, 1993.
 - [22] 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.
 - [23] P. Fua. Reconstructing complex surfaces from multiple stereo views. Dans *Proc. Int. Conference on Computer Vision*, pages 1078–1085, 1995.
 - [24] S. Geman et D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
 - [25] 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.
 - [26] A. V. Goldberg et S. B. Rao. Length functions for flow computations. Rapport Technique 97-055, NEC Research Institute, Princeton NJ, 1997.

- [27] 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.
- [28] 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.
- [29] B. K. P. Horn et E. J. Weldon, Jr. Direct methods for recovering motion. *Int. J. Computer Vision*, 2:51–76, 1988.
- [30] K. Horn et B. Schunck. Determining optical flow. *Artificial intelligence*, 17:185–203, 1981.
- [31] Michal Irani, Benny Rousso, et Shmuel Peleg. Computing occluding and transparent motions. *Int. J. Computer Vision*, 12(1):5–16, 1994.
- [32] H. Ishikawa et D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. Dans *Proc. European Conference on Computer Vision*, Freiburg, Germany, juin 1998.
- [33] 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.
- [34] 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.
- [35] 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.

- [36] 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.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] R. Kumar, P. Anandan, et K. Hanna. Shape recovery from multiple views: a parallax based approach. Dans *ARPA Image Understanding Workshop*, Monterey, CA, 1994.
- [41] 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.
- [42] S. Li. *Markov random field modeling in computer vision*. Springer-Verlag, 1995.
- [43] 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.

- [44] Q.-T. Luong et O. D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *Int. J. Computer Vision*, 17:43–75, 1996.
- [45] D. Marr et T. Poggio. A theory of human stereopsis. *Proceedings of the Royal Society*, B 204:301–328, 1979.
- [46] M. S. Mousavi et R. J. Schalkoff. An implementation of stereo vision using a multi-layer feedback architecture. *IEEE Trans. Systems Man and Cybernetics*, 24(8):1220–1238, 1994.
- [47] S. Negahdaripour et B. K. P. Horn. Direct passive navigation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.
- [48] 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.
- [49] J. Oliensis. Rigorous bounds for two-frame structure from motion. Rapport Technique 95-155, NEC Research Institute, Princeton, NJ, 1993.
- [50] 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.
- [51] 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.
- [52] L. Raffo. Adaptive resistive network for stereo depth estimation. *Electronics Letters*, 31(22):1909–1910, octobre 1995.

- [53] 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.
- [54] 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.
- [55] S. Roy. Analyse d’images stéréoscopiques basée sur la détermination du flux optique. Mémoire de maîtrise, Université de Montréal, Décembre 1992.
- [56] 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.
- [57] 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.
- [58] 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.
- [59] 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.
- [60] A. Scheuing et H. Niemann. Computing depth from stereo images by using optical flow. *Pattern recognition letters*, 4:205–212, 1986.

- [61] 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.
- [62] Y. Shirai. *Three-Dimensional Computer Vision*. Springer-Verlag, Berlin, 1987.
- [63] 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.
- [64] D. Sinclair, A. Blake, et D. Murray. Robust estimation of egomotion from normal flow. *Int. J. Computer Vision*, 13(1):57–69, 1994.
- [65] 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.
- [66] V. Sundareswaran. Egomotion from global flow field data. Dans *Proc. IEEE Workshop on Visual Motion*, pages 140–145, Princeton, NJ, 1991.
- [67] R. Szeliski. Image mosaicing for tele-reality applications. Dans *IEEE Workshop on Applications of Computer Vision*, pages 44–53, 1994.
- [68] 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.
- [69] 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.

- [70] Y. Yang et A. L Yuille. Multilevel enhancement and detection of stereo disparity surfaces. *Artificial Intelligence*, 78:121–145, 1995.
- [71] 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.

Annexe A

CHAMPS ALÉATOIRES DE MARKOV

Les champs aléatoires de Markov (*Markov Random Field*, ou *MRF*) sont principalement utilisés en traitement d'images pour résoudre des problèmes de restauration d'images, d'analyse stéréoscopique, et de segmentation de textures (voir [42] et [24]). Dans ce contexte, le problème consiste à estimer la valeur la plus probable (*MAP*, ou *maximum a posteriori estimate*) d'un paramètre pour chaque pixel de l'image, en minimisant globalement une fonction de coût tout en appliquant un critère de lissage dans un voisinage local de chaque pixel. Dans sa formulation générale, ce problème est difficile à solutionner puisqu'il est NP-complet.

La méthode de flot maximum présentée au chapitre 8 pour la mise en correspondance solutionne un cas particulier du MAP d'un champ aléatoire de Markov.

Ce chapitre présente le contexte général des champs aléatoires de Markov ainsi que les conditions nécessaires pour permettre une estimation efficace et optimale du MAP par la méthode du flot maximum. Il s'inspire de travaux en cours [8] sur la segmentation d'images médicales à partir des champs aléatoires de Markov.

A.1 Étiquetage MAP-MRF

Le problème typique d'étiquetage discret consiste en un ensemble de sites $\mathcal{S} = \{0, \dots, m-1\}$ à associer à un ensemble d'étiquettes $\mathcal{L} = \{0, \dots, n-1\}$. À chaque site $i \in \mathcal{S}$, on doit associer une étiquette $f_i \in \mathcal{L}$ de façon à former une correspondance $f = \{f_0, \dots, f_{m-1}\}$ entre \mathcal{S} et \mathcal{L} , aussi appelée une *configuration*.

Une famille de variables aléatoires $F = \{F_0, \dots, F_{m-1}\}$ définie sur l'ensemble des sites \mathcal{S} est appelée un champ aléatoire, et l'événement joint $F = f$ correspond à une

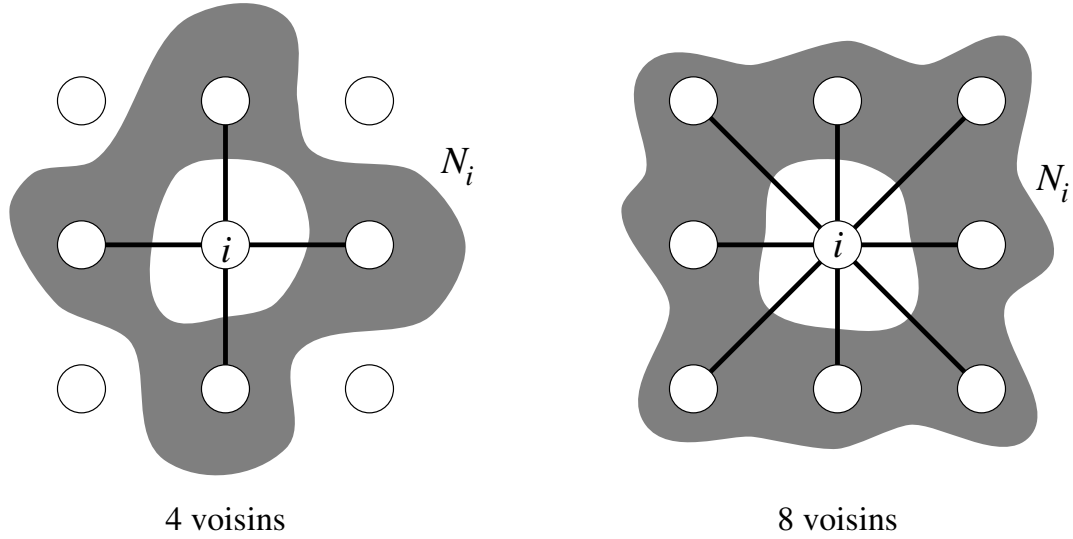


Figure A.1. Système de voisinage d'un champ aléatoire de Markov pour quatre et huit voisins. Le voisinage \mathcal{N}_i (en gris) du site i (au centre) définit la relation d'adjacence (arcs reliant les sites).

réalisation de ce champ.

Pour ce qui va suivre, on utilisera $Pr(F = f)$ pour désigner la probabilité de l'événement joint $F = f$ et $Pr(F_i = f_i)$ pour la probabilité marginale que la variable aléatoire F_i prenne la valeur f_i . La relation d'adjacence entre les sites de \mathcal{S} est décrite par un système de voisinage $\mathcal{N} = \{\mathcal{N}_i : i \in \mathcal{S}\}$ où \mathcal{N}_i est l'ensemble des sites dans le voisinage du site i et tel que $i \notin \mathcal{N}_i$. La figure A.1 donne deux exemples de tels systèmes de voisinages.

Un champ aléatoire F est un champ aléatoire de Markov sur l'ensemble des sites \mathcal{S} par rapport au système de voisinage \mathcal{N} si et seulement si les conditions suivantes sont satisfaites:

$$Pr(F_i = f_i \mid F_j = f_j, j \in \mathcal{S} \setminus \{i\}) = Pr(F_i = f_i \mid F_j = f_j, j \in \mathcal{N}_i) \quad (\text{A.1})$$

$$Pr(F = f) > 0 \quad \forall f \quad (\text{A.2})$$

La condition A.1 est la propriété de Markov. Elle établit que la probabilité que F

prenne une certaine valeur dans un certain site, en assumant les valeurs de F connues partout ailleurs, ne dépend que des valeurs de F dans le voisinage de ce site. La condition de positivité (A.2) garantit que toutes les configurations sont possibles.

Le problème avec la formulation des MRF est que la distribution des probabilités des configurations $Pr(F = f)$ n'est pas évidente, et il est difficile d'établir des probabilités conditionnelles locales de façon à obtenir une distribution jointe valide. Le théorème de Hammersley-Clifford résout ce problème en établissant un lien entre la propriété de Markov (A.1) et la distribution de Gibbs. Le théorème établit que *F est un MRF sur \mathcal{S} par rapport à \mathcal{N} si et seulement si F est un champ aléatoire de Gibbs (GRF, ou Gibbs random field) sur \mathcal{S} par rapport à \mathcal{N} .*

A.1.1 Champ aléatoire de Gibbs

Une famille de variables aléatoires est un GRF sur \mathcal{S} par rapport à \mathcal{N} si et seulement si la probabilité de ses configurations suivent une distribution de Gibbs

$$Pr(F = f) = \frac{e^{-U(f)}}{Z} \quad (\text{A.3})$$

où $Z = \sum_f e^{-U(f)}$ est une constante de normalisation, appelée fonction de répartition, obtenue en sommant le numérateur sur l'ensemble des configurations. La fonction d'énergie $U(f)$ est définie en termes de potentiels de cliques $V_c(f)$ sur l'ensemble des cliques $c \in \mathcal{C}$

$$U(f) = \sum_{c \in \mathcal{C}} V_c(f) \quad (\text{A.4})$$

Une clique c est un sous-ensemble quelconque de sites de \mathcal{S} . L'ensemble \mathcal{C} de toutes les cliques possibles se décompose donc en sous-ensembles \mathcal{C}_k formés uniquement de cliques de k sites. On a

$$\mathcal{C} = \bigcup_{0 < k < |\mathcal{S}|} \mathcal{C}_k$$

Nous allons nous restreindre au sous-ensemble de \mathcal{C}_2 ne contenant que des cliques c de deux sites voisins, définies comme $c = \{i, j\}$ tel que $j \in \mathcal{N}_i$. Le potentiel de cliques

correspondant s'exprime comme $V_{\{i,j\}}(f_i, f_j)$. Tous les autres potentiels de cliques sont fixés à zéro. On peut donc reformuler la fonction d'énergie de l'équation A.4 comme

$$U(f) = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} V_{\{i,j\}}(f_i, f_j) \quad (\text{A.5})$$

L'équivalence MRF-GRF établie par le théorème de Hammersley-Clifford permet d'utiliser cette fonction d'énergie pour exprimer directement la distribution jointe de probabilité $Pr(F = f)$.

A.1.2 Étiquetage bayésien

Pour établir l'étiquetage bayésien d'un MRF, on procède à la maximisation de la distribution *a posteriori* $Pr(F = f \mid X = x)$, où x représente une observation. Selon le théorème de Bayes, on a

$$Pr(F = f \mid X = x) \propto Pr(X = x \mid F = f) Pr(F = f) \quad (\text{A.6})$$

où $Pr(F = f)$ est la distribution *a priori* des configurations et $Pr(X = x \mid F = f)$ est appelé *fonction de vraisemblance* (*likelihood function*). La vraisemblance exprime la probabilité d'obtenir une observation x à partir d'une configuration connue f . La quantification de la vraisemblance requiert un modèle permettant de prédire les observations à partir des valeurs des inconnues tout en tenant compte des caractéristiques statistiques du bruit associé à la mesure de ces observations. La fonction de vraisemblance intègre donc le processus de dégradation des observations. Si on assume un bruit indépendant et uniformément distribué (IUD), la vraisemblance prend la forme

$$Pr(X = x \mid F = f) = \prod_{i \in \mathcal{S}} Pr(X_i = x_i \mid F_i = f_i) \quad (\text{A.7})$$

Finalement, on peut reformuler le problème d'étiquetage bayésien d'un MRF en un problème de minimisation d'une fonction d'énergie. En effet, on peut maintenant combiner les équations A.3, A.5, A.6 et A.7 pour reformuler la maximisation de la

probabilité *a posteriori* $Pr(F = f \mid X = x)$ sur l'ensemble des configurations en une minimisation d'une fonction d'énergie par la transformation

$$\begin{aligned}
& \max [Pr(F = f \mid X = x)] \\
&= \max [Pr(X = x \mid F = f) Pr(F = f)] \\
&= \max [Pr(X = x \mid F = f) e^{-U(f)}] \\
&= \max \left[\prod_{i \in \mathcal{S}} Pr(X_i = x_i \mid F_i = f_i) e^{-U(f)} \right] \\
&= \max \left[\ln \left(\prod_{i \in \mathcal{S}} Pr(X_i = x_i \mid F_i = f_i) e^{-U(f)} \right) \right] \\
&= \min \left[U(f) - \sum_{i \in \mathcal{S}} \ln (Pr(X_i = x_i \mid F_i = f_i)) \right] \\
&= \min \left[\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} V_{\{i,j\}}(f_i, f_j) - \sum_{i \in \mathcal{S}} \ln (Pr(X_i = x_i \mid F_i = f_i)) \right] \\
&= \min [E(f)]
\end{aligned}$$

avec la fonction d'énergie

$$E(f) = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} V_{\{i,j\}}(f_i, f_j) - \sum_{i \in \mathcal{S}} \ln(Pr(X_i = x_i \mid F_i = f_i)) \quad (\text{A.8})$$

A.1.3 Modélisation a priori et potentiels de cliques

La probabilité *a priori* d'une configuration peut être modélisée directement par les potentiels des cliques $V_{\{i,j\}}(f_i, f_j)$ de la distribution de Gibbs. La nature de ces potentiels détermine la complexité de la minimisation de la fonction d'énergie $E(f)$. Certaines formulations des potentiels de cliques permettent de trouver efficacement un minimum global, alors que pour la forme la plus générale, la recherche est exponentielle et seule une solution approximative (i.e. un minimum local) peut être obtenue.

Les potentiels de cliques sont aussi idéalement adaptés à la représentation de contraintes de lissages. Le lissage se définit comme la tendance qu'ont des sites voisins

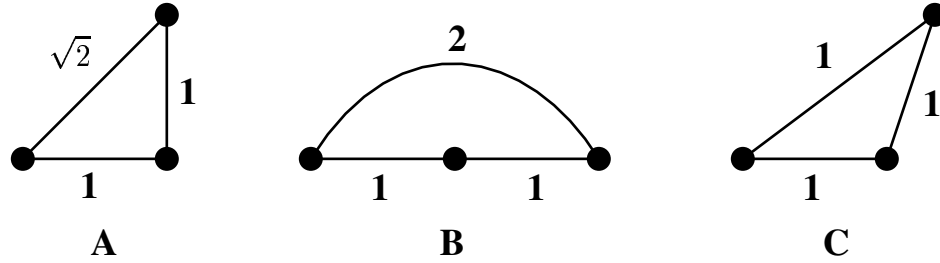


Figure A.2. Ordre des étiquettes. La distance entre étiquettes peut présenter un ordre A) bidimensionnel, B) unidimensionnel, C) non ordonné.

à posséder la même étiquette, ou des étiquettes semblables. La notion de similarité entre deux étiquettes est rattachée au concept de *distance* entre étiquettes. Cette distance est intégrée dans le potentiel de clique et fait office de *coût de discontinuité*. Comme l'illustre la figure A.2, la distance entre étiquettes peut prendre diverses formes et présenter des ordonnancements de différentes dimensions. Par exemple, la profondeur en stéréoscopie et le niveau d'intensité d'un pixel en restauration d'image présentent des étiquettes ordonnées en une dimension (figure A.2-B). Le coût de discontinuité associé à un tel ordre s'exprime par un potentiel de clique de la forme

$$V_{\{i,j\}}(f_i, f_j) = v(i, j) |f_i - f_j| \quad (\text{A.9})$$

où le terme $|f_i - f_j|$ pénalise les discontinuités proportionnellement à leur taille. Le terme $v(i, j)$ est indépendant des étiquettes et permet donc de représenter une variation de l'homogénéité du champ qui ne dépend que des sites. On a

$$v(i, j) = \beta h(i, j) \quad (\text{A.10})$$

où β est un paramètre global de lissage et $h(i, j)$ contrôle localement le degré de lissage pour la pair de sites $\{i, j\}$. Généralement, $h(i, j)$ est constant sur l'ensemble des sites.

Le déplacement libre d'un pixel (dans le calcul du flux optique, par exemple) est ordonné en deux dimensions (figure A.2-A) et ne convient pas à notre méthode.

L'étiquetage des textures d'une image, par exemple, ne présente aucun ordre (figure A.2-C). Les potentiels de cliques seront de la forme

$$V_{\{i,j\}}(f_i, f_j) = v(i, j) (1 - \delta(f_i - f_j))$$

où $\delta(\cdot)$ est la fonction impulsion, qui prend la valeur 0 partout sauf pour $\delta(0) = 1$. Ce cas est discuté par Boykov *et al.* [11], mais ne peut être résolu optimalement par une méthode efficace. Il propose une transformation vers le problème de coupe minimum multi-terminaux (*minimum multi-way cut*), qui lui-même doit être résolu par une méthode approximative puisqu'il est non polynomial.

En fait, seule une distance unidimensionnelle peut être minimisée par la reformulation en flot maximum du chapitre 8, alors que les autres distances imposent une recherche exponentielle.

A.1.4 Optimisation par le calcul du flot maximum

Cette section propose une façon de minimiser efficacement la fonction d'énergie $E(f)$ (équation A.8) utilisant les potentiels de clique de l'équation A.9, c'est-à-dire

$$E(f) = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} \beta h(i, j) |f_i - f_j| - \sum_{i \in \mathcal{S}} \ln(\text{Pr}(X_i = x_i \mid F_i = f_i)) \quad (\text{A.11})$$

en transformant l'optimisation en un problème de calcul du flot maximum dans un graphe.

Soit un graphe $G = (V, E)$. L'ensemble des noeuds V est défini par

$$V = V' \cup \{s, t\}$$

où s est la source, t le drain et V' représente l'ensemble des correspondances possibles entre les sites et les étiquettes

$$V' = \{(i, k) : i \in \mathcal{S}, k \in \mathcal{L} \cup \{M\}\}$$

où M est une étiquette factice (*dummy*).

L'ensemble des arc E est composé de trois différents groupes d'arcs. Premièrement, la source s est connectée aux correspondances dont l'étiquette est 0, et les correspondances dont l'étiquette est M sont connectés au drain t .

Le second groupe est l'ensemble $E_{penalty}$ contenant les pénalités de discontinuités. Il établit la connexion entre deux correspondances qui partagent la même étiquette et dont les sites sont voisins selon \mathcal{N} .

Le dernier groupe est l'ensemble E_{label} exprimant la fonction de vraisemblance. Chaque arc connecte deux correspondances qui partagent un même site et dont les étiquettes sont k et $k + 1$ respectivement. On a donc

$$E = \{(s, (i, 0)) : i \in \mathcal{S}\} \cup \{((i, M), t) : i \in \mathcal{S}\} \cup E_{penalty} \cup E_{label}$$

avec

$$\begin{aligned} E_{penalty} &= \{((i, k), (j, k)) : i \in \mathcal{S}, j \in \mathcal{N}_i, k \in \mathcal{L}\} \\ E_{label} &= \{((i, k), (i, k + 1)) : i \in \mathcal{S}, k \in \mathcal{L}\} \end{aligned}$$

En associant aux arcs de E une capacité de flot, on peut calculer le flot maximum du graphe G et par la même occasion la coupure de coût minimum, c'est-à-dire un sous-ensemble des arcs E qui sépare G en deux sections distinctes, respectivement connexes à la source et au drain, et dont la capacité totale est minimum.

On peut faire correspondre cette coupure minimum du graphe G à la solution de la minimisation de la fonction d'énergie par un choix judicieux des capacités des arcs. Les arcs directement connectés à la source ou au drain ont une capacité infinie, $c(s, (i, 0)) = c((i, M), t) = \infty$. Les arcs qui ne font pas partie de E ont une capacité nulle.

Puisque les arcs de l'ensemble E_{label} expriment la vraisemblance, ils sont associés au second terme de l'équation A.11 pour donner

$$c((i, k), (i, k + 1)) = -\ln(\Pr(X_i = x_i \mid F_i = f_i = k))$$

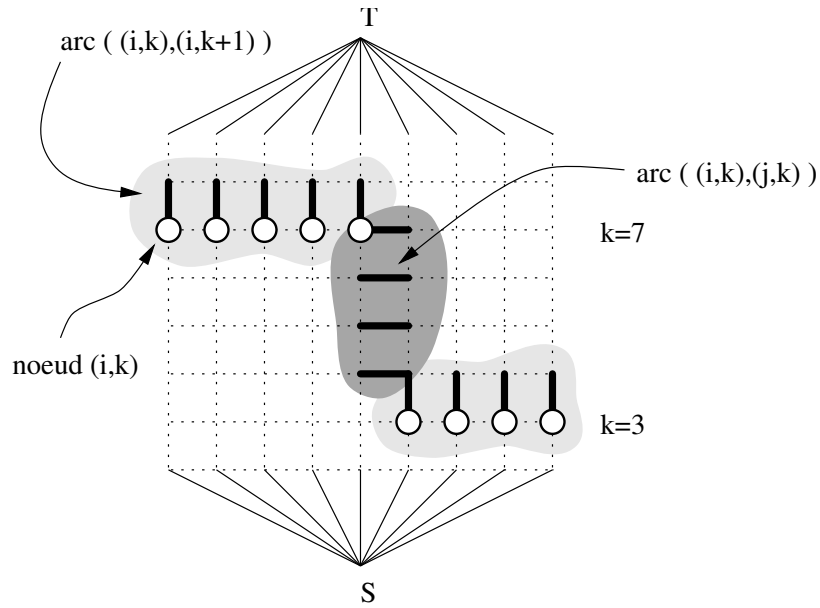


Figure A.3. Représentation des discontinuités. La zone gris foncé contient 4 arcs de type *pénalité* pour exprimer la discontinuité entre les étiquettes $k = 3$ et $k = 7$ de sites voisins. Les zones gris clair contiennent les arcs exprimant la fonction de vraisemblance.

correspondant à la vraisemblance de l'étiquette k pour le site i . Un arc $((i, k), (i, k + 1))$ coupé (i.e. faisant partie de la coupure de coût minimum) indique que l'étiquette k a été assignée au site i .

Si on assigne deux étiquettes différentes à deux sites voisins, on est en présence d'une discontinuité. Dans le contexte des MRF, ces discontinuités sont modélisées par les potentiels de cliques (équation A.9) qui font office de fonction de pénalité en associant un coût à cette discontinuité.

Pour une discontinuité entre un site i associé à l'étiquette k et un site j voisin associé à l'étiquette k' , il y aura $|k - k'|$ arcs qui font partie de la coupe de coût minimum (voir figure A.3). Ainsi, la capacité d'un arc de pénalité est définie à partir

des équations A.9 et A.10 comme

$$c((i, k), (j, k)) = \beta h(i, j) \quad (\text{A.12})$$

de façon à ce que le coût total de la discontinuité $((i, k), (j, k'))$ corresponde au potentiel des cliques, c'est-à-dire

$$\begin{aligned} & \sum_{l=\min(k, k')+1}^{\max(k, k')} c((i, l), (j, l)) \\ = & \sum_{l=\min(k, k')+1}^{\max(k, k')} \beta h(i, j) \\ = & \beta h(i, j) |k - k'| \\ = & V_{\{i, j\}}(k, k') \end{aligned}$$

A.1.5 Connaissances *a priori* et stabilité

Lorsque le paramètre de lissage β est mis à 0, on ne dispose d'aucune connaissance *a priori*, ce qui équivaut à retirer les arcs de pénalité (E_{penalty}) de l'ensemble des arcs E . La configuration résultante est triviale à évaluer et correspond à l'étiquetage de maximum de vraisemblance (ML ou *maximum likelihood*), qui est aussi celui qui contient le maximum de discontinuité.

À l'inverse, si on pose $\beta = \infty$, les connaissances *a priori* forcent l'absence de discontinuité et une seule étiquette sera associée à tous les sites.

Ainsi, la valeur de β est déterminante et il est important de démontrer que la solution évolue de façon ordonnée en fonction de ce paramètre, de façon à garantir une dégradation élégante de la solution. Des expériences préliminaires (voir [8]) montrent que la solution est remarquablement robuste aux variations de β , ce qui rend beaucoup moins critique le choix de β .

A.1.6 Preuve d'exactitude et d'optimalité

Une fois que le graphe G associé au MRF a été créé, il suffit de calculer le flot maximum et d'extraire la coupure de coût minimum pour pouvoir associer une étiquette à chaque site. La coupure minimum est composée d'arcs provenant des ensembles E_{label} et $E_{penalty}$ puisque les autres arcs de E ont une capacité infinie. La solution pour le site i sera l'étiquette k d'un arc $((i, k), (i, k + 1))$ provenant de E_{label} et faisant partie de la coupure minimum.

Certaines conditions sont requises pour garantir l'équivalence de la transformation. D'une part, il faut prouver que toute solution du MRF est exprimable sous forme d'une coupure de flot du graphe G (trivial). D'autre part, il faut aussi prouver que toute coupure minimum de G introduit une (et une seule) solution du MRF. Ceci équivaut à prouver que chaque site recevra une et une seule étiquette, ce qui n'est pas trivial. Une preuve est donnée dans [11] mais n'est pas complète parce qu'elle introduit une restriction sur la capacité des arcs et n'est donc pas nécessairement toujours applicable. Dans [33], le graphe G est explicitement modifié, par l'ajout d'arcs de capacité infinie, pour éliminer les coupures de G qui assignent plusieurs étiquettes à un seul site. Par contre, il est trivial de montrer que chaque site recevra *au moins* une étiquette. Un des développements futurs de cette thèse sera de présenter une preuve que les coupes de G qui assignent plus d'une étiquette à un site ne peuvent être minimales et ce, sans restriction sur la capacité des arcs et sans ajouter d'arcs artificiels au graphe.

A.1.7 Autres modèles a priori

La première référence à l'utilisation du flot maximum est due à Greig *et al.* [27]. Il ne résout que le cas binaire (deux étiquettes) avec des résultats limités à des cas très simples. Après la publication de l'article du chapitre 8, certaines variations et nouvelles applications ont été introduites dans les articles [11, 32, 33].

GLOSSAIRE

ARGUMENT: replacement text which customizes a \TeX macro for each particular usage.

BACK-UP: a copy of a file to be used when catastrophe strikes the original. People who make no back-ups deserve no sympathy.

CONTROL SEQUENCE: the normal form of a command to \TeX .

DELIMITER: something, often a character, that indicates the beginning and ending of an argument. More generally, a delimiter is a field separator.

DOCUMENT STYLE: a file of macros that tailors \LaTeX for a particular document. The macros described by this thesis constitute a document style.

DOCUMENT STYLE OPTION: a macro or file of macros that further modifies \LaTeX for a particular document. The option `[chapternotes]` constitutes a document style option.

FIGURE: illustrated material, including graphs, diagrams, drawings and photographs.