Exploitation de contraintes photométriques et géométriques en vision. Application au suivi, au calibrage et à la reconstruction.

par

Jamil Draréni

Thèse de doctorat effectuée sous une convention de co-tutelle

entre

l'Université de Montréal

et

l'Institut national polytechnique de Grenoble

Thèse présentée à la Faculté des arts et des sciences de l'Université de Montréal

en vue de l'obtention du grade de Philosophiae Doctor (Ph.D.) en informatique

et à

L'école doctorale, Mathématiques, Sciences et Technologie de l'Information

pour obtenir le grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Le travail a été effectué au Département d'informatique et de recherche opérationnelle de l'Université de Montréal et au sein de l'équipe-projet PERCEPTION, Laboratoire Jean Kuntzmann et INRIA Rhone-Alpes

juin, 2010

Université de Montréal

Faculté des arts et des sciences

et

L'institut national polytechnique de Grenoble

Ecole doctorale : Mathematiques, Sciences et Technologie de l'Information

Cette thèse intitulée :

Exploitation de contraintes photométriques et géométriques en vision. Application au suivi, au calibrage et à la reconstruction.

présentée et soutenue à l'Université de Montréal par:

Jamil Draréni

a été évaluée par un jury composé des personnes suivantes:

Max MIGNOTTE ..... président-rapporteur et représentant du doyen.

Sébastien ROY ..... co-directeur de recherche.

Peter STURM ..... co-directeur de recherche.

Robert LAGANIÈRE ..... examinateur externe.

David FOFI ..... examinateur externe.

Jean MEUNIER ..... membre du jury.

Augustin LUX ..... membre du jury.

RÉSUMÉ

Cette thèse s'intéresse à trois problèmes fondamentaux de la vision par ordinateur qui sont le suivi vidéo, le calibrage et la reconstruction 3D. Les approches proposées sont strictement basées sur des contraintes photométriques et géométriques présentent dans des images 2D.

Le suivi de mouvement se fait généralement dans un flux vidéo et consiste à suivre un objet d'intérêt identifié par l'usager. Nous reprenons une des méthodes les plus robustes à cet effet et l'améliorons de sorte à prendre en charge, en plus de ses translations, les rotations qu'effectue l'objet d'intérêt.

Par la suite nous nous attelons au calibrage de caméras; un autre problème fondamental en vision. Il s'agit là, d'estimer des paramètres intrinsèques qui décrivent la projection d'entités 3D dans une image plane. Plus précisément, nous proposons des algorithmes de calibrage plan pour les caméras linéaires (*pushbroom*) et les vidéo projecteurs lesquels étaient, jusque là, calibrés de façon laborieuse.

Le troisième volet de cette thèse sera consacré à la reconstruction 3D par ombres projetée. À moins de connaissance à priori sur le contenu de la scène, cette technique est intrinsèquement ambigüe. Nous proposons une méthode pour réduire cette ambiguïté en exploitant le fait que les spots de lumières sont souvent visibles dans la caméra.

Mots-clés: *vision par ordinateur, suivi automatique 2D, calibrage, caméra linéaire, projecteur vidéo, reconstruction tridimensionelle, auto-calibrage plan.*

ABSTRACT

The topic of this thesis revolves around three fundamental problems in computer vision; namely, video tracking, camera calibration and shape recovery. The proposed methods are solely based on photometric and geometric constraints found in the images.

Video tracking, usually performed on a video sequence, consists in tracking a region of interest, selected manually by an operator. We extend a successful tracking method by adding the ability to estimate the orientation of the tracked object.

Furthermore, we consider another fundamental problem in computer vision: calibration. Here we tackle the problem of calibrating linear cameras (a.k.a: pushbroom) and video projectors. For the former one we propose a convenient plane-based calibration algorithm and for the latter, a calibration algorithm that does not require a physical grid and a planar auto-calibration algorithm.

Finally, we pointed our third research direction toward shape reconstruction using coplanar shadows. This technique is known to suffer from a bas-relief ambiguity if no extra information on the scene or light source is provided. We propose a simple method to reduce this ambiguity from four to a single parameter. We achieve this by taking into account the visibility of the light spots in the camera.

Keywords: *computer vision, tracking 2D, calibration, linear camera, pushbroom, video projector, 3D reconstruction, shape-from-shadows, planar auto-calibration.*

TABLE DES MATIÈRES

LISTE DES FIGURES

à mes parents qui ne quittent jamais mes pensées ni mon coeur.

REMERCIEMENTS

Comme toute thèse, celle-ci a été jonchée de moments de joies et d'enthousiasmes mais aussi de périodes de découragements et de questionnements. La concrétisation de cette thèse n'aurait pas été possible sans la généreuse contribution de personnes fabuleuses que j'ai côtoyées.

En premier lieu, je désire exprimer ma profonde gratitude à mes directeurs de recherche, Sébastien Roy et Peter Sturm, pour m'avoir accueilli au sein de leur équipe respective. Leur soutien, leurs disponibilités et leurs conseils ont permis d'achever cette thèse.

La réussite de ma co-tutelle entre deux continents n'aurait pas été assurée sans le soutient de mes collègues et amis de part et d'autre de l'atlantique. Merci à vous, Mélissa, Caroline, Lucie, Édouard, Mohamed et Vincent d'avoir partager repas, vins-et-fromages et discussions durant mes périodes montréalaises. Une mention spéciale pour mon ami et "frère" de recherche, Nicolas Martin. Merci de ta complicité au niveau scientifique et surtout d'avoir cru aux personnages loufoques, sortis droit de mon imagination (l'épouse de N., le villageois, Mme Poudre, la loutre...)!

Mes séjours grenoblois n'auraient pas été aussi formidable sans l'escouade volley-ball/BBQ composée d'Amaël, Régis, Antoine, Simone, Visesh, Avinesh, Miles, Ramia, Gaetan et Michael. Un gros merci pour cela!

Mes remerciements s'adressent également aux formidables dames et demoiselles qui m'ont guidé à travers les dédales de l'administration. Un gros merci à Virginie Allard-Caméus, Mariette Paradis, Manon Lajeunesse, Anne Pasteur et à Marie-Eve Morency.

À vous aussi, les cinq membres de ma famille. Sachez que votre pensé a été la clé

de ma réussite.

Enfin, cette thèse est aussi dédiée à une personne merveilleuse qui a vu ce projet naître et y avait cru autant que moi : Marilèna Liguori. Spero che tu sia fiera di me e ricordati che ti voglio sempre bene.

Chapitre 1

INTRODUCTION

La vision est sans aucun doute le sens le plus complexe chez l'être humain. Les avancées en neurosciences ont confirmé cette thèse en évaluant la proportion du cortex visuel par rapport aux autres régions du cerveau. C'est dire à quel point on voit avec notre cerveau et non pas avec les yeux ! Il n'est donc pas étonnant que toute entreprise visant à imiter le système visuel humain par un ordinateur serait laborieuse, voire même vaine s'il on n'émet pas d'hypothèses simplificatrices sur les phénomènes de physique et d'optique associés à la formation de l'image.

L'image captée par l'œil est riche en contenu, diverse en nature et redondante en information. Le système visuel humain ne garde qu'une infime partie de cette information pour des traitements de haut niveau (reconnaissance de formes, de mouvements, ...) et ce, à partir de concepts de bas niveau tels que les orientations, les contours, ...etc. Ce traitement cognitif "étagé" est orchestré de sorte à inférer une information intelligible le plus vite possible aux amygdales (circuit de la peur) afin d'identifier les situations hostiles.

Ce même schéma a inspiré le neuroscientifique Anglais David Marr à la fin des années 1970 lorsqu'il a établi sa théorie de la vision artificielle qu'on appelle communément le paradigme de Marr. À partir des images sources, le modèle de Marr propose d'extraire d'abord des primitives 2D simples afin d'esquisser une première ébauche de la scène (*primal sketch*). Cette ébauche permet de créer une représentation centrée sur l'observateur par le truchement de stéréoscopie, d'analyse de mouvement, ombrages et autres propriétés de la scène. Marr désigne cette étape, l'ébauche 2.5D.

Combinée à des connaissances 3D, l'ébauche 2.5 D permet une représentation continue centrée sur la scène. Une telle représentation inclut les relations entre différents objets de la scène, les positions absolues, les angles, les orientations,... etc. Il est intéressant de noter, qu'en dépit de son âge, le paradigme de Marr est toujours d'actualité.

Marr note que le système visuel humain effectue la même tâche cognitive décrite par son paradigme à condition que l'observateur humain ait acquis ou compris au préalable, des notions de physique et de géométrie tels que les principes de la perspective qui associe la taille des objets à leur distance. D'une certaine façon, l'œil doit être "éduqué" pour interpréter la scène. Ces notions s'acquièrent en vision par ordinateur à travers des modèles mathématiques dont la formulation doit garantir un compromis entre faisabilité et réalisme. Dans certains cas, un tel compromis n'est pas garanti. On se retourne alors vers des approches inspirées de l'apprentissage cognitif où il est question de concevoir des algorithmes qui "apprennent" à partir de modèles et tentent d'en isoler des structures (patterns). Ces méthodes se regroupent sous le thème de l'apprentissage machine.

Cette thèse présente nos contributions à trois problèmes fondamentaux reliés à la vision par ordinateur. Il sera question de suivi d'objets, de calibrage de caméras et de reconstruction 3D.

Le suivi d'objet (ou *tracking* en anglais) occupe une place fondamentale en vision par ordinateur. Le but est d'estimer le déplacement d'un ou plusieurs pixels dans le temps à partir de flux vidéo. La surveillance, l'asservissement de robots et le suivi d'acteurs n'en sont que quelques applications. Notre contribution dans ce domaine est l'estimation de l'orientation de l'objet d'intérêt en plus de sa position. En général, le suivi d'objets ne s'intéresse pas aux propriétés tridimensionnelles de la scène et donc peut se faire sans aucune connaissance a priori sur la caméra. On parle alors de vision non-calibrée. Cependant, si le but d'un algorithme est de mesurer des propriétés métriques de la scène (distances, angles, ...), les paramètres de la caméra deviennent alors nécessaires. En vision, ces paramètres s'estiment par un processus

qu'on appelle *calibrage*. Notre intérêt pour le calibrage se limitera à deux sortes de dispositifs : les caméras linéaires et les vidéo projecteurs.

Le schéma adopté à la résolution de chacun de ces problèmes, suit les grandes lignes du paradigme de Marr. En effet, quelle que soit la nature du problème abordé, on essaiera le plus possible de dégager des contraintes géométriques ou photométriques à partir des images. Ces contraintes seront par la suite combinées aux formalismes propres à chaque problème afin d'y apporter une solution nouvelle.

Organisation de la thèse. Dans cette introduction, nous avons brièvement introduit quelques concepts de la vision ainsi que les sujets étudiés.

Le chapitre 2 présente les éléments de base utile à la lecture de cette thèse ainsi que les notations adoptées.

Chaque sujet traité sera réparti sur deux chapitres. Un premier qui vise à expliquer les concepts essentiels et un second qui étaye notre contribution sous forme d'article scientifique.

Ainsi, aux chapitres 3 et 4 il sera question de suivi de mouvement à l'aide du paradigme *mean-shift*. Le calibrage de caméra en général et des caméras linéaires en particulier, fera l'objet des chapitres 5 et 6. Nous entamerons par la suite un autre volet sur le calibrage des vidéo projecteurs aux chapitres 7 et 8.

L'auto-calibrage plan des projecteurs fera l'objet des chapitres 9 et 10.

L'estimation de la structure 3D à partir d'ombres projetées constitue le dernier sujet traité dans cette thèse et sera détaillée aux chapitres 11 et 12.

Notre conclusion se fera au chapitre 13. On y résume le travail présenté et y suggérons de nouvelles perspectives pour les recherches futures.

Chapitre 2

NOTATIONS ET ÉLÉMENTS DE BASE

Dans ce chapitre, nous mettons en exergue les éléments de base utiles à la lecture de cette thèse. En plus de définir les notations et les conventions employées, nous introduirons aussi des concepts de géométrie nécessaires à la résolution de problèmes abordés dans cette thèse. Nous renvoyons le lecteur au livre de Hartley et Zisserman [30] pour plus de détails.

Les entités géométriques (points, plans, ...) sont considérées dans des espaces à dimensions différentes. Dans cette thèse, on côtoyera souvent les espaces à 2 et à 3 dimensions. Sauf mention contraire, ces entités seront représentées par des vecteurs en coordonnées homogènes et sont donc, définies à un facteur d'échelle près. L'opérateur $\sim$ indique cette égalité entre coordonnées homogènes.

Les points 3D seront notés en majuscule et en gras, comme $\mathbf{Q}$, etl'image 2D correspondante sera notée en minuscule $\mathbf{q}$. On utilisera souvent des indices pour différencier les points 3D et leurs images dans différentes caméras. Ainsi, la projection du point 3D $\mathbf{Q}_i$ dans la j$^{\text{e}}$ caméra se notera $\mathbf{q}_{ij}$.

Les vecteurs seront aussi représentés par des caractères gras minuscule, comme $\mathbf{v}$. À moins de mention contraire, $\mathbf{v}$ représente un vecteur colonne et sa transposition, $\mathbf{v}^\mathsf{T}$, un vecteur ligne.

Les matrices seront représentées par des caractères sans-sérif droits, comme H et A. Pour une matrice A donnée, ses matrices transposée, inverse et adjointe seront notées respectivement A^T, A^{-1} et $\mathsf{A}^{-\mathsf{T}}$. Une sous-matrice de A sera notée $\bar{\mathsf{A}}$. Nous tenterons le plus possible d'inclure ses dimensions en indice, par exemple, une matrice identité 3×3 amputée de sa dernière colonne sera notée $\bar{\mathsf{I}}_{3\times 2}$:

$$\bar{\mathsf{I}}_{3\times 2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Le produit vectoriel entre deux vecteurs $\mathbf{p}$ et $\mathbf{q}$ sera noté soit $\mathbf{p} \times \mathbf{q}$ ou encore $[\mathbf{p}]_\times \mathbf{q}$, où $[\mathbf{p}]_\times$ représente la matrice anti-symétrique associée au vecteur $\mathbf{p}$ de longueur 3 :

$$[\mathbf{p}]_\times = \begin{pmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{pmatrix}$$

Les plans seront désignés par des caractères grecs majuscules, comme Π. Ils sont définis par un vecteur normal et par leur distance par rapport à l'origine. Ainsi un point 3D $\mathbf{Q}$ appartient au plan défini par la normale $\mathbf{n}$ et placé à une distance d, ssi :

$$\mathbf{n} \cdot \mathbf{Q} + d = 0$$

Les coniques seront aussi notées en caractères grecs minuscules. Les coniques sont une famille de courbes planes résultant de l'intersection d'un plan avec un cône de révolution. Dans cette thèse, on travaillera avec des coniques 2D, décrites par une équation du second degré. Un point 2D, $\mathbf{q} = (x, y, 1)$, appartient à une conique, ω, définie par (a, b, c, d, e, f), ssi :

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

Le plus souvent, on privilégiera une notation matricielle de la conique, ainsi l'exemple ci-dessus s'écrirait :

$$\mathbf{q}^\mathsf{T} \omega \mathbf{q} = 0$$

Avec :

$$\omega = \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix}$$

Chapitre 3

SUIVI DE MOUVEMENT ET MEAN-SHIFT

Dans ce chapitre, nous présentons des notions générales de suivi de mouvement (SM) et nous mettrons l'accent sur un algorithme en particulier, le **mean-shift.**

De façon générale, le SM consiste à reconstruire la trajectoire d'un ou plusieurs points dans une séquence vidéo au fil du temps. Très souvent, on applique le SM à des objets ou des régions d'intérêt présélectionnés par un usager tel que les visages, des personnes, des véhicules, etc.

Le suivi de mouvement a énormément d'applications dans les médias ou la surveillance. Flouter une partie d'une vidéo à des fins d'anonymat, suivre des athlètes lors d'un évènement sportif ou détecter des intrusions ne sont qu'une partie de ces applications.

La difficulté associée au suivi de mouvement est surtout dûe à l'absence d'une structure ou un modèle connu au préalable, ce qui limite l'information exploitable aux distributions locales des luminances [1, 2]. Ceci, oblige les chercheurs à employer des méthodes locales sujettes à des instabilités numériques.

Comaniciu *et al.* [10] ont proposé un algorithme de suivi en temps réel et robuste aux changements d'illumination et aux déformations géométriques. Leur algorithme est basé sur une approche non-paramétrique, **mean-shift**. Cependant, cet algorithme ne prend en compte que la composante translationnelle de l'objet d'intérêt (changement de position 2D dans l'image) et néglige ses changements d'orientation.

Afin de mieux comprendre les limitations de la formulation originale du suivi par mean-shift [10], nous allons dans les deux prochaines sections, présenter l'algorithme mean-shift et son application pour le suivi de mouvement.

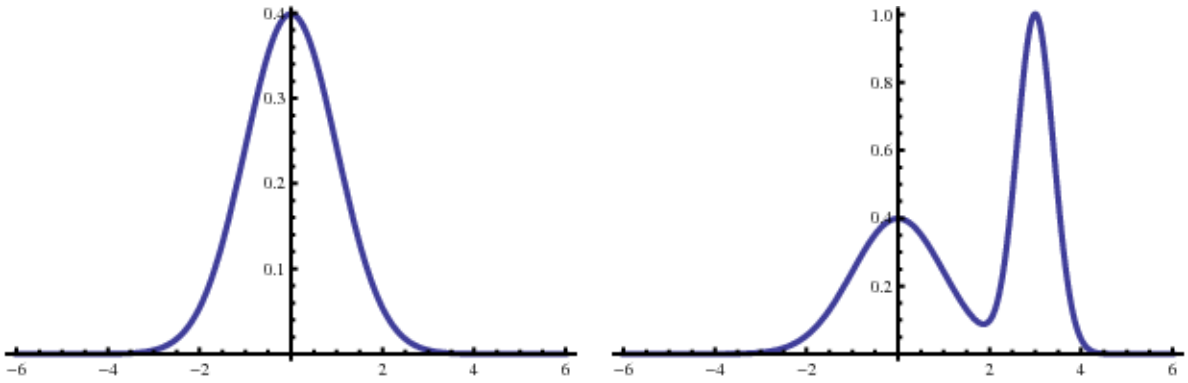

Figure 3.1. Exemples de distributions 1-D. Gauche) unimodale. Droite) bi-modale.

3.1 Mean-Shift

Mean-Shift (MS) est un algorithme non-paramétrique qui permet d'estimer récursivement le mode d'une distribution de points en k dimensions. En statistiques, le mode d'une distribution est défini comme la valeur la plus représentée dans une distribution de points. Ce qui ne doit pas être confondu avec la moyenne ou la médiane définies comme :

Moyenne : Somme des valeurs, divisée par le nombre des valeurs.

Médiane : Valeur qui partage une série numérique ordonnée en deux parties de même nombre d'éléments.

Une distribution n'a qu'une moyenne et une médiane, mais peut avoir plusieurs modes. On parle alors d'une distribution multi-modale (voir la figure 3.1).

Mean-shift repose sur une hypothèse simple. Chaque donnée représente un échantillon d'une densité de probabilité sous-jacente. Ainsi, l'ensemble des données est une représentation discrète d'une densité dont on veut estimer le mode.

Soient $\mathbf{S} = \{\mathbf{x}_i\}_{i=1...n}$ un ensemble de n points à k-dimension et $K(X)$ une fonction noyau de rayon h. La fonction noyau est une fonction à k-dimension aussi qui associe à chaque élément de $\mathbf{S}$ un poids.

On définit la moyenne pondérée $m(\mathbf{x})$ au point $\mathbf{x}$ par :

$$m(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in S} K(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{x}_i}{\sum_{\mathbf{x}_i \in S} K(\mathbf{x} - \mathbf{x}_i)}$$

Le vecteur $\overrightarrow{m(\mathbf{x}) - \mathbf{x}}$, appelé vecteur *mean-shift*, pointe vers le centre de masse local. Il a été démontré dans [11] que si le noyau $K(\mathbf{x})$ est convexe décroissant alors le vecteur mean-shift pointe toujours dans la direction de la pente ascendante de la densité. Donc, ce théorème nous assure que, la direction du vecteur mean-shift mène récursivement au maximum local (mode local) de la distribution $\mathbf{S}$. Cette procédure est illustrée à la figure 3.2. La force de mean-shift réside dans sa capacité de traiter les données de façon non paramétrique. Ainsi, la nature et la dimension de l'espace des points (position 2d, espace des couleurs RGB, ...) importent peu. Cette particularité a permis à Comaniciu *et al.* de concevoir un algorithme de suivi de mouvement robuste basé sur mean-shift.

3.2 Suivi de mouvement par Mean-Shift

Dans [10], Comaniciu *et al.* ont proposé un algorithme de suivi vidéo robuste basé sur Mean-Shift. L'histogramme des couleurs de la région d'intérêt en constitue le descripteur, ce qui permet le suivi d'objets non rigides grâce à la persistance de l'information colorimétrique et aussi une robustesse car, l'histogramme est insensible aux changements d'éclairage (moyennant une normalisation).

L'algorithme commence par estimer la représentation de l'objet d'intérêt dans un espace de propriétés (*feature space*), tel que l'espace quantifié des couleurs. La "signature" dans ce cas particulier n'est rien d'autre que l'histogramme des couleurs. Le suivi peut alors commencer à partir de la position initiale du modèle. Afin de

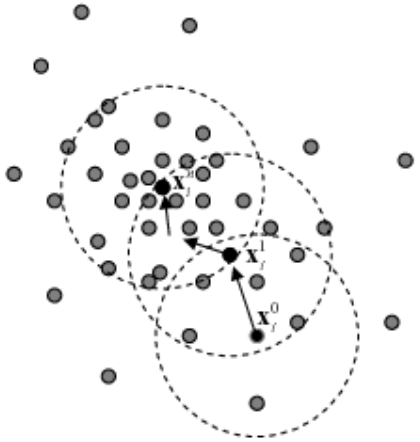

Figure 3.2. Exemple d'itérations de Mean-Shift. Chaque cercle gris représente un échantillon d'une distribution de points 2D dont on veut retrouver le mode. La moyenne pondérée des points locaux au cercle centré en $\mathbf{X}_i^0$ donne $\mathbf{X}_i^1$ qui à son tour devient le nouveau point de départ. Récursivement, on aboutit au mode local, $\mathbf{X}_i^n$.

localiser l'objet dans l'image suivante, l'algorithme de Comaniciu *et al*. estime les histogrammes autour de chaque nouvel emplacement potentiel et pour chacun d'entre eux, une mesure de ressemblance est calculée avec l'histogramme du modèle. La mesure de comparaison est la distance de Bhattacharya [7] qui est maximisée lorsque les deux histogrammes sont identiques.

Cette analyse de voisinage génère une distribution de mesures de ressemblance dont le maximum local mène au nouvel emplacement de l'objet suivi. C'est à cette étape précise de l'algorithme que mean-shift intervient pour estimer le mode local.

Tel que proposé, l'algorithme n'a pas été formulé pour estimer les changements d'orientations de la région d'intérêt. La raison est simple, toute tentative d'estimation d'orientation à l'aide d'histogramme de couleur est vouée à l'échec car ce dernier est invariant à la rotation.

Notons que, si la zone d'intérêt est carrée ou circulaire, cela ne risque pas de poser de problèmes. Dans le cas où elle serait allongé ou oblongue, la situation peut être problématique car la région d'intérêt doit être assez grande pour accommoder les différentes orientations de l'objet suivi au point d'inclure des zones supplémentaires susceptibles d'altérer la qualité du suivi (voir figure 3.3), car l'histogramme de cette région "grossie" n'est plus représentatif de l'objet original.

Au chapitre suivant, nous présentons une extension du suivi vidéo par Mean-Shift [10] qui prend en compte la rotation de l'objet d'intérêt. L'idée consiste à générer toutes les orientations possibles de l'objet et pour chacune d'entre elles, l'histogramme des orientations du gradient de l'image est calculé et utilisé comme descripteur. Ainsi, on construit un dictionnaire où chaque "mot" est une orientation de l'objet. L'estimation de nouvelles orientations de l'objet au fil du temps se fait par recherche directe dans le dictionnaire. Le temps d'exécution supplémentaire est très négligeable compte tenu de la modeste taille du dictionnaire.

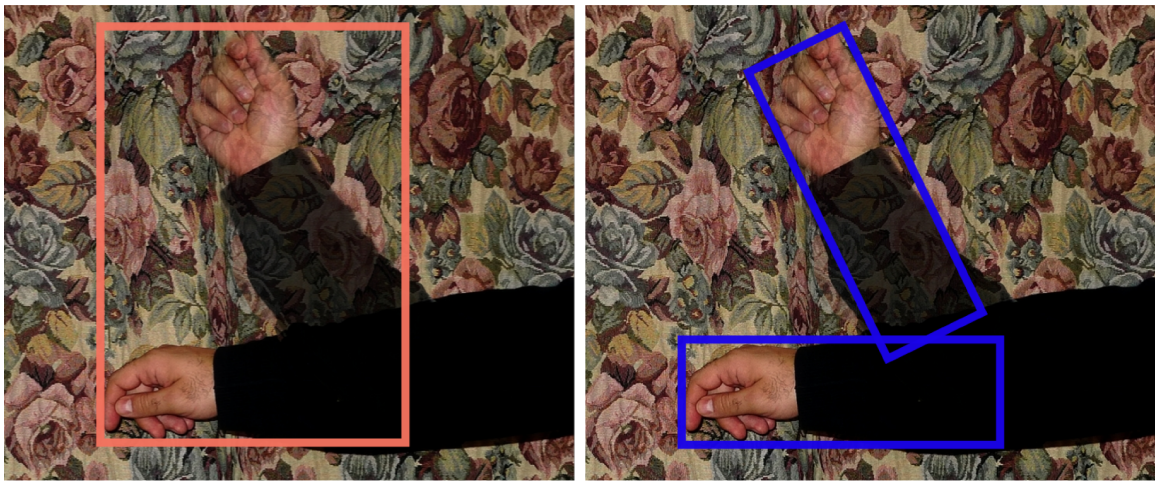

Figure 3.3. La prise en compte de la rotation lors du suivi vidéo influence la compacité de la zone d'intérêt. Dans cet exemple, l'histogramme des couleurs est utilisé comme signature. Gauche) Sans prise en compte de la rotation. Droite) Avec prise en compte de la rotation.

Chapitre 4

A SIMPLE ORIENTED MEAN-SHIFT ALGORITHM FOR TRACKING

Cet article [18] a été publié comme l'indique la référence bibliographique

Jamil Draréni et Sébastien Roy. A Simple Oriented Mean-Shift Algorithm for Tracking. Dans *Mohamed S. Kamel et Aurélio C. Campilho, editeurs, ICIAR, volume 4633 de Lecture Notes in Computer Science*, pages 558–568. Springer, 2007.

Cet article est présenté ici dans sa version originale.

Abstract

Mean-Shift tracking gained a lot of popularity in computer vision community. This is due to its simplicity and robustness. However, the original formulation does not estimate the orientation of the tracked object. In this paper, we extend the original mean-shift tracker for orientation estimation. We use the gradient field as an orientation signature and introduce an efficient representation of the gradient-orientation space to speed-up the estimation. No additional parameter is required and the additional processing time is insignificant. The effectiveness of our method is demonstrated on typical sequences.

4.1 Introduction

Object tracking is a fundamental and challenging task in computer vision. It is used in several applications such as surveillance [59], eye tracking [85] and object based

video compression/communication [12].

Although many tracking methods exist, they generally fall into two classes, *bottom-up* and *top-down* [47]. In a bottom-up approach, objects are first identified and then tracked. The top-down approach instead, uses hypotheses or signatures that discriminate the object of interest. The tracking is then performed by hypotheses satisfaction.

Recently, a *top-down* algorithm based on mean-shift was introduced for blob tracking [10]. This algorithm is non-parametric and relies solely on intensities histograms. The tracking is performed by finding the mode of a statistical distribution that encodes the likelihood of the original object's model and the model at the probing position. Because it is a *top-down* approach and it does not rely on a specific model, the mean-shift tracker is well adapted for real-time applications and robust to partial occlusions.

In [9], an extension was proposed to cope with the scale variation. However, little has been done to extend the tracker for rotational motions[84]. In fact, the original mean-shift tracker as proposed in [10] is invariant to rotations and thus, does not provide information on the target's orientation. This property is induced by the inherent spaceless nature of the histograms. While this may not be problematic for objects with symmetrical dimensions like circles or squares, it is no longer valid when the tracked objects are "thin" [84]. An example of a tracked thin object (an arm) is illustrated in Fig.4.1.

In [84], the authors used a simplified form of *correlograms* to encode pixels positions within the region of interest. Pairs of points at an arbitrarily fixed distance along the principal axis vote with their joint intensities and their angle relative to the patch's origin to generate an orientation-intensity correlogram. Once the correlogram is estimated, it is used in the mean-shift's main loop just like a regular histogram. Unfortunately no method was proposed to automatically select the fixed distance for pairs sampling. Furthermore, since the pairs are only picked along the principal axis of the object, the generated correlogram does not encode a global representation of

Figure 4.1. Result of tracking an arm using the presented oriented mean-shift tracker.

the object.

In this paper, we propose a fast and simple algorithm for an oriented mean-shift tracking. We use the original mean-shift formulation to estimate the object's translation and for the rotational part, a histogram of the orientations of the spatial gradients (within the region of interest) is used to assign an orientation according to a previously computed set of possible orientations' histogram. The effectiveness of the proposed method is demonstrated in experiments with various types of images. Our method can also be applied to video stabilization as our experiments will show. Real-time applications are still possible since the additional processing time is negligible.

The rest of the paper is organized as follows, in section 2, mean-shift tracking is summarized. Section 3, presents the gradient-orientation representation using histogram's LUT. The implementation of the proposed method is described in section 4. The experiments and results are reported in section 5 and we finally summarize our conclusion in section 6.

4.2 Mean-shift and Limitations

The mean-shift algorithm, as initially proposed in [23], is a non-parametric method to estimate the mode of a density-function. Let $S = \{x_i\}_{i=1..n}$ a finite set of r-dimensional data and $K(x)$ a multivariate kernel with window radius h. The sample mean at x is defined as:

$$m(x) = \frac{\sum_{x_i \in S} K(x - x_i).x_i}{\sum_{x_i \in S} K(x - x_i)}$$

The quantity $m(x) - x$ is called the mean-shift vector. It has been proven that if $K(x)$ is an isotropic kernel with a convex and monotonic decreasing profile, the mean-shift vector always points in the direction of the maximum increase in the density. Thus, following this direction recursively leads to the local maximum of the density spanned by S. Examples of such kernels are the gaussians and Epanechnikov kernels.

The reader is referred to [23, 8, 10] for further details on the mean-shift algorithm and related proof of convergence.

4.2.1 Mean-Shift for Tracking

Comaniciu et al.[10] took advantage of the mean-shift's property and proposed an elegant method to track blobs based on intensities histograms. The algorithm finds the displacement $\triangle y$ of the object of interest S as a weighted sum:

$$\triangle y = \frac{\sum_{x_i \in S} w_i.K(x - x_i).x_i}{\sum_{x_i \in S} w_i.K(x - x_i)}$$

Where w_i are weights related to the likelihood of the model and the target's intensities histograms. The estimation is recursive until the displacement's magnitude $||\triangle y||$ vanishes (or reaches a predefined value).

Unfortunately, the mean-shift tracker can not infer the orientation of an object based on its intensity histogram. To overcome this limitation, the tracker must use clues related to the spatial organization of the pixels or parameters that describe textures. Among those clues, image gradients are good candidates because their orientations vary when the image undergoes a rotation and are easy to compute.

4.2.2 Gradients and Gradients Histogram

Let I be an image. The first order gradient of I at position (x, y), noted ∇I_{xy} is defined as:

$$\nabla I_{xy} = [I_x, I_y]^T = \begin{bmatrix} I(x+1, y) - I(x-1, y) \\ I(x, y+1) - I(x, y-1) \end{bmatrix} \tag{4.1}$$

The orientation and the magnitude of the gradient vector I_{xy} are given by:

$$\theta(x, y) = tan^{-1}\left(\frac{I_y}{I_x}\right) \; ; \; mag(x, y) = \sqrt{I_x^2 + I_y^2}$$

It is clear that the orientation is independent of the image translation. However, a rotation of the image yield a rotation of the gradient field by the same amount. This property can be used to assign an orientation to the object of interest. Instead of keeping track of the gradient field itself, it is more convenient to build a histogram of gradient's orientations. This representation has been used in Lowe's SIFT [43] to assign an orientation to the keypoints.

In the present work, the m-bin orientation histogram O of an object is computed as:

$$O_m = C \sum_{i=1}^{i=n} mag(p_i) \cdot \delta[\theta(p_i) - m] \tag{4.2}$$

Where $p_0, p_1, ...p_n$ are the n pixels of the object of interest and the normalization constant C is computed as to insure that $\sum_{u=1}^{u=m} O_u = 1$. δ is the Kronecker delta function.

$\theta(p_i)$ and $mag(p_i)$ are functions that return the orientation and the magnitude of the gradient at pixel p_i as defined in (4.2) As opposed to a regular intensity histogram, each sample modulates its contribution with its magnitude. The reason behind this choice is two-fold: first, we generally observe that gradients with larger magnitudes tend to be more stable ; second, the gradient is known to be very sensitive to noise, thus weighting the votes with their magnitudes is like privileging samples with a good signal to noise ratio.

As opposed to [43], we do not extract a dominant orientation from the histogram. Rather, we keep the whole histogram as an orientation signature.

Histograms and bin width

One of the major problem that arises when estimating a histogram (or any density function) from a finite set of data is to determine the bin width of the histogram. A large bin width gives an over-smoothed histogram with a coarse block look, whereas

a small bin width results in an under-smoothed and jagged histogram [74]. In [58], Scott showed that the optimal bin width W, which provides an unbiased estimation of the probability density is given by:

$$W = 3.49 \cdot \sigma \cdot N^{-1/3} \tag{4.3}$$

Where N is the number of the samples and σ is the standard deviation of the distribution. We used a more robust formulation described in [35]:

$$W = 2 \cdot IQR \cdot N^{-1/3} \tag{4.4}$$

The interquartile range (IQR) is the difference between the 75^{th} and 25^{th} percentile of the distribution. Note that (4.4) does not contain σ, thereby reducing the risk of bias. The bin width computed with (4.4) is the one we use throughout our experiments.

4.3 Tracking with Gradient Histograms

A single orientation histogram encodes only the gradient distribution for one specific orientation. Thus, to infer the orientation from a gradient histogram, a LUT of gradient histograms corresponding to all image orientations must be built beforehand (at the initialization step). During the tracking process, the gradient histogram of the object must be compared against the histograms in the LUT. The sought orientation is the one that corresponds to the closer histogram in the LUT. A histogram's likelihood can be computed in different ways. We used the histogram intersection as introduced in [68] for its robustness and ease of computation.

The intersection of two m-bins histograms h_1 and h_2 is defined as:

$$h_1 \cap h_2 = \sum_{i=1}^{i=m} Min(h_1[i], h_2[i]) \tag{4.5}$$

Where $Min()$ is a function that returns the minimum of its arguments. It is clear that the closer the histograms, the bigger the intersection score. The look-up table

of histograms captures the joint orientation-gradient space of the object and can also be seen as a 2D histogram.

In the following subsections, two methods are introduced to construct a histogram gradient table: *Image-Rotation Voting* and *Gradient-Rotation Voting*.

4.3.1 Image-Rotation Voting

This is the simplest way to gather histograms of gradients for different orientations. The image of the tracked object is rotated by 360° around its center. The rotation is performed by a user-defined step (2° in our experiments) and an orientation histogram is computed at each step. The resulting histograms are stored in a stack and they form the gradient's histograms LUT. To reduce noise due to the intensity aliasing , rotations are performed with a bi-cubic interpolation. This method is outlined in the algorithm below:

Given: Original image, target's pixels $\{p_i\}_{i=1...n}$ and a rotation step $\triangle rot$.

1. $step \leftarrow 0$, $ndx \leftarrow 0$.
2. Apply a gaussian filter on $\{p_i\}_{i=1...n}$ to reduce noise (typically 3×3).
3. Compute $\{mag_i\}_{i=1...n}$ and $\{\theta_i\}_{i=1...n}$ the orientation and magnitudes of gradients at $\{p_i\}_{i=1...n}$ according to (4.1).
4. Derive the orientation histogram O_m using $\{mag_i\}$ and $\{\theta_i\}$ according to (4.2).
5. $LUT[ndx] \leftarrow O_m$
6. $ndx \leftarrow ndx + 1$, $step \leftarrow step + \triangle rot$.
7. Rotate $\{p_i\}_{i=1...n}$ by $step$ degrees.
8. If $step < 360$ go to step 3.
9. return LUT

4.3.2 Gradient-Rotation Voting

The second method is faster and produces better results in practice. Instead of rotating the image itself, the computed gradient field of the original image is incrementally rotated and the result of each rotation votes in the proper histogram. Note that due to histogram descretization, rotating a gradient field is not exactly equivalent to shifting the histogram by the same amount. This is due to the fact that histogram sampling is generally not the same as the rotation sampling. For instance, after rotating the gradient field some samples that vote for a specific bin might still vote for the same bin whereas others may jump to an adjacent bin. They would be equivalent if the gradient histogram had a bin width of 1 (i.e 360 bins), which is not the case in practice. The gradient-rotation voting is outlined below:

Given: Original image, target's pixels $\{p_i\}_{i=1...n}$ and a rotation step $\triangle rot$.

1. $step \leftarrow 0$, $ndx \leftarrow 0$.
2. Apply a gaussian smoothing on $\{p_i\}_{i=1...n}$ to reduce noise.
3. Compute $\{mag_i\}_{i=1...n}$ and $\{\theta_i\}_{i=1...n}$ the orientation and magnitudes of gradients at $\{p_i\}_{i=1...n}$ according to (4.1).
4. Derive the orientation histogram O_m using $\{mag_i\}$ and $\{\theta_i\}$ according to (4.2).
5. $LUT[ndx] \leftarrow O_m$
6. $ndx \leftarrow ndx + 1$, $step \leftarrow step + \triangle rot$.
7. For each $\{\theta_i\}_{i=1...n}$

   Do $\theta_i \leftarrow \theta_i + \triangle rot$
8. If $step < 360$ go to step 4.
9. return LUT

4.4 Implementation

We implemented the proposed oriented mean-shift tracker as an extension to the original mean-shift tracker. The user supplies the initial location of the object to track, along with its bounding-box and an initial orientation. Images are first smoothed with a gaussian filter to reduce the noise (typically a 3×3 gaussian mask). Note that the smoothing is only applied in the neighborhood of the object. Orientations's look-up table are generated using the *Gradient-Rotating* method with a 2° step. The orientation estimation can either be nested within the original mean-shift loop or performed separately after the estimation of the translational part. The complete algorithm is outlined below:

Given: The original sequence, the initial object's position (y_0) and orientation (θ_0).

1. Compute the LUT of histograms orientations at y_0 (see section 3).
2. Initialize the mean-shift algorithm.
3. For each frame f_i
   (a) Update the object's position using the original mean-shift.
   (b) compute the gradient and estimate H the orientation histogram using (4.2).
   (c) $h_{max} \leftarrow \ Max \ \ [H \cap h_i];\ h_i \in LUT$.
   (d) update the object's orientation by the orientation that corresponds to h_{max}.

Notice that the orientations are estimated relatively to the initial orientation θ_0.

Even though histogram intersection is a fast operation, processing time can still be saved at step 4.c by limiting the search in a specific range instead of the entire LUT. Typical range is $\pm 20^\circ$ from the object's previous orientation.

4.5 Experimental Results

We tested the proposed oriented mean-shift algorithm on several motion sequences. Since we propose an orientation upgrade to the original mean-shift tracker, we mostly considered sequences with dominating rotational motion. We first ran our tracker on a synthetic sequence that was generated by fully rotating a real image (a chocolate box). The figure fig.4.2 shows some frames from the synthetic sequence.

The error of rotation estimation using different bin size is reported in figure fig.**??**. The green curve represents the error with a LUT generated by the *image-rotation* method whereas the red curve is the error using a LUT generated by the *gradient-rotation* method. In both cases the computed optimal bin size was 14. As we can see, the *gradient-rotation* method gives better results and is less sensitive to the bin

size variation. For the rest of the experiments, gradient's LUT were generated using the *gradient-rotation* method.

We further tested our method for face tracking. As the face underwent an almost perfect roll, we computed the orientation estimations at each frame. We observe that the face is tracked accurately, although no exact ground truth is available in this case. The results are shown in figures fig.4.4 and fig.4.5.

Aerial surveillance is another field where the tracking is useful. Due to the rectangular shape of common vehicles, an oriented tracking is suitable as shown in figure fig.4.6. However, notice that the orientation is not truly 2D, as the view angle induces some perspective distortions that is not handled in our method.

Finally, we illustrate the effectiveness of the proposed method for video rectification. A hand-held camera was rotated by hand around its optical axis while gazing at a static scene (see figure fig.4.7, left column). We tracked a rigid object attached to the scene and used the recovered motion to rectify and cancel the rotation in the video sequence. The results of tracking/rectification are shown in the figure fig.4.8 and the estimated orientations are plotted in the figure fig.4.7. The rotation is well recovered, as can be seen in the estimated curve of figure fig.4.7 and the rectified images of figure fig.4.8. Notice that the rectified images are sometimes distorted by parallax effects that are not modelled by our algorithm.

Figure 4.2. Some frames from the manually rotated sequence (with a fixed background).

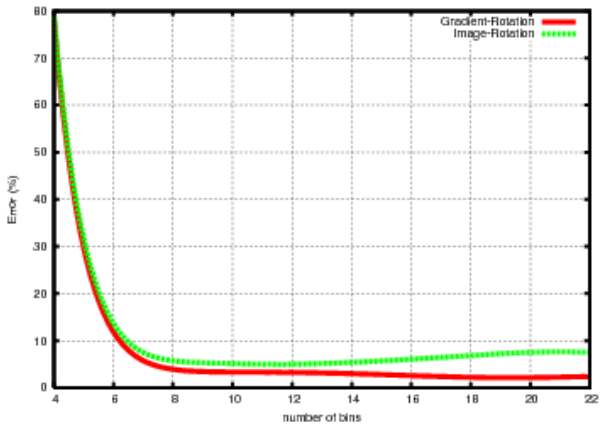

Figure 4.3. Errors in orientation estimation as a function of histogram samples.

Figure 4.4. Results of tracking a rotating face. Sample frames: 78, 164 and 257

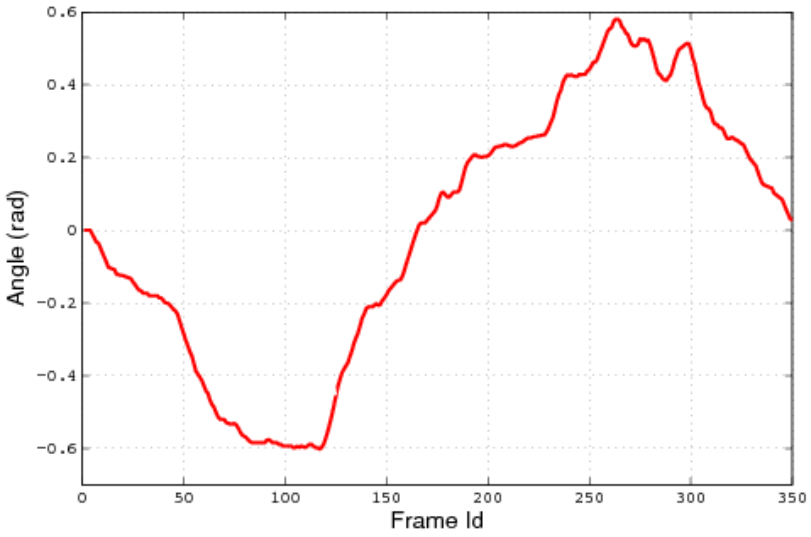

Figure 4.5. Estimated orientation for the rotating face sequence.

Figure 4.6. Tracking results for the car pursuit sequence.

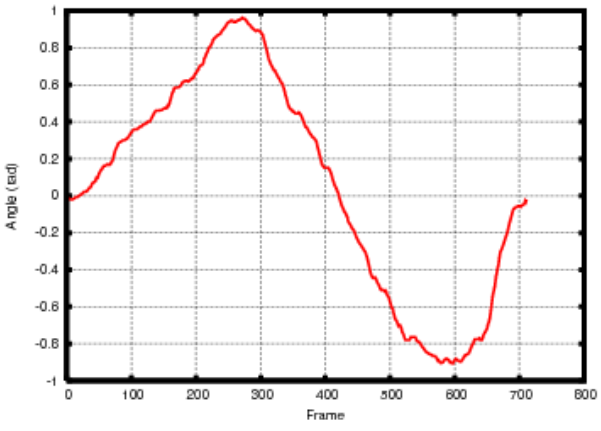

Figure 4.7. Estimated orientation for the shelf sequence.

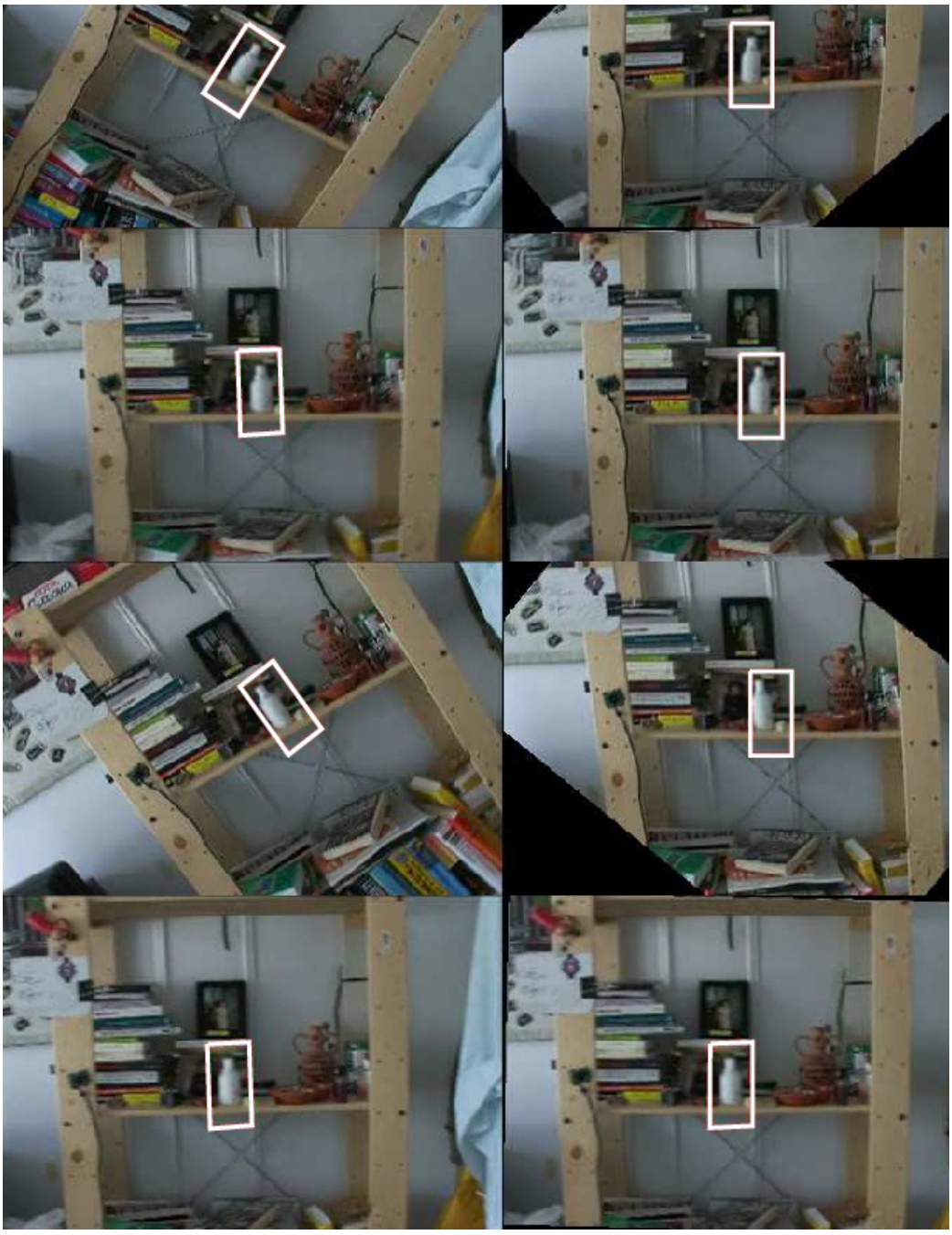

Figure 4.8. Results of tracking and rectifying images from a rolling camera sequence. $left$) results of the original tracking. $right$) rectified sequence after rotation cancellation. Shown frames are 0, 69,203,421,536 and 711.

4.6 Conclusion

We have presented a fast and simple extension to the original mean-shift tracker, to allow the estimation of the orientation. This rotation parameter is crucial when the tracked objects have a "thin" shape. We introduced the idea of the gradient-orientation space represented by the gradient look-up tables. Of course, the LUT can be extended to other cues related to the texture or pixels positions. This representation proved to be efficient as our experiments depicted. The proposed method ran comfortably on a regular PC in real time. Tracking was performed at 10-25 frames per second for typical 2000 pixels objects. In our implementation, the orientation was estimated independently from the translation shift. However, performing a combined mean-shift on histograms intensities and gradient LUT is possible. In the future, we plan on adding support for perspective deformation to better handle different type of rotations.

Chapitre 5

INTRODUCTION AU CALIBRAGE

En vision par ordinateur, la caméra est l'outil de choix pour "observer" le monde extérieur. En soi, c'est un dispositif qui fait passer des entités géométriques d'un monde tridimensionnel à une représentation planaire 2D. Ce processus optique est modélisé mathématiquement par une projection qui décrit la relation entre les coordonnées des points 3D de la scène et de leurs projections 2D dans l'image. Ce chapitre fournit les éléments de base qui permettront de comprendre le processus de formation de l'image ainsi que le calibrage de la caméra. Essentiellement, calibrer une caméra c'est d'abord choisir un modèle de projection et déterminer ensuite les paramètres de ce modèle à partir d'images. La connaissance de ces paramètres est cruciale dès lors qu'on souhaite inférer une information 3D ou des mesures métriques à partir d'une collection d'images 2D.

5.1 Formation géométrique de l'image

Pour la suite de ce chapitre, nous nous intéressons uniquement à la projection perspective et ce à travers un modèle très courant en vision par ordinateur : le modèle **sténopé**.

Comme l'illustre la figure 5.1, réduit à sa plus simple expression, le sténopé forme une image sur un **plan-image** Π situé devant un **centre optique**, **C**. La projection orthogonale de **C** sur le plan-image est le **point principal**. Normalement, le plan-image est placé derrière le centre optique et qui a pour effet de produire une image inversée. Ici nous maintenons son emplacement à l'avant pour des besoins illustratifs. Concrètement, le plan image et le centre optique se matérialisent dans nos caméras

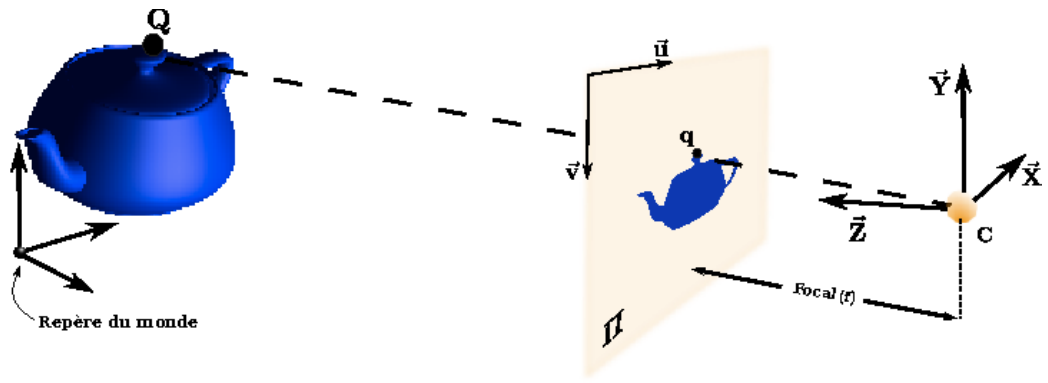

Figure 5.1. Caméra sténopé. La projection d'un point 3D $\mathbf{Q}$ se trouve à l'intersection de la droite $< \mathbf{C}, \mathbf{Q} >$ avec le plan image $\mathbf{\Pi}$

modernes sous la forme de capteurs numériques (CCD ou CMOS) et d'objectifs.

Dans un modèle sténopé, le point 3D $\mathbf{Q}$ (noté en coordonnées homogènes) se projette dans le plan image en $\mathbf{q}$ qui est défini comme l'intersection de la droite reliant $\mathbf{C}$ et $\mathbf{Q}$ avec le plan image. Ce concept est représenté par une transformation projective P, appelée matrice de projection :

$$\mathbf{q} \sim \mathsf{P}_{3\times4}\mathbf{Q}$$

La transformation projective $\mathsf{P}_{3\times4}$ "encapsule" en une seule matrice plusieurs transformations intermédiaires dont la décomposition en révèle la structure et nous aide à mieux comprendre la relation 3D-2D des points. Chacune de ces transformations implique un passage d'un espace à un autre et chacun d'entre eux sera associé à un repère.

Nous allons décrire ces transformations successives à partir d'un point 3D, exprimé dans un référentiel du monde.

- **Du repère monde au repère caméra :** Un point 3D $\mathbf{Q}$ est exprimé dans un repère global dit **repère du monde**. Une transformation rigide sous forme

d'une rotation R autour des 3 axes du référentiel et d'une translation T ramène **Q** dans le repère de la caméra positionné en **C** et orienté selon $(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{Y}}, \overrightarrow{\mathbf{Z}})$. Les paramètres qui définissent cette transformation sont appelés **paramètres extrinsèques**. Ils représentent essentiellement la position et l'orientation de la caméra dans le monde et s'expriment sous forme matricielle comme suit :

$$\mathsf{RT} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- **Projection sur le plan-image :** Ici il s'agit de projeter le point 3D sur le plan-image **Π** orienté par, $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}})$. C'est une transformation projective perspective. Le résultat sera un point en 2 dimensions, $(x, y, 1)$, exprimé dans le repère du plan-image :

$$x = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z}$$

Ou encore, sous forme matricielle :

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX/Z \\ fY/Z \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{Q}$$

Nous rappelons que le symbole $\sim$ désigne l'égalité à un facteur d'échelle.

- **Du repère image au repère pixel :** Il s'agit d'appliquer une transformation affine 2D aux pixels de l'image. Décrite par les paramètres intrinsèques de la caméra, cette transformation passe le point $(x, y, 1)$ du repère image au repère pixel. Concrètement, le centre est déplacé à un coin de l'image (souvent supérieur gauche) et un facteur d'échelle est appliqué. En supposant les axes de l'image orthogonaux, cette troisième et dernière transformation s'écrit :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Ici, (u_0, v_0) désignent les coordonnées du point principal dans le repère pixel et (k_u, k_v) la taille des pixels. Avec la focale f, ils constituent les **paramètres intrinsèques** de la caméra.

Nous avons expliqué le processus de projection d'un point 3D dans l'image de la caméra en employant trois sortes de transformations (rigide, projective et affine). Les caractéristiques de ces transformations sont comme suit :

- **Transformation rigide**
  - Implique uniquement les translations et les rotations.
  - Les propriétés géométriques des entités (points, objets,...) ne sont pas modifiées.
- **Transformation affine**
  - En plus des rotations et des translations, les mises à l'échelle sont impliquées.
  - Préserve le parallélisme.
  - Ne préserve pas les longueurs ni les angles.
  - Elle est réversible.
- **Transformation projective**
  - Préserve les droites, les coniques et les intersections.
  - Le birapport est invariant à la projection.

– Cette transformation est n'est pas toujours inversible.

À présent, il nous est possible d'exprimer la projection $\mathsf{P}_{3\times4}$ sous forme de produit matriciel :

$$\mathsf{P}_{3\times4} = \begin{pmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \end{pmatrix}$$

Ou encore de façon plus compacte :

$$\mathsf{P}_{3\times4} = \underbrace{\begin{pmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Paramètres Intrinsèques}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{11}t_x + r_{12}t_y + r_{13}t_z \\ r_{21} & r_{22} & r_{23} & r_{21}t_x + r_{22}t_y + r_{23}t_z \\ r_{31} & r_{32} & r_{33} & r_{31}t_x + r_{32}t_y + r_{33}t_z \end{pmatrix}}_{\text{Paramètres Extrinsèques}} \tag{5.1}$$

5.2 Calibrage de Caméra

Nous avons vu précédemment que la transformation projective $\mathsf{P}_{3\times4}$ était composée de paramètres extrinsèques et intrinsèques. Les premières définissent l'orientation de la caméra et les secondes ses propriétés physiques et dont l'estimation fait l'objet du calibrage de caméra. Presque tout a été fait et dit sur le calibrage de caméra sténopé et donc face à cette profusion de littérature, on propose de classifier les méthodes de calibrage par la dimension des entités de calibrage.

5.2.1 Objet 3D

En vision, les premiers travaux sur le calibrage de caméra ont été proposés par Tsai [72] et par Faugeras-Toscani [48]. Le calibrage est fait en calculant explicitement les coefficients de la matrice de projection P. Cette matrice de projection est estimée à partir d'un appariement de points d'une mire 3D[1] et de leur projection dans l'image

[1] Généralement deux plans disposés en angle droit.

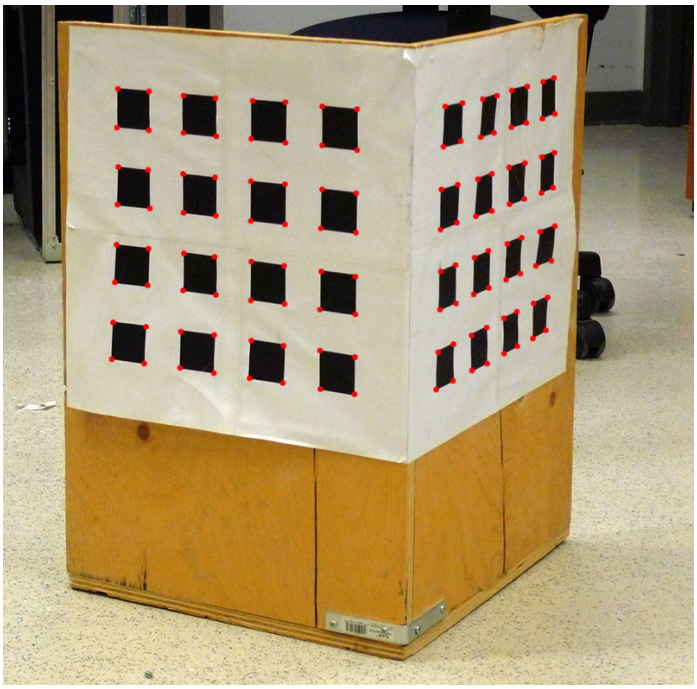

Figure 5.2. Mire composée de deux plans utilisée pour le calibrage 3D.

de la caméra [30]. Un exemple de mire 3D est illustré à la figure 5.2. La méthode proposée par Tsai [72] peut s'appliquer à des mire planaires à condition de connaitre le mouvement de la caméra, ce qui revient à connaitre les coordonnées de points en 3D [82].

5.2.2 Mire plane

Le principal défaut des méthodes de calibrage par objet 3D se situe au niveau de la conception de la mire elle-même. Elle doit être montée avec soin et nécessite un minimum d'assemblage. Sturm [67] et Zhang [83] ont présenté indépendamment une méthode pour calibrer une caméra à partir d'une mire planaire (voir Fig.5.3). La popularité de cette méthode est due à la disponibilité de son implémentation ainsi qu'à sa facilité d'usage. Effectivement, le calibrage ne nécessite qu'une mire plane marquée (tel un échiquier) qui peut être affichée sur un écran plat d'ordinateur.

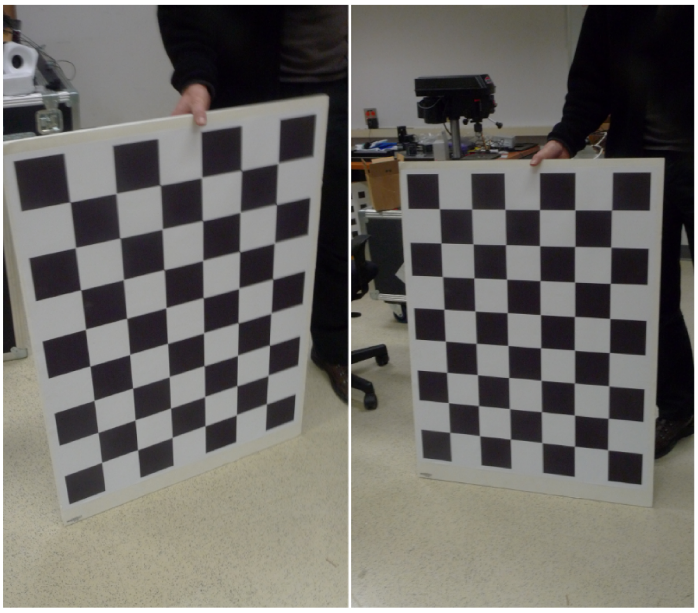

Figure 5.3. Un damier imprimé utilisé comme mire de calibrage 2D. Illustration de deux poses différentes.

Étant donné que nous avons le choix du repère du monde, nous allons le faire correspondre avec celui de la mire. Vu que tous les points de la mire sont coplanaires, ils ont la même coordonnées Z, en l'occurrence 0. Soit un point $\mathbf{Q}_i = (a_i, b_i, 0, 1)$ de la mire exprimé dans le repère du monde qu'on a établi plus haut et $\mathbf{q}_i = (u_i, v_i, 1)$ sa projection dans l'image de la caméra. Soient K, R et $\mathbf{t}$ les matrices de paramètres internes, la matrice de rotation et le vecteur translation, $\mathbf{Q}_i$ et $\mathbf{q}_i$ sont reliés par :

$$\mathbf{q}_i \sim \mathsf{K}\mathsf{R}\left[\bar{\mathsf{I}}_{4\times 3}|\mathbf{t}\right]\mathbf{Q}_i$$

Étant donné que la 3$^{\text{e}}$ coordonnée des points $\mathbf{Q}_i$ est nulle, l'équation précédente se réécrit en fonction des deux premières colonnes de R comme :

$$\mathbf{q}_i \sim \underbrace{\mathsf{K}\left[\mathbf{r}_1\mathbf{r}_2\mathbf{t}\right]}_{\mathsf{H}}\bar{\mathbf{Q}}_i$$

où, $\bar{\mathbf{Q}}_i = (a_i, b_i, 1)$.

La matrice H est une homographie qui fait correspondre les points de la mire à leur projection dans l'image de la caméra et peut être estimée linéairement à partir de 4 correspondances mire-caméra (Voir Annexe A). En notant la i$^{\text{e}}$ colonne de H par $\mathbf{h}_i$, on remarque que $[\mathbf{h}_1\mathbf{h}_2] \sim \mathsf{K}\,[\mathbf{r}_1\mathbf{r}_2]$. Ce qui est équivalent à $\mathsf{K}^{-1}\,[\mathbf{h}_1\mathbf{h}_2] \sim [\mathbf{r}_1\mathbf{r}_2]$. Ici, en remarquant que $\mathbf{r}_1$ et $\mathbf{r}_2$ sont orthonormaux, deux contraintes peuvent être imposées sur ω en fonction des éléments de l'homographie:

$$\mathbf{h}_1^{\mathsf{T}}\omega\mathbf{h}_2 = 0, \quad \mathbf{h}_1^{\mathsf{T}}\omega\mathbf{h}_1 = \mathbf{h}_2^{\mathsf{T}}\omega\mathbf{h}_2$$

Avec au moins trois homographies il est possible d'estimer ω. Par la suite, ω peut etre factorisée à l'aide de la factorisation de Cholesky afin d'extraire la matrice de paramètres intrinsèques K.

La matrice ω a une signification géométrique très intéressante et qui s'avérera utile pour l'auto-calibrage. Pour l'illustrer, notons que toutes les sphères intersectent le plan infini en un ensemble de points [30], $\mathbf{P}_i$, dont la forme en coordonnées homogènes est :

$$\mathbf{P}_i \sim (X_i, Y_i, Z_i, 0)$$

Ces points vérifient aussi l'équation :

$$\mathbf{P}_i{}^{\mathsf{T}}\mathbf{P}_i = X_i^2 + Y_i^2 + Z_i^2 = 0$$

Ces points peuvent être vus comme appartenant à une conique spéciale, $\Omega_\infty \sim diag(1, 1, 1, 0)$, car ils en vérifient l'équation (voir chapitre 2) :

$$\mathbf{P}_i{}^{\mathsf{T}}\mathbf{P}_i = \mathbf{P}_i{}^{\mathsf{T}}\Omega_\infty\mathbf{P}_i = 0$$

L'image de la conique absolue dans une caméra est exactement la matrice ω qui est définie comme suit [30] :

$$\omega \sim \left(\mathsf{K}\mathsf{K}^{\mathsf{T}}\right)^{-1}$$

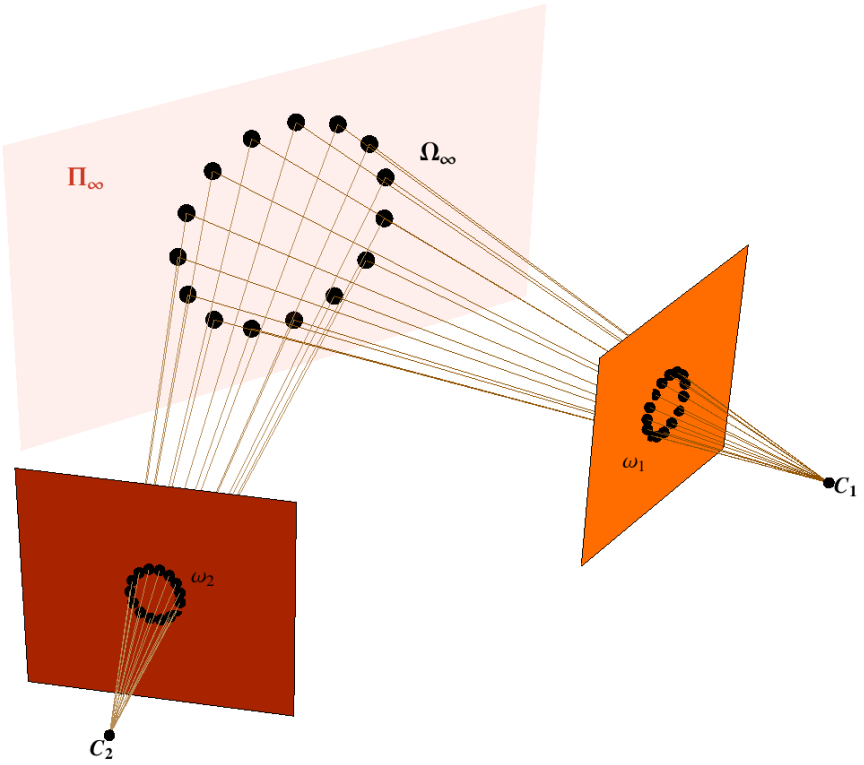

Figure 5.4. La conique absolue, Ω_∞, se projette dans l'image de la caméra C_i en ω_i. Cette projection ne dépend que des paramètres intrinsèques de la caméra.

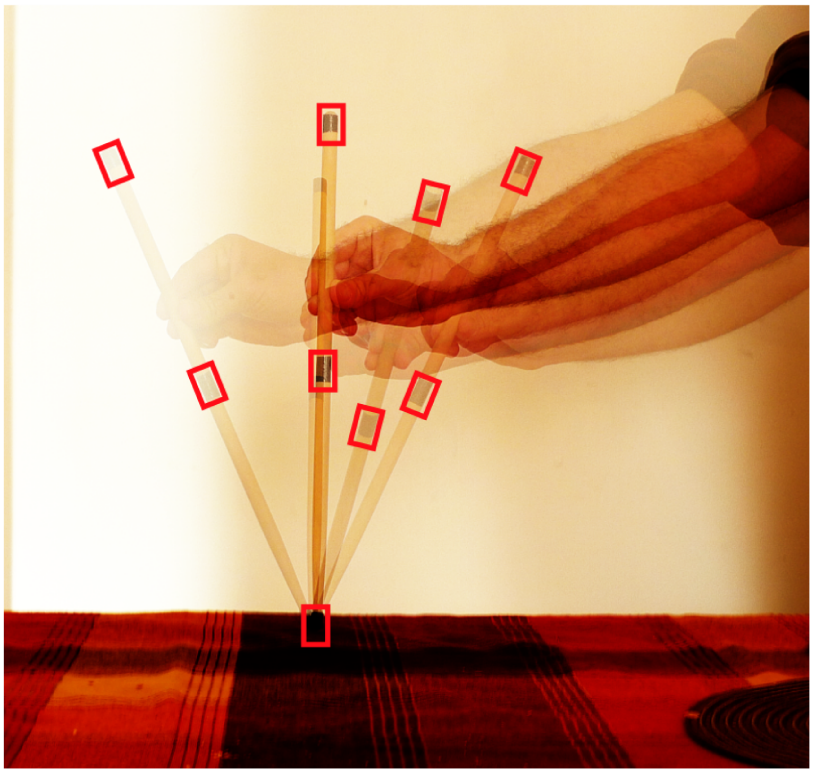

Figure 5.5. Superposition d'images de bâton utilisé pour le calibrage 1D. La pointe inférieure est maintenue fixe.

Une propriété intéressante de la conique absolue est que son image dans une caméra n'est contrainte que par les paramètres intrinsèques de la caméra (voir figure 5.4).

5.2.3 Objet linéaire (1D)

Le calibrage de caméra à l'aide d'un objet 1D (un bâton) a été proposé en premier par Zhang dans [82]. L'auteur démontre que le calibrage est possible à l'aide de 3 points colinéaires si la position des points est connue sur la ligne. De plus, l'un des points doit demeurer fixe, ce qui revient à acquérir les images de 3 points sur un bâton qui pivote autour d'un point (Voir Fig.5.5).

Soient **A** et **B** deux points sur le bâton de calibrage et dont la distance L est

connue :

$$||\mathbf{B} - \mathbf{A}|| = L \tag{5.2}$$

La position du troisième point $\mathbf{C}$ peut s'exprimer en fonction de la position de $\mathbf{A}$ et de $\mathbf{B}$:

$$\mathbf{C} = \lambda_A \mathbf{A} + \lambda_B \mathbf{B} \tag{5.3}$$

Les constantes $\lambda_{A,B}$ sont connues du fait que la position des points $\mathbf{A}$ et $\mathbf{B}$ est connue [2].

Sans perte de généralité, on utilise le repère de la caméra pour définir les points $\mathbf{A}, \mathbf{B}$ et $\mathbf{C}$. En notant leur profondeur $z_{A,B,C}$ et leur projection dans l'image (en coordonnées homogènes) $\mathbf{a}, \mathbf{b}$ et $\mathbf{c}$, on a que :

$$\begin{aligned} \mathbf{A} &= z_A \mathsf{K}^{-1}\mathbf{a} \\ \mathbf{B} &= z_B \mathsf{K}^{-1}\mathbf{b} \\ \mathbf{C} &= z_C \mathsf{K}^{-1}\mathbf{c} \end{aligned} \tag{5.4}$$

En substituant l'équation (5.4) dans (5.3) et en éliminant K^{-1} de part et d'autre, on obtient :

$$z_c \mathbf{c} = z_a \mathbf{a} + z_b \mathbf{b} \tag{5.5}$$

L'application aux deux côtés de l'équation précédente du produit vectoriel avec $\mathbf{c}$ permet d'isoler z_B :

$$z_B = -z_A \frac{\lambda_A (\mathbf{a} \times \mathbf{c}).(\mathbf{b} \times \mathbf{c})}{\lambda_B (\mathbf{b} \times \mathbf{c})(\mathbf{b} \times \mathbf{c})} \tag{5.6}$$

À partir de l'équation (5.2), on déduit que $L = ||\mathsf{K}^{-1}(z_B \mathbf{b} - z_A \mathbf{a})||$ qui nous permet de réécrire l'équation (5.6) comme :

[2] Remarquez que si $\mathbf{C}$ coupe la ligne $\mathbf{AB}$ en deux alors $\lambda_A = \lambda_B = 0.5$.

$$z_A \left|\left|\mathsf{K}^{-1}\mathbf{h}\right|\right| = L \tag{5.7}$$

ce qui est équivalent à :

$$z_A^2 \mathbf{h}^\mathsf{T} \underbrace{\mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}}_{\mathsf{S}} \mathbf{h} = L^2 \tag{5.8}$$

avec :

$$\mathbf{h} = \mathbf{a} + \frac{\lambda_A(\mathbf{a}\times\mathbf{c}).(\mathbf{b}\times\mathbf{c})}{\lambda_B(\mathbf{b}\times\mathbf{c})(\mathbf{b}\times\mathbf{c})}\mathbf{b}$$

Comme dans le cas du calibrage planaire, avec un nombre suffisant de poses (6 dans ce cas-ci), on résout S qui permet de calculer la matrice des paramètres intrinsèques K par une décomposition de Cholesky.

5.2.4 Auto-Calibrage

Les méthodes décrites précédemment ont en commun deux aspects. Elles sont supervisées et nécessitent l'utilisation d'une mire (ou d'un étalon) dont la métrique est connue. Ceci implique, qu'avant toute session de travail un calibrage doit être fait au préalable. Cette approche est peu commode (voire même inapplicable) pour les systèmes de vision actifs ou modulaires (zoom, optique interchangeable,...).

L'auto-calibrage répond précisément à ce besoin. En soi, le terme désigne tout processus de calibrage qui ne requiert pas de mire ou de métrique connue au préalable, d'ailleurs par abus de langage l'auto-calibrage est désigné par calibrage 0D.

De façon générale, l'auto-calibrage exploite la rigidité de la scène [22], des mouvement de caméras spéciaux [32], ou des scènes structurées tels que des plans [70]. Nous aurons l'occasion de reparler d'auto-calibrage lorsque nous traiterons l'auto-calibrage de projecteurs vidéo au chapitre 9.

5.3 Caméra Linéaire

Un autre modèle de caméra très utilisé en vision est le modèle linéaire 1D (**Push Broom**). Ici le capteur matriciel des caméras conventionnelles est remplacé par un capteur sous forme de barrette 1D, ce qui peut facilement doubler la résolution et la vitesse d'acquisition des caméras matricielles. L'acquisition se fait par balayage de la scène ou de façon réciproque, par défilement de scène devant un capteur fixe. On comprend dès lors, l'utilité de ces caméras dans l'inspection industrielle (rapidité), imagerie satellite (stockage réduit) ou encore les numériseurs de documents (coût très faible).

L'image 2D est formée en "empilant" les images linéaires les unes par dessus les autres, la géométrie de la projection ne peut qu'être hétérogène. Le long de la barrette, la projection demeure tributaire du choix de l'optique, dans l'autre sens elle dépend du mouvement du capteur. Une modélisation générale des caméras linéaires est présentée dans [33].

Dans ce qui suit, nous n'utiliserons que le modèle sténopé pour la projection le long de la barrette et nous supposerons que le mouvement de la caméra linéaire est constant, ce qui revient à adopter un modèle **orthographique à l'échelle**.

5.3.1 Calibrage de Caméra linéaire

Avec les hypothèses ci-haut mentionnées, le calibrage de caméra linéaire consiste à estimer la longueur focale (f), la position du point principal (u_0) et le facteur d'échelle (s), rattaché au mouvement du capteur [24].

Une étude moderne comprenant le calibrage et la géométrie qui relie deux caméras linéaires (géométrie épipolaire) a été présentée par Hartley et Gupta [52]. En montrant que dans un cas général, une ligne 3D se projette dans une caméra linéaire sous la forme d'une hyperbole, les auteurs élaborent la géométrie épipolaire pour ce modèle ainsi qu'une méthode de calibrage basée sur la factorisation de la matrice de

projection. Cette dernière est calculée à partir d'une association de points 3D (connus avec précision) et de leur projection 2D dans l'image. Cette méthode partage les qualités et défauts de son équivalent pour les caméras matricielles. L'implémentation est facile mais nécessite une mire 3D qui peut s'avérer complexe à confectionner.

Afin de pallier à ce problème, nous proposons une méthode pour calibrer une caméra linéaire à l'aide de mire 2D. Cette contribution fait l'objet du prochain chapitre.

Chapitre 6

PLANE-BASED CALIBRATION FOR LINEAR CAMERAS

Cet article [17] a été publié comme l'indique la référence bibliographique

J. Draréni, P.F. Sturm, et S. Roy. Plane-based calibration for linear cameras. Dans *OMNIVIS'2008, the Eighth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, in conjunction with ECCV 2008, Marseille, France*.

Cet article, a été accepté aussi pour publication dans le journal scientifique *International Journal of Computer Vision* au mois de Mai 2010. Il est présenté ici dans sa version originale.

Abstract

Linear or 1D cameras are used in several areas such as industrial inspection and satellite imagery. Since 1D cameras consist of a linear sensor, a motion (usually perpendicular to the sensor orientation) is performed in order to acquire a full image. In this paper, we present a novel linear method to estimate the intrinsic and extrinsic parameters of a 1D camera using a planar object. As opposed to traditional calibration scheme based on 3D-2D correspondences of landmarks, our method uses homographies induced by the images of a planar object. The proposed algorithm is linear, simple and produces good results as shown by our experiments.

6.1 Introduction

Pushbroom cameras or linear scanners are a one-dimensional imaging devices. They are preferred over conventional 2D cameras when it comes to *scan* a static scene like airborne landscapes and urban scapes reconstruction [27]. This choice is motivated by the need for a higher frame rate and a better resolution. At the time of writing, existing pushbroom cameras embed sensors up to 8192 pixels and delivers 1D images at a stunning frame-rate of 140Khz [4].

If the acquired images are meant for a 3D euclidean reconstruction or metrology purposes [28] [63], a camera calibration is necessary. As detailed in section 6.2 linear cameras have a specific model thus, standard 2D camera calibration methods can no longer be used to recover internal parameters.

Classical calibration methods use mappings of 3D feature points on a calibration rig and their projections on the image to infer the internal parameters of a camera [72, 48]. These methods are not very flexible because they use a specially designed calibration rig and often, features are manually selected.

In the last decade, new plane-based calibration methods have been introduced [67, 83]. They enjoyed a growing popularity in the computer vision community due to their stability and their higher ease of use. In fact, the calibration can be done with an off-the-shelf planar object and a printed checkerboard.

Despite the several improvements that plane-based calibration methods went through [25, 75] [54] , none of these works tackled the calibration of linear cameras. In fact, the predominant method for 1D camera calibration was proposed by Hartley et al. [52, 24] and supposes a mapping between 3D landmarks and their projections in the image.

In this paper, we present a novel method to fully calibrate a pushbroom camera using a planar object. Here, the considered camera model is the translational pushbroom camera. Our method is linear, fast and simple to implement. To the best of

our knowledge, the presented plane-based calibration is the first of its kind.

For the rest of the paper, the terms 1d camera, linear camera and pushbroom camera will be used equally.

The remaining of the paper is organized as follows, in section 6.2, the linear camera model is described. Section 6.3, presents the mathematical derivation and the algorithm of the plane-based calibration for linear cameras . The experiments and results are reported in section 6.6 and we finally summarize our conclusion in section 6.7.

6.2 Camera Model

Although the motivation behind our work was to calibrate a flatbed scanner using a pushbroom model, the presented algorithm along with the mathematical derivations still hold for any linear camera provided that the sensor undergoes a motion orthogonal to its orientation.

In general, a 1D camera consists of a linear array of sensors (such as CCD) recording an image projected by an optical system. A displacement of the sensor (usually orthogonal to the sensor) is required. We make the same reasonable assumption as in [52] regarding the sensor motion. We assume its velocity constant.

We set up the local camera coordinate system as depicted in the figure 6.1. Let the point $(u, v, 1)^\mathsf{T}$ be the projection of the 3D point $(X, Y, Z)^\mathsf{T}$ in the camera image plane. The perspective projection of the coordinate u along the sensor can me modelled with a 2×3 projection matrix P:

$$\begin{pmatrix} u \\ 1 \end{pmatrix} \sim \underbrace{\begin{pmatrix} f & u_0 & 0 \\ 0 & 1 & 0 \end{pmatrix}}_{\mathsf{P}} \begin{pmatrix} X \\ Z \\ 1 \end{pmatrix} \tag{6.1}$$

The parameters f and u_0 are respectively the focal length and the optical center of the linear sensor.

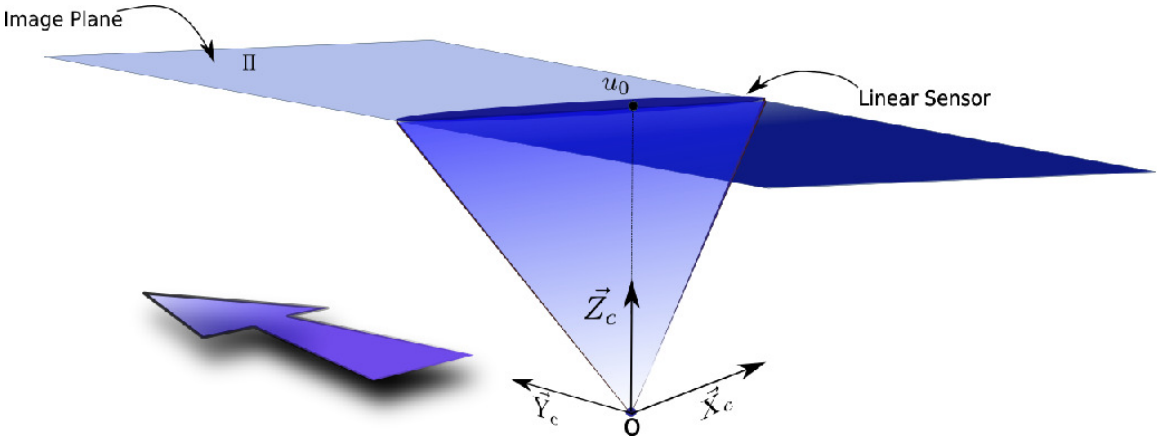

Figure 6.1. A typical linear camera. A sensor, linear along the X axis, undergoes motion along the Y axis.

As the sensor sweeps the scene, a 2D image is formed by stacking the 1D images obtained through the successive camera positions. Since the speed of the camera is assumed constant, the v coordinates is related to Y by a scaling factor s that depends on the speed of the sensor:

$$v = sY \tag{6.2}$$

If we combine (6.1) and (6.2) in a single matrix, the complete projection of a 3D point $(X, Y, Z)^{\mathsf{T}}$ is expressed as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fX + u_0 Z \\ sYZ \\ Z \end{pmatrix} \sim \underbrace{\begin{pmatrix} f & 0 & u_0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathsf{K}} \begin{pmatrix} X \\ YZ \\ Z \end{pmatrix} \tag{6.3}$$

where K represents the sought intrensic camera matrix. We can see from the above equation that the perspective coordinate u depends solely on X and its depth Z, whereas v the orthographic coordinate is directly related to Y and the scaling factor s. One can also observe the non-linearity of the projection equation in the

3D coordinates due to the YZ term. This is not surprising, since the projection is non-central. This precludes the use of a pinhole-based camera calibration.

6.3 Calibration With a Planar Grid

Let us consider a point $(a, b, 0)^\mathsf{T}$ on the grid. It is mapped into the camera's coordinate system as $(X, Y, Z)^\mathsf{T}$ by a rigid transform:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathsf{R} \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} + \mathbf{t} \tag{6.4}$$

where R is a 3×3 rotation matrix and $\mathbf{t}$ a translation vector. Notice that, since the considered point lies on the grid, its third coordinate is null. Hence, the entries of the third column of R are zeroed and the Eq.6.4 in homogeneous coordinates simplifies as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \mathbf{t} \end{pmatrix} \begin{pmatrix} a \\ b \\ 1 \end{pmatrix} = \begin{pmatrix} ar_{11} + br_{12} + t_1 \\ ar_{21} + br_{22} + t_2 \\ ar_{31} + br_{32} + t_3 \end{pmatrix} \tag{6.5}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are the first two columns of R.

As stated before, the non-central nature of the camera makes it impossible to establish a linear mapping between points on the grid and their images on the camera plane. For instance, $(u, v, 1)^\mathsf{T}$ is expressed from Eq.6.3 and Eq.6.5 as:

$$\begin{aligned} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} &\sim \mathsf{K} \begin{pmatrix} X \\ YZ \\ Z \end{pmatrix} \\ &= \mathsf{K} \begin{pmatrix} ar_{11} + br_{12} + t_1 \\ a(r_{21}t_3 + r_{31}t_2) + b(r_{22}t_3 + r_{32}t_2) + t_2t_3 + a^2r_{21}r_{31} + b^2r_{22}r_{32} + ab(r_{21}r_{32} + r_{22}r_{31}) \\ ar_{31} + br_{32} + t_3 \end{pmatrix} \end{aligned} \tag{6.6}$$

An approach to circumvent this problem is to express the points in a higher dimensional space via the so-called "lifted" coordinates. In our case, the point $(a, b, 1)^{\mathsf{T}}$ "lifts" (according to their Veronese mapping) to $(a, b, 1, a^2, b^2, ab)^{\mathsf{T}}$. Thus, the Eq.6.7 becomes:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathsf{K} \cdot \underbrace{\begin{pmatrix} r_{11} & r_{12} & t_1 & 0 & 0 & 0 \\ r_{21}t_3 + r_{31}t_2 & r_{22}t_3 + r_{32}t_2 & t_2t_3 & r_{21}r_{31} & r_{22}r_{32} & r_{21}r_{32} + r_{22}r_{31} \\ r_{31} & r_{32} & t_3 & 0 & 0 & 0 \end{pmatrix}}_{\mathsf{T}} \begin{pmatrix} a \\ b \\ 1 \\ a^2 \\ b^2 \\ ab \end{pmatrix} \tag{6.7}$$

which represents the complete projection equation of a point on the grid expressed in its lifted coordinates.

The homography $\mathsf{H} \sim \mathsf{KT}$ that maps point on the grid and its image has 6 zeroed entries. The remaining 12 non-zero entries can be estimated up to a scale factor using 6 or more point matches as explained in the next subsection.

6.3.1 Estimate the Homography

We recall from Eq.6.7 that the mapping between grid points and image points is represented by the homography H as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathsf{H} \cdot \begin{pmatrix} a \\ b \\ 1 \\ a^2 \\ b^2 \\ ab \end{pmatrix} \tag{6.8}$$

If we multiply both hands of the above equation by $\left[\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}\right]_\times$, the cross product skew matrix, we get a homogeneous equation system that upon simplifications yields the following linear and homogeneous equation system in the entries of H:

$$\begin{pmatrix} 0 & 0 & 0 & a & b & 1 & a^2 & b^2 & ab & -av & -bv & -v \\ a & b & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -au & -bu & -u \\ -av & -bv & -v & au & bu & u & a^2u & b^2 & abu & 0 & 0 & 0 \end{pmatrix} \mathbf{h} = \mathbf{0} \qquad (6.9)$$

where $\mathbf{h}^\mathsf{T} = (\mathbf{h}_{11}, \mathbf{h}_{12}, \mathbf{h}_{13}, \mathbf{h}_{21}, \mathbf{h}_{22}, \mathbf{h}_{23}, \mathbf{h}_{24}, \mathbf{h}_{25}, \mathbf{h}_{26}, \mathbf{h}_{31}, \mathbf{h}_{32}, \mathbf{h}_{33})$ is the vector that contains the non-zero entries of H.

It is easy to see that only two equations are linearly independent. For instance, the third row can be obtained by adding the first and the second row, scaled respectively by u and $-v$. Thus, given at least 6 matches between grid points and their images, H can be solved using 2 equations from the system Eq.6.9 per match.

6.3.2 Extracting the Principal Point and the Focal Length

We shall now show how the camera's internal parameters are extracted from the homographies computed in the previous subsection. Let us recall the explicit form of the homography H:

$$\mathsf{H} = \lambda \begin{pmatrix} fr_{11} + u_0 r_{31} & fr_{12} + u_0 r_{32} & ft_1 + u_0 t_3 & 0 & 0 & 0 \\ s(r_{21}t_3 + r_{31}t_2) & s(r_{22}t_3 + r_{32}t_2) & st_2 t_3 & sr_{21}r_{31} & sr_{22}r_{32} & s(r_{21}r_{32} + r_{22}r_{31}) \\ r_{31} & r_{32} & t_3 & 0 & 0 & 0 \end{pmatrix}$$

The scalar λ is added because the homography H can only be retrieved up to a scale factor. One can notice that $\bar{\mathsf{R}}$, the two first rotation's columns can be expressed

as:

$$\bar{\mathsf{R}} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \\ r_{31} & r_{32} \end{pmatrix} = \begin{pmatrix} \frac{h_{11}-u_0 h_{31}}{\lambda f} & \frac{h_{12}-u_0 h_{32}}{\lambda f} \\ \frac{h_{24}}{s h_{31}} & \frac{h_{25}}{s h_{32}} \\ \frac{h_{31}}{\lambda} & \frac{h_{32}}{\lambda} \end{pmatrix}$$

From the above equation, $\bar{\mathsf{R}}$ can be expressed as a product of two matrices (up to a scale factor) L that depends on internal parameters and M:

$$\mathsf{L} = \begin{pmatrix} s & 0 & -su_0 \\ 0 & \lambda f & 0 \\ 0 & 0 & sf \end{pmatrix} \qquad \mathsf{M} = \begin{pmatrix} H_{11} & H_{12} \\ H_{24}/H_{31} & H_{25}/H_{32} \\ H_{31} & H_{32} \end{pmatrix}$$

The product of $\bar{\mathsf{R}}$ with its transpose is a 2×2 identity matrix due to the orthogonality of its columns. Thus, we have:

$$\bar{\mathsf{R}}^{\mathsf{T}} \bar{\mathsf{R}} = \mathtt{I}_{2\times 2} \sim \mathsf{M}^{\mathsf{T}} \mathsf{L}^{\mathsf{T}} \mathsf{L} \mathsf{M}$$

The matrix L is related to the above calibration matrix K, with the notable fact that it also includes the scalar λ. Note that λ will be different for each view, as opposed to the 3 intrinsic parameters f, s and u_0 which remain the same. Let us define the matrix X as:

$$\mathsf{X} = \mathsf{L}^{\mathsf{T}} \mathsf{L} = \begin{pmatrix} s^2 & 0 & -s^2 u_0 \\ 0 & \lambda^2 f^2 & 0 \\ -s^2 u_0 & 0 & s^2(u_0^2 + f^2) \end{pmatrix} = \begin{pmatrix} v_1 & 0 & v_2 \\ 0 & v_4 & 0 \\ v_2 & 0 & v_3 \end{pmatrix}$$

where the intermediate variables v_1, v_2, v_3, v_4 were introduced for ease of notation. The equation Eq.6.3.2 gives 2 constraints on X that can be written as:

$$\left.\begin{aligned} \left(\mathsf{M}^{\mathsf{T}} \mathsf{X} \mathsf{M}\right)_{12} &= 0 \\ \left(\mathsf{M}^{\mathsf{T}} \mathsf{X} \mathsf{M}\right)_{11} - \left(\mathsf{M}^{\mathsf{T}} \mathsf{X} \mathsf{M}\right)_{22} &= 0 \end{aligned}\right\} \tag{6.10}$$

Which in turn can be expressed in terms of the intermediate variables $v_{1,2,3,4}$ as:

$$\begin{pmatrix} m_{11}m_{12} & m_{11}m_{32}+m_{12}m_{31} & m_{31}m_{32} & m_{21}m_{22} \\ m_{11}^2-m_{12}^2 & 2(m_{11}m_{31}-m_{12}m_{32}) & m_{31}^2-m_{32}^2 & m_{21}^2-m_{22}^2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{6.11}$$

With at least two different views of a grid, the $v_{1,2,3,4}$ can be computed up to a scaling factor. Bare in mind that v_4 is different at each view because of the homography scaling factor λ. Once the $v_{1,2,3,4}$ computed, the principal point and the focal length are simply computed as:

$$u_0 = -\frac{v_2}{v_1} \tag{6.12}$$

$$f = \sqrt{\frac{v_3}{v_1}-u_0^2} = \sqrt{\frac{v_3}{v_1}-\frac{v_2^2}{v_1^2}} = \sqrt{\frac{v_1v_3-v_2^2}{v_1^2}} \tag{6.13}$$

6.3.3 Extracting the Scaling factor and the Extrinsic Parameters

Now that we have extracted the focal length and the principal point, we will show how the scaling factor s along with the extrinsic parameters (rotation matrix R and translation vector $\mathbf{t}$) can be computed using more constraints. Let us define a matrix A_i as:

$$\mathsf{A}_i = \lambda_i \begin{pmatrix} r_{11} & r_{12} & t_1 & 0 & 0 & 0 \\ s(r_{21}t_3+r_{31}t_2) & s(r_{22}t_3+r_{32}t_2) & st_2t_3 & sr_{21}r_{31} & sr_{22}r_{32} & s(r_{21}r_{32}+r_{22}r_{31}) \\ r_{31} & r_{32} & t_3 & 0 & 0 & 0 \end{pmatrix} \tag{6.14}$$

The subscript i refers to the i^{th} view of the calibration grid. We can first notice that:

$$\begin{aligned} t_{1i} &= a_{13}/\lambda_i \\ t_{2i} &= a_{23}/sa_{33} \\ t_{3i} &= a_{33}/\lambda_i \end{aligned} \tag{6.15}$$

It's easy to see that $\bar{\mathsf{R}}_i$ (the two first columns of R_i as defined in the previous subsection) can be expressed as:

$$\bar{\mathsf{R}}_i = \begin{pmatrix} \frac{1}{\lambda_i} & & \\ & \frac{1}{sa_{33i}} & \\ & & \frac{1}{\lambda_i} \end{pmatrix} \begin{pmatrix} a_{11i} & a_{12i} \\ a_{21i} - a_{31i}\frac{a_{23i}}{a_{33i}} & a_{22i} - a_{32i}\frac{a_{23i}}{a_{33i}} \\ a_{31i} & a_{32i} \end{pmatrix} \tag{6.16}$$

$$= \begin{pmatrix} \frac{1}{\lambda_i} & & \\ & \frac{1}{s} & \\ & & \frac{1}{\lambda_i} \end{pmatrix} B_i \tag{6.17}$$

where a_{xyi} are the elements of the matrix A_i and the matrix B_i defined as:

$$B_i = \begin{pmatrix} a_{11i} & a_{12i} \\ \frac{a_{21i}a_{33i} - a_{31i}a_{23i}}{a_{33i}^2} & \frac{a_{22i}a_{33i} - a_{32i}a_{23i}}{a_{33i}^2} \\ a_{31i} & a_{32i} \end{pmatrix}$$

As in the previous subsection, we once again make use of the orthogonality of the rotation matrix $\bar{\mathsf{R}}_i$ to gain constraints on λ_i and $\frac{1}{st_{3i}}$. For instance, one notices that:

$$\bar{\mathsf{R}}_i^{\mathsf{T}}\bar{\mathsf{R}}_i = \mathsf{B}_i{}^{\mathsf{T}} \begin{pmatrix} \frac{1}{\lambda_i^2} & & \\ & \frac{1}{s^2} & \\ & & \frac{1}{\lambda_i^2} \end{pmatrix} \mathsf{B}_i = \mathtt{I}_{2\times 2}$$

The above result gives 3 linear equations in $\frac{1}{\lambda_i^2}$ and $\frac{1}{s^2}$. Since we solved for $\frac{1}{\lambda_i^2}$, the scaling factor λ_i is extracted up to a sign.

So far, only the 3 first columns of A_i have been used. In order to extract the real λ_i from the 2 possible solutions, the last 3 columns of A_i will be used. We proceed with the following simple steps for each possible solution:

- Compute $\bar{\mathsf{R}}_i$ and A_i from eq.6.16.
- Compute t_{1i} and t_{3i} as defined in eq.6.15.

- Compute the residual term:

$$\Delta = (a_{24i} - sr_{21i}r_{31i})^2 + (a_{25i} - sr_{22i}r_{32i})^2 + (a_{26i} - s(r_{21i}r_{32i} + r_{22i}r_{31i}))^2$$

The ideal solution is the one that leads to the smallest Δ. By the definition of a_{24i}, $a25i$ and a_{26i} (see Eq.6.14) and in an ideal noiseless case, Δ vanishes. Notice that if a couple (s, λ_i) minimizes Δ, then $(-s, -\lambda_i)$ also minimizes Δ. This ambiguity corresponds to the mirror-pose solution. Given our choice of coordinate system, visible points must have positive Z-coordinate, thus we pick the solution that gives a positive t_{3i}.

Finally, t_{2i} is computed from eq.6.15 and the third column of the rotation matrix is obtained by a simple cross-product of the two columns of $\bar{\mathsf{R}}_i$. The orthonormality of the final rotation matrix R_i can be enforced using SVD.

Notice that, as opposed to the reference calibration method [52], the proposed method estimates the scaling factor s related to the speed of the linear sensor.

6.3.4 Non-Linear Optimization

In this subsection, we give the details of a non-linear optimization procedure through bundle adjustment for our calibration method. Though optional, such optimization is highly recommended and as shown later, is fast and reduces the reprojection error.

Once an initial estimation of the internal parameters has been carried out (using the linear method described earlier), an optimization procedure can be applied in order to minimize the reprojection error in the camera and represented by following cost function:

$$\min_{\mathsf{K},\bar{\mathsf{R}}_i,\mathbf{t}_i} \sum_{i,j} dist^2 \left((u_{ij}, v_{ij}, 1)^{\mathsf{T}}, \mathsf{K}\bar{\mathsf{R}}_i\mathbf{t}_i(a_j, b_j, 1)^{\mathsf{T}} \right) \tag{6.18}$$

where $(a_j, b_j, 1)^{\mathsf{T}}$ represents the j^{th} feature on the calibration plane and $(u_{ij}, v_{ij}, 1)^{\mathsf{T}}$ its projection in the i^{th} camera.

For each camera pose, we must optimize the 3 intrinsic parameters (supposed fixed) and the 6 extrinsic parameters (different at each pose). Thus, for n camera poses, we have $3 + 6n$ parameters to optimize. In our implementation, we used the Levenberg-Marquardt method for the optimization. Usual implementations take advantage of the sparsity of the problem to gain time on matrix operations such as inversions. However, given the small size of our problem, we used standard SVD routines to inverse matrices. Indeed, for a typical calibration process using 10 poses, solving for the normal equation involves inverting matrices of 63×63.

We give the formulation of the error function derivatives and the form of the jacobian matrix in the next section.

Before, we should mention that, the whole bundle adjustement process runs in less than 2 seconds on a 2ghz PC.

6.4 Bundle Adjustment

In this section, we give the details of our bundle adjustment implementation. The emphasis is given to the partial derivatives formulation to estimate the jacobian. Details of the bundle adjustment algorithm itself can be found in [30] and [71].

6.4.1 Parametrization

The entries of the intrinsic matrix are parameterized by the focal length f, the principal point u_0 and the scale factor s. For each pose i, the translation vector t_i is represented by its entries (t_{1i}, t_{2i}, t_{3i}) and the rotation R_i is parameterized by computing, at each iteration, update rotations with small angles Δ_i, relative to the rotations of the previous iteration S_i:

$$\mathsf{R}_i = \Delta_i \mathsf{S}_i$$

The matrix Δ_i is the skew-symmetric matrix that encodes the cross product with the vector (w_{1i}, w_{2i}, w_{3i}). Its direction represents the axis of the rotation update and its norm represents the update rotation angle.

6.4.2 Partial Derivatives

We define the residual e of the cost function we wish to minimize (see (6.18)) in terms of its components $e1$ and $e2$ as follows:

$$\begin{aligned} e1 &= u_{ij} - \frac{\left(\mathsf{K}\left[\bar{\mathsf{R}}_i(a_j, b_j, 1)^{\mathsf{T}} - \mathbf{t}_i\right]\right)_1}{\left(\mathsf{K}\left[\bar{\mathsf{R}}_i(a_j, b_j, 1)^{\mathsf{T}} - \mathbf{t}_i\right]\right)_3} \\ e2 &= v_{ij} - \left(\mathsf{K}\left[\bar{\mathsf{R}}_i(a_j, b_j, 1)^{\mathsf{T}} - \mathbf{t}_i\right]\right)_2 \end{aligned}$$

We recall that $(u_{ij}, v_{ij})^{\mathsf{T}}$ is the projection of the j^{th} grid point in the camera i. The derivatives of the residual error w.r.t the intrinsic parameters are:

$$\begin{pmatrix} \frac{\partial e1}{\partial f, u_0, s} \\ \frac{\partial e2}{\partial f, u_0, s} \end{pmatrix} = \begin{pmatrix} \frac{-\lambda_1}{\lambda_2} & -1 & 0 \\ 0 & 0 & -t_3 - \lambda_3 \end{pmatrix}$$

Notice that the derivatives are obtained after setting the values for the update angles (w_1, w_2, w_3) at zero. The derivatives w.r.t the translations are:

$$\begin{pmatrix} \frac{\partial e1}{\partial t_1, t_2, t_3} \\ \frac{\partial e2}{\partial t_1, t_2, t_3} \end{pmatrix} = \begin{pmatrix} -\frac{f}{\lambda_2} & 0 & \frac{f\lambda_1 + u_0\lambda_2}{\lambda_2^2} - \frac{u_0}{\lambda_2} \\ 0 & -s & 0 \end{pmatrix}$$

The derivatives of the residuals w.r.t the rotation updates are:

$$\begin{pmatrix} \frac{\partial e1}{\partial w_1, w_2, w_3} \\ \frac{\partial e2}{\partial w_1, w_2, w_3} \end{pmatrix} = \begin{pmatrix} \frac{u_0\lambda_3}{\lambda_2} + \frac{-\lambda_3\lambda_1 f - \lambda_3\lambda_2}{\lambda_2^2} & \frac{\lambda_1\lambda_4 f + \lambda_4\lambda_2 u_0}{\lambda_2^2} - \frac{\lambda_2 f - t_3 f + \lambda_4 u_0}{\lambda_2} & \frac{\lambda_3 f}{\lambda_2} \\ -s(t_3 - \lambda_2) & 0 & -s(\lambda_1 - t_1) \end{pmatrix}$$

where the intermediate variables $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are defined as follows:

$$\begin{aligned}
\lambda_1 &= ar_{11} + br_{12} + t_1 \\
\lambda_2 &= ar_{31} + br_{32} + t_3 \\
\lambda_3 &= ar_{21} - br_{22} \\
\lambda_4 &= ar_{11} - br_{12}
\end{aligned}$$

6.4.3 The jacobian

Using the partial derivatives of the residual, the formulation of the jacobian $\mathbf{J}$ of the cost function defined in (6.18) is straightforward. We give the example of the jacobian for $i = 3$ camera poses:

$$\mathbf{J} = \left(\begin{array}{c|c|c|c} \frac{\partial e}{\partial f, u_0, s} & \frac{\partial e}{\partial t_{11}, t_{21}, t_{31}, w_{11}, w_{21}, w_{31}} & 0 & 0 \\ \hline \frac{\partial e}{\partial f, u_0, s} & 0 & \frac{\partial e}{\partial t_{12}, t_{22}, t_{32}, w_{12}, w_{22}, w_{32}} & 0 \\ \hline \frac{\partial e}{\partial f, u_0, s} & 0 & 0 & \frac{\partial e}{\partial t_{13}, t_{23}, t_{33}, w_{13}, w_{23}, w_{33}} \end{array} \right)$$

6.5 Complete Plane-Based Calibration Algorithm

In this section we present the complete plane-based algorithm for linear cameras calibration. From n view of a calibration grid:

1. Estimate the projection matrices H_i for all n views (see section 6.3.1), using

point matches and the relation

$$\begin{pmatrix} u_{ij} \\ v_{ij} \\ 1 \end{pmatrix} \sim \mathsf{H}_i \begin{pmatrix} a_j \\ b_j \\ 1 \\ a_j^2 \\ b_j^2 \\ a_j b_j \end{pmatrix}$$

where j is an index for calibration points. The estimation of H_i is equivalent to the so-called DLT (Direct Linear Transform) and can be done by solving a linear equation system.

2. Compute matrices M_i according to Eq.6.11.

3. Form the matrix S of dimension $2n \times (3+n)$:

$$\mathsf{S} = \begin{pmatrix} m_{1,11}m_{1,12} & m_{1,11}m_{1,32} + m_{1,12}m_{1,31} & m_{1,31}m_{1,32} & m_{1,21}m_{1,22} & & \\ \vdots & \vdots & \vdots & & \ddots & \\ m_{n,11}m_{n,12} & m_{n,11}m_{n,32} + m_{n,12}m_{n,31} & m_{n,31}m_{n,32} & & & m_{n,21}m_{n,22} \\ M_{1,11}^2 - M_{1,12}^2 & 2(m_{1,11}m_{1,31} - m_{1,12}m_{1,32}) & M_{1,31}^2 - M_{1,32}^2 & M_{1,21}^2 - M_{1,22}^2 & & \\ \vdots & \vdots & \vdots & & \ddots & \\ M_{n,11}^2 - M_{n,12}^2 & 2(m_{n,11}m_{n,31} - m_{n,12}m_{n,32}) & M_{n,31}^2 - M_{n,32}^2 & & & M_{n,21}^2 - M_{n,22}^2 \end{pmatrix}$$

4. Solve the following system to least squares:

$$\mathsf{S} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_{4,1} \\ v_{4,2} \\ \vdots \\ v_{4,n} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

5. From the $v_{1,2,3}$, extract the intrinsic parameters f and u_0 according to Eq.6.12 and Eq.6.13.

6. Compute s and the extrinsic parameters according to the algorithm of subsection 6.3.3.

7. Optional but recommended: non-linear optimization of all unknowns, i.e. intrinsic and extrinsic parameters, by minimizing the reprojection errors (see subsection 6.4).

6.6 Experimental Results

The proposed algorithm has been tested on both synthetic data and real data. Both tests are detailed in the next two subsections.

6.6.1 Computer Simulations

We performed several tests of our algorithm using synthetic data. Throughout all the experiments, we used a planar calibration grid of $10 \times 10 = 100$ corners. The virtual camera has a 1000×1000 image resolution, a focal length of 1000, and its optical center at the image center, at pixel $(500, 500)$.

We refer to the *"calibration volume"* as the bounding box that encloses all the calibration grids. Actually the most relevant parameter is not the bounding box volume itself but its height. In our experiments, the volume height is expressed as a percentage of the grid's length. Some configuration examples with several calibration volumes are depicted in Fig.6.2.

Sensitivity to noise level

For this test, we used 10 planes oriented randomly in a calibration volume of 100% the size of the calibration grid. After projection, a gaussian noise with mean 0 and

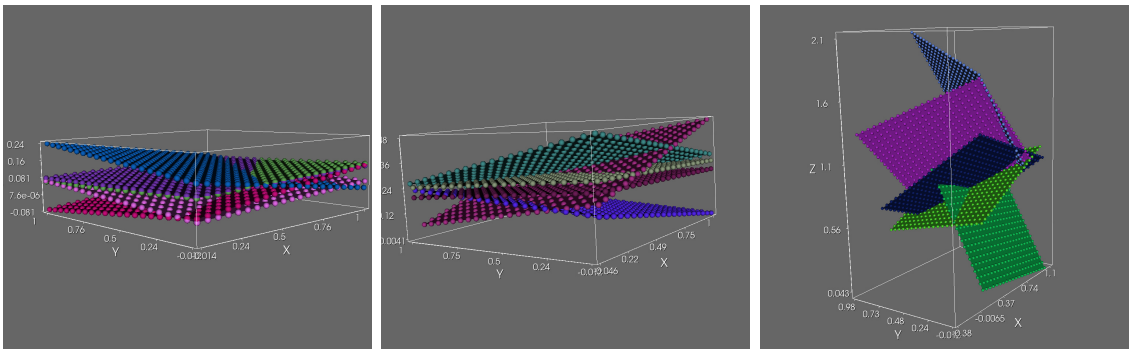

Figure 6.2. An example of 3 calibration volume with increasing height. From left to right, 25%**,** 50% **and** 200% **of the calibration length.**

increasing standard deviation was added to the image points. The standard deviation σ varied from 0.2 to 2. As in [83], we performed 100 independent runs for each noise level and computed the average errors for both the focal length and the principle point. As we can see from Fig.6.3 the error increases almost linearly for both the focal and the optical center. For an noise level of $\sigma = 0.5$ the errors in the focal and the optical center is less than 4 pixels which represents (given our camera characteristics) less than 0.8%.

Sensitivity to the number of planes

In this test, the sensitivity of our method w.r.t the number of planes is investigated. We set the calibration volume height to 100% of the grid's length and we varied the number of planes from 2 to 20. The average errors (from 100 independent runs) for both the focal length and the optical center were estimated and reported on Fig.6.4 for a noise level of $\sigma = 0.5$ and $\sigma = 1.0$. We notice that the errors decrease as more planes are used.

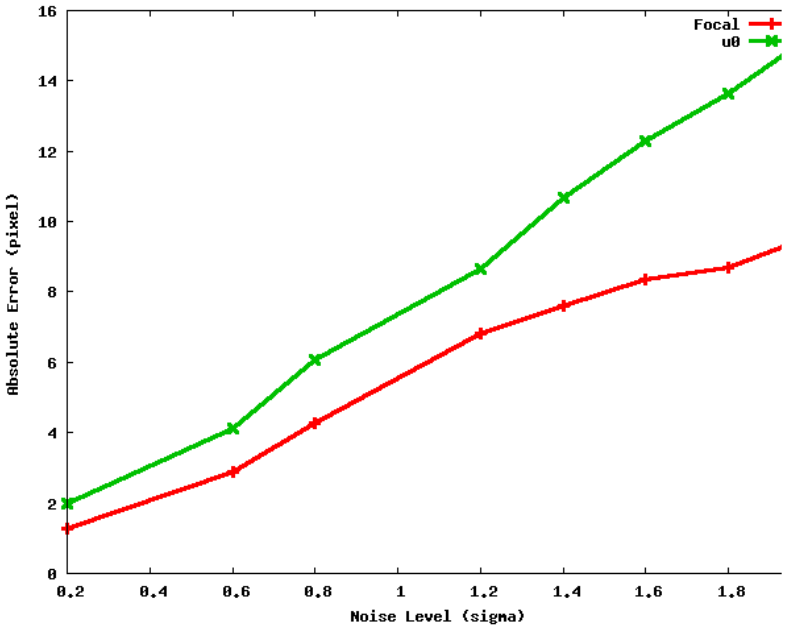

Figure 6.3. Focal length and optical center errors w.r.t the noise level in the image points.

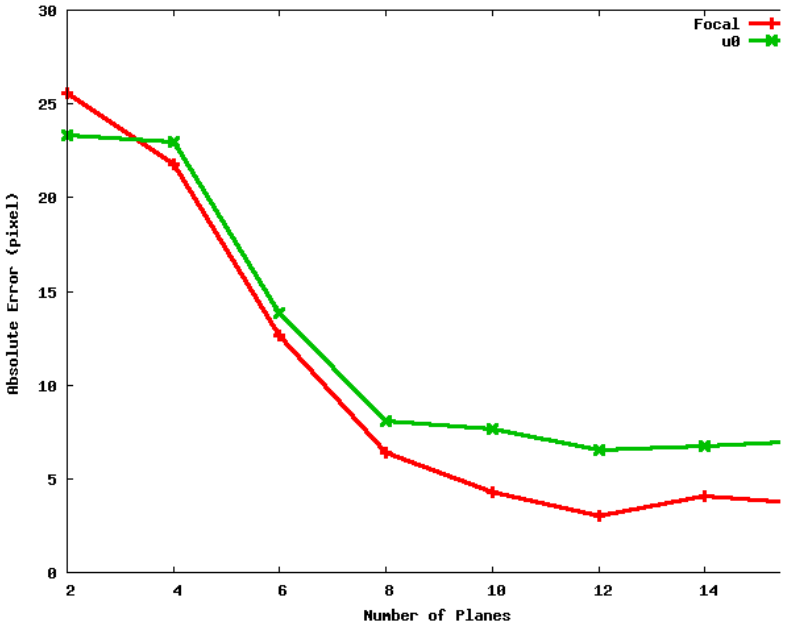

Figure 6.4. Focal length and optical center errors vs. the number of planes used ($\sigma = 0.5$).

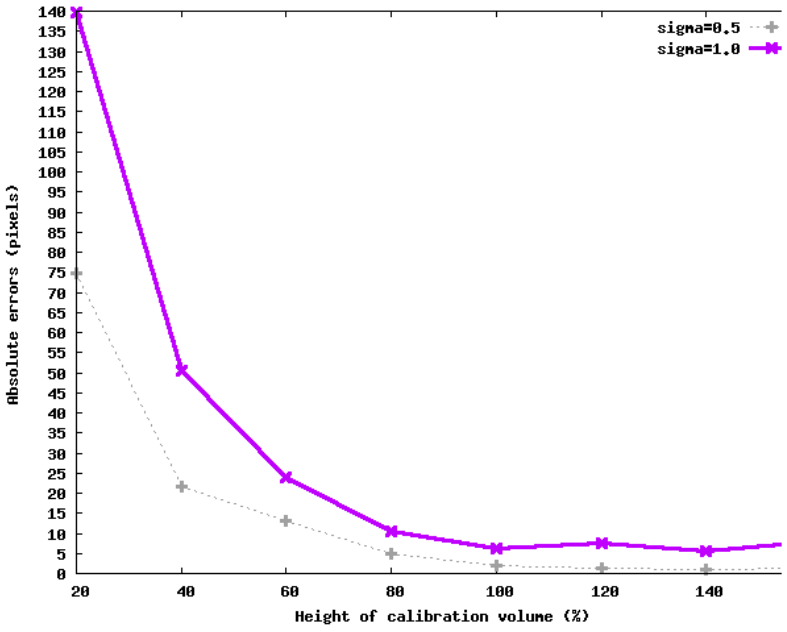

Figure 6.5. Focal length error vs. the height of calibration volume.

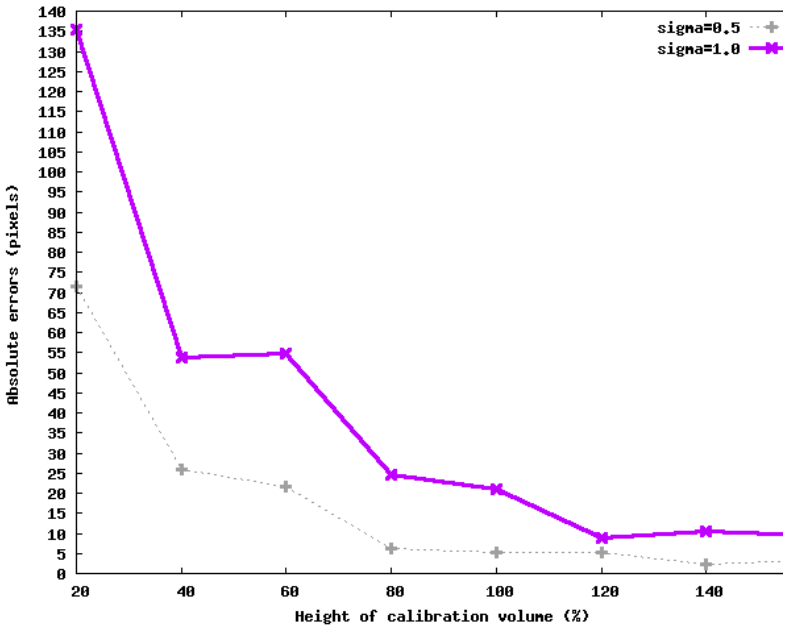

Figure 6.6. Optical center error vs. the height of calibration volume.

Sensitivity w.r.t the reconstruction volume

In this last synthetic experiment we analyse the performance of our method with respect to the calibration volume, or more precisely the volume's height. For this test we used 10 calibration grids oriented randomly and varied the calibration volume height from 20% to 160% of the grid's length (we remind that the grid is squared). This test was performed with a noise level of $\sigma = 0.5$ and $\sigma = 1.0$ (which is larger than the noise observed in a typical calibration [83]). We can see from Fig.6.5 and Fig.6.6 that the volume's height affects the quality of the calibration. In fact the errors decrease when a higher reconstruction volume is used. This is primarily due to the fact that a higher reconstruction volume permits a higher motion degree which guarantees a better sampling of the rotation space.

6.6.2 Real Data

Experiments on real data were conducted on two setups. The first one consists of a regular perspective camera mounted on a linear stage to simulate a pushbroom camera. In the second experiment we will show how a consumer flatbed scanner can be modeled as a pushbroom sensor. Because flatbed scanners are widely available and very affordable, they make a perfect device for high resolution measurements.

Camera + Linear Stage

For this experiment, we mounted a Prosilica camera on a controllable linear stage (see Fig.6.7). The camera was set to deliver images of 1360×1024 pixels at 5 frame per second. The speed of the stage was set to $4mm/s$. The size of the squares on the calibration plane were 1×1 inch.

From each image delivered by the camera, we extracted the column that passes by the principal point and form a *panorama* by stacking them on top of each other. Hence, the resulting panorama is akin to an image shot with a push-broom camera

Figure 6.7. Our setup to simulate a pushbroom camera. The camera (Prosilica) is mounted on a programmable linear stage. The accuracy of the stage is in the 100th of millimeter.

Table 6.1. Results of the camera calibration as Push-Broom and fully perspective (see text).

Parameter	Perspective	Push-Broom	Error (%)
Focal Length	1983.98	1998.32	0.7
Optical Center	554.81	549.68	0.9
Scale Factor	*31.75*	32.56	2.4

[60]. This procedure was repeated to acquire 10 images of the calibration plane under several orientations. The results of our calibration are shown in Table. 6.1. To assess the quality of our calibration, we also included the intrinsic parameters of the camera when calibrated as fully perspective. The later has been performed using the *OpenCV* library plane-based calibration routines.

We can see that the estimated perspective parameters (focal and principal point) are compatible with the results obtained using a standard plane-based calibration. The scale factor in the perspective column (6.1) is the expected value given our settings and is computed as follow. Within one second the camera acquires 5 frames, thus 5 columns of the push-broom image. In this same second the camera would have translated by $4mm = \frac{4}{25.4}in$. Since one unit of the calibration grid is 1 inch, the scale factor is $\frac{1\times5}{\frac{4}{25.4}} = 31.75$. Our method estimated a scale factor of 32.56, yielding an error of 2.4%.

Flatbed Scanner

We tested the proposed algorithm on an Epson V200 flatbed scanner. The manufacturer claims that the scanner is suited for scanning 3D objects thanks to its depth of field and adapted optic. We thus, modeled the scanner as a push-broom camera and used the proposed algorithm to retrieve its intrinsic parameters using a planar grid.

Table 6.2. Flatbed scanner calibration results (see text).

Parameter	**DLT**	**Plane-Based**	Error (%)
Focal Length	2673.4	2659.7	0.57
Optical Center	1315.2	1299.5	1.2
Scale Factor	—	146.48	2.4

The scans were done at a resolution of 300dpi (dot per inch), the grid's squares were half inch long each. Resulting images had a resolution of 2538x2328. Homographies were estimated by first detecting grid's features using *OpenCV* routines. To ensure a better numerical stability, points were normalized as suggested in [29]. We also calibrated the same scanner using the DLT method proposed by Hartley [52] [24]. In the later case, we scanned a 3D calibration rig and features were manually selected. Results and comparisons are reported in Table. 6.2

Since no ground truth was available, we took as a reference the classical calibration method proposed by Hartley et al.[52, 24] and we can see that the focal length and the principle point estimated by our method are very close to the estimation made by Hartley's method (both parameters differ by less than 1.5%). Further, each square of the calibration grid measured 0.5 inch length and giving the fact that the tests were made at a resolution of 300dpi, the scaling factor s should be $s = 300 \times 0.5 = 150$ which differs by only 2.4% from the scaling factor computed using our method.

6.7 Conclusion

In this paper we have presented a simple algorithm to calibrate a linear camera. The calibration is done using images of a planar grid acquired under different orientations. The proposed method is based on a closed-form solution with an optional non-linear refinement. Both synthetic and real experiments proved the effectiveness and the

quality of our procedure. As opposed to the reference method, the proposed one estimates all three internal parameters including the scaling factor, and the calibration tool is as simple as a planar grid.

Chapitre 7

MODÉLISATION ET CALIBRAGE DE PROJECTEURS

Afin de faciliter la résolution de certains problèmes fondamentaux en vision, tel que le problème d'appariement, la communauté de vision a créé un sous-domaine de recherche : la vision active. La vision active renvoie à un paradigme qui consiste à contrôler activement la scène ou les paramètres du capteur d'images [73]. Parmi ces interventions actives on retrouve le contrôle d'éclairage, la projection de motifs spéciaux, l'usage de capteurs/instruments de mesures (accéléromètres, odomètres,. . .).

Au début de la vision active, on avait recours à des dispositifs à faible coût mais aux performances limitées, tel que les acétates imprimées de motifs ou des faisceaux laser. Cependant, l'arrivée des vidéo-projecteurs (VP) n'a pas fait que le bonheur des cinéphiles-maison. Les chercheurs en vision par ordinateurs s'en sont aussi accaparés pour mieux *contrôler* les environnements de travail (contrôle de l'éclairage, réalité augmentée, . . .).

En plus de leur usage conventionnel comme dispositifs de projections, on retrouve les VP dans moult d'applications. En voici quelques-unes :

Multi-projection. Il est possible de combiner plusieurs projecteurs pour combler des besoins de projection à grande échelle ou encore pour créer des environnements immersifs (voir figure 7.1).

Lumière structurée. Ici le principe de base consiste à projeter une ou plusieurs images dans une scène afin d'en extraire la géométrie. Les images projetées ont une structure définie au préalable afin de permettre l'encodage directe d'informations sur la scène. L'observation de ces codes peut être exploitée par la suite afin d'inférer la structure 3D de la scène. Un exemple de lumière

Figure 7.1. Projection multiple. L'adjonction de 2 ou plusieurs projecteurs permet de couvrir de grandes surfaces de projection.

structurée utilisant des codes de Grey est illustré à la figure 7.2.

Stéréo Photométrique. Sous l'hypothèse d'un modèle de réflexion lambertien, il est possible de reconstruire des objets en 3D en les éclairant sous des angles différents et bien sûr, à condition de connaitre l'orientation de chaque source de lumière (voir figure 7.3). Pour cela, les VP constituent un excellent outil car après calibrage, il est possible d'estimer la pose du projecteur et donc la direction de l'éclairage.

La plupart de ces applications opèrent au niveau métrique (euclidien) et pour cela, le VP doit être calibré au même titre qu'une caméra afin d'en estimer les propriétés géométriques de projection. Nous allons donc nous atteler au problème du calibrage géométrique du vidéo-projecteur. Mais avant toute chose, il nous faut présenter le modèle de projection du VP.

Figure 7.2. Lumière structurée. La déformation des motifs peut être exploitée pour recouvrir la structure 3D.

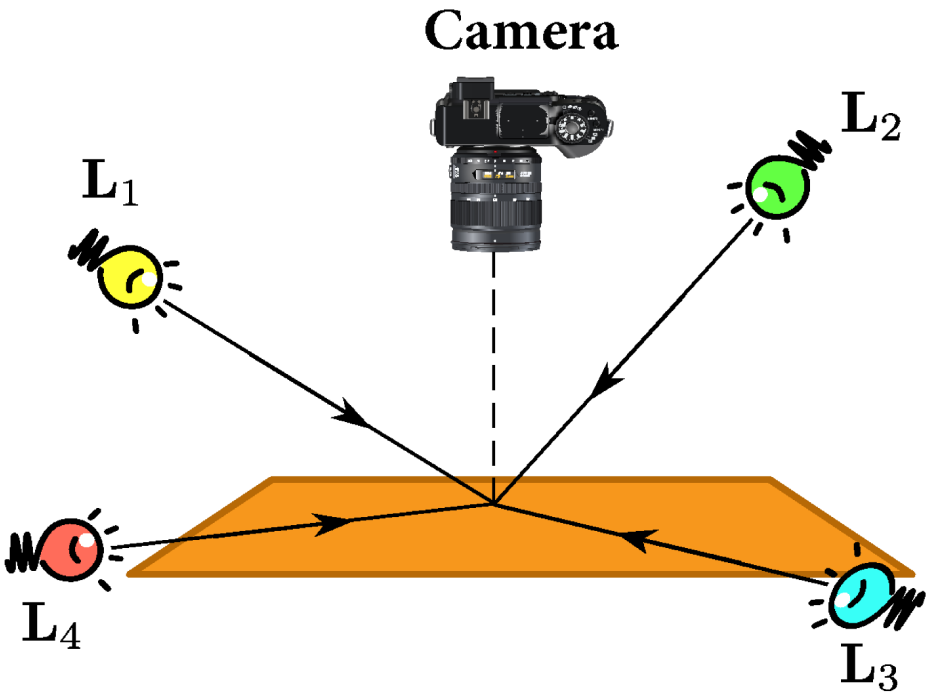

Figure 7.3. Principe de la stéréo photométrique. Avec au moins 3 lumières non coplanaires, il est possible de reconstruire la surface d'un objet.

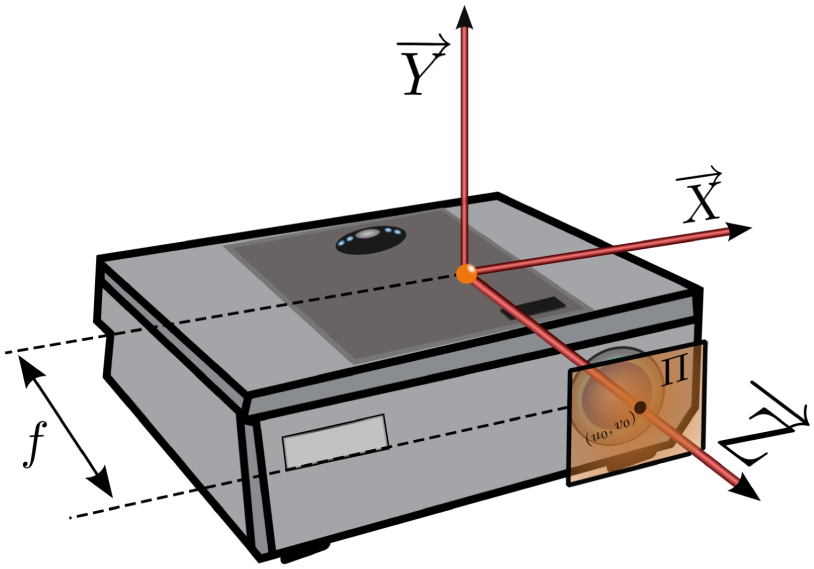

Figure 7.4. Anatomie d'un vidéo-projecteur. Le point principal se trouve en (u_0, v_0) la distance qui le sépare du plan focal Π est la longueur focale f.

7.1 Géométrie du vidéo-projecteur

En vision par ordinateur, il est d'un commun usage de considérer un VP comme une caméra *inversée*. Ici, le terme renvoie au chemin inversé qu'empruntent les rayons de lumière par rapport à une caméra : de l'intérieur vers l'extérieur. Ici nous adopterons, une fois de plus, le modèle sténopé pour modéliser la géométrie interne du projecteur. Les paramètres intrinsèques sont au nombre de quatre. Les coordonnées du point principal (u_0, v_0), la distance focale, f, qui sépare le centre de projection du plan image Π et un rapport d'échelle ρ. Ce dernier paramètre représente le rapport entre les dimensions d'un pixel et vaut 1 pour des pixels carrés. Ce modèle est illustré à la figure 7.4.

7.2 Calibrage du vidéo-projecteur

Le problème que pose le calibrage du projecteur par rapport à la caméra est que le VP ne "*regarde*" pas la scène mais y diffuse un contenu. L'usage d'une caméra ou d'un capteur externe est donc essentiel à la procédure de calibrage pour observer l'interaction entre le projecteur et la scène. Nous allons à présent passer en revue les principales méthodes de calibrage de VP qu'on retrouve dans la littérature. Elles ont en commun l'usage d'une mire plane comme dispositif de calibrage et se distinguent par la façon d'estimer l'orientation relative entre le projecteur et la mire. Dans ce qui suit, le terme "projecteurs" au pluriel désigne un même projecteur physique (mêmes paramètres intrinsèques) mais sous différentes orientations (paramètres extrinsèques).

Shen *et al.* [62] proposent d'utiliser une mire de calibrage plane montée sur un plan de projection dont le déplacement est contrôlé électriquement. Ceci permet de connaitre avec précision l'orientation de la mire par rapport au VP. Après quoi, la mire est ôtée et une caméra enregistre les coordonnées de points projetés par le VP sur le plan de projection. En appliquant ces deux étapes, en alternance, tout en changeant la pose du VP, on peut calculer les homographies qui relient la mire et les projecteurs. Une fois ces homographies estimées, le schéma de calibrage est identique au calibrage planaire de caméra [67, 83].

Afin d'estimer l'orientation du plan de projection (appelons-le *mur*) sans avoir recours à un dispositif mécanique, Sadlo *et al.* [55] utilisent une mire physique attachée au mur. À l'aide d'une caméra calibrée et de la mire physique, l'orientation caméra-mur est estimée à partir des paramètres extrinsèques de la caméra. Ceux-ci permettent par la suite de contraindre l'orientation mur-projecteur. Le reste de la manipulation consiste à calculer les homographies entre la caméra qui demeure fixe et le projecteur dont la pose change. Ces homographies, combinées à l'orientation du mur, seront converties en homographies mur-projecteur lesquelles permettent d'estimer les paramètres intrinsèques du projecteur [67, 83] (voir aussi Eq.5.2). On nommera cette

méthode "calibrage linéaire direct" et nous présenterons au chapitre suivant le détail d'une variante proposée par les nous.

Les caméras ne sont pas l'unique capteur d'acquisition pour calibrer un VP. Raskar *et al*. [40] ont montré qu'il était possible de calibrer un projecteur en utilisant des capteurs de lumières (types photo-transistor) encastrés dans un plan et dont la position est connue [40]. Pour une position donnée du plan et des pixels de projecteurs allumés, les capteurs indiquent quels pixels du projecteur éclairent quelle portion du plan. Même si à la base, ces travaux visaient la correction d'effets d'alignements dans un système multi-projecteurs, ils peuvent facilement être adaptés au calibrage géométrique de VP.

À travers cette revue de littérature, on voit clairement le problème sous-jacent auquel on se heurte lors du calibrage d'un VP : l'estimation de l'orientation du mur. Ceci n'est pas surprenant car l'orientation du mur permet de définir une métrique et donc d'assigner des coordonnées 3D au points du VP. À défaut, aucune relation mur-projecteur ne peut être établie.

Malheureusement, pour y parvenir, les méthodes citées ci-haut ne sont pas souples, du moins en pratique, car elles utilisent des éléments intermédiaires (mire physique, plan de projection contrôlé, ...) susceptibles d'introduire des erreurs supplémentaires ou encore qui nécessitent un usinage particulier (capteurs disposés avec précision).

Au chapitre suivant, nous présenterons un calibrage simple et pratique pour estimer les paramètres intrinsèques d'un VP. Cette nouvelle méthode, mise au point par l'auteur, ne nécessite qu'une caméra partiellement calibrée.

Chapitre 8

PROJECTOR CALIBRATION USING A MARKERLESS PLANE

Cet article [16] a été publié comme l'indique la référence bibliographique :

J. Draréni, P.F. Sturm, et S. Roy. Projector Calibration Using a Markerless Plane. Dans *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, pages 377–382. IEEE Computer Society, 2009.

Cet article est présenté ici dans sa version originale.

Abstract

In this paper we address the problem of geometric video projector calibration using a markerless planar surface (wall) and a partially calibrated camera. Instead of using control points to infer the camera-wall orientation, we find such relation by efficiently sampling the hemisphere of possible orientations. This process is so fast that even the focal of the camera can be estimated during the sampling process. Hence, physical grids and full knowledge of camera parameters are no longer necessary to calibrate a video projector.

8.1 Introduction

With the recent advances in projection display, video projectors (VP) are becoming the devices of choice for active reconstruction systems. Such systems like Structured Light [56] and Photometric Stereo [76, 3] use VP to alleviate the difficult task of

establishing point correspondences. However, even if active systems can solve the matching problem, calibrated VP are still required. In fact, a calibrated projector is required to triangulate points in a camera-projector structured light system, or to estimate the projector's orientation when the latter is used as an illuminant device for a photometric stereo system.

Since a video projector is often modeled as an inverse camera, it is natural to calibrate it as part of a structured light system rather than as a stand alone device. In order to simplify the calibration process, a planar surface is often used as a projection surface on which features or codified patterns are projected. The projector can be calibrated as a regular camera, except for the fact that a regular *accessory* camera must be used to see the projector patterns. The way patterns are codified and the projection surface orientation is estimated will distinguish the various calibration methods from each other.

In [62], a VP projects patterns on a plane mounted on a mechanically controlled platform. Thus, the orientation and position of the projection plane is known and is used to calibrate the structured light system using conventional camera calibration techniques.

Other approaches use a calibrated camera and a planar calibration chessboard attached to the projection surface [50, 55].

For convenience and because the projection surface is usually planar, we will refer to it as the *wall*. The attached chessboard is used to infer the orientation and the position of the wall w.r.t the camera. This relation is then exploited, along with the images of the projected patterns to estimate the intrinsic parameters of the projector.

In order to measure the 3D position of the projected features, [55] estimates the homography between the attached chessboard and the camera. This allows the computation of the extrinsic parameters of the camera. It is important to mention that the camera must be fully calibrated in this case. With at least three different orientations, a set of 3D-2D correspondences can be obtained and then used to estimate the

VP parameters with standard plane-based calibration methods [66, 81]. We refer to this method as Direct Linear Calibration (DLC). To increase accuracy of the DLC, a printed planar target with circular markers is used in [50], to calibrate the camera as well as the projector.

In [40], a structured light system is calibrated without using a camera. This is made possible by embedding light sensors in the target surface. Gray-coded binary patterns are then projected to estimate the sensor locations and prewarp the image to accurately fit the physical features of the projection surface. The VP parameters are not explicitly estimated but the method could easily be extended for that purpose.

In this paper, a new projector calibration method is introduced. The proposed method does not require a physical calibration board nor a full knowledge of the camera parameters.

We overcome the problem of determining the camera-wall homography $\mathsf{H}_{w\to c}$ by exploring the space of all acceptable homographies and consider the one that minimizes the reprojection error (see Figure.8.1). Since $\mathsf{H}_{w\to c}$ depends only on the orientation between the camera and the wall, the space of acceptable homographies can be parameterized with only 2 angles: the elevation and the azimuth angles that define the normal vector at the wall.

Finding the normal of the wall consists then in sampling the space of orientations on a unit sphere. For each orientation sample, a DLC is performed and we select the homography that minimizes the reprojection errors in the images. It is worth mentioning that our DLC implementation differs slightly from the one used in [55] as explained in the next section.

Our proposed method is fully automatic, fast and produces excellent results as shown in our experiments. We also show that when the camera is not fully calibrated, projector calibration is still tractable. This is done by making the common assumptions that the pixels are square and that the center of projection coincides with the image center [64]. Thus, the only unknown camera parameter left to estimate is the

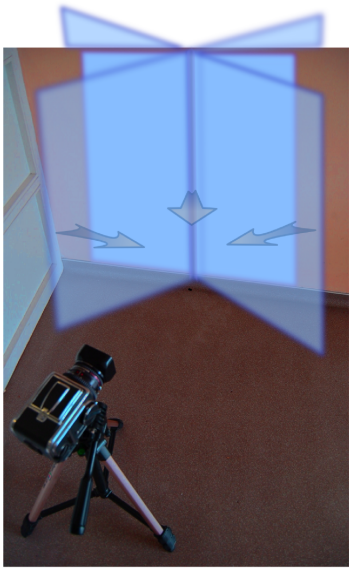

Figure 8.1. The homography wall-camera is defined by the orientation of the wall.

focal length, which is estimated by sampling.

The rest of this paper is organized as follows. Section 8.3 presents our variant of the direct linear calibration for a projector. Section 8.4 details our orientation sampling calibration (OSC) using only a (partially calibrated) camera and a marker-less projection plane.

Section 8.5 presents the results of our calibration method, followed by a discussion of limitations and future work in Section 8.6.

8.2 Video Projector Model

We model the video projector as an inverse camera. Therefore, we intend to compute the intrinsic and extrinsic parameters. Without loss of generality, we consider in this paper a 4 parameters projector model, namely: the focal length, the aspect ratio and

the principal point. Thus, the projector matrix K_p is defined as:

$$\mathsf{K}_p = \begin{pmatrix} \rho f & 0 & cx \\ 0 & f & cy \\ 0 & 0 & 1 \end{pmatrix}$$

The extrinsic parameters that describe the i^{th} projector pose are the usual rotation matrix R^i and the translation vector $\mathbf{t}^i$.

8.3 Direct Linear Calibration

In this section, we review the details of the Direct Linear Calibration for projectors. This method is used as a reference for our benchmark test. As opposed to [55], the variant presented here is strictly based on homographies and does not require a calibrated camera.

If a static camera observes a planar surface (or a wall), a homography is induced between the latter and the camera image plane. This linear mapping ($\mathsf{H}_{w\to c}$) relates a point P_w on the wall to a point P_c in the camera image as follows:

$$\mathbf{P}_c \sim \mathsf{H}_{w\to c} \cdot \mathbf{P}_w \tag{8.1}$$

Where $\sim$ denotes equality up to a scale. Details on homography estimation can be found in [30].

The video projector is used afterward to project patterns while it is moved to various positions and orientations. For a given projector pose i, correspondences are established between the camera and the VP, leading to a homography $\mathsf{H}^i_{c\to p}$. A point $\mathbf{P}^i_c$ in the image i is mapped into the projector as:

$$\mathbf{P}^i_p \sim \mathsf{H}_{c\to p} \cdot \mathbf{P}^i_c \tag{8.2}$$

Combining Eq.10.4 an Eq.10.5, a point $\mathbf{P}_w$ on the wall is mapped into the i^{th} projector as:

$$\mathbf{P}_p^i \sim \underbrace{\mathsf{H}_{c\to p}^i \cdot \mathsf{H}_{w\to c}}_{\mathsf{H}_{w\to p}^i} \cdot \mathbf{P}_w \tag{8.3}$$

On the other hand, $\mathbf{P}_p^i$ and $\mathbf{P}_w$ are related through a perspective projection as:

$$\mathbf{P}_p^i \sim \mathsf{K}_p \cdot \left[\mathbf{R}_1^i \mathbf{R}_2^i \mathbf{t}^i\right] \cdot \mathbf{P}_w \tag{8.4}$$

Where K_p, $\mathbf{R}_{1,2}^i$ and $\mathbf{t}^i$ are respectively the projector intrinsic parameters, the two first vectors of the rotation matrix R^i, and the translation vector. From Eq.10.6 and Eq.10.7, a relation between $\mathsf{H}_{w\to p}^i$ and the extrinsic parameters of the projector is derived as follows:

$$K_p^{-1} \cdot \mathsf{H}_{w\to p}^i \sim \left[\mathbf{R}_1^i \mathbf{R}_2^i \mathbf{t}^i\right] \tag{8.5}$$

With at least two different orientations, one can solve for K_p^{-1} by exploiting the orthonormal property of the rotation matrix as explained in [66].

8.4 Orientation Sampling Calibration

In this section we give the details of our proposed video projector calibration method. As discussed earlier, the justification for using an attached calibration rig to the wall is to infer the homography *wall-camera* in order to estimate the 3D coordinates of the projected features. We propose to estimate this *wall-camera* relation by exploring the space of all possible orientations since only the orientation of the wall w.r.t the camera matters and not its position.

Another way to look at this orientation space is to consider all vectors lying on a unit hemisphere placed on the wall, as depicted on Figure 8.1.

The calibration process can be outlined in three main steps:

- Pick a direction on the hemisphere.
- Compute the corresponding homography.

- Use the homography to perform a DLC calibration (Section 8.3).

The above steps are repeated for all possible directions and the direction that minimizes the reprojection errors is selected as the correct plane orientation. The first two steps are detailed in the next subsections. The third one is straightforward from section 8.3.

8.4.1 Sampling a Hemisphere

The problem of exploring the set of possible orientations is dependent on the problem of generating uniformly distributed samples on the unit sphere (hemisphere in our case).

Uniform sphere sampling strategies can be random or deterministic [79]. The first class are based on random parameters generation, followed by an acceptance/rejection step depending on whether the sample is or not on the sphere. Deterministic methods produce valid samples on a unit sphere from uniformly distributed parameters, such method include (but not limited to) quaternion sampling [34], normal-deviate methods [37] and methods based on Archimedes theorem [46]. We chose to use the latter method for its simplicity and efficiency. As the name suggests, this method is based on Archimedes theorem on the sphere and cylinder which states that the area of a sphere equals the area of every right circular cylinder circumscribed about the sphere excluding the bases. This argument leads naturally to a simple sphere sampling algorithm based on cylinder sampling [46]. Uniformly sampling a cylinder can be done by uniformly choosing an orientation $\theta_i \in [0, \pi]$ (we call it azimuth) to obtain a directed vector $d(\theta_i, 0)$ (See Figure.8.2). After that, a height h_i is uniformly chosen in the range $[-1, 1]$. The resulting vector, noted $d_i(\theta_i, h_i)$, is axially projected on the unit sphere. According to the above theorem, if a point is uniformly chosen on a cylinder, its inverse axial projection will be uniformly distributed on the sphere as well, see [46] for further details.

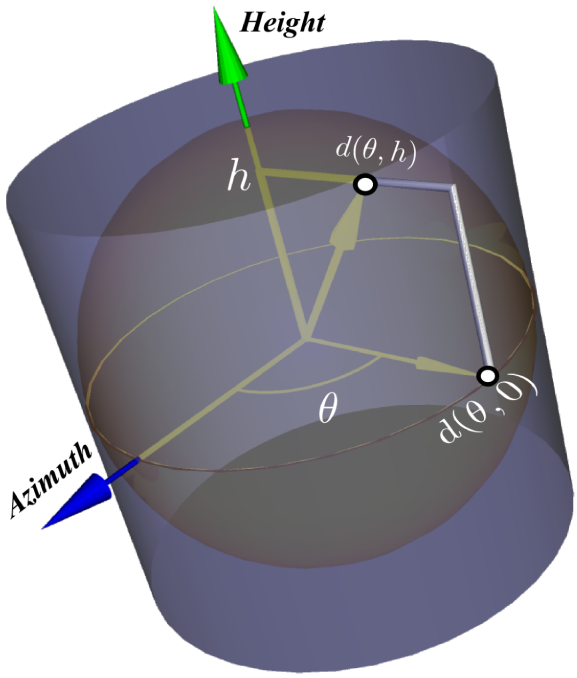

Figure 8.2. Orientation space sampling.

In our case, we only need to sample the hemisphere facing the camera. Thus the span of the points that must be visited is limited to the range $[-1,+1] \times [0,\pi]$.

8.4.2 Homography From an Orientation Sample

The homography wall-camera $\mathsf{H}^i_{w\to c}$ induced by a wall whose normal is a direction $\mathbf{d}_i$ (as defined in the previous subsection), is defined by:

$$\mathsf{H}^i_{w\to c} \sim \mathsf{K}_{cam}.\left[\mathbf{R}^i_1\mathbf{R}^i_2\mathbf{t}\right] \tag{8.6}$$

Where K_{cam}, $\mathbf{R}^i_1$, $\mathbf{R}^i_2$ and $\mathbf{t}$ are respectively the intrinsic camera matrix, the first two vectors of the rotation corresponding to the direction $\mathbf{d}_i$, and the translation vector. Without loss of generality and for the sake of simplicity, we fix the projection of the origin of the wall $P^0_w = (0,0)^\mathsf{T}$ into the camera at the image center. With this convention, the translation vector $\mathbf{t}$ simplifies to $(0,0,1)^\mathsf{T}$.

The rotation matrix R^i is computed via Rodrigues formula, which requires a rotation axis and a rotation angle. The rotation axis is simply the result of the cross product between $\mathbf{d}_i$ and the vector $(0,0,1)^\mathsf{T}$ whereas the rotation angle α_i is obtained from the dot product of the same vectors:

$$\alpha_i = \mathbf{cos^{-1}} \left(\mathbf{d}_i{}^\mathsf{T} \cdot (0,0,1)^\mathsf{T} \right) \tag{8.7}$$

8.4.3 Complete Algorithm

We are now ready to give the complete algorithm of our video projector calibration. We assume the existence of two supporting functions, *ReprojError* that returns a reprojection error for a given projector parameters and *DLC* a function that estimate the projector parameters using the DLC method (see Section.8.3).

Algorithme 1 : Orientation Sampling Calibration

Data : $\mathsf{H}^k_{c\to p}$, the k camera-projector homographies and K_{cam} Camera intrinsic matrix (optional).

foreach $(h_i,\theta_i)\in[-1,1]\times[-\pi/2,\pi/2]$ **do**

Estimate direction $\mathbf{d}_i(\theta_i, h_i)$ (sec.8.4.1)

if K_{cam} *is undef* **then**

Initialize elements of K_{cam} using image center and f_i

end

Estimate $\mathsf{H}^i_{w\to c}$ from $\mathbf{d}_i$ and f_i (sec.8.4.2)

foreach $\mathsf{H}^k_{c\to p}$ **do**

$\mathsf{H}^k_{w\to p} = \mathsf{H}^k_{c\to p}\cdot\mathsf{H}^i_{w\to c}$

end

$\mathsf{K}^i_{proj} \leftarrow \mathtt{DLC}(\mathsf{H}^k_{c\to p})$ (sec.8.3)

$\mathsf{Error} \leftarrow \mathtt{ReprojError}(\mathsf{K}^i_{proj})$

if $\mathsf{Error} < \mathsf{BestError}$ **then**

$\mathsf{K}_{proj} \leftarrow \mathsf{K}^i_{proj}$

$\mathsf{BestError} \leftarrow \mathsf{Error}$

end

end

return *Projector calibration matrix* K_{proj}

8.5 Experiments

We have evaluated the proposed calibration method with both a calibrated and an uncalibrated cameras. The results were also compared to the DLC method. The evaluation platform consists of a Mitsubishi pocket projector of 800×600 pixels resolution and a digital camera (Nikon D50). A $50mm$ lens was used on the camera and the resolution was set to 1500×1000. The calibration of the camera using the

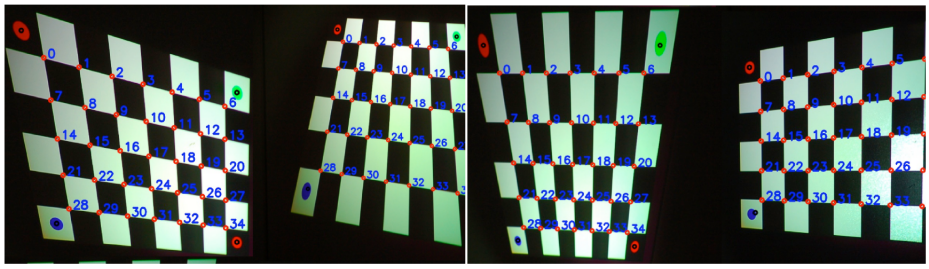

Figure 8.3. Images of projected patterns and detected features. The numbers and small red dots are added for illustration only. The large dots in the 4 corners are part of the projected pattern.

Matlab toolbox gave the following intrinsic matrix K_{cam}:

$$\mathsf{K}_{cam} = \begin{pmatrix} 3176.3115 & 0 & 790.6186 \\ 0 & 3172.4809 & 495.3829 \\ 0 & 0 & 1 \end{pmatrix}$$

To include the DLC algorithm in our benchmark, the camera was mounted on a tripod and was first registered to the wall using an attached printed chessboard. Images of projected chessboard using the video projector under several orientations were then acquired using the camera. We took precaution to remove the attached chessboard form the wall before acquiring the projector images to avoid overlaps between the projected patterns and the rigidly attached pattern.

Some images of the projected chessboard along with detected features are depicted on Figure.10.8.

Notice the presence of colored dots on the chessboard. Those were used to compute a rough estimate of the homography (which will be refined) and to eliminate the orientation ambiguity of the chessboard while assigning 3D coordinates to the detected features.

Our benchmark includes a projector calibration using the DLC method, the proposed method with both a calibrated and an uncalibrated camera. In the first case,

we used the image of the attached checker to infer the *wall-camera* homography and calibrated as explained in Section.8.3. For the second method, we used a multi-resolution strategy to sample the azimuth angles and heights. The conditions of the third method were identical to the second one except that the camera parameters were ignored and were estimated as follows:

- The focal length estimation was included in the sampling process. The sampling range was $[0, 10000]$.
- The pixels are assumed square.
- The center of projection is assumed to coincides with the image center.

Table 8.1. Projector calibration benchmark: Direct method, Orientation sampling with a calibrated camera (Sampling-C) and Orientation sampling with an uncalibrated camera (Sampling-U).

Method	f_{proj}	ρ	cx	cy	$estf_{cam}$	Error	Error B.A
Direct	1320.13	1.02	382.1	368	-	4.35	0.47
Sampling-C	1327.30	1.01	377.4	366	-	0.43	0.22
Sampling-U	1322.15	1.00	376	360	3108	0.16	0.09

The result of this benchmark is outlined on the Table.1. The table provides the estimated parameters, the reprojection errors in pixels (*Error*), and the error difference comparing before and after applying a bundle adjustment refinement (*Error B.A*). Technical and implementation details on the latter can be found in [42].

The running times for a data set of 20 images on an 1.5 Ghz computer are provided in Table.2.

From this test, we can see that our method, even in the absence of camera parameters knowledge, out-perform the Direct Linear Method at the expenses of a higher

Table 8.2. Execution time for Direct method, Sampling with calibrated camera, and Sampling with uncalibrated camera.

Method	Time (seconds)
Direct	0.18
Sampling-C	1.23
Sampling-U	6.2

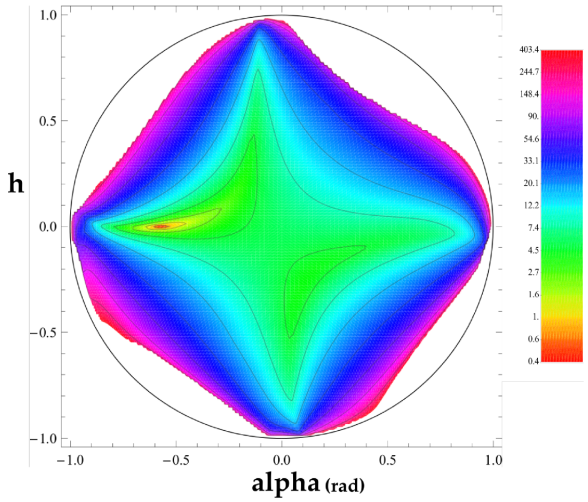

Figure 8.4. Reprojection error in terms of the orientation parameters h and α. The error computation does not include bundle adjustment refinement

running time. However, we are convinced that the performance of our implementation could be further improved by choosing a better multi-scale sampling strategy. We also consider that not requiring a printed chessboard attached to the wall is a major advantage, especially when the wall surface is large or unaccessible.

A plot of the reprojection error in terms of the orientation parameters h and α is provided in Figure.8.4. We can clearly see that the function is very well behaved and easy to minimize.

As a last test, we wanted to assess the stability of the focal length estimate. We thus fixed the value of the wall orientation at the value obtained in the first experiment

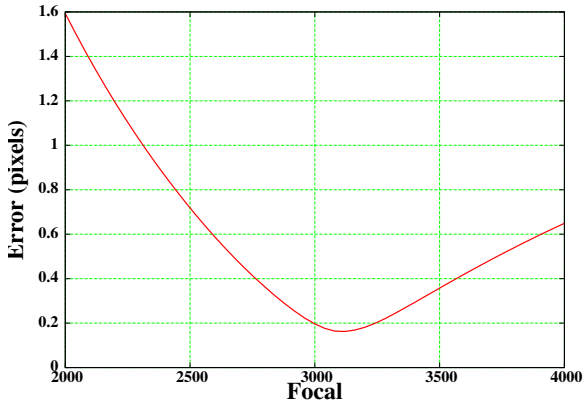

Figure 8.5. Reprojection error in terms of the camera focal length values (prior to bundle adjustment procedure). The minimum is reached at 3034.4**, the off-line camera calibration estimated a camera focal of** 3176**.**

and varied the focal length. The plot of the reprojection error as a function of the sampled focal length of the camera is shown on Figure.8.5. As we can see the error function is smooth and convex, suggesting that the lack of knowledge of the focal length can easily be circumvented in practice.

8.6 Conclusion

In this paper we presented a new video projector calibration method. Contrary to most methods, we showed that a physical target attached to a projection surface is not necessary to achieve an accurate projector calibration. We also suggest that full knowledge of camera parameters is not strictly required and can be relaxed into a set of commonly used assumptions regarding the camera geometry. Very simple to implement, the proposed method is fast and will handle large projector-camera systems that were previously impossible to calibrate due to the impractical chessboard.

Chapitre 9

INTRODUCTION À L'AUTO-CALIBRAGE PLAN

Nous avons présenté au chapitre précédent une nouvelle méthode pour calibrer un vidéo projecteur dont l'innovation résidait dans sa simplicité tant théorique que pratique. Nous avons montré que l'usage d'une mire physique n'était plus nécessaire. Ceci n'aurait pas été possible si on n'avait pas émis d'hypothèses sur la caméra, à savoir : des pixels carrés et un point principal confondu avec le centre de l'image.

Ces hypothèses, bien que raisonnables pour des caméras de haut de gamme, constitueraient une aberration si émises sur des caméras bon marché. Afin de calibrer notre projecteur sans mire physique, donc en l'absence d'une métrique, et à l'aide d'une caméra non calibrée il nous faut exploiter la rigidité de la scène et la co-planarité des points projetés. C'est exactement le besoin auquel répond l'auto-calibrage plan.

De façon générale, l'auto-calibrage est une procédure qui vise à estimer les paramètres intrinsèques d'une caméra à partir d'images acquises et sans aucune information sur la structure de la scène (angles, distances, métrique ...). Ceci s'avère bénéfique pour la reconstruction 3D dans le cas où nous avons affaire à des images acquises par des caméras non calibrées ou encore par des caméras dont la géométrie variable invaliderait tout calibrage fait d'avance (changement de zoom , mise au point, ...).

Le point de départ commun à toutes les méthodes d'auto-calibrage est l'exploitation de la rigidité de la scène. Cependant, nous nous intéresserons uniquement à l'auto-calibrage plan. Nous renvoyons le lecteur intéressé à l'auto-calibrage général à [30, 44, 65].

9.1 Auto-Calibrage plan pour les caméras

L'auto-calibrage plan s'applique à une scène constituée de points coplanaires mais de structure inconnue[1]. Supposons qu'une caméra observe cette scène sous n poses différentes et que des points d'intérêt soient extraits dans les n images de la séquence. La structure de la scène étant inconnue, il n'est plus possible de calculer les homographies scène-images comme pour le calibrage plan, par contre l'appariement de points saillants entre des caméras i et j est possible. Ceci permet l'estimation d'homographies dites inter-images ou inter-vues qu'on notera H_{ij} et qui serviront de paramètres d'entrée aux méthodes d'auto-calibrage plan que nous présenterons. Mais avant celà, nous allons présenter quelques entités géométriques utiles.

Conique absolue Nous avons vu précédement (§5.2.2) que la conique absolue, Ω, était une quadrique définie sur le plan infini Π_∞. Nous avons vu que son image, ω_∞, dans une caméra ne dépend que des paramètres internes de cette dernière.

Points cycliques L'intersection d'un plan Π avec la conique absolue résulte en deux points sur la droite à l'infini l_∞, nommés points cycliques du plan $Z = 0$. Les coordonnées de ces points vérifient l'équation suivante :

$$X^2 + Y^2 = 0$$

Il s'en suit que les points cycliques, au nombre de deux, sont complexes et conjugués : $\mathbf{J}_\pm = (1, i_\pm, 0, 0)$. Étant donné que les points cycliques $\mathbf{J}_\pm$ appartiennent à Ω_∞, leurs images $\mathbf{j}_\pm$ appartiennent à l'image de la conique absolue ω :

$$\mathbf{j}_\pm^\mathsf{T} \omega \, \mathbf{j}_\pm = 0$$

Ceci est illustré à la figure 9.1.

[1] Le contraire implique un repère euclidien connu attaché à la scène.

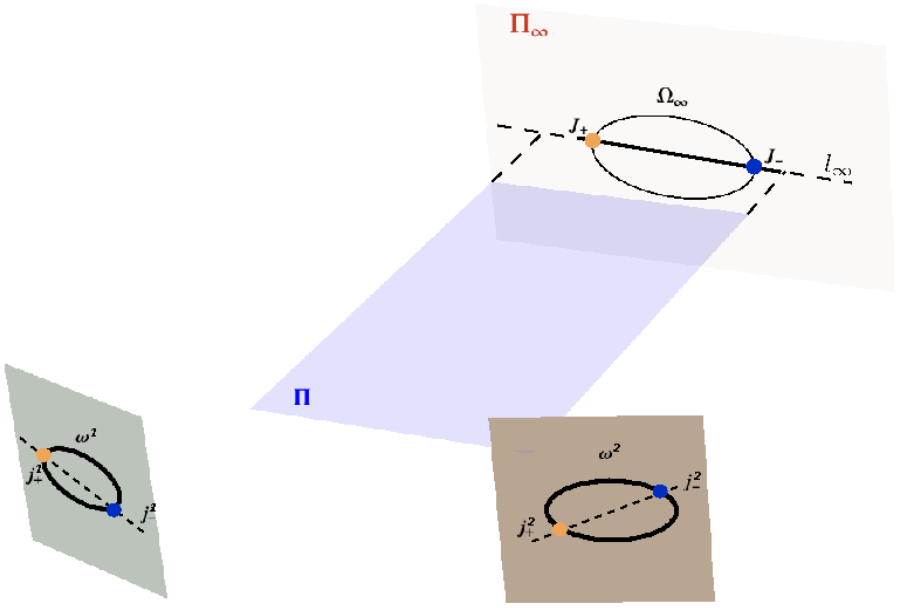

Figure 9.1. Le plan Π coupe la conique absolue Ω_∞ en deux points cycliques, J_+ et J_-. Ces mêmes points se reprojettent dans la caméra i en $j^i_\pm$.

9.1.1 Approche de B. Triggs

Dans l'approche présenté par Triggs [70], les images des points cycliques (IPC) sont exploitées pour contraindre l'image de la conique absolue. Contrairement au calibrage plan (voir §5), ces images sont désormais inconnues. Cependant, comme le note Triggs, si ces images sont connues dans une vue clef, il est possible de les transférer vers une vue j grâce à l'homographie inter-vues H_{1i} entre l'image clef (d'indice 1) et l'image j. Effectivement, si l'on note par $\mathbf{j}_\pm$, les IPC de la vue clef, $\mathsf{H}_{1i}\mathbf{x}^j_\pm$ représentent les IPC de la vue j. Par définition, les IPC de la vue i appartiennent à ω_i, l'image de la conique absolue dans la vue i, donc :

$$(\mathsf{H}_{1i}\,\mathbf{j}^i_\pm)^\mathsf{T}\omega_i(\mathsf{H}_{1i}\,\mathbf{j}^i_\pm) = 0 \tag{9.1}$$

En théorie, 4 images permettent d'estimer la matrice de la conique absolue définie

en fonction des paramètres de la caméra : focal, rapport d'échelle et point principal.

Afin de déterminer les IPC de l'image clef, Triggs propose une paramétrisation de l'homographie H_1 qui relie le plan de calibrage et la vue clef. Cette paramétrisation permet d'extraire les images initiales des IPC avec une connaissance a priori de la pose de la caméra. En pratique, une pose fronto-parallèle de la caméra procure une bonne initialisation [26].

Un point sensible de la méthode est sa nature non-linéaire. Les méthode itératives employées nécessitent de bonnes valeurs initiales des paramètres intrinsèques notamment celle de la focale [26].

9.1.2 Approche de E. Malis

Pour sa part, Malis [45] démontre une propriété intéressante de la matrice $\mathsf{H}_{ij}\left[\mathbf{n}_j\right]_\times$ (3×3), où $\mathbf{n}_j$ est la normale du plan dans le repère de la caméra j. Il prouve que les deux premières valeurs singulières sont égales et que la troisième est nulle. Ainsi, dans le cadre qu'il a présenté, la matrice de calibrage K doit minimiser :

$$\min_{\mathsf{K}} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{\sigma_1^{ij} - \sigma_2^{ij}}{\sigma_1^{ij}}$$

Où, les $\sigma_{1,2}^{ij}$ désignent les valeurs singulières de la matrice H_{ij}. L'avantage de la méthode de Malis par rapport à celle de Triggs réside dans la prise en compte en simultanée de toutes les vues sans en privilégier une en particulier.

9.2 Auto-calibrage Plan appliqué au Projecteur

Nous sommes à présent prêts à formuler et à résoudre le problème d'auto-calibrage plan pour le vidéo projecteur. Inspiré du travail de Gurdjos et Sturm [26], notre initialisation se fera à partir d'une pose, à peu près, fronto-parallèle du projecteur vis-à-vis de la surface de projection. Afin d'assurer un résultat de bonne qualité, il est possible de prendre plusieurs photos de la pose fronto-parallèle du projecteur

et de sélectionner celle qui donne les plus petites erreurs de reprojections. Ceci est expliqué en détail au chapitre suivant.

Chapitre 10

GEOMETRIC VIDEO PROJECTOR AUTO-CALIBRATION

Cet article [14] a été publié comme l'indique la référence bibliographique

J. Draréni, P.F. Sturm, et S. Roy. Geometric Video Projector Auto-Calibration. Dans *Proceedings of the IEEE International Workshop on Projector-Camera Systems*, IEEE Computer Society, 2009.

Cet article est présenté ici dans sa version originale.

Abstract

In this paper we address the problem of geometric calibration of video projectors. Like in most previous methods we also use a camera that observes the projection on a planar surface. Contrary to those previous methods, we neither require the camera to be calibrated nor the presence of a calibration grid or other metric information about the scene. We thus speak of geometric auto-calibration of projectors (GAP). The fact that camera calibration is not needed increases the usability of the method and at the same time eliminates one potential source of inaccuracy, since errors in the camera calibration would otherwise inevitably propagate through to the projector calibration. Our method enjoys a good stability and gives good results when compared against existing methods as depicted by our experiments.

10.1 Introduction

With the recent advances in projection display, video projectors are becoming the devices of choice for active reconstruction systems and 3D measurement. Such systems like Structured Light [56] and also Photometric Stereo [76, 3] use video projectors to alleviate the difficult task of establishing point correspondences. However, even if active systems can solve the matching problem, calibrated video projectors are still required. In fact, a calibrated projector is required to triangulate points in a camera–projector structured light system, or to estimate the projector's orientation when the latter is used as an illumination device for a photometric stereo system.

The projection carried out by a video projector is usually modeled as the inverse projection of a pin-hole camera, and thus considered as a perspective projection.

In order to simplify the calibration process, a planar surface is often used as projection surface, onto which features or codified patterns are projected. The way patterns are codified and the projection surface orientation is estimated distinguishes most previous calibration methods from one another.

In [62, 69], a video projector projects patterns on a plane mounted on a mechanically controlled platform. Thus, the orientation and position of the projection plane is known and is used to calibrate the structured light system using conventional camera calibration techniques.

For convenience and because the projection surface is usually planar, we will also refer to it as the *wall*.

In [55], a planar calibration grid is attached to the wall and observed by a calibrated camera. Due to the camera's calibration information and the metric information about the grid, the grid's and thus the wall's orientation and distance relative to the camera can be computed by classical pose estimation. After this, the 3D positions of features projected onto the wall by the video projector, can be easily computed. If this is done for three or more positions of the video projector, a set of

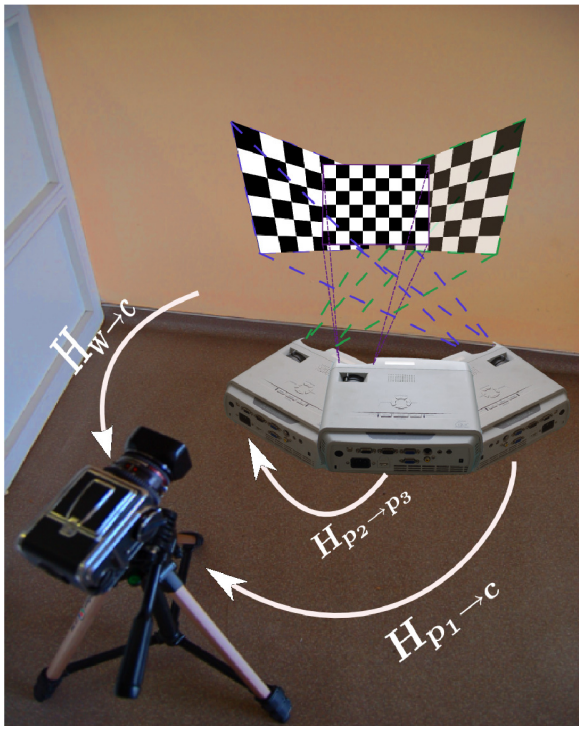

Figure 10.1. A Camera-Projector setup and its homographies (see text).

correspondences between the wall and the "projector images" can be obtained and then used to estimate the projector parameters with standard plane-based calibration methods [66, 81]. We refer to this method as Direct Linear Calibration (DLC). Note that all this could actually be done without pre-calibrating the camera, purely based on plane homographies, as explained in section 10.3. Further, to increase accuracy of the DLC, a printed planar target with circular markers is used in [50], to calibrate the camera as well as the projector.

In [40], a structured light system is calibrated without using a camera. This is made possible by embedding light sensors in the target surface (the wall). Gray-coded binary patterns are then projected to estimate the sensor locations and prewarp the image to accurately fit the physical features of the projection surface. The projector parameters are not explicitly estimated but the method could easily be extended for that purpose.

In [51], an auto-calibration method for multi-projector display walls is proposed. The authors focus more on estimating the relative orientations of the projectors w.r.t a camera to achieve a large seamless display. The method does not require fiducial points but makes assumptions on the projector intrinsic parameters and the camera must be calibrated. Further, the method assumes the x-axis of the projectors aligned.

Okatani *et al.* [49] presented a work on video projector auto-calibration but their work is meant for multiple projectors alignment and keystoning, provided that the intrinsic parameters of the projectors are known.

Kimura *et al.* [36] proposed a calibration method based on the camera-projector epipolar geometry. Again, the camera must be fully calibrated.

In this paper, a new projector calibration method is introduced. As opposed to most existing methods, the proposed method does not require a physical calibration grid nor any knowledge about the camera parameters. Indeed, our method imposes only two constraints on the calibration setup. Namely, the camera should remain static while the video projector displays patterns onto a planar surface and the user

must put the projector once in a roughly fronto-parallel position relative to the wall. The latter constraint does not have to be exact and serves only as a starting point for a non-linear minimization as explained below.

The rest of the paper is organized as follows. In section 10.2, our model for the geometric transformation associated with the video projector, is described. In section 10.3, we explain the above mentioned DLC (direction linear calibration) approach, which serves as an introduction to the proposed auto-calibration method, described in section 10.4. Experimental results are presented in section 10.5 and conclusions are drawn in section 10.6.

10.2 Projector Model

Throughout this paper, the projector is assumed to have a perspective projection model like a pin-hole camera, with the slight difference that here the projection direction is reversed [36]. Based on this assumption, a 3D point $P = [X, Y, Z, 1]^{\mathsf{T}}$ is mapped to $p_p = [x, y, 1]^{\mathsf{T}}$ in the projector as:

$$p_p \sim \mathsf{K}_p \begin{pmatrix} \mathsf{R}_p & \mathbf{t}_p \end{pmatrix} P \tag{10.1}$$

where $\sim$ stands for equality up to scale between homogeneous coordinates. These 2D points p_p live in what we refer to by the "projector image".

The matrix R_p and the vector $\mathbf{t}_p$ represent the extrinsic parameters of the projector. The calibration matrix K_p is described by the sought internal parameters and is defined as follows:

$$\mathsf{K}_p = \begin{pmatrix} \rho f & 0 & u \\ 0 & f & v \\ 0 & 0 & 1 \end{pmatrix} \tag{10.2}$$

where f, ρ and (u, v) are respectively the focal length, the aspect ratio and the

principal point coordinates.

Consider a camera imaging what is projected by the projector onto the wall. Since we assume the wall to be planar, it induces an homography $\mathsf{H}_{p\to c}$ between the projector and the camera image. Without loss of generality, we may assume that the world coordinate system is aligned with the wall, such that points on the wall have coordinates $Z = 0$. Then, the homography between projector and camera can be written as:

$$\mathsf{H}_{p\to c} \sim \underbrace{\mathsf{K}_c \begin{pmatrix} \bar{\mathsf{R}}_c & \mathbf{t}_c \end{pmatrix}}_{\mathsf{H}_{w\to c}} \underbrace{(\mathsf{K}_p \begin{pmatrix} \bar{\mathsf{R}}_p & \mathbf{t}_p \end{pmatrix})^{-1}}_{\mathsf{H}_{p\to w}} \tag{10.3}$$

where $\bar{\mathsf{A}}$ refers to the first two columns of a 3×3 matrix A. K_c is the camera's calibration matrix and R_c and $\mathbf{t}_c$ represent its extrinsic parameters. The homography $\mathsf{H}_{p\to c}$ can also be seen as the product of the homography $\mathsf{H}_{p\to w}$ that maps the projector image plane to the wall with $\mathsf{H}_{w\to c}$, the homography that relates the wall to the camera image plane.

10.3 Direct Linear Calibration

In this section, we review the details of the Direct Linear Calibration for projectors. This method is used as a reference for our experiments. As opposed to [55], the variant presented here [16] is strictly based on homographies and does not require a calibrated camera.

A planar calibration grid is attached to the wall. This allows to estimate the homography $\mathsf{H}_{w\to c}$ between the wall and the camera, introduced above. It relates a point p_w on the wall to a point p_c in the camera image as follows:

$$p_c \sim \mathsf{H}_{w\to c} p_w \tag{10.4}$$

Once this homography is computed (details on homography estimation can be found in [30]), the video projector is used to project patterns while it is moved to

various positions and orientations. For each projector pose i, correspondences are established between the camera and the video projector, leading to an homography $\mathsf{H}_{c\to p_i}$. A point p_c in the camera image is mapped into the projector at pose i as:

$$p_p^i \sim \mathsf{H}_{c\to p_i} p_c \tag{10.5}$$

Combining (10.4) and (10.5), a point p_w on the wall is mapped into the i^{th} projector as:

$$p_p^i \sim \underbrace{\mathsf{H}_{c\to p_i}\mathsf{H}_{w\to c}}_{\mathsf{H}_{w\to p_i}} p_w \tag{10.6}$$

We thus can compute the wall-to-projector homography for each pose i. It has the following form (see above):

$$\mathsf{H}_{w\to p_i} \sim \mathsf{K}_p \begin{pmatrix} \bar{\mathsf{R}}_p^i & \mathbf{t}_p^i \end{pmatrix} \tag{10.7}$$

It is now straightforward to apply classical plane-based calibration methods [66, 81] to calibrate the projector and, if necessary, to compute its extrinsic parameters, from two or more poses.

10.4 Projector Auto-Calibration

10.4.1 Basic Idea

The approach described in the previous section requires a calibration grid to be attached to the wall and, in the version of [55], the camera to be calibrated. In this section, we show that these requirements may be avoided and propose a true geometric video projector auto-calibration approach.

The key observation underlying the auto-calibration approach is as follows. It is "easy" to compute homographies between the projector image and the camera image, induced by the projection surface. There are indeed many possibilities to do so, the

simplest ones consisting in projecting a single pattern such as a checkerboard and extracting and identifying corners in the camera image. More involved ones could make use of multiple patterns, sequentially projected from each considered projector pose, such as Gray codes, allowing for robust and dense matching. From the obtained matches, the computation of the homography is straightforward.

Consider now homographies associated with two poses of the projector, $\mathsf{H}_{c\to p_i}$ and $\mathsf{H}_{c\to p_j}$. From these we can compute an homography between the two projector images, induced by the planar projection surface:

$$\begin{aligned}\mathsf{H}_{p_i\to p_j} &\sim \mathsf{H}_{w\to p_j}\mathsf{H}_{w\to p_i}^{-1}\\ &\sim \mathsf{H}_{c\to p_j}\mathsf{H}_{w\to c}\left(\mathsf{H}_{c\to p_i}\mathsf{H}_{w\to c}\right)^{-1}\\ &\sim \mathsf{H}_{c\to p_j}\mathsf{H}_{c\to p_i}^{-1}\end{aligned}$$

We are now in the exact same situation as an uncalibrated perspective camera taking images of an unknown planar scene: from point matches, the associated plane homographies can be computed and it is well-known that camera auto-calibration is possible from these, as first shown by Triggs [70]. We may thus apply any existing plane-based auto-calibration method, e.g. [70, 45, 26] to calibrate the projector. Compared to auto-calibration of cameras, the case of projectors has an advantage; many and highly accurate point matches can be obtained since the scene texture is controlled, by projecting adequate patterns onto the wall.

Plane-based auto-calibration comes down to a non-linear optimization problem, even in the simplest case when only the focal length is unknown. To avoid convergence problems, we adopt an approach suggested in [26] that requires to take one image in a roughly fronto-parallel position relative to the scene plane. Here, this means of course by analogy that the projector should once be positioned in a roughly fronto-parallel position relative to the wall; subsequent poses can (and should) then be different. This allows for a closed-form initial solution to the auto-calibration problem, which may then be refined by a non-linear optimization (bundle adjustment). Note that

the assumption of fronto-parallelism for one of the images is only required for the initialization; during optimization, this is then no longer enforced.

10.4.2 Initialization Procedure

We derive the initialization procedure in a different and simpler way compared to [26]. Let the fronto-parallel view correspond to pose 1; in the following we only consider homographies between that view and all the others. Consider first the wall-to-projector homography of the fronto-parallel view, $\mathsf{H}_{w\to p_1}$. So far, we have assumed that the world coordinate system is such that the wall is the plane $Z = 0$ (see section 10.2). Without loss of generality, we may assume that the X and Y axes are aligned with those of the fronto-parallel view and that the optical center of that view is located at a distance equal to 1 from the wall. Note that these assumptions are not required to obtain the below results, but they simply make the formulae simpler. With these assumptions, the wall-to-projector homography for the fronto-parallel pose is simply:

$$\mathsf{H}_{w\to p_1} \sim \mathsf{K}_p$$

Consider now the homography between the fronto-parallel view and another view j:

$$\begin{aligned}\mathsf{H}_{p_1\to p_j} &\sim \mathsf{H}_{w\to p_j}\mathsf{H}_{w\to p_1}^{-1} \\ &\sim \mathsf{K}_p \begin{pmatrix}\bar{\mathsf{R}}_p^j & \mathbf{t}_p^j\end{pmatrix} \mathsf{K}_p^{-1}\end{aligned}$$

In the following let us, for simplicity, drop all indices:

$$\mathsf{H} \sim \mathsf{K} \begin{pmatrix}\bar{\mathsf{R}} & \mathbf{t}\end{pmatrix} \mathsf{K}^{-1}$$

It follows that:

$$\mathsf{K}^{-1}\mathsf{H} \sim \begin{pmatrix}\bar{\mathsf{R}} & \mathbf{t}\end{pmatrix} \mathsf{K}^{-1}$$

Let us now multiple each side of the equation from the left with its own transpose:

$$\mathsf{H}^T\mathsf{K}^{-T}\mathsf{K}^{-1}\mathsf{H} \sim \mathsf{K}^{-T} \begin{pmatrix}\bar{\mathsf{R}} & \mathbf{t}\end{pmatrix}^T \begin{pmatrix}\bar{\mathsf{R}} & \mathbf{t}\end{pmatrix} \mathsf{K}^{-1}$$

Since $\bar{\mathsf{R}}$ consists of the first two columns of the rotation matrix R, we have $\bar{\mathsf{R}}^T\bar{\mathsf{R}} = \mathtt{I}$ and thus:

$$\mathsf{H}^T\mathsf{K}^{-T}\mathsf{K}^{-1}\mathsf{H} \sim \mathsf{K}^{-T}\begin{pmatrix} 1 & 0 & \times \\ 0 & 1 & \times \\ \times & \times & \times \end{pmatrix}\mathsf{K}^{-1}$$

where entries marked as $\times$ depend on $\mathbf{t}$ and are irrelevant for the following. Due to the form of K, this becomes:

$$\mathsf{H}^T\mathsf{K}^{-T}\mathsf{K}^{-1}\mathsf{H} \sim \begin{pmatrix} 1 & 0 & \times \\ 0 & \rho^2 & \times \\ \times & \times & \times \end{pmatrix} \tag{10.8}$$

Let us use the image of the absolute conic (IAC) to parameterize the projector's intrinsic parameters, defined as $\omega \sim \mathsf{K}^{-T}\mathsf{K}^{-1}$. From (10.8) we can now deduce the following two equations on the intrinsic parameters, which are similar to those of calibration based on a planar calibration grid [66, 81]:

$$h_1^T\omega h_2 = 0 \tag{10.9}$$

$$\rho^2 h_1^T\omega h_1 - h_2^T\omega h_2 = 0 \tag{10.10}$$

where h_k denotes the kth column of H. Let us note that $\rho^2 = \omega_{11}/\omega_{22}$; hence, equation (10.10) can be written:

$$\omega_{11}h_1^T\omega h_1 - \omega_{22}h_2^T\omega h_2 = 0 \tag{10.11}$$

Equation (10.9) is linear in ω, whereas (10.11) is quadratic. There are different ways of using these equations to compute the IAC ω and from this, the intrinsic parameters. If the aspect ratio ρ is known beforehand, both equations are linear and thus easy to solve. If ρ is unknown, one can either use only the linear equation (10.9), which requires five views (the fronto-parallel one and four others), or compute ω from three views only. In the latter case, we have two linear and two quadratic equations and a "closed-form" solution in the form of a degree-4 polynomial in one of the unknowns, is straightforward to obtain.

10.4.3 Non-linear Optimization

Once an initial solution of the projector calibration is computed using the above approach, a non-linear optimization through bundle adjustment may be carried out. Let us briefly outline its peculiarities, compared to plane-based auto-calibration of a camera. Note that the only noisy observations in our scenario are features in the camera image: those in the projector "images" are perfectly known and noisefree! Hence, the cost function of the bundle adjustment should be based on the reprojection error in the camera image. The following formulation is one possible option:

$$\min_{\mathsf{H}_{w\to c},\mathsf{K}_p,\mathsf{R}_p^i,\mathbf{t}_p^i} \sum_{i,j} dist^2\left(p_c^{ij}, \mathsf{H}_{w\to c}\mathsf{H}_{p_i\to w}p_p^{ij}\right)$$

where i stands for projector poses and j for points. I.e. we optimize the wall-to-camera homography, the intrinsic projector parameters and its extrinsic parameters for all views, by minimizing the reprojection error when mapping from the projector images into the camera image (the $\mathsf{H}_{p_i\to w}$ are parameterized by K_p and the extrinsic projector parameters).

Another option would be to include camera intrinsics and extrinsics in the optimization instead of the "black-box" homography $\mathsf{H}_{w\to c}$, but since the camera is static in our case, at most two intrinsics can be estimated [66, 81].

Let us briefly describe the gauge freedom in our problem. Everything is defined up to a 3D similarity transformation, i.e. 7 degrees of freedom (rotation, translation, and scale). We fix 3 of those by letting the projector screen be the plane $Z = 0$. We may fix 3 others by imposing an arbitrary position for one of the projector images. The remaining degree of freedom corresponds to rotation about the normal of the projector screen. This may be fixed by imposing e.g. an X-coordinate of the position of a second projector image.

Overall, for n projector images, we thus have $8 + m + 6n - 4$ parameters to optimize, where m is the number of estimated projector intrinsics (usually, 3) and the 8 correspond to the coefficients of the wall-to-camera homography.

In our implementation, we use the Levenberg-Marquardt method for the optimization and make use, as is common practice, of the sparsity of the problem's normal equations. At each iteration, solving the normal equations comes down to inverting 6×6 symmetric matrices (blocks corresponding to extrinsic parameters of individual projector images), and inverting one 11×11 symmetric matrix (a block corresponding to homography and intrinsic parameters). The whole bundle adjustment takes far less than a second on a standard PC.

10.4.4 Estimation of Focal Length Changes

The above paragraphs constitute our auto-calibration approach. Here, we describe another method that allows to estimate the change of the projector's intrinsics caused by zooming. If the projector has been calibrated beforehand, this allows to update its calibration. We suppose that a zoom causes, besides the focal length, also the principal point to change (especially its vertical coordinates is likely to change in practice), but that the aspect ratio ρ remains constant.

We also suppose here that both the camera and the projector remain static. Let H be the projector-to-camera homography before zooming and H' the one afterwards. The inter-image homography between the two projector images is then given by:

$$\begin{aligned} \mathsf{M} &\sim (\mathsf{H}')^{-1}\,\mathsf{H} \\ &\sim \mathsf{K}'_p\,(\mathsf{K}_p)^{-1} \\ &\sim \begin{pmatrix} f' & 0 & u'f - uf' \\ 0 & f' & v'f - vf' \\ 0 & 0 & f \end{pmatrix} \end{aligned}$$

It is straightforward to compute the intrinsic parameters after zooming:

$$\begin{aligned} f' &= \frac{M_{11}}{M_{33}} f \\ u' &= \frac{M_{13} + uM_{11}}{M_{33}} \\ v' &= \frac{M_{23} + vM_{11}}{M_{33}} \end{aligned}$$

Note that M depends only on the three unknown intrinsic in K'_p and can thus be computed from two points matches already. If the principal point can be assumed to remain constant, a single match is sufficient. A single match is also sufficient if only one coordinate of the principal point is supposed to change due to zooming (which is often the case for video projectors).

10.5 Experiments

The proposed algorithm has been tested on synthetic and real data. Both tests are detailed in the next two subsections.

10.5.1 Synthetic Data

We performed several tests of our algorithm using synthetic data to assess its sensitivity to noise, number of projector poses and fronto-parallelism inaccuracy. Throughout all the synthetic experiments, we used a camera panned at 30 degrees w.r.t the projection surface. The camera resolution was set to 1000×1000 and its calibration matrix defined as:

$$\mathsf{K}_c = \begin{pmatrix} 1000 & 0 & 500 \\ 0 & 1000 & 500 \\ 0 & 0 & 1 \end{pmatrix} \tag{10.12}$$

The projector parameters are identical to the camera parameters.

Sensitivity to noise level. For this test, we used 20 inter-image homographies computed by orienting the projector at random. The range of the orienta-

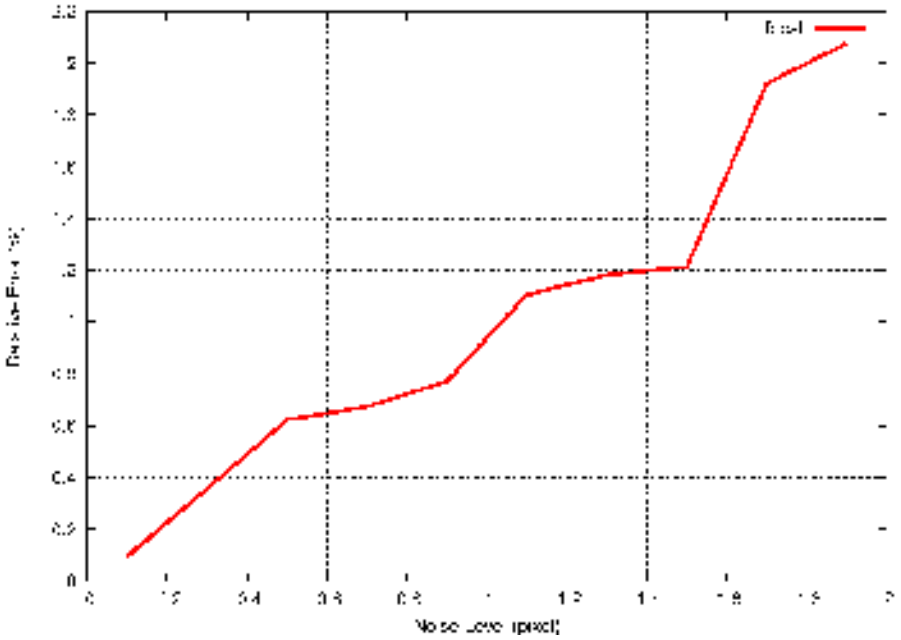

Figure 10.2. Focal length error vs. noise level

tions was ±20 degrees w.r.t the projection surface. Projector points were then imaged by the camera, and a gaussian noise with mean 0 and increasing standard deviation was added to the image points. The standard deviation σ varied from 0.1 to 1.5. As in [81], we performed 100 independent runs for each noise level and computed the average errors for both the focal length and the principal point. As we can see from Fig. 10.2 and Fig. 10.3 the error increases almost linearly for both the focal length and the principal point. For a noise level of $\sigma = 0.5$ the error in the focal length is about 0.6% and the error in the coordinates of the principal point is less than 3 pixels which represents, or less than 0.7% relative error.

Sensitivity to the number of projector poses. We set the amount of noise to $\sigma = 1$ and we varied the number of projector poses from 2 to 20 in a range of ±20 degrees w.r.t the projection surface. The average errors (from 100 independent

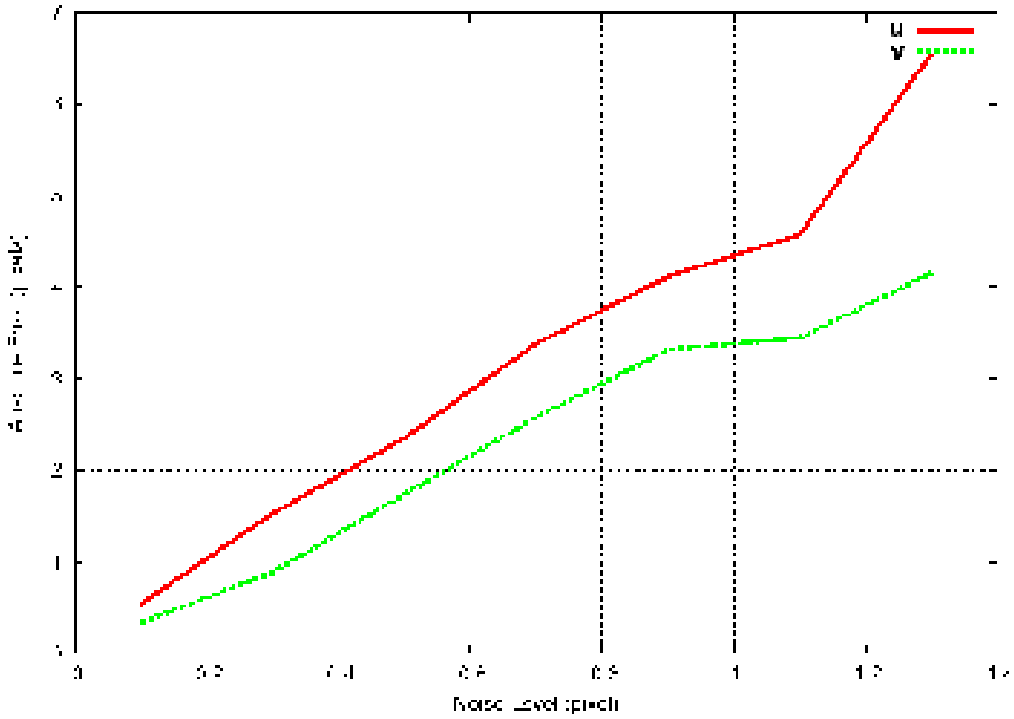

Figure 10.3. Principal point error vs. noise level

runs) for both the focal length and the principal point are reported in Fig. 10.4 and Fig. 10.5 . We notice that, as may be expected, the results gain stability when the number of projector poses is increased.

Sensitivity to fronto-parallelism inaccuracy. We conclude these synthetic experiments by assessing the sensitivity of our algorithm to the fronto-parallelism assumed in one of the images. The standard deviation of the noise added to the point coordinates was 0.5. We altered the orientation of the projector fronto-parallel to the projection surface. The resulting errors on the focal length and the principal point are reported in Fig. 10.6 and Fig. 10.7

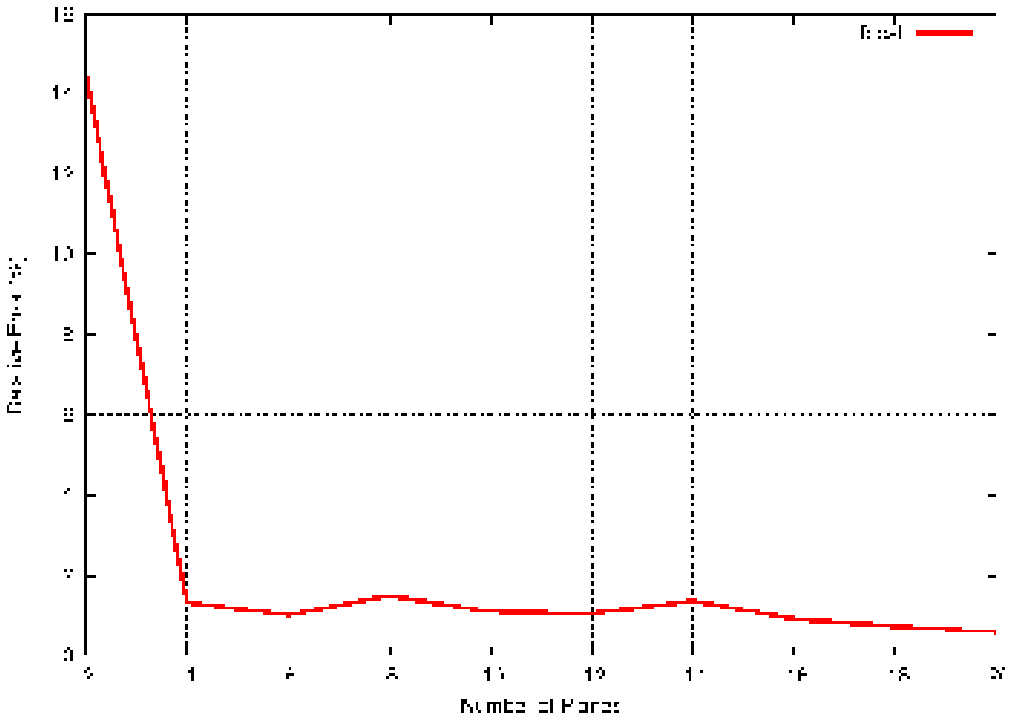

Figure 10.4. Focal length error vs. nb poses ($\sigma = 1$).

10.5.2 Real Images

We tested our algorithm on a Mitsubishi Pocket Projector and compared it to our variant of the DLC method, described in section 10.3. The projector has a native resolution of 800×600 and a fixed focal length. The acquisition device was a Nikon D50 camera. A $50mm$ lens was used on the camera and the resolution was set to 1500×1000.

We acquired 20 images of projected patterns while the projector underwent several orientations. Some images of the projected chessboard along with detected features are depicted on Figure.10.8.

We calibrated the projector with the proposed method and with our implementation of the DLC. The result of this benchmark is outlined in Table 10.1.

The table provides the estimated parameters and the reprojection error in pixels.

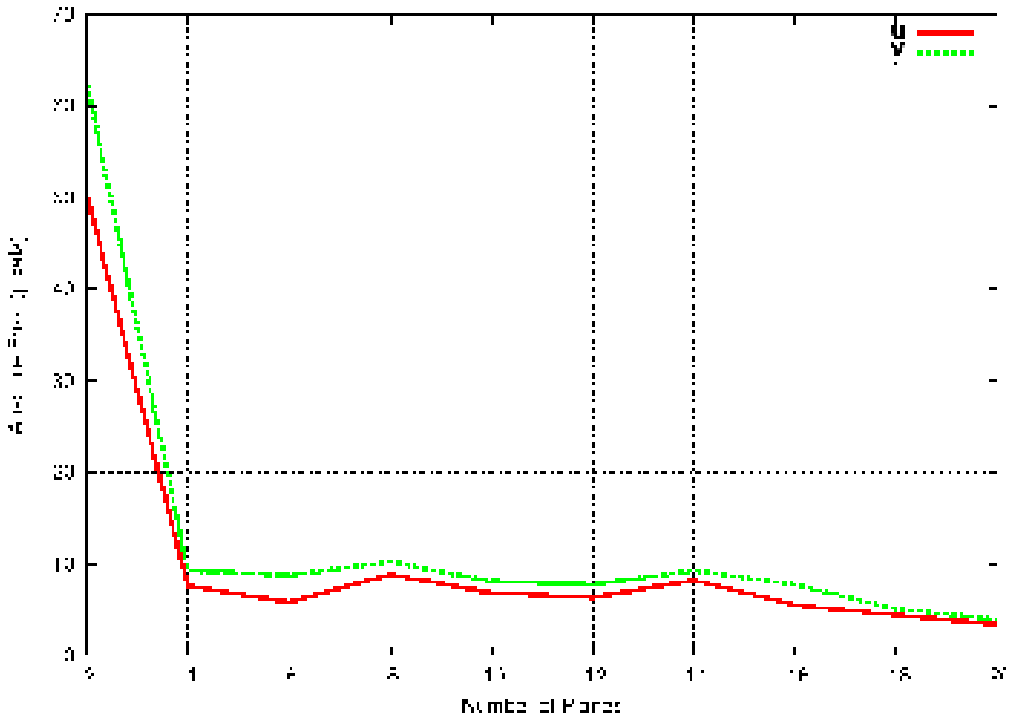

Figure 10.5. Principal point errors vs. nb poses ($\sigma = 1$).

Because our method was initialized with several fronto-parallel images we reported the range of reprojection error instead of an error average.

Table 10.1. Projector calibration benchmark: Direct method and the proposed Auto-Calibration method.

Method	$\mathbf{f_{proj}}$	ρ	u	v	Error
DLC	1320.13	1.002	382.1	448	0.46
Auto-Calib	1312.27	1.007	370.28	466	$0.42 - 0.27$

We performed a second calibration test on a video projector (Mitsubishi XD430U) with a zooming capability and a native resolution of 1024×768. For this test, we estimated the intrinsic parameters with two different zoom settings and the results were compared to the predictions obtained using the method introduced in section

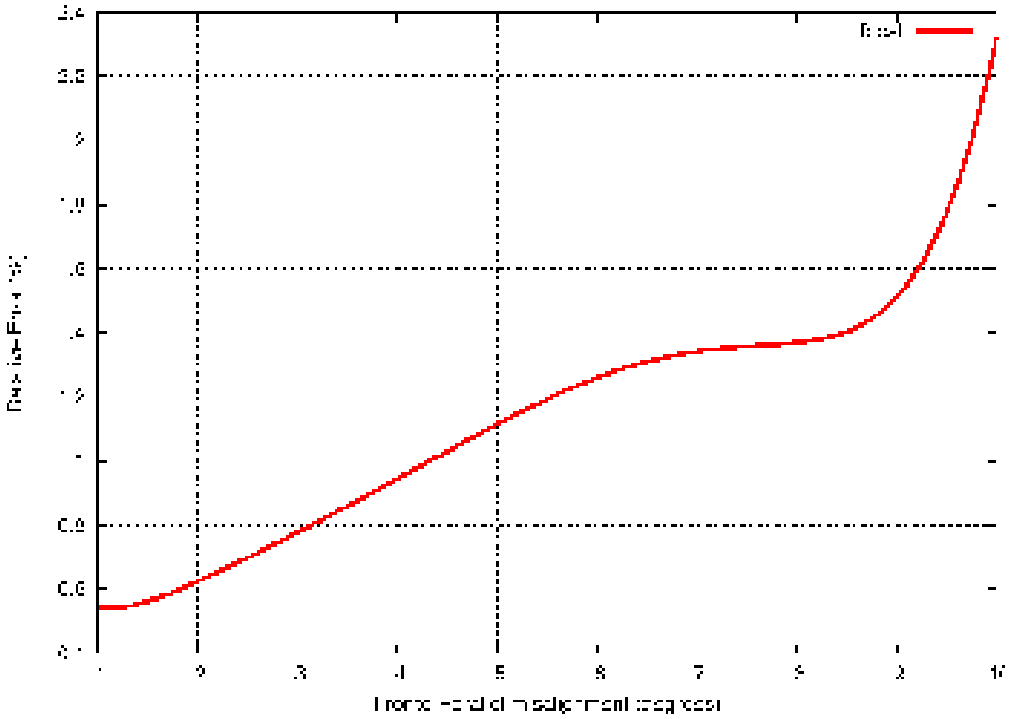

Figure 10.6. Focal length error vs. fronto-parallel misalignment.

10.4.4.

We observed that both methods are consistent as reported in Table 10.2.

Table 10.2. Calibration results with varying parameters.

Method	$\mathbf{f_{proj}}$	ρ	u	v
Zoom 1	2292.29	1.045	584.42	969.36
Zoom 2 (pred)	1885.7	1.045	587.64	949.55
Zoom 2 (est)	1873.14	1.045	590.9	944

10.6 Conclusion

In this paper we presented a new video projector auto-calibration method. It does not require a physical calibration grid or other metric information on the scene. Also, the

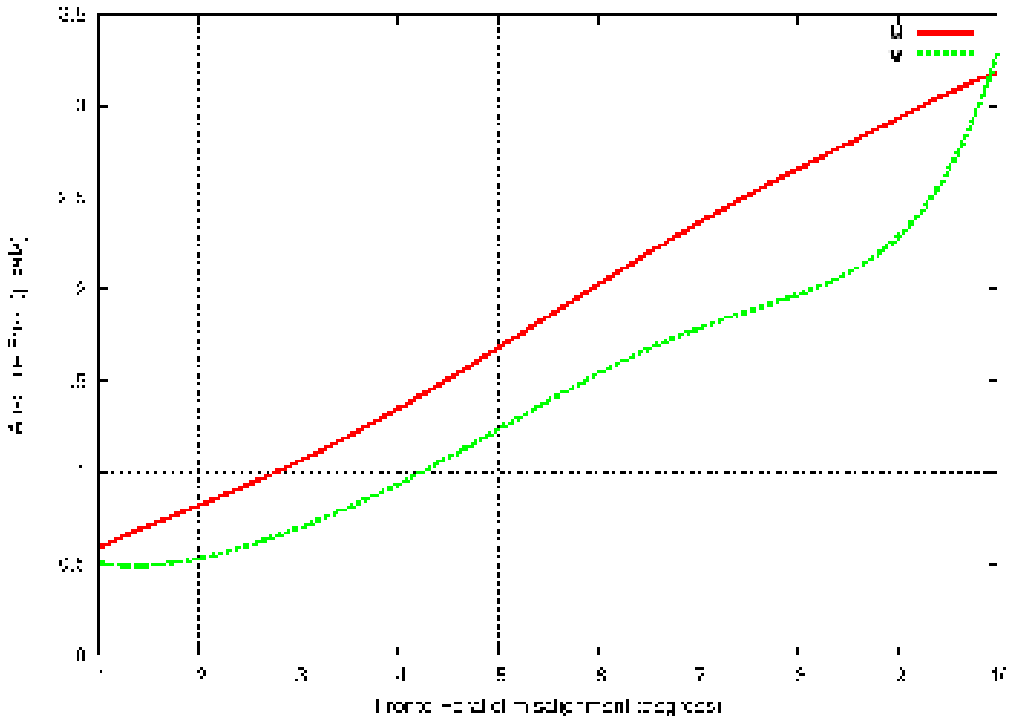

Figure 10.7. Principal point error vs. fronto-parallel misalignment.

camera used together with the projector, does not need to be calibrated; it is indeed merely used to get plane homographies between "images" of the projector associated with different poses. To the best of our knowledge, there are no other techniques that can work with the same input.

We believe that this aspect of our method increases its stability, otherwise the error of the camera calibration would affect the accuracy of the projector calibration [55]. Of course, as usual with auto-calibration methods, a certain number of poses, and especially a sufficient variety of poses (especially orientation), are required to get good results. In our synthetic experiments, results are very good with 4 poses or more.

Very simple to implement, the proposed method is fast, gives good results and is completely linear if one uses common assumptions regarding the projector aspect

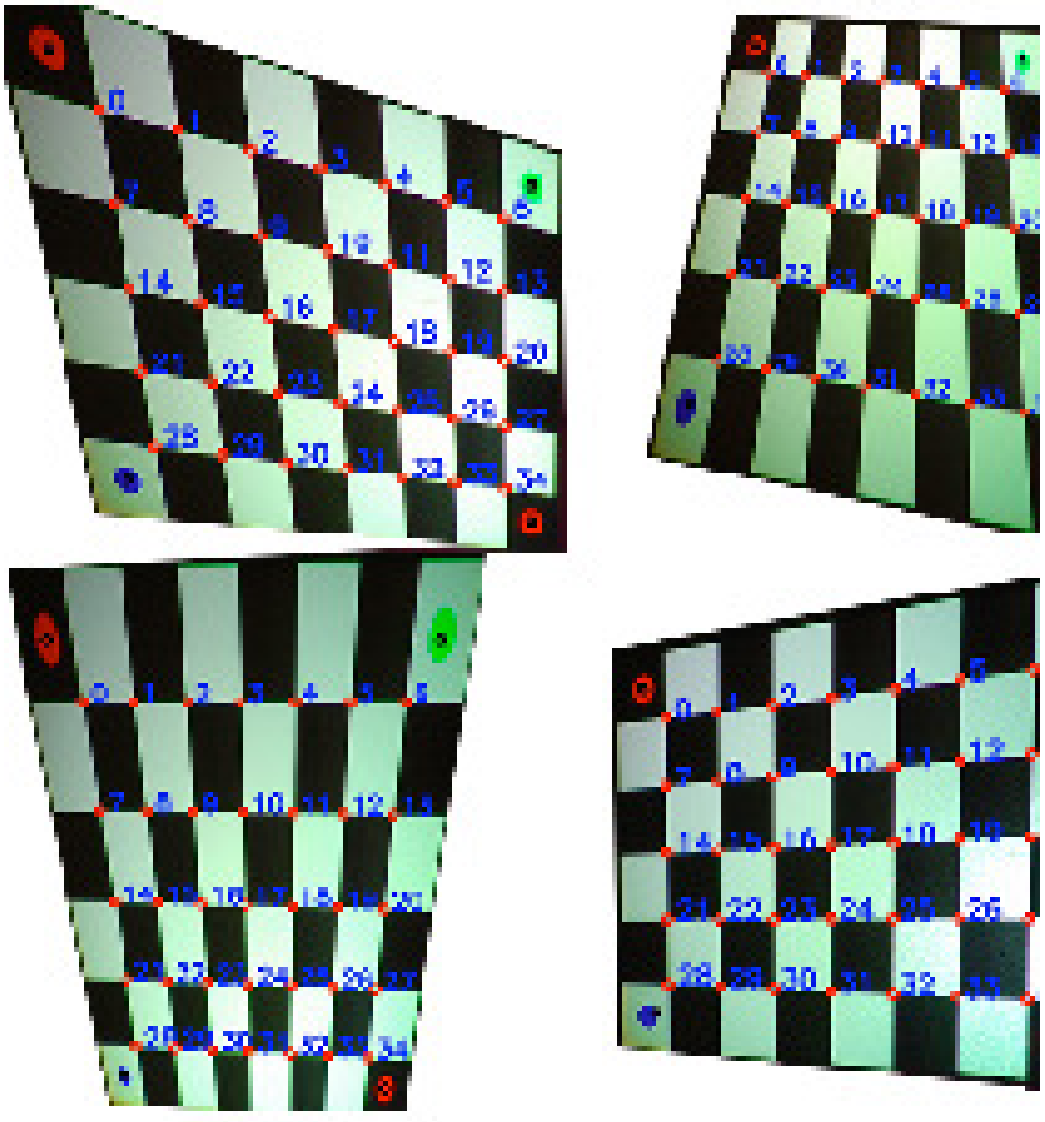

Figure 10.8. Images of projected patterns and detected features. The numbers and small red dots are added for illustration only. The large dots in the 4 corners are part of the projected pattern.

ratio. In the near future we will implement and test the bundle adjustment procedure outlined in the paper. This is straightforward and is expected to further improve our results.

More generally, we believe that our method will enable to handle large projector-camera systems that were previously impossible to calibrate due to cumbersome calibration chessboards required by previous methods.

Chapitre 11

GÉOMÉTRIE ÉPIPOLAIRE: CAMÉRA ET LUMIÈRE PONCTUELLE

Ce chapitre présente les fondements de la géométrie épipolaire entre deux caméras et par extension entre deux lumières ponctuelles. Ce dernier aspect est essentiel à la compréhension de la reconstruction 3D basée sur les ombres projetées.

11.1 Géométrie épipolaire de caméras

La géométrie épipolaire décrit la relation entre deux images, prises de points de vue différents, d'une même scène. Algébriquement, elle est entièrement représentée par une transformation 3×3 qu'on appelle **matrice essentielle** dans le cas d'une caméra calibrée ou **matrice fondamentale**, sinon. Dans ce qui suit, nous allons présenter les fondements de la géométrie épipolaire. Le lecteur pourra se référer à la figure 11.1 comme complément à nos explications.

Soient $\mathbf{Q}$ un point dans l'espace 3D et $\mathbf{q}_{1,2}$ ses projections dans les caméras situées en $\mathbf{C}_1$ et $\mathbf{C}_2$. Les points $\mathbf{Q}$, $\mathbf{C}_1$ et $\mathbf{C}_2$ forment un plan qu'on appelle **plan épipolaire** de $\mathbf{Q}$. L'intersection du plan épipolaire avec les plans image donne deux droites, appelées **droites épipolaires**. Le point d'intersection d'un faisceau de droites épipolaires dans une image est appelé **épipole**. Il représente la projection du point focal d'une caméra du point de vu de l'autre caméra. On notera $\mathbf{e}_{ij}$ le centre optique de la j^{e} caméra vue dans l'image de la caméra i. Les épipoles jouent un rôle majeur dans l'estimation de l'orientation relative des caméras. Dans le cas où les 2 caméras sont fronto-parallèles, les droites épipolaires sont parallèles ; Les épipoles se situent donc à l'infini.

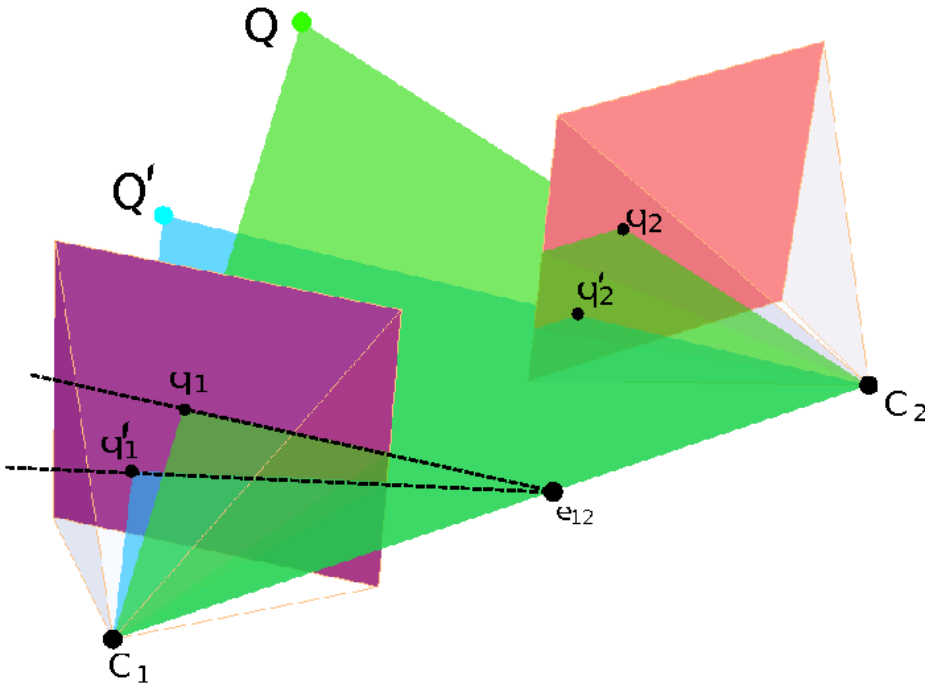

Figure 11.1. Géométrie épipolaire de caméras.

On remarque que $\mathbf{q}_1$ et $\mathbf{q}_2$, les projections de $\mathbf{Q}$ dans chacune des caméras, appartiennent à leurs droites épipolaires respectives. On voit vite l'intérêt de la contrainte épipolaire pour l'appariement de primitives dans deux images : pour un point donné dans une image, la recherche de sa projection dans l'autre image se fait sur une ligne au lieu de l'image au complet.

Les matrices Fondamentales et Essentielles

La relation entre des droites épipolaires correspondantes est algébriquement représentée par une matrice 3×3. Pour un système non-calibré, elle se nomme **matrice fondamentale** et opère en coordonnées pixels. Elle a été introduite indépendamment par Faugeras [21] et Hartley [31].

Ainsi, si l'on note par F_{12} la matrice fondamentale reliant les vues 1 et 2, la relation

entre les points pixels $\mathbf{q}_1$ et $\mathbf{q}_2$ s'exprime :

$$\mathbf{q}_2^{\mathsf{T}} \mathsf{F}_{12} \mathbf{q}_1 = 0$$

Pour mieux comprendre cette équation, il faut remarquer que $\mathbf{l}_2 = \mathsf{F}_{12}\mathbf{q}_1$ est l'équation d'une droite, $\mathbf{l}_2$, qui n'est rien d'autre que la droite épipolaire correspondante de $\mathbf{q}_1$ dans la vue 2. L'appartenance de $\mathbf{q}_2$ à $\mathbf{l}_2$ se traduit tout simplement par $\mathbf{q}_2^{\mathsf{T}}\mathbf{l}_2 = 0$, d'où, $\mathbf{q}_2^{\mathsf{T}}\mathsf{F}_{12}\mathbf{q}_1 = 0$.

La contrainte épipolaire inverse, F_{21}, est donnée par la transposée de la matrice F_{12} :

$$\mathsf{F}_{21} = \mathsf{F}_{12}^{\mathsf{T}}$$

Les épipoles $\mathbf{e}_1$ et $\mathbf{e}_2$ sont donnés par les noyaux droit et gauche de la matrice F. Ceci traduit bien le fait que les épipoles appartiennent à toutes les droites épipolaires.

Lorsque les paramètres intrinsèques des caméras sont disponibles, la géométrie épipolaire est représentée par une matrice 3×3 qu'on nomme **matrice essentielle** et qui met en relation deux points dans le repère des caméras. La matrice essentielle possède moins de degrés de liberté et contient les informations sur la pose relative des deux caméras. La matrice essentielle E d'un système est reliée à la matrice fondamentale F du même système par :

$$\mathsf{E} = \mathsf{K}_2^{\mathsf{T}} \mathsf{F} \mathsf{K}_1$$

Où, K_1 et K_2 représentent les matrices des paramètres intrinsèques des caméras. On peut voir la matrice essentielle, proposée par Longuet-Higgins [41], comme une adaptation de la matrice fondamentale dans un cadre de coordonnées images normalisées.

11.2 Géométrie épipolaires et lumières ponctuelles

Nous allons à présent, présenter la relation qui existe entre deux sources de lumière ponctuelles et les ombres qu'elles projettent. Ces notions seront utiles à la recon-

struction 3D d'objets à partir de leurs ombres projetées. Mais avant toute chose définissons une source de lumière ponctuelle.

Définition: *Une source ponctuelle modélise une source lumineuse sans dimension. Elle émet des rayons dans toutes les directions de l'espace et n'est caractérisée que par sa position dans l'espace.*

Une source ponctuelle $\mathbf{l} = (u, v, w, 1)$ éclaire un objet $\mathcal{O}$ lequel, en obstruant les rayons lumineux , génère une ombre portée sur un plan Π [1]. Sans perte de généralité, le repère du monde est attaché à Π et est situé à $z = 0$.

L'ombre formée par $\mathcal{O}$, $\mathcal{S}^{\mathcal{O}}$, est appelée **shadowgram** [77]. Elle est la projection perspective du contour apparent de $\mathcal{O}$:

$$\mathcal{S}^{\mathcal{O}} = \mathsf{P}(\mathbf{l}) \cdot \mathcal{O}$$

Avec, $\mathsf{P}(\mathbf{l})$ la matrice de projection qui consiste en une transformation perspective dans un repère où $\mathbf{l}$ est l'origine :

$$\begin{aligned}\mathsf{P}(\mathbf{l}) &= \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{Eliminer\ le\ \ z} \cdot \underbrace{\begin{pmatrix} \mathsf{I}_{3\times 3} & \mathbf{l} \\ \mathbf{0}_3^{\mathsf{T}} & 1 \end{pmatrix}}_{translation\ vers\ \mathbf{l}} \cdot \underbrace{\begin{pmatrix} -w\mathsf{I}_{3\times 3} & \mathbf{0}_3 \\ (0,0,1) & 0 \end{pmatrix}}_{projection\ sur\ \Pi} \\ &= \begin{pmatrix} -w & 0 & u & 0 \\ 0 & -w & v & 0 \\ 0 & 0 & 1 & -w \end{pmatrix} \end{aligned} \tag{11.1}$$

À présent, considérons deux lumières ponctuelles $\mathbf{l}_{1,2}$ éclairant $\mathcal{O}$ ainsi que les ombres qui en résultent, $\mathcal{S}^{\mathcal{O}}_{1,2}$ (voir figure 11.2).

[1] En pratique, le plan Π est un écran placé devant une caméra qui observe les ombres.

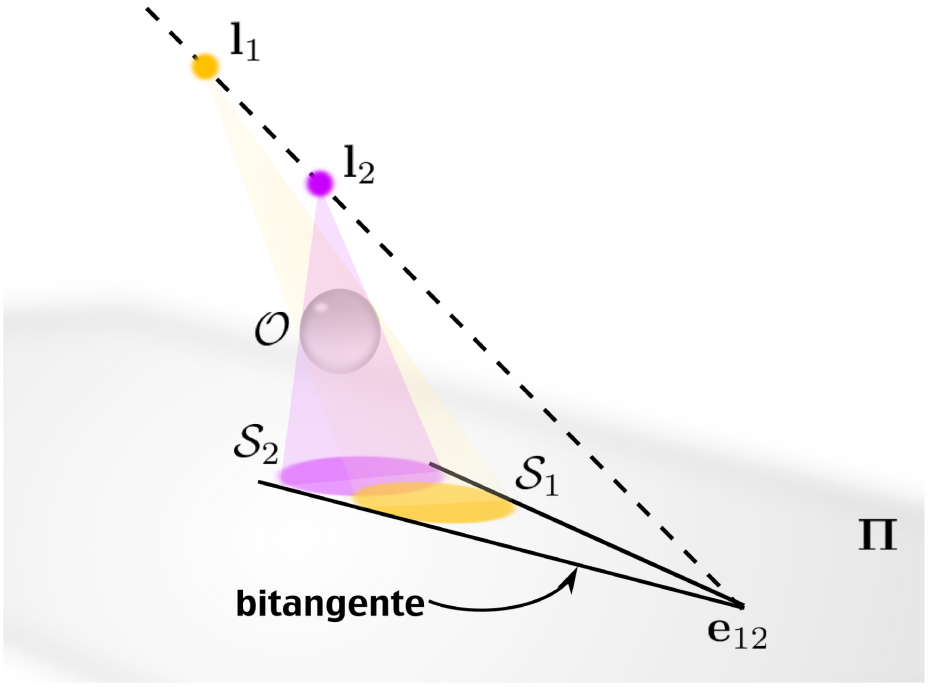

Figure 11.2. Géométrie épipolaire entre lumières. $\mathcal{S}_1$ **et** $\mathcal{S}_2$ **sont les ombres générées par les lumières** l_1 **et** l_2**.**

La droite qui relie $\mathbf{l}_1$ et $\mathbf{l}_2$ (*baseline*, en anglais) intersecte Π en un point $\mathbf{e}_{12}$: l'épipole. Tout plan qui passe par les sources $\mathbf{l}_{1,2}$ et $\mathcal{O}$ est un plan épipolaire. Cependant, une seule sorte de plan épipolaire nous est utile. Il s'agit de plans qui touchent $\mathcal{O}$ en un seul point. Ce sont en fait des plans tangents dont l'intersection avec Π donne une ligne épipolaire appelée aussi **bitangente épipolaire**[2].

Tel qu'illustré à la figure 11.2, les droites épipolaires se déduisent des ombres observées car par définition, ce sont les droites bitangentes à $\mathcal{S}_1^{\mathcal{O}}$ et $\mathcal{S}_2^{\mathcal{O}}$. Les deux droites épipolaires se croisent à l'épipole $\mathbf{e}_{1,2}$.

[2] Une bitangente est une droite qui touche deux courbes en deux points distincts.

Les éléments que nous avons présentés, permettent de reconstruire des objets en 3D à partir d'images de shadowgram [78]. Cependant, cette technique n'est qu'une variante de la reconstruction par silhouettes [39] et comme toute méthode de cette famille, elle souffre d'une ambiguité inhérente, appelée ambiguité de bas-relief [6].

Ainsi, pour une reconstruction 3D, $\tilde{\mathcal{O}}$, et un ensemble de lumières $\mathsf{P}(\mathbf{l}_i)$ compatibles avec les silhouettes $\mathcal{S}_i^{\mathcal{O}}$ observées, on peut écrire :

$$\mathcal{S}_i^{\mathcal{O}} = \mathsf{P}(\mathbf{l}_i) \cdot \tilde{\mathcal{O}}$$

Cependant, pour toute transformations projective A, $\mathsf{A}^{-1}\tilde{\mathcal{O}}$ et $\mathsf{P}(\mathbf{l}_i)\mathsf{A}$ sont aussi compatibles avec les observations, car :

$$\begin{aligned}\mathcal{S}_i^{\mathcal{O}} &= \mathsf{P}(\mathbf{l}_i) \cdot \tilde{\mathcal{O}} \\ &= \mathsf{P}(\mathbf{l}_i) \cdot (\mathsf{A}\mathsf{A}^{-1}) \cdot \tilde{\mathcal{O}} \\ &= (\mathsf{P}(\mathbf{l}_i)\mathsf{A}) \cdot (\mathsf{A}^{-1}\tilde{\mathcal{O}})\end{aligned}$$

Fort heureusement, dans le cadre des shadowgrams, la matrice A a une forme spécifique à 4 degrés de libertés [78] :

$$\mathsf{A} = \begin{pmatrix} 1 & 0 & a_1 & 0 \\ 0 & 1 & a_2 & 0 \\ 0 & 0 & a_3 & 0 \\ 0 & 0 & a_4 & 1 \end{pmatrix}$$

Dans le prochain chapitre, nous démontrons que cette ambiguité se réduit à un seul paramètre si les sources de lumière sont visibles dans l'image de la caméra. En pratique, les sources apparaissent sous la forme de spots brillants.

Chapitre 12

BAS-RELIEF AMBIGUITY REDUCTION IN SHAPE FROM SHADOWGRAMS

Cet article [15] a été publié comme l'indique la référence bibliographique

J. Draréni, S. Roy et P. Sturm. Bas-relief ambiguity reduction in shape from shadowgrams. Dans *3DPVT, Paris, May 2010.*

Abstract

Coplanar shadowgrams provide an affordable mean to retrieve the 3d shape of an object especially when classical stereopsis fails (eg:textureless objects). Its principles are similar to the concepts used for Shape-From-Silhouettes with the only exception that here, light sources and cameras are interchanged. However, it is well known that any attempt to use the shadowgram to retrieve light sources positions is subject to a 4-parameter ambiguity. In this paper, we show how using the light spot visible in the camera reduces this ambiguity to a single parameter. We also suggest some practical solutions to gain a supplemental constraint on light sources positions and break the ambiguity. We demonstrate the effectiveness of our method using synthetic and real images.

12.1 Introduction

Many cues have been used in computer vision in order to infer and understand the 3d shape of objects and visual scenes in general. The resulting methods are quoted as *Shape-From-X*, where X refers to the main cue used for the reconstruction pro-

cess. Stereoscopic disparity, apparent contour, shading, motion and shadows are some examples.

Shape from shadows received a great attention in the computer vision community, this is not surprising because valuable information on objects can be revealed from their cast shadows along with the corresponding light source. An other appealing aspect of the techniques based on shadows is that they do not rely on correspondences or a matching process like with classical stereopsis.

Early work on using shadows for structure retrieval dates back to Shafer and Kanade [61] who first established constraints on the orientations of the surfaces of interest in terms of observed cast shadows. Multiple light sources framework was proposed by [53] where a directional light moves in arc around the object of interest to acquire a sequence of planar cast shadows. The planar cast shadows are referred to as *planar shadowgrams*.

Later, Daum and Dudek [13] extended this framework to non single arc light trajectories. In [80], Yu *et al*. proposed a graph-based method to represent the constraints on the surfaces from light orientations. However, this method works on terrain-like surfaces lit by a directional source.

Following the success of space carving [38], Savarese *et al*. in the same vein proposed a shadow carving approach [57] where both silhouettes and shadows are used to infer the shape of an object. This is done by first building an occupancy volume from the silhouettes and carving away the regions that are not consistent with the observed shadows.

Recently, Shuntaro *et al*. [36] proposed a theory of shape from coplanar shadowgrams using a moving light source with no constraints on the trajectory of the light source. Once the positions of the light sources recovered, the convex hull of the object [5] is computed from the shadowgram using the classical shape from silhouette approach [39]. However, as proven by the authors themselves, when the shape of the object is unknown, the location of all the points sources can be recovered from

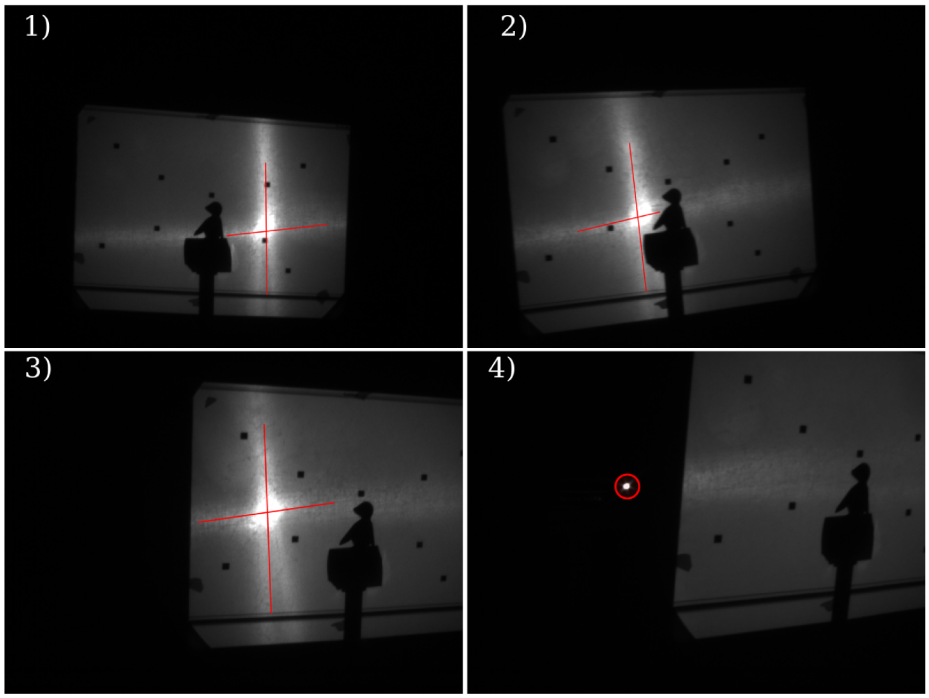

Figure 12.1. The projection of the light source is seen as a white spot through the screen. As the camera moves to the left (1,2,3), so the spot until the light source is directly visible (4).

the coplanar shadowgrams, only up to a four parameter perspective transformation related to the Generalized Perspective Bas-Relief ambiguity [6].

To break this ambiguity and to infer the location of the light sources, Shuntaro *et al*. use the shadowgrams of two additional spheres placed adjacent to the object of interest.

In this paper we propose a Shape-From-Silhouette (SFS) method based on shadowgrams. The proposed method exploits the projection of the light sources in the camera, visible through the translucent shadowgrams screen. We will show how the four parameter ambiguity described in [36] drops to a single parameter ambiguity by using the images of the light spots readily available in the image. In addition to

this theoretical result, the proposed method enjoys some interesting practical aspect by not relying on additional calibration objects, as their shadows may interact with the shadow of the object of interest. In deed, distinguishing a spot from a shadowgram silhouette can be done by a simple image thresholding even under sever camera distortions. We will also suggest some simple procedures that can be done at the acquisition time to constrain the remaining ambiguity and to fully solve the problem. We should point out that our method does not require the visibility of every spot corresponding to the light source we wish to estimate. Only two spots are needed and the rest are estimated using epipolar geomtry.

The rest of the paper is organized as follow. We outline the concept of epipolar geometry of shadowgrams in Section 12.2. Our main framework based on light triplets is presented in Section 12.3. We propose two simple methods to break the final ambiguity in Section 12.4. Experiments and conclusion are the subject of sections 12.5 and 12.6 respectively.

12.2 Shadowgrams and Epipolar Geometry

We assume the world's coordinate system attached to the shadowgram plane $\mathbf{\Pi}$ and the latter is located at $Z = 0$.

We represent the location of a light source $\mathbf{L}_i \in \mathbb{RP}^3$ in the global coordinate system by $\mathbf{L}_i = (X_i, Y_i, Z_i, 1)^\mathsf{T}$ and it's projection in the camera by $\mathbf{l}_i = (u_i, v_i, 1)$. The shadowgrams acquired by the camera are related to the shadowgrams on $\mathbf{\Pi}$ by a homography that remains fixed and can be estimated independently using standard computer vision techniques. Thus, we may assume the camera aligned with $\mathbf{\Pi}$ and express the projection of the light source $\mathbf{L}_i$ as:

$$\mathbf{l}_i \sim [\mathtt{I}_{3\times 3} | -\mathbf{t}]\, \mathbf{L}_i$$

Where $\mathbf{t}$ is the position of the camera's center of projection.

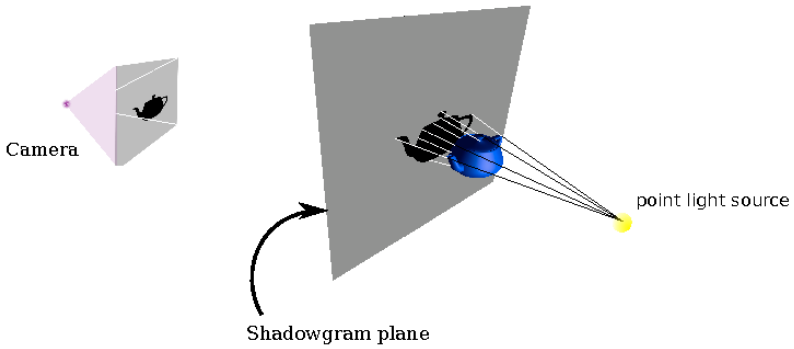

Figure 12.2. Setup to implement SFS. A point light source lit an object that in turn cast a shadow on a screen. A camera , placed on the other side of the screen, captures the shadowgram.

When two light sources $\mathbf{L}_i$ and $\mathbf{L}_j$ are considered, an epipolar geometry is defined akin to the classical binocular stereo configuration. In this case, the light sources are analogous to the centers of projection of the cameras and the line that joins them is the baseline. The intersection of the baseline with $\mathbf{\Pi}$ is the epipole $\mathbf{e}_{ij}$. The homogeneous coordinates of $\mathbf{e}_{ij}$ can be expressed in terms of $\mathbf{L}_{i,j}$ coordinates as:

$$\mathbf{e}_{ij} = \begin{pmatrix} X_j Z_i - X_i Z_j \\ Y_j Z_i - Y_i Z_j \\ 0 \\ Z_i - Z_j \end{pmatrix} \tag{12.1}$$

This result stems form the intersection of the line joining $\mathbf{L}_{i,j}$ and $\mathbf{\Pi}$. It is worth noting that here, we only have one epipole as opposed to the classic stereo setup.

The same epipole can be estimated using the observed shadowgrams casted by an object lit by $\mathbf{L}_i$ and $\mathbf{L}_j$. In fact, the epipole is estimated from the intersection of two bitangent lines to the shadowgrams as depicted in Fig.12.2.

In case the light sources are located at the same distance from $\mathbf{\Pi}$, the epipole is located at infinity, and so the bitagent intersection.

Using only the information from the shadowgrams, any attempt to extract light positions is subject to a 4-parameters Bas-Relief ambiguity as shown in [36]. In the next section, we will show that considering the relation between three light sources reduces the ambiguity up to a single parameter.

12.3 Three Light Source Relation

Because three light sources are always coplanar and for the sake of simplicity, let us consider light sources $\widetilde{\mathbf{L}}_\mathbf{i} = (0, Y_i, Z_i, 1)^\mathsf{T}$ that falls in the YZ-plane. From (12.1), the resulting epipoles $\tilde{\mathbf{e}}_\mathbf{ij}$ read off:

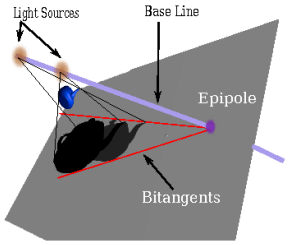

Figure 12.3. Shadowgrams from different light sources. The white spot is the projection of the spot.

$$\tilde{\mathbf{e}}_{\mathbf{ij}} = \begin{pmatrix} 0 \\ Y_j Z_i - Y_i Z_j \\ 0 \\ Z_i - Z_j \end{pmatrix} \tag{12.2}$$

It's easy to see that the three $\tilde{\mathbf{e}}_{\mathbf{ij}}$ are colinear on $\mathbf{\Pi}$ along an epipolar line $\phi(Y)$ defined as:

$$\phi(Y) : Y = Y_j Z_i - Y_i Z_j / (Z_i - Z_j)$$

Let $\mathbf{\Gamma}$ represents a 3D plane that passes by $\phi(Y)$ and whose normal makes an angle of α with the XZ-plane:

$$\mathbf{\Gamma} \sim \begin{pmatrix} \cos\alpha \\ 0 \\ \sin\alpha \\ 0 \end{pmatrix}$$

The observed spots from the camera image constrain the sought light source $\mathbf{Q}_i$ on a line:

$$\mathbf{Q}_i(\lambda) = \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix} + \lambda \left[\begin{pmatrix} \tilde{\mathbf{l}}_\mathbf{i} \\ 1 \end{pmatrix} - \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix} \right]$$

The light source Q_i must project into $\tilde{\mathbf{l}}_\mathbf{i}$ and also lies on the plane $\mathbf{\Gamma}$, which leads to the following constraint in terms of λ:

$$\mathbf{\Gamma}^\mathsf{T} \mathbf{Q}_i(\lambda) = 0 \tag{12.3}$$

Which yield the following value of λ:

$$\lambda = \frac{(\mathbf{t}_1 \cos\alpha + \mathbf{t}_3 \sin\alpha)}{(\mathbf{t}_1 \cos\alpha + (\mathbf{t}_3 - \mathbf{Z}_i)\sin\alpha)}$$

The line that joins two light sources $\mathbf{Q}_i$ and Q_j is expressed as $\gamma\mathbf{Q}_i + \rho\mathbf{Q}_j$ and the coordinate of the related epipole $\mathbf{e}'_{ij}$ must satisfy:

$$\mathbf{e}'_{ij} \sim \gamma\mathbf{Q}_i + \rho\mathbf{Q}_j = \begin{pmatrix} 0 \\ \cdots \\ 0 \\ \cdots \end{pmatrix} \tag{12.4}$$

By Zeroing the first and the third component, we force the epipole to be the projection of a point on the YZ-plane according to the assumptions we made on the light sources locations. In a general context, one must fix the appropriate constraints to ensure that the epipoles lie into the observed epipolar line.

Further, to satisfy the equation (12.4), the scalars γ and ρ read off:

$$\begin{pmatrix} \gamma \\ \rho \end{pmatrix} \sim \begin{pmatrix} Z_j \\ -Z_i \end{pmatrix}$$

Substituting the values of γ and ρ in (12.4) gives:

$$
\begin{aligned}
\mathbf{e}'_{ij} &\sim \begin{pmatrix} 0 \\ (\mathbf{Y}_i\mathbf{Z}_j - \mathbf{Y}_j\mathbf{Z}_i)(\mathbf{t}_3 \sin\alpha + \mathbf{t}_1 \cos\alpha) \\ 0 \\ (\mathbf{Z}_j - \mathbf{Z}_i)(\mathbf{t}_3 \sin\alpha + \mathbf{t}_1 \cos\alpha) \end{pmatrix} \\
&\sim \begin{pmatrix} 0 \\ \mathbf{Y}_i\mathbf{Z}_j - \mathbf{Y}_j\mathbf{Z}_i \\ 0 \\ \mathbf{Z}_j - \mathbf{Z}_i \end{pmatrix} \\
&\sim \mathbf{e}_{ij}
\end{aligned}
\tag{12.5}
$$

Thus, for each angle α we can compute an epipole e'_{ij} that verifies consistent with the epipolar geometry and the observed spots.

This ambiguity is materialized by a single parameter, namely the swiveling angle α as illustrated in Fig.4.

12.4 Solving the Ambiguity

In the previous section we showed that when the light spots are identified in the shadowgram images, the relationship between three light sources via their mutual epipolar geometry constrain the light positions up to a 1 parameter ambiguity. In order to break this ambiguity, an extra "information" on the light sources configuration is mandatory. Bare in mind, like it will be shown, that we do not need to observe every spot to infer the locations of all light sources; two spots are sufficient.

We propose two simple procedures that a user can perform while moving the light source in order to estimate the location of three light sources and alleviate the ambiguity of the whole system.

Moving the Camera. After acquiring the desired sequence, a user can move the camera while the light source remain static. It is then possible to triangulate the

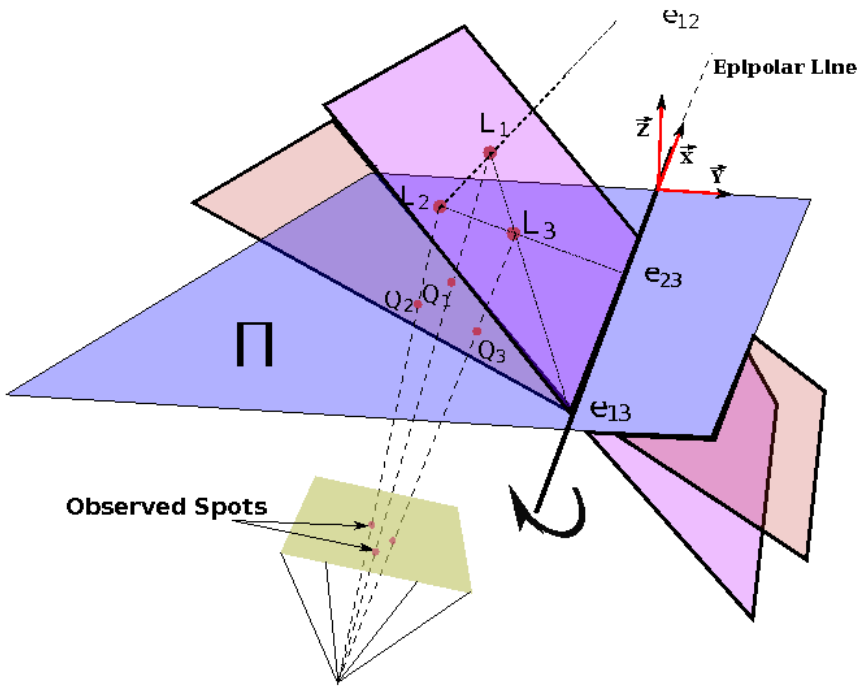

Figure 12.4. The 1-parameter ambiguity. When the deprojection plane swivel around the epipolar line, new light sources Q_1, Q_2 and Q_3 can be infered with the same properties as the real one.

last light position since the camera has a direct observation of the spot. Here, moving the camera can be avoided if one can afford using a second camera. Let us denote the reconstructed light source by L_1, we can see from Fig.12.3 that a light source L_3 can be reconstructed by intersecting the line $L_1 - e_{13}$ and the line that joins e_{23} and the spot of the light L_2. The remaining lights can be reconstructed the same way we did for L_3.

Baseline Normal to the Screen. If the user identifies a pair of light with a baseline orthogonal to the screen (pure front/back motion w.r.t the screen), an additional constraint on light locations is gained. In deed, a plane defined by these light and any other light is fully constrained.

12.5 Experiments

In this section we present our experiments involving synthetic and real images. Here we asses the presented method using the first solution to alleviate the ambiguity ; by moving the camera we fully reconstruct a light source and deduce the others using the epipolar geometry.

12.5.1 Synthetic Tests

Our synthetic test consist of 10 lights positioned randomly. We computed the 3D location of one light source by triangulating its spots observed in 2 virtual cameras. Along with another spot, we estimated the location of the remaining light sources using epipolar geometry. The sensitivity of the process was measured by adding zero mean gaussian noise to the virtual spots. The result of this experiment with different noise level is depicted in Fig.5 .

The reported error is the average distance of the reconstructed lights with the original one. We can see that the error grows linearly in terms of noise level.

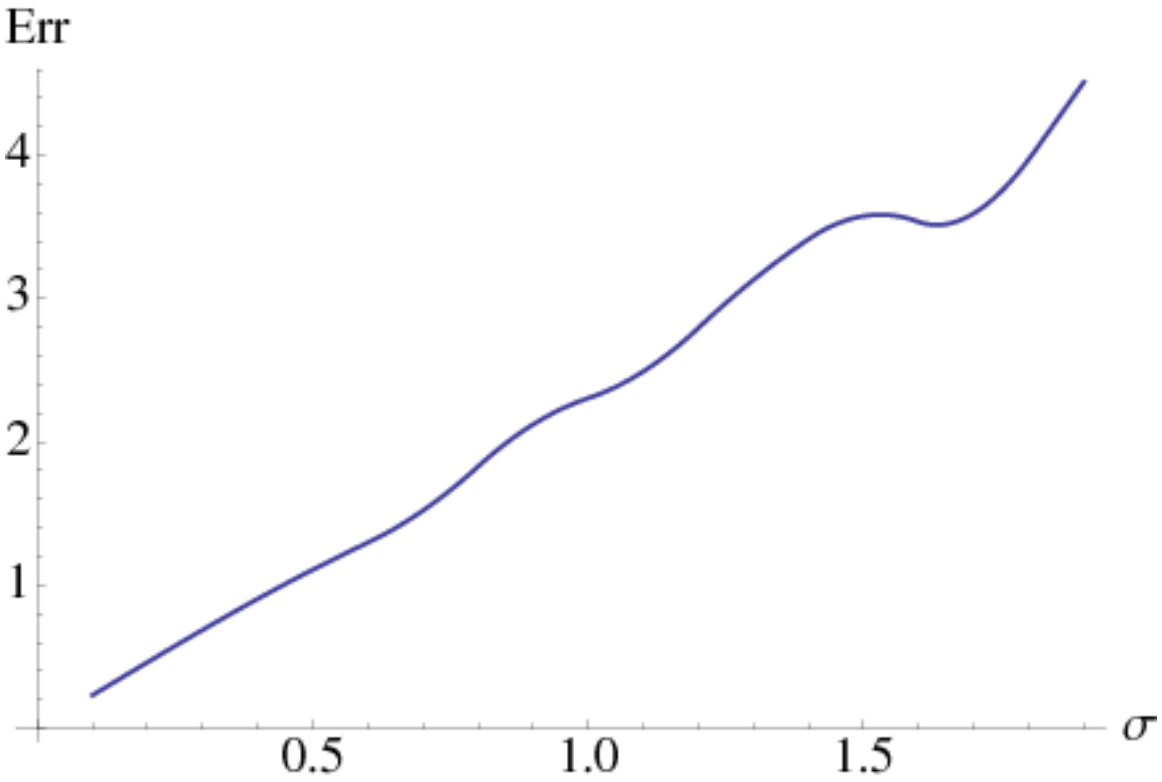

Figure 12.5. Sensitivity of the method in terms of noise level.

12.5.2 Real Tests

We tested our method on real shadowgrams of a penguin figurine. The figurine was lit by a moving projector and the resulting shadows were recorded by a video camera through a screen. The resolution of the camera is 640×480 pixels. At the end of the sequence the camera underwent a motion while the projector remained fixed. Hence, the 3D position of the last projector could be reconstructed by triangulating the observed spots using conventional structure from motion techniques.

Once this done, the remaining light sources were estimated one by one using the 3-light sources geometry presented in this paper combined with the epipolar geometry. The triplets consist of the fully reconstructed light source (the last one), a light source with a visible spot and the light source of interest.

The resulting 3D model using 10 shadowgrams is depicted in Fig.6 .

12.6 Conclusion

In this paper we presented a method to reconstruct objects from their shadowgrams. As opposed to previous works, the presented method does not use a calibration rig or an object that may interfere with the object of interest. Instead, we exploited the images of the light sources visible in the camera as bright spots easily identified. We also showed that using these spots alone can only reduce the ambiguity from 4 to 1 parameter. The later can be solved using simple procedures during the acquisition to yield a 3D model.

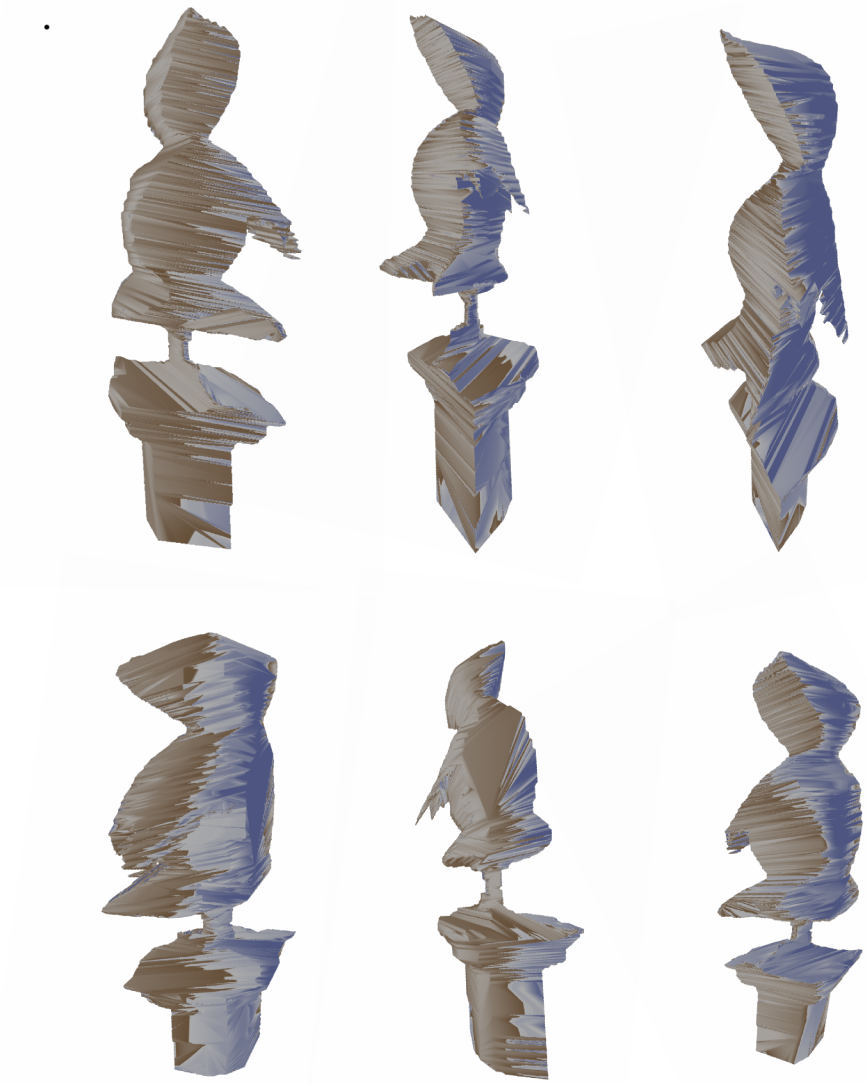

Figure 12.6. Snapshots of the reconstructed penguin using silhouettes from Fig.12.1

Chapitre 13

CONCLUSION

Cette thèse a présenté nos contributions pour divers domaines de la vision par ordinateur. Nous avons abordé ces problèmes d'un point de vue théorique et nous avons tenté d'apporter, le plus possible, des solutions mathématiquement élégantes tout en gardant à l'esprit leur faisabilité et leur applicabilité. Car, n'oublions pas que tel est le dilemme auquel fait face la vision par ordinateur ou toute autre science qui exploite des données réelles : savoir conjuguer avancée théorique et réalisation pratique.

Nous avons commencé par présenter le problème du suivi vidéo (*Video Tracking*) et de l'algorithme Mean-Shift. Nous avons abordé sa robustesse aux déformations géométriques ainsi que sa rapidité. Cependant, dans sa formulation originale, le suivi par Mean-Shift n'estime que les changements de positions d'un l'objet d'intérêt. Notre contribution a été d'augmenter le descripteur de l'objet de sorte à pouvoir estimer, en plus, les changements d'orientations.

Dans un deuxième volet, nous avons abordé le calibrage de camera linéaire et de vidéo projecteur. Ce sont là des dispositifs très utilisés en vision mais dont le calibrage était jusqu'ici fastidieux. Dans le cas des caméras linéaires, nous avons proposé un calibrage plan dont la commodité est indéniable. Le vidéo projecteur est un dispositif qui nécessite un calibrage aussi dès lors qu'il est utilisé comme instrument de mesure. La difficulté à laquelle on se heurte lors du calibrage d'un projecteur vient du fait qu'il est actif et ne "voit" pas la scène. L'usage d'une caméra est alors impératif. Nous avons proposé trois solutions à ce problème dont l'application dépend de l'information disponible sur la caméra (calibrée ou pas) et la scène (plan marqué, pose initiale connue).

Le troisième thème de cette thèse était consacré à la reconstruction 3D par ombres projetées. Nous avons présenté la relation géométrique entre deux sources de lumière ponctuelles et nous avons montré comment ces notions pouvaient servir à l'estimation de la structure tridimensionnelle. Cette technique partage les fondements et les limitations de la reconstruction par enveloppe visuelle, soit l'incontournable ambiguïté de bas-relief à 4 paramètres. Nous avons démontré que cette ambiguïté pouvait être réduite à un seul paramètre si les sources de lumière étaient visibles dans la caméra.

Finalement, rappelons que toutes ces contributions ont été publiées dans les actes de conférences internationales [18, 16, 17, 14, 15] et que deux d'entre elles le sont dans deux revues scientifiques [20, 19].

RÉFÉRENCES

[1] Elise Arnaud, Etienne Memin, et Bruno Cernuschi-Frias. Filtrage conditionnel pour le suivi de points dans des sequences d'images. Dans *Congres Francophone de Vision par Ordinateur, ORASIS*, may 2003.

[2] Elise Arnaud, Etienne Memin, et Bruno Cernuschi-Frias. Conditional filters for image sequence based tracking - application to point tracking. *IEEE Transactions on Image Processing*, 14(1), 2005.

[3] Svetlana Barsky et Maria Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1239–1252, 2003.

[4] Basler. Basler vision technologies, 2009.

[5] B.G Baumgard. *Geometric Modeling for Computer Vision.* PhD thesis, University of Stanford, 1974.

[6] Peter N. Belhumeur, David J. Kriegman, et Alan L. Yuille. The bas-relief ambiguity. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1060, 1997.

[7] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by probability distributions. *Bulletin of the Calcutta Math Society*, (35):99–109, 1943.

[8] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[9] R. T. Collins. Mean-shift blob tracking through scale space. Dans *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 234–240, 2003.

[10] D. Comaniciu, V. Ramesh, et P. Meer. Real-time tracking of non-rigid objects using mean shift. Dans *Conference on Computer Vision and Pattern Recognition*, pages 142–151, 2000.

[11] Dorin Comaniciu et Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[12] J. Crowley et K. Schwerdt. Robust tracking and compression for video communication. Dans *IEEE Computer Society Conference on Computer Vision, Workshop on Face and Gesture Recognition*, 1999.

[13] M. Daum et G. Dudek. On 3-d surface reconstruction using shape from shadows. Dans *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 461, Washington, DC, USA, 1998. IEEE Computer Society.

[14] J. Draréni, S. Roy, et P. Sturm. Geometric video projector auto-calibration. Dans *Proceedings of the IEEE International Workshop on Projector-Camera Systems*, Miami Beach, USA, June 2009.

[15] J. Draréni, S. Roy, et P. Sturm. Bas-relief ambiguity reduction in shape from shadowgrams. Dans *3DPVT*, Paris, France, May 2010.

[16] J. Draréni, Peter Sturm, et S. Roy. Projector calibration using a markerless plane. Dans *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, volume 2, pages 377–382, feb 2009.

[17] J. Draréni, P.F. Sturm, et S. Roy. Plane-based calibration for linear cameras. Dans *OMNIVIS'2008, the Eighth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, in conjunction with ECCV 2008, Marseille, France*, pages xx–yy. IEEE Computer Society, 2008.

[18] Jamil Draréni et Sébastien Roy. A simple oriented mean-shift algorithm for tracking. Dans Mohamed S. Kamel et Aurélio C. Campilho, editeurs, *ICIAR*, volume 4633 de *Lecture Notes in Computer Science*, pages 558–568. Springer, 2007.

[19] Jamil Draréni, Sébastien Roy, et Peter Sturm. Methods for geometrical video projector calibration. *Machine Vision and Applications, Journal of*, 2010.

[20] Jamil Draréni, Sébastien Roy, et Peter Sturm. Plane-based calibration for linear cameras. *International Journal of Computer Vision*, May 2010.

[21] Olivier D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. Dans *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 563–578, London, UK, 1992. Springer-Verlag.

[22] Olivier D. Faugeras, Quang-Tuan Luong, et Stephen J. Maybank. Camera self-calibration: Theory and experiments. Dans *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 321–334, London, UK, 1992. Springer-Verlag.

[23] K. Fukunaga et L. Hostetler. The estimation of the gradient of a density function with applications in pattern recognition. Dans *IEEE Transactions on Information Theory*, pages 32–40, 1975.

[24] Rajiv Gupta et Richard I. Hartley. Camera estimation for orbiting pushbrooms.

Dans *The Proc. of Second Asian Conference on Computer Vision*, volume 3, 1995.

[25] P. Gurdjos et R. Payrissat. Calibration of a moving camera using a planar pattern: A centre line-based approach for varying focal length. Dans *British Machine Vision Conference*, page Session 7: Geometry &. Structure, 2001.

[26] Pierre Gurdjos et Peter Sturm. Methods and geometry for plane-based self-calibration. Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 491–496. IEEE, 2003. Madison, Wisconsin.

[27] F. Scholten H. Hirschmüller et G. Hirzinger. Stereo vision based reconstruction of huge urban areas from an airborne pushbroom camera (hrsc). Dans Walter G. Kropatsch, Robert Sablatnig, et Allan Hanbury, editeurs, *DAGM-Symposium*, volume 3663 de *Lecture Notes in Computer Science*, pages 58–66. Springer, 2005.

[28] H. Yamamoto H. Ishiguro et S. Tsuji. Omni-directional stereo. *IEEE, Transactions on Pattern Analysis and Machine Intelligence*, pages 257–262, 1992.

[29] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997.

[30] Richard Hartley et Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, second édition, 2003.

[31] Richard I. Hartley. Estimation of relative camera positions for uncalibrated cameras. Dans Giulio Sandini, editeur, *ECCV*, volume 588 de *Lecture Notes in Computer Science*, pages 579–587. Springer, 1992.

[32] Richard I. Hartley. Self-calibration from multiple views with a rotating camera. Dans *ECCV '94: Proceedings of the third European conference on Computer*

vision (vol. 1), pages 471–478, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[33] Radu P. Horaud et Olivier Monga. *Vision par ordinateur: outils fondamentaux*. 'Editions Hermès, Paris, 1995.

[34] Berthold K. P. Horn. *Robot Vision (MIT Electrical Engineering and Computer Science)*. The MIT Press, mit press ed édition, March 1986.

[35] A. J. Izenman. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, 86(413):205–224, 1991.

[36] Makoto Kimura, Masaaki Mochimaru, et Takeo Kanade. Projector calibration using arbitrary planes and calibrated camera. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–2, 2007.

[37] Donald E. Knuth. *Art of Computer Programming, Volume 2: Seminumerical Algorithms (3rd Edition)*. Addison-Wesley Professional, November 1997.

[38] Kiriakos N. Kutulakos et Steven M. Seitz. A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3):199–218, 2000.

[39] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2):150–162, 1994.

[40] Johnny C. Lee, Paul H. Dietz, Dan Maynes-Aminzade, Ramesh Raskar, et Scott E. Hudson. Automatic projector calibration with embedded light sensors. Dans *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 123–126. ACM, 2004.

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

[42] M.I.A. Lourakis et A.A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Rapport Technique 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004. Available from `http://www.ics.forth.gr/~lourakis/sba`.

[43] D. Lowe. Distinctive image features from scale-invariant keypoints. Dans *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[44] Yi Ma, Stefano Soatto, Jana Kosecka, et S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[45] E. Malis et R. Cipolla. Camera self-calibration from unknown planar structures enforcing the multi-view constraints between collineations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 4(9), 2002.

[46] Norman Badler Min-Zhi Shao. Spherical sampling by archimedes' theorem. Technical Report 184, University of Pennsylvania, jan 1996.

[47] Katja Nummiaro, Esther Koller-Meier, et Luc J. Van Gool. An adaptive color-based particle filter. *Image Vision Comput.*, 21(1):99–110, 2003.

[48] G. Toscani O. Faugeras. Camera calibration for 3d computer vision. Dans *Int. Workshop on Machine Vision and Machine Intelligence*, pages 240–247, 1987.

[49] T. Okatani et K. Deguchi. Autocalibration of a projector-camera system. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1845–1855, Dec. 2005.

[50] Jean-Nicolas Ouellet, Félix Rochette, et Patrick Hébert. Geometric calibration of

a structured light system using circular control points. Dans *3D Data Processing, Visualization and Transmission*, pages 183–190, 2008.

[51] A. Raij et M. Pollefeys. Auto-calibration of multi-projector display walls. Dans *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 14–17 Vol.1, Aug. 2004.

[52] Richard I. Hartley Rajiv Gupta. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):963–975, Sept 1997.

[53] D. Raviv, Y.H. Pao, et K.A. Loparo. Reconstruction of three-dimensional surfaces from two-dimensional binary images. *Robotics and Automation*, 5(5):701–710, May 1989.

[54] C. Ricolfe Viala et A.J. Sanchez Salmeron. Improving accuracy and confidence interval of camera parameters estimated with a planar pattern. Dans *International Conference on Image Processing*, pages II: 1142–1145, 2005.

[55] Filip Sadlo, Tim Weyrich, Ronald Peikert, et Markus Gross. A practical structured light acquisition system for point-based geometry and texture. Dans *Proceedings of the Eurographics Symposium on Point-Based Graphics*, pages 89–98, 2005.

[56] J. Salvi, J. Pagés, et J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, April 2004.

[57] Silvio Savarese, Marco Andreetto, Holly Rushmeier, Fausto Bernardini, et Pietro Perona. 3d reconstruction by shadow carving: Theory and practical evaluation. *Int. J. Comput. Vision*, 71(3):305–336, 2007.

[58] D. Scott. On optimal and data-based histograms. *Biometrika*, 3(66):605–610, 1979.

[59] J. Segen et S. G. Pingali. A camera-based system for tracking people in real time. Dans *International Conference on Pattern Recognition*, page 63, 1996.

[60] Steven M. Seitz et Jiwon Kim. The space of all stereo images. *International Journal of Computer Vision*, 48(1):21–38, 2002.

[61] Steven Shafer et Takeo Kanade. Using shadows in finding surface orientations. *Computer Vision, Graphics, and Image Processing*, 22:145–176, 1983.

[62] T.S Shen et C.H Meng. Digital projector calibration for 3-d active vision systems. *Journal of Manufacturing Science and Engineering*, 124(1):126–134, February 2002.

[63] H. Y. Shum et R. Szeliski. Stereo reconstruction from multiperspective panoramas. Dans *International Conference on Computer Vision*, pages 14–21, 1999.

[64] Noah Snavely, Steven M. Seitz, et Richard Szeliski. Photo tourism: Exploring photo collections in 3d. Dans *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press.

[65] Peter Sturm. *Vision 3D non calibrée : contributions à la reconstruction projective et étude des mouvements critiques pour láuto-calibrage*. PhD thesis, Institut National Polytechnique de Grenoble, December 1997.

[66] Peter Sturm et Steve Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. Dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, USA*, pages 432–437, Juin 1999.

[67] P.F. Sturm et S.J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. Dans *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages I: 432–437, 1999.

[68] Michael J. Swain et Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[69] J. Tao. Slide projector calibration based on calibration of digital camera. Dans *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6788 de *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, Novembre 2007.

[70] B. Triggs. Autocalibration from planar scenes. Dans *Proceedings of the 5th European Conference on Computer Vision, Freiburg, Germany*, 1998.

[71] Bill Triggs, Philip F. Mclauchlan, Richard I. Hartley, et Andrew W. Fitzgibbon. *Bundle Adjustment – A Modern Synthesis*, volume 1883. Springer-Verlag London, UK, January 2000.

[72] R.Y. Tsai. An efficient and accurate camera calibration technique for 3-d machine vision. Dans *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 364–374, 1986.

[73] Thierry Vieville. *A Few Steps Towards 3d Active Vision*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.

[74] M. P. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59, 1997.

[75] J.H. Wang, F.H. Shi, J. Zhang, et Y.C. Liu. Camera calibration from a single

frame of planar pattern. Dans *Advanced Concepts for Intelligent Vision Systems*, pages 576–587, 2006.

[76] R. J. Woodham. Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from a Single View. Dans *Proceedings of the 22nd SPIE Annual Technical Symposium*, volume 155, pages 136–143, San Diego, California, USA, Août 1978.

[77] Shuntaro Yamazaki, Srinivasa G. Narasimhan, Simon Baker, et Takeo Kanade. Coplanar shadowgrams for acquiring visual hulls of intricate objects. Dans *Proc. International Conference of Computer Vision*, pages 1–8, 2007.

[78] Shuntaro Yamazaki, Srinivasa G. Narasimhan, Simon Baker, et Takeo Kanade. Coplanar shadowgrams for acquiring visual hulls of intricate objects. Dans *Proc. International Conference of Computer Vision*, pages 1–8, 2007.

[79] Anna Yershova et Steven M. LaValle. Deterministic sampling methods for spheres and so(3). Dans *ICRA*, pages 3974–3980, 2004.

[80] Yizhou Yu et Johnny T. Chang. Shadow graphs and 3d texture reconstruction. *Int. J. Comput. Vision*, 62(1-2):35–60, 2005.

[81] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1:666–673 vol.1, 1999.

[82] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:892–899, 2004.

[83] Z.Y. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. Dans *International Conference on Computer Vision*, pages 666–673, 1999.

[84] Qi ZHao et Hai Tao. Object tracking using color correlogram. Dans *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 263–270, 2005.

[85] Zhiwei Zhu, Qiang Ji, et Kikuo Fujimura. Combining kalman filtering and mean shift for real time eye tracking under active ir illumination (oral). Dans *International Conference on Pattern Recognition*, 2002.

Annexe A

ESTIMATION D'UNE HOMOGRAPHIE

Une homographie est une transformation bijective entre deux plans de même dimension. Cette relation entre deux plans de dimensions d est représentée par une matrice de dimension $d \times d$.

Ici nous nous intéressons aux homographies entre plans 3D. Soient, H l'homographie qui relie les points $\mathbf{q}_i = (x_i, y_i, w_i)^\mathsf{T}$ et $\mathbf{q}'_i = (x'_i, y'_i, w'_i)^\mathsf{T}$ appartenant respectivement aux plans Π et Π', alors:

$$\mathbf{q}'_i \sim \mathsf{H}\mathbf{q}_i \tag{A.1}$$

En exprimant l'équation précédente en terme de produit vectoriel on obtient:

$$\mathbf{q}'_i \times \mathsf{H}\mathbf{q}_i = 0 \tag{A.2}$$

Il nous sera plus facile d'exprimer une solution à H sous cette nouvelle forme. En notant la j^{e} ligne de H par ${\mathbf{h}^j}^\mathsf{T}$, nous avons:

$$\mathsf{H}\mathbf{q}_i = \begin{pmatrix} {\mathbf{h}^1}^\mathsf{T}\mathbf{q}_i \\ {\mathbf{h}^2}^\mathsf{T}\mathbf{q}_i \\ {\mathbf{h}^3}^\mathsf{T}\mathbf{q}_i \end{pmatrix} \tag{A.3}$$

En substituant (A.3) dans (A.2) on obtient:

$$\mathbf{q}'_i \times \mathsf{H}\mathbf{q}_i = \begin{pmatrix} y'_i{\mathbf{h}^3}^\mathsf{T}\mathbf{q}_i - w'_i{\mathbf{h}^2}^\mathsf{T}\mathbf{q}_i \\ w'_i{\mathbf{h}^1}^\mathsf{T}\mathbf{q}_i - x'_i{\mathbf{h}^3}^\mathsf{T}\mathbf{q}_i \\ x'_i{\mathbf{h}^2}^\mathsf{T}\mathbf{q}_i - y'_i{\mathbf{h}^1}^\mathsf{T}\mathbf{q}_i \end{pmatrix} \tag{A.4}$$

À partir de (A.4), on voit que chaque correspondance $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$ procure trois équations en termes de H, dont deux sont indépendantes:

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w'_i\mathbf{x}_i{}^\mathsf{T} & y'_i\mathbf{x}_i{}^\mathsf{T} \\ -w'_i\mathbf{x}_i{}^\mathsf{T} & \mathbf{0}^\mathsf{T} & x'_i\mathbf{x}_i{}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = 0 \tag{A.5}$$

Avec quatre correspondances, il est possible d'estimer les paramètres de H à un facteur d'échelle près en résolvant le système homogène (A.5).