Advances on Pose Estimation and 3D Resconstruction of 2 and 3-View Scenes

Laura Fernandez Julia

École Doctorale Paris-Est
Mathématiques & Sciences et Technologies
de l'Information et de la Communication

Thèse de doctorat
de l'Université Paris-Est
Domaine : Traitement du Signal et des Images

Présentée par

Laura Fernàndez Julià

pour obtenir le grade de

Docteur de l'Université Paris-Est

Advances on Pose Estimation and 3D Reconstruction of 2 and 3-View Scenes

---

Soutenue publiquement le 13 décembre 2018 devant le jury composé de :

Pascal Monasse	École des Ponts ParisTech	Directeur de thèse
Marc Pierrot-Deseilligny	IGN	Co-directeur de thèse
Coloma Ballester	Universitat Pompeu Fabra	Rapporteur
Peter Sturm	Inria Grenoble Rhône-Alpes	Rapporteur
Gabriele Facciolo	ENS Paris-Saclay	Président
Ewelina Rupnik	IGN	Examinateur

Acknowledgements

This thesis would not have been possible without the support, collaboration and love of many people. To name all of them would be impossible but I want to try my best to recognize the impact they had in my life and, consequently, contributed to this work.

Every *thésard* needs a supervisor to guide them through the rough path that can be the completion of a PhD. Thank you Pascal for your support throughout this thesis and for offering me the opportunity to be a part of the Imagine group. I also want to thank my co-advisor Marc for always bringing new ideas and a different point of view to my research.

I would like to thank the members of the jury: Peter Strum and Coloma Ballester, who accepted the time consuming task to be reviewers, and Gabriele Facciolo and Ewelina Rupnik, who agreed to be examiners.

I have the incredible luck to have an amazing family and I want to thank them for believing in me during every step of this experience and supporting me in every way imaginable. Thanks specially to my father Xavier, my mother Olga and my brother Carles. We will now finally be *els quatre doctors*.

To all my friends from Barcelona, I want to thank them for being a constant in my life. I missed you during all my years in Paris but never felt far away. *Birretes*, you have always seen the best potential in me and given me confidence. To my friends from uni (A. H. I. D.), thanks for always inspiring me and making me more ambitious. Most of all, thank you Andrea, for being the best *best friend* I could ever ask for, always being there for me with the best advise and maintaining my sanity in the hard times, I could not have made it without you.

As one gets older, it becomes harder to make friends in a new place, but in Paris I had the chance to find amazing friends that became an important part of my life. I have really missed them since my return to Barcelona. To my group of friends that originated from the MVA (D. M. F. M.J. S.), thank you for becoming my first rock in a new country. Mainly, thank you Despoina for your invaluable friendship and making me a part of your life.

To all the Imagine crew (M. M. S. S. P. F. R. S. M. T. . . .), you have made my days at the lab better with coffee breaks, cinema club, after-work beers, running groups... I had the best times hanging out with all of you, you broaden my horizons with our discussions and improved my life in many ways. I have to specially thank my colleague, flatmate for 2+ years and friend Marina, I cannot thank you enough for all the good times we had and the daily talks that helped me get through.

Abstract

The study of cameras and images has been a prominent subject since the beginning of computer vision, one of the main focus being the pose estimation and 3D reconstruction. The goal of this thesis is to tackle and study some specific problems and methods of the structure-from-motion pipeline in order to provide improvements in accuracy, broad studies to comprehend the advantages and disadvantages of the state-of-the-art models and useful implementations made available to the public. More specifically, we center our attention to stereo pairs and triplets of images and discuss some of the methods and models able to provide pose estimation and 3D reconstruction of the scene.

First, we address the depth estimation task for stereo pairs using block-matching. This approach implicitly assumes that all pixels in the patch have the same depth producing the common artifact known as the "foreground fattening effect". In order to find a more appropriate support, Yoon and Kweon introduced the use of weights based on color similarity and spatial distance, analogous to those used in the bilateral filter. We present the theory of this method and the implementation we have developed with some improvements. We discuss some variants of the method and analyze its parameters and performance.

Secondly, we consider the addition of a third view and study the trifocal tensor, which describes the geometric constraints linking the three views. We explore the advantages offered by this operator in the pose estimation task of a triplet of cameras as opposed to computing the relative poses pair by pair using the fundamental matrix. In addition, we present a study and implementation of several parameterizations of the tensor. We show that the initial improvement in accuracy of the trifocal tensor is not enough to have a remarkable impact on the pose estimation after bundle adjustment and that using the fundamental matrix with image triplets remains relevant.

Finally, we propose using a different projection model than the pinhole camera for the pose estimation of perspective cameras. We present a method based on the matrix factorization due to Tomasi and Kanade that relies on the orthographic projection. This method can be used in configurations where other methods fail, in particular, when using cameras with long focal length lenses. The performance of our implementation of this method is compared to that given by the perspective-based methods, we consider that the accuracy achieved and its robustness make it worth considering in any SfM procedure.

Keywords

computer vision; structure from motion; pose estimation; 3D reconstruction; stereovision; disparity map; multi-view geometry; trifocal tensor; fundamental matrix; orthographic projection; long focal length.

Résumé Étendu

L'étude des caméras et des images a été un sujet prédominant depuis le début de la vision par ordinateur, l'un des principaux axes étant l'estimation de pose et la reconstruction 3D. Au début, l'accent était mis sur l'estimation de la profondeur en stéréovision, plus tard, l'extension aux vues $N \geq 3$ a été de plus en plus explorée par des méthodes de *structure-from-motion* (SfM). Il y a deux variantes principales du pipeline SfM : l'approche incrémentale, où le processus d'estimation commence avec une paire de vues et continue d'ajouter une par une les vues restantes ; et l'approche globale, qui vise à estimer toutes les poses en même temps. Les deux s'appuient en grande partie sur l'étalonnage à deux vues.

Les applications de la SfM sont vastes et variées, de l'archéologie et l'urbanisme jusqu'au cinéma et la réalité augmentée. Les problèmes rencontrés dans ce domaine sont aussi vastes que ses applications. La précision et la robustesse sont essentielles pour assurer le succès de la tâche d'estimation de pose dans n'importe quelle application. Dans cette optique, nous cherchons à évaluer l'utilité et les avantages des différentes méthodes impliquées dans le processus de la SfM, ainsi qu'à introduire de nouvelles approches pour surmonter les obstacles spécifiques qui sont généralement difficiles à affronter avec les procédures standard. Parmi les différentes étapes impliquées dans tout processus de reconstruction 3D, nous nous concentrerons sur certaines des questions qui se posent dans les domaines suivants : la correspondance stéréo des paires d'images et l'estimation de pose à trois vues à partir des points correspondants.

La reconstruction 3D à deux vues peut être réduite à un calcul de profondeur lorsque les images sont rectifiées. Pour résoudre le problème, il est nécessaire de faire correspondre les pixels d'une image à l'autre, généralement en utilisant des patches et une distance de similarité. Cette approche suppose implicitement que tous les pixels du patch ont la même profondeur et crée l'artefact connu sous le nom de phénomène *foreground-fattening*.

Les contraintes géométriques reliant les points correspondants des N vues sont déduites du modèle de la perspective, dérivé du modèle de caméra à sténopé, et elles constituent la base de la plupart des méthodes d'estimation de pose. Pour exactement $N = 3$ vues, les contraintes géométriques entre les points d'appariement sont plus fortes que lorsque seulement deux vues sont considérées. Pour cette raison, il est raisonnable de supposer que la prise en compte de ces contraintes à 3 vues pourrait avoir un impact positif sur la précision et la robustesse des méthodes d'estimation de pose. Cependant, les contraintes basées sur la projection en perspective s'affaiblissent lorsque l'objet reconstruit est éloigné par rapport à la distance entre les caméras, du fait qu'elles reposent sur l'intersection de rayons qui deviennent presque parallèles. Par conséquent, les méthodes basées sur des contraintes de perspective sont instables dans ce type de situations, par exemple, lorsque l'on utilise des caméras à longue distance focale. Dans de telles situations, un autre modèle de projection

vaut la peine d'être étudié, de sorte qu'une méthode d'estimation de pose plus robuste puisse être développée.

Le but de cette thèse est d'aborder et d'étudier ces problèmes et méthodes spécifiques du pipeline de la *structure-from-motion*. Nos contributions consistent en des études approfondies pour comprendre les avantages et les inconvénients des différentes méthodes, des nouvelles approches pour améliorer la précision et la robustesse dans des scènes spéciales et une attention particulière pour fournir des implémentations disponibles au public. Nous précisons ci-dessous le contenu de cette thèse chapitre par chapitre.

Chapitre 2 Une brève introduction des fondements de la *structure-from-motion* et des concepts et notations importants qui seront utilisés tout au long de la thèse sont présentés. La tâche de reconstruction 3D à partir de photographies peut être divisée en plusieurs étapes de la SfM : recherche de points de correspondance entre les paires d'images, élimination des correspondances mal détectées, estimation de la pose et reconstruction 3D, et raffinement final de la solution par optimisation.

Chapitre 3 Nous abordons la tâche d'estimation de la profondeur pour les paires stéréoscopiques à l'aide de la correspondance de blocs. Nous présentons un étude et implémentation d'une méthode particulière d'estimation de profondeur par Yoon et Kweon [YK06]. Cette approche de comparaison de blocs utilise des poids basés sur la similarité des couleurs et la distance spatiale, analogues à ceux utilisés dans le filtre bilatéral, pour éliminer le *foreground-fattening effect*. Pour deux pixels $\mathbf{p}, \mathbf{q}$ dans du même patch, le poids assigné est

$$w(\mathbf{p},\mathbf{q}) = w_{col}(\mathbf{p},\mathbf{q}) \cdot w_{pos}(\mathbf{p},\mathbf{q}) = \exp\Big(- \Big(\frac{\Delta c_{\mathbf{pq}}}{\gamma_{col}} + \frac{\Delta g_{\mathbf{pq}}}{\gamma_{pos}} \Big) \Big).$$

où $\Delta c_{\mathbf{pq}}$ est la similarité des couleurs, $\Delta g_{\mathbf{pq}}$ est la distance spatiale et γ_{col} et γ_{pos} sont de paramètres positifs fixés.

Grâce à de nombreuses expériences, les paramètres utilisés dans cette méthode sont analysés et des valeurs optimales sont trouvées en fonction de la précision des résultats. Des variantes de la méthode ont été examinées et testées ainsi que d'autres méthodes pour assigner des poids aux blocs ont été explorées. Nous avons mis à disposition notre implémentation qui inclut quelques améliorations et toutes les variantes de la méthode.

Chapitre 4 Pour l'estimation de la position des vues, on dépend généralement de la matrice fondamentale et des relations epipolaires entre deux vues pour obtenir un premier étalonnage externe. Néanmoins, d'autres contraintes géométriques dérivent de la relation entre plusieurs vues [HZ04]. Yi Ma et al. [MHV$^+$04] prouvent que les relations entre plus de trois images dépendent uniquement des relations entre triplets et paires d'images. De ce fait, on s'intéresse aux contraintes géométriques qui lient trois vues de la même scène, à savoir le tenseur trifocal et les équations trilinéaires. Avec ces relations on peut obtenir un étalonnage initial de trois vues sans passer par la matrice fondamentale. Nous explorons les avantages offerts par le tenseur trifocal dans la tâche d'estimation de pose d'un triplet de caméras par opposition au calcul des poses relatives paire par paire.

Le tenseur trifocal est un tenseur $3 \times 3 \times 3$, $\mathbf{T} = [T_1, T_2, T_3]$, composé alors de 27 paramètres, mais avec seulement 18 degrés de liberté. Pour un triplet de points correspondants

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ dans les trois images, des contraintes sont données par le tenseur $\mathbf{T}$,

$$[\bar{\mathbf{x}}_2]_\times \Big(\sum_i (\bar{\mathbf{x}}_1)_i T_i \Big) [\bar{\mathbf{x}}_3]_\times = 0_{3\times 3}$$

qui composent 9 équations trilinéaires dont seulement 4 sont indépendants.

Plusieurs façons d'estimer le tenseur trifocal à partir de triplets de points correspondants existent dans la littérature, nous présentons une étude et implémentation de quatre de ces paramétrisations minimales : les *correlation slices* de C. Ressl [Res02], les matrices orthogonales de K. Nordberg [Nor09], les contraintes des déterminants des *slices* du tenseur données par O. D. Faugeras et T. Papadopoulo [FP98] et les matrices Π de J. Ponce and M. Hebert [PH14]. En étudiant la dernière des paramétrisations, nous avons proposé de nouvelles contraintes pour surmonter certaines limitations.

Nous avons comparé les méthodes d'estimation de pose à 3 vues basées sur le tenseur trifocal ainsi que l'estimation de pose à 2 vues avec la matrice fondamentale et testé leurs performances dans différents contextes. Avec ces expériences, nous avons montré que l'amélioration initiale offerte par l'avantage théorique du tenseur trifocal n'est pas suffisante pour avoir un impact remarquable sur la précision de l'estimation de pose après le raffinement avec ajustement de faisceaux. Pour sa simplicité et son temps de calcul plus court, l'option recommandée est de ne considérer que les contraintes par paires et la matrice fondamentale, à condition d'utiliser un ajustement de faisceau à la fin (ce qui est aussi fortement recommandé, car cela peut réduire l'erreur par un facteur significatif).

Chapitre 5 Nous proposons d'utiliser un modèle de projection différent de celui de la caméra à sténopé pour l'estimation de pose des caméras perspectives. Nous abordons le modèle de la projection orthographique comme une approximation du modèle de la perspective pour des cameras loin de la scène photographiée. La projection orthographique suppose que le centre de la camera est à l'infini, et donc la projection des points de l'espace est orthogonale au plan de l'image. Pour un point de l'espace $\mathbf{X} \in \mathbb{R}^3$ sa projection orthogonale $\mathbf{x}$ est calculé

$$\mathbf{x} = s \begin{pmatrix} \vec{i}^\top \\ \vec{j}^\top \end{pmatrix} \mathbf{X} + \begin{pmatrix} a \\ b \end{pmatrix} \quad , \tag{1}$$

où $s > 0$ est un facteur d'échelle, $\vec{i}$ et $\vec{j}$ caractérisent la rotation et direction de projection $\vec{k} = \vec{i} \times \vec{j}$ et $(a, b)^\top$ est l'origine des coordonnées image. En particulier, nous utilisons le modèle de projection orthographique avec échelle proposé par Poelman et Kanade [PK97], qui définit les paramètres de la projection orthogonale a partir des paramètres d'une caméra perspective pour émuler son comportent lorsque son centre est à l'infinie.

La méthode de factorisation matricielle de Tomasi et Kanade [TK92] nous permet d'obtenir une méthode d'estimation de la rotation et de la translation des caméras même si les observations obéissent au modèle perspectif central. Cette méthode s'avère comme une approche robuste dans des configurations où d'autres méthodes échouent, en particulier lorsque l'on utilise des caméras avec des objectifs à longue distance focale. L'estimation de pose avec cette méthode a besoin d'un minimum de $M = 3$ vues et $N = 4$ points correspondants visibles dans toutes les images. Nous prouvons aussi que les scènes plates ne peuvent pas être reconstruites par le modèle orthographique.

La performance de notre implémentation de cette méthode est comparée à celle des méthodes fondées sur la perspective. Nos expériences prouvent que cette méthode a de bonnes performances dans l'estimation de pose dans le cas de caméras avec des objectifs à longue distance focale avec une grande robustesse au bruit. Nous montrons qu'en utilisant cette estimation comme estimation initiale pour une procédure d'ajustement de faisceau, la minimisation converge plus rapidement. Cette méthode basée sur le modèle orthographique est bien adaptée à l'étalonnage externe de caméras en perspective avec des distances focales relativement longues surpassant l'estimation de pose basée sur la perspective et étant comparable en complexité computationnelle au calcul linéaire de la matrice fondamentale. Nous constatons alors que la méthode mérite être prise en considération dans toute procédure de *structure-from-motion*.

Contents

Chapter 1

Introduction

In Computer Vision, the field of 3D Pose Estimation and Reconstruction has been widely studied for decades now. In the beginning, the focus was depth estimation in stereo-vision, later on, the extension to $N \geq 3$ views was more and more explored by structure-from-motion (SfM) methods. There are two main variants of the SfM pipeline: the incremental approach, where the estimation process starts with a pair of views and continues adding one view at a time; and the global approach, which aims at estimating all poses at once. Both rely substantially in the 2-view calibration.

The applications are broad and diverse, from Archeology and City Planning all the way to Cinema and Augmented Reality. The problems faced in this field are as broad as its applications. Accuracy and robustness are essential to ensure the success of the pose estimation task in any application. With this in mind, we aim to evaluate the usefulness and advantages of different methods involved in the SfM process as well as to introduce new approaches to endure specific obstacles which are generally difficult to address with the standard procedures.

From the several steps involved in any 3D reconstruction process, we will focus on some of the issues that arise in the following areas:

- **Stereo matching.** The 3D reconstruction of two views can be reduced to depth computation when the images are rectified. To solve the problem it is necessary to match pixels from one image to the other, generally using patches and a similarity distance. This approach implicitly assumes that all pixels in the patch have the same depth and creates the artifact known as the *foreground-fattening* phenomena.

- **3-view pose estimation from matching points.** The geometric constraints linking matching points in N views are deduced from the perspective model, which is derived from the pinhole camera model, and they are the base for most of the pose estimation methods. For exactly $N = 3$ views, the geometric constraints between the matching points are stronger than when only two views are considered. For this reason, it is a reasonable assumption that taking these 3-view constrains into account could have a positive impact in the accuracy and robustness of pose estimation methods. However, constraints based on the perspective projection become weak when the object reconstructed is far away with respect to the distance between the cameras, due to the fact that they rely on the intersection of projection rays which become almost parallel. In consequence, methods based on perspective constraints are unstable

in these kind of scenes, for instance, when taking long focal length cameras. In such situations, another projection model is worth studying, so that a more robust pose estimation method can be developed.

1.1 Thesis Contributions

In this thesis we address and study these specific problems and methods of the structure-from-motion pipeline. Our contributions consist of extensive studies to comprehend the advantages and disadvantages of different methods, the addition of new approaches to improve accuracy and robustness in special scenes and a special focus on providing implementations available to the public. We specify below the main contributions of this thesis,

- A study and implementation of a particular depth estimation method by Yoon and Kweon. The block-matching approach uses weights based on color similarity and spatial distance, analogous to those used in the bilateral filter to eliminate the *foreground fattening* effect. Through many experiments, the parameters used in this method were analyzed and optimal values were found based on the accuracy of the results. Variants of the method were explored and tested. We made our implementation available which includes some improvements and all the variants of the method.

- A broad review of different trifocal tensor parameterizations and the pose estimation methods for three views. While studying one of the parameterizations we were able to propose new constraints to overcome certain limitations. We have implemented the 3-view pose estimation methods based on the trifocal tensor as well as the 2-view pose estimation with the fundamental matrix and tested their performances in different settings. With these experiments we have shown that the theoretical advantage of using the trifocal tensor does not translate on a remarkable impact in the accuracy of the pose estimation.

- A robust approach for the pose estimation of cameras with long focal length lenses. By using an orthographic projection model instead of the perspective one to represent pinhole cameras at "infinite" distance, we were able to implement a robust pose estimation method based on the matrix factorization due to Tomasi and Kanade. We evaluated the performance of our implementation of this method comparing it to the perspective-based methods to show that the accuracy achieved and its robustness make it worth considering in any SfM procedure.

The publications associated to the contributions of this thesis are listed below. For some of the published work we specify where to find the available implementation that we have developed.

1. On line Journal: *Bilaterally Weighted Patches for Disparity Map Computation*, Laura F. Julià and Pascal Monasse, Image Processing On Line, 2015.

    * C++ code and on line demo for Disparity Map Computation is available at the same web page than the paper: `https://doi.org/10.5201/ipol.2015.123`

2. Colloque: *Estimation de pose à trois vues: les mérites respectifs du tenseur trifocal et du modèle orthographique*, Laura F. Julià, Pascal Monasse and Marc Pierrot-Deseilligny, comunication at Photogrammétrie Numérique et Perception 3D: les Nouvelles Conquêtes organized by the SFPT on March 2016.

3. International Conference (Poster): *A Critical Review of the Trifocal Tensor Estimation*. Laura F. Julià and Pascal Monasse, PSIVT'17, The Eighth Pacific-Rim Symposium on Image and Video Technology, Nov 2017, Wuhan, China.

    * MATLAB code for pose estimation for triplets of views using trifocal tensor estimation and fundamental matrix computation is available at the Github repository `https://github.com/LauraFJulia/TFT_vs_Fund.git`

4. Submission to IPOL: *The Orthographic Projection Model for Pose Calibration of Long Focal Images*, Laura F. Julià, Pascal Monasse and Marc Pierrot-Deseilligny.

    * MATLAB code is now available at the Github repository `https://github.com/LauraFJulia/OrthographicPE.git` and will also appear along with an on line demo in the IPOL web page once the paper is published.

1.2 Manuscript Organization

The structure of the manuscript of this thesis is the following. In Chapter 2 a brief introduction of the basis of Structure-from-Motion and important concepts that will be used throughout the thesis are presented. Then, in Chapter 3 we describe our implementation an study of a block-matching algorithm for disparity map computation based on bilateral weights. Chapter 4 presents our review of the trifocal tensor and the experiments to compare it to the fundamental matrix in the pose estimation task. Finally, in Chapter 5 an approach for pose estimation of long focal length images based on the orthographic model is described and compared through synthetic and real experiments to the perspective model approach.

Chapter 2

Introduction to Structure-from-Motion

In this chapter we will introduce the notation used throughout, the main basic concepts used in the thesis, as well as an overview of each step involved in the Structure from Motion (SfM) process. The task of 3D reconstruction from photographs can be broken down into the following SfM steps: search for matching points between image pairs (Section 2.7), discard incorrectly detected matches (Section 2.8 and Section 2.9), estimation of pose (Section 2.4.1) and 3D reconstruction (Section 2.4.2) and final refinement of the solution using optimization (Section 2.6).

Contents

2.1 Notation

$\mathbf{X}$, $\bar{\mathbf{X}}$	3D point in $\mathbb{R}^3$ and equivalent point in $\mathbb{P}^3$.
$\mathbf{x}$, $\bar{\mathbf{x}}$	2D point in $\mathbb{R}^2$ and equivalent point in $\mathbb{P}^2$.
$\sim$	proportional, equality in homogeneous coordinates.
K	(external) calibration matrix.
R, Q	rotation matrices.
$\vec{i}$, $\vec{j}$, $\vec{k}$	normal vectors in $\mathbb{R}^3$ representing rotation axes.
$\vec{t}$, $\vec{l}$	translation vectors in $\mathbb{R}^3$.
$\mathbf{C}$	camera center.
$\mathbf{O}$	center of the scene, i.e. centroid of the 3D points.
E	essential matrix.
F	fundamental matrix.
$\mathbf{T}$	trifocal tensor.
$[v]_\times$	3×3-matrix form of the cross product on the left by vector $v \in \mathbb{R}^3$, i.e. $[v]_\times w = v \times w$ for $w \in \mathbb{R}^3$.
$\lvert M \rvert$	determinant of a matrix M.
$\lVert v \rVert$	L^2-norm for a vector $v \in \mathbb{R}^3$.
$\lVert M \rVert$, $\lVert \mathbf{T} \rVert$	L^2-norm of the vector built from the coefficients of the matrix or tensor.

2.2 Euclidean and Projective Space

Throughout the thesis, both Euclidean spaces $\mathbb{R}^n$ and projective spaces $\mathbb{P}^n$ will be used. Projective geometry can be understood as an extension of Euclidean geometry, the basic intuition being that projective space has more points than Euclidean space for the same dimension, which are called *points at infinity*, and geometric transformations are allowed to transform the extra points to Euclidean points, and vice versa.

In the projective space $\mathbb{P}^n$, points are represented in homogeneous coordinates $\bar{\mathbf{x}} = (x_1, \ldots, x_{n+1}) \in \mathbb{P}^n$, that means that two points with proportional coordinates are considered the same,

$$(x_1, \ldots, x_{n+1}) \sim \lambda(x_1, \ldots, x_{n+1}) \qquad \forall \lambda \in \mathbb{R} - \{0\} \tag{2.1}$$

which explains that points have $n+1$ coordinates in a space of dimension n. We will note "$\sim$" as the equality in this context to avoid confusion. The point 0 (represented by all homogeneous coordinates to 0) does not exist in the projective space.

An explicit inclusion of the Euclidean space $\mathbb{R}^n$ to the projective space $\mathbb{P}^n$ can be done by adding 1 as an extra coordinate at the end of each point coordinates,

$$\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n \;\longrightarrow\; \bar{\mathbf{x}} = (x_1, \ldots, x_n, 1) \in \mathbb{P}^n \;. \tag{2.2}$$

In the same way, for any point in the projective space $\mathbb{P}^n$ with non-zero last coordinate, the Euclidean equivalent can be recovered by dividing the homogeneous coordinates by the last coordinate,

$$\bar{\mathbf{x}} = (x_1, \ldots, x_{n+1}) \in \mathbb{P}^n \text{ s.t. } x_{n+1} \neq 0 \;\longrightarrow\; \mathbf{x} = \left(\frac{x_1}{x_{n+1}}, \ldots, \frac{x_n}{x_{n+1}}\right) \in \mathbb{R}^n \;. \tag{2.3}$$

The projective points that have last coordinate zero, $x_{n+1} = 0$, are the *points at infinity* and do not correspond to any Euclidean point.

From here onwards, we will work with points in the plane ($\mathbb{R}^2$ and $\mathbb{P}^2$) and in the 3D space ($\mathbb{R}^3$ and $\mathbb{P}^3$).

2.3 Functioning of a general camera

A general photographic camera is an optical device able to capture an image of an object or a scene and record it on a digital sensor or photographic film. In Figure 2.1 a simple representation of its functioning is presented. The basic design of a camera consists of an enclosed box where light enters through a lens and is recorded on a light-sensitive medium.

Lens are used to capture the light from the subject of the photograph and bring it to a focus on the sensor. There is a wide range of types of lenses serving different purposes (normal, long focus, wide angle, fisheye). In digital cameras, image sensors are composed of cells measuring the light that hits them. Each cell corresponds to a pixel in the final digital image. In computer vision, the **focal length** refers to the distance from the center of the lens to the image plane, i.e. the sensor. This distance can be modified in most of cameras with the optical zoom and it has an impact on the *perspective deformation* of the image.

The most commonly used model to explain the geometry of a standard camera is the **pinhole camera**. In this ideal model the camera lens is omitted and the light is considered to go through a tiny hole, a single point in space that we call the camera center. Due to equivalence, this model is usually represented with the image plane between the space points and the camera center, instead of *behind* the center. This way the image is no longer upside-down, while the coordinates of the projected points remain unchanged.

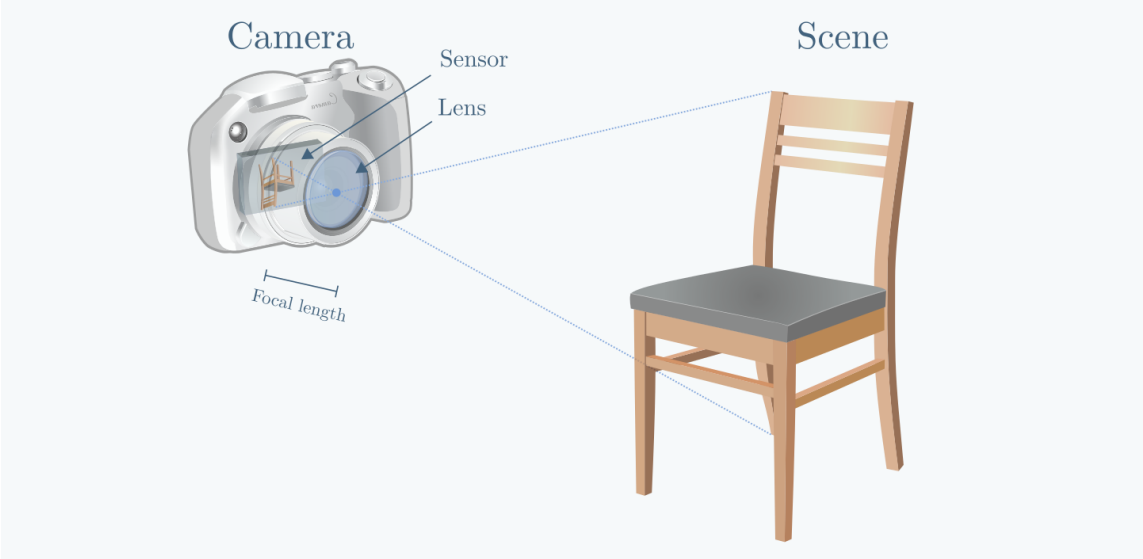


Figure 2.1 – Basic sketch of the functioning and parts of a photographic camera.

The mathematical function linking the space points to the image points corresponds to a central projection with center $\mathbf{C}$, the **center of the camera**, and projection plane the image plane (sensor), see Figure 2.2. When the coordinate frame is centered on $\mathbf{C}$ and the camera axes, the projection equation for a 3D point $\mathbf{X} = (X, Y, Z)^\top \in \mathbb{R}^3$ can be written,

$$\mathbf{x} = \rho_{\mathrm{P}}(\mathbf{X}) = \frac{f}{Z}(X, Y) \tag{2.4}$$

where f is the focal length of the camera. The projection equation is not linear, the relative depth of the point $\mathbf{X}$ appears dividing in the expression and it contributes to the *perspective effect*. It is clear that from the image point $\mathbf{x}$ it is impossible to recover uniquely the space point $\mathbf{X}$, since the depth Z cannot be inferred even knowing f.

For other coordinate frames, $\mathbf{X}$ has to be expressed in the camera system coordinates. Let us note the axes setting the image coordinates in the plane as $\vec{i}$ and $\vec{j}$ and $\vec{k}$ the plane's normal. Then, we call the **rotation matrix** of the camera to $R = (\vec{i}, \vec{j}, \vec{k})^\top$ and the **translation vector** to $\vec{t} = -R\mathbf{C}$. Applying the change of coordinates to $\mathbf{X}$ we get $R(\mathbf{X} - \mathbf{C}) = R\mathbf{X} + \vec{t}$. After that, we can apply the central projection. This transformation is better expressed in homogeneous coordinates as follows,

$$\bar{\mathbf{x}} \sim K \, [R \, \vec{t}] \, \bar{\mathbf{X}} \tag{2.5}$$

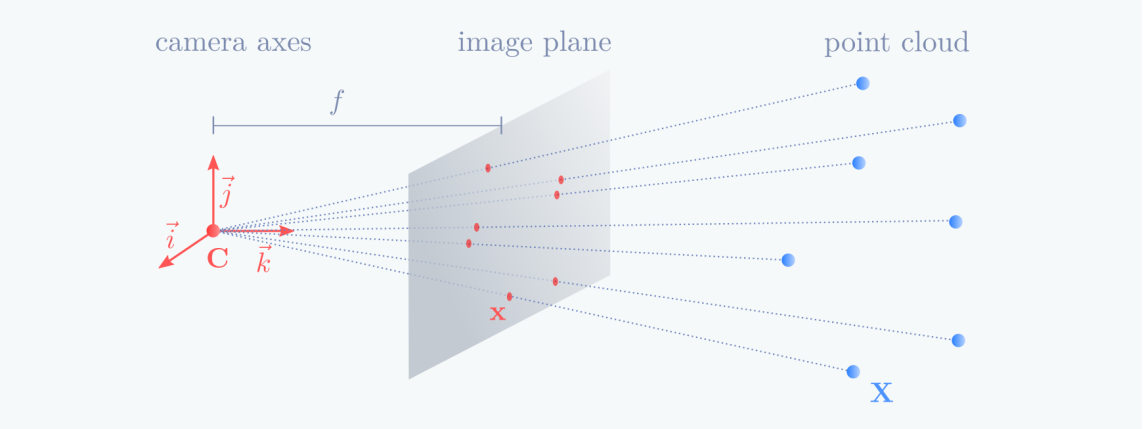


Figure 2.2 – Central projection of 3D points to a plane representing the pinhole model.

where K is the **calibration matrix** of the camera,

$$K = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.6}$$

$c = (c_x, c_y)$ is called the **principal point** and it indicates the point where the principal axis $\vec{k}$ meets the image plane, in image coordinates.

When the internal parameters of the camera, i.e. the calibration matrix, are unknown, the common approach is to consider the **projection matrix** $P = K\,[R\,\vec{t}]$ usually written $P = (A\ a_4)$ for the pose estimation, and working on the projective space instead of the affine.

2.4 Geometry of Two Views

When two images of the same scene are considered, it is possible to recover the space points from their respective projections in both images. Let us have two cameras with centers $\mathbf{C}_1$ and $\mathbf{C}_2$, and projection matrices $P_1 = K_1\,[R_1\ \vec{t}_1]$ and $P_2 = K_2\,[R_2\ \vec{t}_2]$. Then for a point $\mathbf{X} \in \mathbb{R}^3$ its two projections are given by

$$\bar{\mathbf{x}}_1 \sim P_1\,\bar{\mathbf{X}}\ , \qquad \bar{\mathbf{x}}_2 \sim P_2\,\bar{\mathbf{X}}\ . \tag{2.7}$$

Eliminating 3D points from this model we obtain the **epipolar equations** which define the **fundamental matrix** F_{21} for the two views, a 3×3 matrix verifying

$$\bar{\mathbf{x}}_2^\top F_{21} \bar{\mathbf{x}}_1 = 0 \tag{2.8}$$

for any two projections $\mathbf{x}_1$, $\mathbf{x}_2$ of the same point $\mathbf{X}$. This matrix is rank deficient, $\text{rank}(F_{21}) \leq 2$, and up-to-scale [HZ04].

The line connecting the two camera centers intersects with the image planes in the epipoles $\mathbf{e}_1$ and $\mathbf{e}_2$. For an image point $\mathbf{x}_1$, the line $F_{21}\bar{\mathbf{x}}_1$ in the second image represents all the possible points corresponding to $\mathbf{x}_1$. It is called the **epipolar line** associated to $\mathbf{x}_1$.

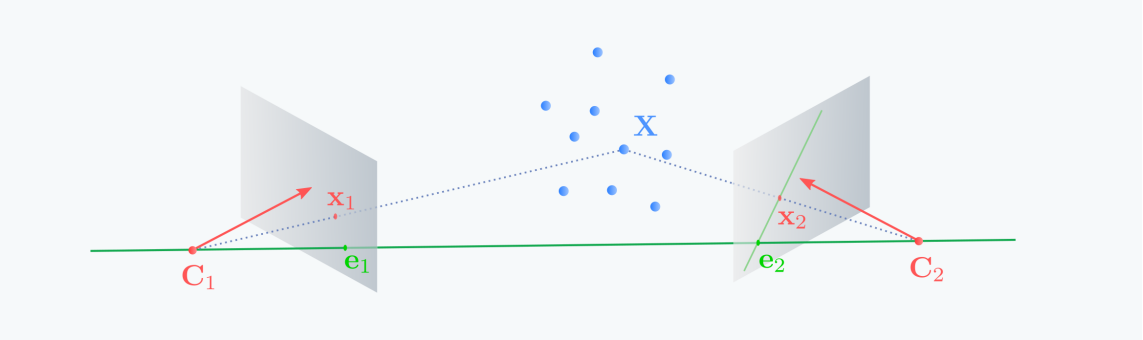


Figure 2.3 – Two views of the same scene and the two projections $\mathbf{x}_1, \mathbf{x}_2$ of space point $\mathbf{X}$.

The fundamental matrix can be estimated from 7 or more pairs of image correspondences. The matrix has 9 parameters up-to-scale and verifies $\det(F_{21}) = 0$, hence it has 7 degrees of freedom. Each pair of corresponding image points gives one equation (2.8) linear on the fundamental matrix parameters. Therefore, F_{21} can be estimated algebraically with 7 points and taking into account the rank deficiency (7-point algorithm [HZ04]) or linearly with 8 points imposing the rank deficiency to the matrix found by the linear system (8-point algorithm [LH81]).

When using normalized coordinates for the image points, i.e. $\hat{\mathbf{x}} = K^{-1}\bar{\mathbf{x}}$, the equivalent to the fundamental matrix in this case is the **essential matrix** E_{21},

$$\hat{\mathbf{x}}_2^\top E_{21} \hat{\mathbf{x}}_1 = 0 \tag{2.9}$$

for corresponding points $\mathbf{x}_1$ and $\mathbf{x}_2$. It follows that the relationship between the essential and fundamental matrices is $E_{21} \sim K_2^\top F_{21} K_1$. Moreover, it can be seen that

$$E_{21} \sim [t_{21}]_\times R_{21} \tag{2.10}$$

where $R_{21} = R_2 R_1^\top$ and $t_{21} = t_2 - R_2 R_1^\top t_1$ describe the relative rotation and translation between the two views.

2.4.1 Pose Estimation

The pose of the cameras can only be determined to some degree, only the relative rotation R_{21} and translation t_{21} between them can be inferred. The reconstruction will be ambiguous by global rotation, translation and scaling of the scene if no other real data (measurements of the scene, control points, etc.) is given.

Once the fundamental matrix F_{21} has been estimated and if the calibration matrices K_1 and K_2 are known, the essential matrix can be computed as $E_{21} = K_2^\top F_{21} K_1$, from which the relative orientation (R_{21}, t_{21}) can be retrieved by the singular value decomposition of E_{21} (using property (2.10)), the translation vector being up to unknown scale. The overall scale is commonly fixed by setting $\|t_{21}\| = 1$, the first camera pose to $(\mathrm{Id}, \mathbf{0})$ and the second camera to (R_{21}, t_{21}).

2.4.2 Triangulation

With the camera poses estimated we can triangulate any 3D point $\mathbf{X} \in \mathbb{R}^3$ from its projections in the two images $\mathbf{x}_1$, $\mathbf{x}_2$. Because of the noise in the measured points and the inaccuracy in the estimation of the poses, the estimated 3D point will not exactly satisfy the geometric relations. The linear triangulation can be done with the DLT algorithm [HZ04]; from the projection equations (2.7) the homogeneous scale factor can be eliminated by applying a cross product,

$$\mathbf{0} = \bar{\mathbf{x}}_i \times \bar{\mathbf{x}}_i = \bar{\mathbf{x}}_i \times (P_i \bar{\mathbf{X}}) \Rightarrow \begin{cases} x_i(p_i^3 \bar{\mathbf{X}}) - p_i^1 \bar{\mathbf{X}} &= 0 \\ y_i(p_i^3 \bar{\mathbf{X}}) - p_i^2 \bar{\mathbf{X}} &= 0 \\ x_i(p_i^2 \bar{\mathbf{X}}) - y_i(p_i^1 \bar{\mathbf{X}}) &= 0 \end{cases} \quad i = 1, 2. \tag{2.11}$$

where p_i^k is the k-th row of the projection matrix P_i. The three equations in (2.11) are linear in $\bar{\mathbf{X}}$ coordinates but only two are independent, so taking the first two given by each view we can compose a 4×4 matrix A so that $\bar{\mathbf{X}}$ is the solution of the linear system $A\bar{\mathbf{X}} = \mathbf{0}$,

$$A = \begin{pmatrix} x_1\, p_1^3 - p_1^1 \\ y_1\, p_1^3 - p_1^2 \\ x_2\, p_2^3 - p_2^1 \\ y_2\, p_2^3 - p_2^2 \end{pmatrix} . \tag{2.12}$$

Hence, the solution $\bar{\mathbf{X}}$ can be computed as the unit singular vector corresponding to the smallest singular value of the matrix A.

This method can be generalized to M image points in M views by composing the $2M \times 4$ matrix A with two equations for each view,

$$A = \begin{pmatrix} x_1\, p_1^3 - p_1^1 \\ y_1\, p_1^3 - p_1^2 \\ \vdots \\ x_M\, p_M^3 - p_M^1 \\ y_M\, p_M^3 - p_M^2 \end{pmatrix} . \tag{2.13}$$

2.5 Errors for Pose Estimation

To evaluate and compare the accuracy of pose estimation methods mainly the following two kind of errors are used: the **reprojection error** and the **angular error** in rotations and translations, since only translation direction can be estimated, not its norm.

The first does not depend on a known ground truth and can always be computed. For one image point $\mathbf{x}$, the reprojection error given the corresponding estimations for the camera pose, $P = K[R\,t]$ and 3D point $\mathbf{X}$, is the distance from the original measured point $\mathbf{x}$ to the projection of $\mathbf{X}$ using the estimated parameters, $\bar{\mathbf{x}}_r \sim K[R\,t]\bar{\mathbf{X}}$ (in homogeneous coordinates). The total reprojected error would be then the mean of the reprojection errors for all measured image points in all views, although the RMS is usually used instead for computational purposes. For N correspondences and M cameras,

$$e_{\text{repr}} = \sqrt{\frac{1}{N} \sum_{j=1}^{N} \left(\frac{1}{M} \sum_{i=1}^{M} \|\mathbf{x}_i^j - (\mathbf{x}_i^j)_r\|^2 \right)} . \tag{2.14}$$

For a configuration in which the 3D structure has not been estimated, or only the pose estimation is being evaluated, the reprojection error is computed by triangulating the 3D points (eq. (2.11)) using all the views available for each point.

On the other hand, the angular error compares the estimated poses to the ground truth. To do the comparison one of the cameras is aligned with the known true position and the scale is adjusted, then the rotations and translations defining the other cameras are compared to the true ones and the angle of their differences is computed. For instance, if the true poses for the cameras with respect to the first one are $\{[R^0_{i1},\ \vec{t}^{\,0}_{i1}]\}_{i=2,\dots,M}$ and the estimated poses are $\{[R_{i1},\ \vec{t}_{i1}]\}_{i=2,\dots,M}$, then the angular errors are

$$e_{\text{rot}} = \frac{1}{M-1} \sum_{i=2,\dots,M} \angle(R^0_{i1}, R_{i1}) \tag{2.15}$$

$$e_{\text{trans}} = \frac{1}{M-1} \sum_{i=2,\dots,M} \widehat{\vec{t}^{\,0}_{i1}\ \vec{t}_{i1}} \tag{2.16}$$

where $\angle(R, R')$ denotes the angle of the rotation corresponding to the matrix $RR'^{\top}$. It can be computed using the trace $\angle(R, R') = \arccos((\text{tr}(RR'^{\top}) - 1)/2)$.

It is worth noting that while the angular error in rotations is usually proportional to the reprojection error, the angular error in translations highly depends on the norm of the translation vectors. This translates into having errors of different order even in images taken with the same camera, of the same scene, which relative poses have been estimated in the same way, and have similar reprojection errors. In Appendix A we present a computation of the approximated pixel errors expected for a variation in the pose parameters to justify this phenomena.

2.6 Bundle Adjustment

Usually, the estimation of the camera poses obtained by singular decomposition of the essential matrix will not be very accurate and will leave much room for improvement. For this reason a common last step in pose estimation is Bundle Adjustment, first introduced by Triggs [TMHF00], a refinement of the camera orientations and structure using optimization. It is generalized to M views, and it is used at each step of incremental methods and at the end of global methods.

The optimization of Bundle adjustment minimizes the total reprojection error (2.14) over the possible cameras orientations and space points, thus it solves the equivalent minimization problem,

$$\min_{\substack{\{R_i,t_i\}_{i=1,\dots,M} \\ \{X^j\}_{j=1,\dots,N}}} \sum_{j=1}^{N} \sum_{i=1}^{M} \|\mathbf{x}_i^j - (\mathbf{x}_i^j)_r\|^2 \ . \tag{2.17}$$

The Levenberg-Marquardt algorithm [Lev44] for non-linear least squares minimization problems is the preferred method to solve the bundle adjustment. An initialization close to the final solution is needed to launch the iterative process.

2.7 Feature detection and Matching

Pose estimation is impossible without the knowledge of corresponding points between images. A minimum number of correspondences is needed to successfully estimate a relative pose between two views. For this reason detection and matching algorithms are one of the most important fields of computer vision and one of the most studied subjects in image processing.

The search for correspondences is subdivided in three stages: feature detection, feature description and feature matching. Since it is generally impossible or computationally too expensive to find correspondences for every pixel of each image (except in specific well posed scenes) the first step is to detect and select candidate pixels, called **features**, with specific characteristics that make them easier to match (textured areas, edges, corners...). Once the features have been selected they have to be described in order to be able to define a measure for comparison. The **descriptor**, usually a real vector, encodes all the relevant information around the feature and should be robust to light changes, rotation, translation and scale. After a descriptor is defined and given for each feature, the similarity between features in different images can be measured by computing the distance between their descriptors. Then, different strategies exist to pick the optimal match, for example the first nearest neighbor (FNN).

Earlier detection algorithms include Harris corners [HS88] and Harris-Laplace detector [MS01], but the most commonly used and widespread method for detection and description of features is D. G. Lowe's SIFT [Low04], based on an image pyramid to represent image scales and Gaussian convolutions to compute invariant blob-like features. The SIFT descriptor is a 128-float vector consisting in 16 histograms discretized into 8-bins.

2.8 Ransac

In feature matching we can find some noisy or completely false correspondences. Even if they represent a small percentage of the total of correspondences found, taking them into account when estimating the fundamental matrix will yield very inaccurate or simply erroneous pose estimation. This is a general problem that appears in any kind of parameter estimation where low precision or false data is present. In order to ensure a robust and accurate estimation, the data that is too noisy or false, the **outliers**, needs to be rejected so that only the data accurate enough, the **inliers**, are used by the estimation method.

RANSAC (RANdom SAmpling Consensus), first published by Fischler and Bolles [FB81], is the widespread method in computer vision for outlier detection. It is an iterative algorithm applicable to any estimation problem with the following characteristics: a certain model M has to be estimated from noisy data $D = \{d^1, \ldots, d^N\}$ containing outliers, with a method E that estimates the model from data fitting the model, $E(D_0) = M_0$, and an error distance can be defined to measure how much an observation from the data fits a certain model, $e(d^j, M_0)$, so that the distance is low if it fits it and high otherwise.

In the case of the fundamental matrix estimation, the observations are the correspondences between the two images found by a matching algorithm, the estimation method can be either the 7-point algorithm [HZ04] or the 8-point algorithm [LH81] and the error distance is defined by the distance to the epipolar lines given by the fundamental matrix.

The algorithm is described in Algorithm 1. It consists in repeating the same steps for each iteration (until an established maximum of iterations max_it). First, a set of n_E observations S_i is randomly selected from the observations dataset, where n_E is the minimal number of observations needed for the estimation E. Secondly, the model is computed using this sample and the estimation method, $M_i = E(S_i)$. In third place, the inliers I_i and outliers O_i for this model M_i are selected by computing the fitting error for each observation with respect to the estimated model, $e(d^j, M_i)$, if it surpasses the error threshold δ it will be labeled as an outlier, otherwise it will be labeled as inlier. Finally, if the total of inliers found for this model is higher than the highest found until this iteration, we keep this model, subset of observations and inlier set as the optimal ones.

Algorithm 1: RANSAC algorithm

input : Observations dataset $D = \{d^1, \ldots, d^N\}$,
estimation method E and minimal number of observations needed n_E,
error distance e and threshold δ,
maximum iterations max_it.
output: Inliers set I, estimated model M.

Initialize $N_{\max} \leftarrow 0$.
foreach $1 \leq i \leq max\_it$ **do**
  Randomly select a sample of n_E observations $S_i \subseteq D$.
  Compute estimated model M_i with this sample, $M_i \leftarrow E(S_i)$.
  Initialize inliers $I_i \leftarrow \varnothing$.
  foreach $1 \leq j \leq N$ **do** `// find inliers for` M_i `and` δ
    if $e(d^j, M_i) < \delta$ **then**
      $d^j \rightarrow I_i$
  Count number of inliers $N_i \leftarrow \#I_i$.
  if $N_i > N_{max}$ **then**
    $N_{\max} \leftarrow N_i$
    $I \leftarrow I_i$
    $M \leftarrow M_i$

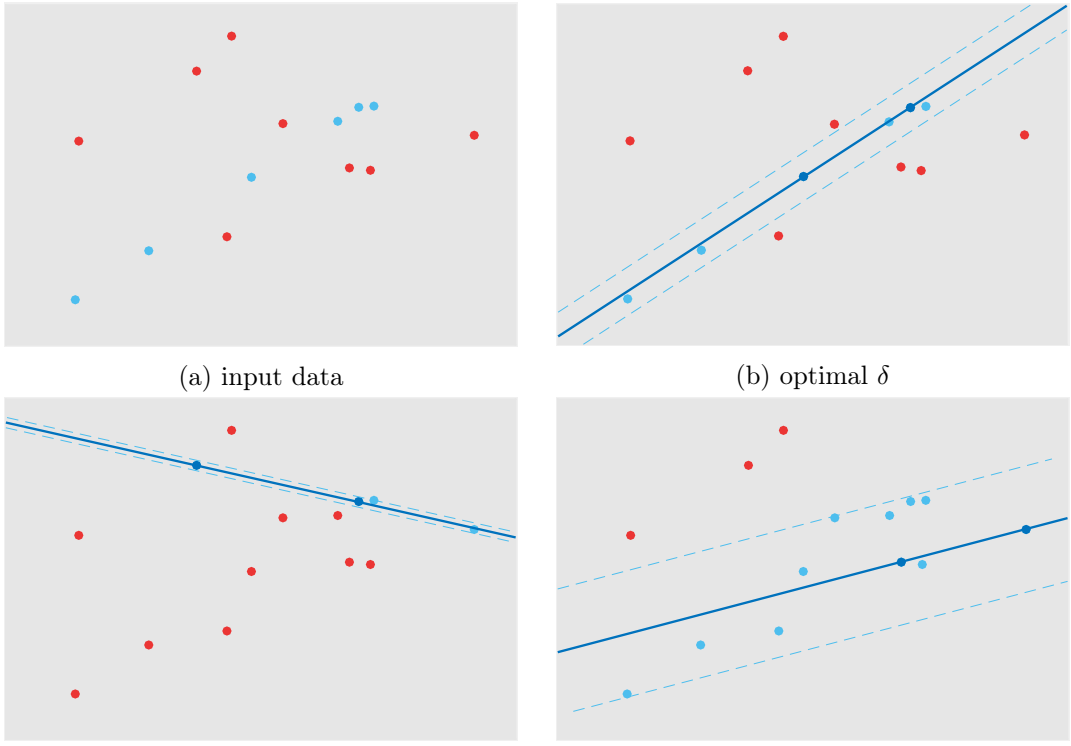
The idea of the RANSAC algorithm is that by repeating this process many times, the small subset of observations chosen randomly will eventually only contain observations fitting the model to estimate, therefore giving an accurate estimation of the model that will hopefully label as outliers only the false or too noisy data. It is to expect that the set of inliers found by this model will be the largest one due to the fact that the outliers can not be explained by any other model.

The algorithm has two parameters that have to be manually set, the maximum of iterations and the error threshold. The number of iterations will directly affect the accuracy of the model since the more iterations done, the higher the probability of finding the right model. If the proportion of inliers present in the data ρ is known or can be roughly estimated, an optimal number of iterations can be given with respect to the desired probability

of success p of the RANSAC algorithm,

$$\text{max\_it} = \frac{\log(1-p)}{\log(1-\rho^{n_E})} \quad . \tag{2.18}$$

On the other hand, the threshold δ is a parameter much more tricky to choose, it depends on the error distance defined, the data, and the model. Moreover, the choice of threshold can directly affect the possibility of estimating the right model and it is necessary to know the amount of noise present in the data in order to be able to make a good choice. This can be visualized with a basic example, fitting a line in 2D to a set of observations in Figure 2.4. It becomes obvious that the problematic selection of the threshold δ is the main drawback of the algorithm.



(a) input data (b) optimal δ

(c) δ too small (d) δ too large

Figure 2.4 – Different solutions found by the RANSAC algorithm for a line estimation problem with the input data shown. The data is composed by 6 inliers corresponding to noisy observations of points in the line, and 8 outliers. When an optimal value for δ is chosen the two inliers are selected by the algorithm to estimate the line and only inlier points lie inside the margin given by δ. When the threshold is too small or too large the line estimated is incorrect as is the classification of inliers and outliers.

2.9 A Contrario Ransac

The *a contrario* (AC) methodology relies on the Helmholtz principle: "an observed strong deviation from the background model is relevant information", i.e. if a configuration is

unlikely to arise by chance, it is notable. The first use of this theory was applied to detection in images by Desolneux et al. in [DMM08].

For model estimation this methodology can be applied to measure the meaningfulness of a certain estimated model, evaluating the likelihood of the data to fit such model under the hypothesis of the data being random. Following this idea, Moisan and Stival [MS04] proposed a variant of RANSAC for fundamental matrix estimation where the threshold δ is chosen automatically. It has been applied to many model estimation problems ([RDGM10, MMM16, MMM12a, MMM12b]) under many names, the most commonly used being AC-RANSAC.

The *a contrario* RANSAC algorithm controls the Number of False Alarms (NFA). A false alarm is a model due to chance and not really corresponding to the data. The NFA for an estimated model M and a number of inliers k is defined

$$\mathrm{NFA}(M,k) = n_{\mathrm{out}}(N-n_E)\binom{N}{k}\binom{k}{n_E}(e_k(M)^d\alpha_0)^{k-n_E} \tag{2.19}$$

where

- N is the number of total data elements.
- n_E minimum number of data necessary to estimate a model.
- n_{out} is the number of solution models given by the estimation method.
- α_0 probability of a random data having error ≤ 1.
- $e_k(M)$ the k-th lowest error to the model M among all data.
- d is the error dimension.

The optimal number of inliers $\hat{k}$ will be found by minimizing $\mathrm{NFA}(M,k)$

$$\hat{k} = \underset{k=n_E+1,\ldots,N}{\operatorname{argmin}} \mathrm{NFA}(M,k) \tag{2.20}$$

so that the chosen threshold will be $e_{\hat{k}}(M)$. A model M will be considered valid if

$$\mathrm{NFA}(M) = \mathrm{NFA}(M,\hat{k}) \leq \epsilon \tag{2.21}$$

The parameter ϵ is usually set to 1.

The *a contrario* methodology for model estimation consists in finding the best model with respect to the NFA, that is from all possible n_E samples of data possibles, finding the one that gives an estimated model minimizing $\mathrm{NFA}(M)$. While the complexity of computing $\mathrm{NFA}(M)$ for a given model is $O(N\log N)$, the number of possible models to be computed, however, is $n_{\mathrm{out}}\binom{N}{n_E}$ that is very large for $n_E > 2$. For this reason, the random sampling of RANSAC comes in hand to avoid computing all possible models.

The AC-RANSAC algorithm, described in Algorithm 2, follows the same steps than standard RANSAC but searches the model M with lower $\mathrm{NFA}(M)$, instead of maximizing the number of inliers with a fixed threshold δ. The advantage of such a strategy is that the precision threshold $e_k(M)$ that replaces δ adapts to the data.

This algorithm can be applied to the estimation of the fundamental matrix from N corresponding pairs of points, by using the following parameters,

E	7-point or 8-point algorithm.
n_E	7 or 8 respectively.
n_{out}	3 or 1 respectively.
$e(d, M)$	distance from image point to epipolar line.
d	1 (dimension of point-to-line distance).
α_0	$\frac{2D}{A}$ where D and A are the diameter and area of the image (upper bound).

Algorithm 2: AC-RANSAC algorithm

input : Observations dataset $D = \{d^1, \ldots, d^N\}$,
estimation method E and minimal number of observations needed n_E,
error distance e, parameter ϵ,
maximum iterations max_it.
output: Inliers set I, estimated model M, final threshold δ.

Initialize $\text{NFA}_{\min} \leftarrow \infty$.
foreach $1 \leq i \leq max\_it$ **do**
  Randomly select a sample of n_E observations $S_i \subseteq D$.
  Compute estimated model M_i with this sample, $M_i \leftarrow E(S_i)$.
  Initialize inliers $I_i \leftarrow \varnothing$.
  foreach $1 \leq j \leq N$ **do** `// Compute error vector` e
    $e_i(j) \leftarrow e(d^j, M_i)$
  Sort vector e.
  foreach $n_E \leq k \leq N$ **do** `// Find optimal` k
    $\text{nfa}_k \leftarrow \text{NFA}(M_i, k)$ `// with eq.` (2.19)
    if $nfa_k < NFA_i$ **then**
      $\text{NFA}_i \leftarrow \text{nfa}_k$
      $k_i \leftarrow k$
  if $NFA_i < NFA_{min}$ **then**
    $\text{NFA}_{\min} \leftarrow \text{NFA}_i$
    $I \leftarrow$ first k_i observations as sorted by e_i
    $M \leftarrow M_i$
    $\delta \leftarrow e_i(k_i)$

Chapter 3

Bilaterally Weighted Patches for Disparity Map Computation

Visual correspondence is the key for 3D reconstruction in binocular stereovision. Local methods perform block-matching to compute the disparity, or apparent motion, of pixels between images. The simplest approach computes the distance of patches, usually square windows, and assumes that all pixels in the patch have the same disparity. A prominent artifact of the method is the "foreground fattening effect" near depth discontinuities. In order to find a more appropriate support, Yoon and Kweon introduced the use of weights based on color similarity and spatial distance, analogous to those used in the bilateral filter. This chapter presents the theory of this method and the implementation we have developed. Moreover, some variants are discussed and improvements are used in the final implementation. Several examples and tests are presented and the parameters and performance of the method are analyzed.

Contents

3.1 Introduction

From the many configurations that two different views could have, there is a particular case that is specially interesting due to the simplified way to compute depth from two corresponding points and the limited neighborhood where the matching points have to be searched in: when cameras have same internal and external parameters but the position varies, the relative translation between them being parallel to their image planes.

(a) Reference image. (b) Target image. (c) Ground-truth disparity map.

Figure 3.1 – *Tsukuba* image pair and its ground truth disparity map: the intensity is proportional to the apparent horizontal displacement, the black border meaning "no data".

When the camera makes a fronto-parallel motion between the two images, depth estimation is done by estimating the **disparity** of each pixel, that is its apparent displacement from the first image (*reference* image) to the second image (*target* image). This disparity permits to recover the 3D position of the point that was photographed in this pixel since disparity is inversely proportional to depth.Using the similarity between triangles in the representation in Figure 3.2 we can see that the disparity d between two projections of the same point $\mathbf{X}$ in space is inversely proportional to its relative depth, Z,

$$d := x_1 + x_2 = \frac{X}{Z}f + \frac{D-X}{Z}f = \frac{Df}{Z} \tag{3.1}$$

since Df is a constant of the configuration, it is the product of the focal distance f and the distance between the two camera centers D.

From any two-view configuration, an equivalent pair of transformed images can be found that verify the fronto-parallel motion configuration by rectification of the images [LZ99]. In such a configuration the line joining the camera centers does not intersect with the image planes (in the affine world) so the epipolar lines become all parallel to each other in the image planes. In a rectified image the displacement between cameras will be horizontal, that is parallel to the x-axis of the image plane, so the epipolar lines will be too.

The core of disparity map computation methods is pixel correspondence. In order to compute the disparity of a pixel in the reference image, the corresponding pixel in the target image has to be found. With rectified pairs of images the search of the corresponding pixel is limited to the pixels on the same horizontal line in the target image, since that corresponds to the epipolar line of the pixel in the reference image. Therefore, the correspondence search can be done measuring the similarity between pixels and their neighborhoods and choosing the best match.

A key challenge in the correspondence search is image ambiguity, which results from the ambiguous local appearances of image pixels due to image noise and insufficient (or

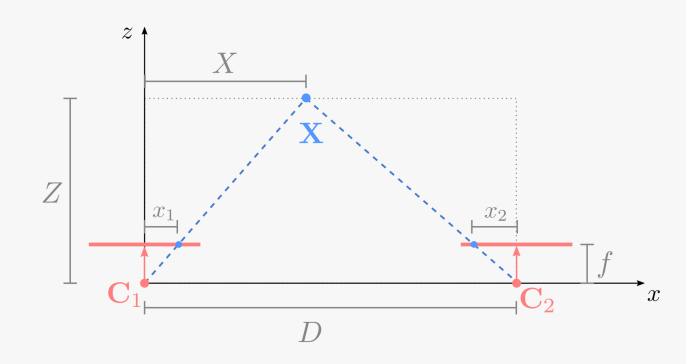
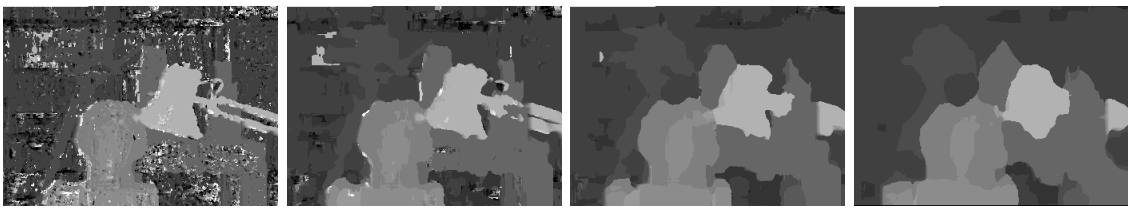


Figure 3.2 – Geometric sketch representing two cameras, with centers C_1 and C_2, in rectified position. The view of the scene is orthogonal to the plane parallel to the camera displacement and perpendicular to the image planes. From the figure, one can notice two pairs of similar triangles, from which we deduce $\frac{x_1}{f} = \frac{X}{Z}$ and $\frac{x_2}{f} = \frac{D-X}{Z}$.

repetitive) texture. By using local support windows, called patches, the image ambiguity is reduced efficiently while the discriminative power of the similarity measure is increased. In this approach, it is implicitly assumed that all pixels in a patch are at a similar depth in the scene and, therefore, that they have similar disparities. The artifact resulting from the violation of this assumption is the "foreground-fattening" phenomenon.

This effect appears when a pixel is wrongly assigned the disparity corresponding to other pixels in its patch. For this reason, although taking larger patches leads to smoother results, it also increases the fattening effect, producing errors near the depth discontinuities (this phenomenon occurring on the data of Figure 3.1 is illustrated in Figure 3.3). To avoid this phenomenon, adaptive-window methods try to find an optimal support patch for each pixel. Because of the advantages of this methods and their simplicity we decided to study and implement the correspondence search method proposed by Yoon and Kweon [YK06].

Yoon and Kweon's method consists in computing the support weights of the pixels in a patch using color similarity and geometric proximity to the center. The weights permit using larger patches, getting better results in homogeneous regions, while not provoking the fattening effect near depth discontinuities.



(a) 3×3 (b) 9×9 (c) 20×20 (d) 35×35

Figure 3.3 – Resulting disparity maps for *Tsukuba* obtained with a simple block-matching method (using SAD, sum of absolute differences, for patch similarity) for several patch sizes. The visible dilation of foreground shapes appearing with the increase of the patch size is the fattening phenomenon.

The weights introduced by this method are analogous to the ones used in the bilateral filter applied to image denoising [TM98]. Recall that the resulting image I_{BF} obtained by applying the bilateral filter to the original image I in a pixel $\mathbf{p}$ is computed as

$$I_{BF}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in N_{\mathbf{p}}\cap I} w(\mathbf{p},\mathbf{q})\, I(\mathbf{q})}{\sum_{\mathbf{q}\in N_{\mathbf{p}}\cap I} w(\mathbf{p},\mathbf{q})}, \tag{3.2}$$

where $N_{\mathbf{p}} = [x-r, x+r] \times [y-r, y+r]$ is a neighborhood of $\mathbf{p} = (x, y)$ and the weights are parameterized by positive real numbers σ_d and σ_r

$$w(\mathbf{p},\mathbf{q}) = \exp\Big(- \frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma_d^2} - \frac{\|I(\mathbf{p})-I(\mathbf{q})\|^2}{2\sigma_r^2}\Big). \tag{3.3}$$

We have slightly abused the notation in (3.2), by noting also I the rectangle domain of the image I.

The Middlebury Datasets

Throughout this chapter, the images from the Middlebury 2001 [SSZ01] and 2003 [SS03] datasets are used to illustrate the results and compare them to those provided in the original article by Yoon and Kweon [YK06]. These datasets contain the four pairs of stereo images: *Tsukuba, Venus, Teddy* and *Cones* shown in Figure 3.4 and their disparity range and ground truth disparity maps are provided. Other Middlebury datasets released more recently are tested at the end of the chapter in Section 3.8. The evaluation is generally done in three different groups of pixels: non-occluded pixels, all pixels and pixels near depth discontinuities. The pixels considered with a wrong disparity assignation are those with disparity error strictly greater than a threshold (generally 1 pixel). The evaluation can be done automatically submitting the resulting disparity maps in the Middlebury website and an average error percentage along with a general ranking are given[1].

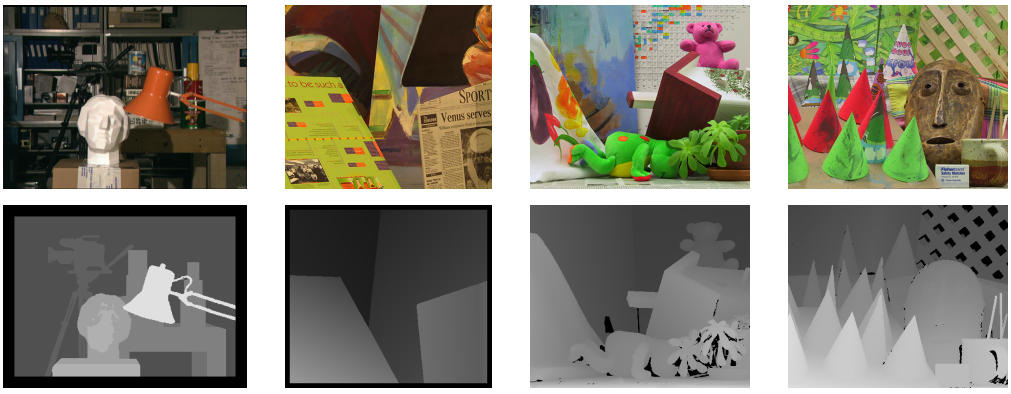


Figure 3.4 – Left images and ground-truth disparity maps for the stereo pairs of Middlebury datasets. From left to right: *Tsukuba, Venus* from the 2001 datasets, *Teddy* and *Cones* from the 2003 datasets.

[1]The evaluation website used for the experiments of this Chapter is no longer available, we present the results obtained at the time of the publication of the IPOL article associated with this work in 2015.

3.2 The Method

Let us note I_1 the reference image and I_2 the target image of a rectified stereo pair. Then, $I_1^c(\mathbf{p}_1)$ will be the color intensity in the color channel c ($c = r$ for red, $c = g$ for green, $c = b$ for blue) of the pixel $\mathbf{p}_1$ in the reference image I_1. Also, given a pixel $\mathbf{p}_1 = (x, y)$ in the image I_1 and a disparity d, then ${\mathbf{p}_2}^d$ will be the pixel in I_2 with coordinates ${\mathbf{p}_2}^d = (x+d, y)$.

The method is composed of three parts: adaptive support-weight computation, dissimilarity computation based on the support-weights and disparity selection.

3.2.1 Weight Assignation Based on Gestalt Grouping

Like in other block-matching methods, we take into account a support window of neighbor pixels to compare it to another patch in the target image. But not all the pixels in these patches may be relevant to do the comparison. Ideally, only pixels belonging to the same object and at the same depth should be taken into account. Considering the difficulty of explicitly segmenting an image, a pragmatic approach is adopted, assigning weights or probabilities to the neighbor pixels, trying to imitate the human visual system mechanism for grouping similar points. Therefore, the similarity of pixels in the patch is measured and weights corresponding to the probability to be grouped with the central pixel are assigned.

The strongest Gestalt grouping principles of the visual system are the color similarity and the spatial proximity [Met36]. Following these two principles, the support-weight of a pixel $\mathbf{p}$ with respect to another pixel $\mathbf{q}$ can be written as

$$w(\mathbf{p}, \mathbf{q}) = f(\Delta c_{\mathbf{pq}}, \Delta g_{\mathbf{pq}}) = w_{\text{col}}(\Delta c_{\mathbf{pq}}) \cdot w_{\text{pos}}(\Delta g_{\mathbf{pq}}), \tag{3.4}$$

where $\Delta c_{\mathbf{pq}}$ and $\Delta g_{\mathbf{pq}}$ are the color distance and the spatial distance between $\mathbf{p}$ and $\mathbf{q}$, respectively. The function f gives the strength of grouping by similarity and proximity, and if we consider $\Delta c_{\mathbf{pq}}$ and $\Delta g_{\mathbf{pq}}$ as independent events, the strength can be measured separately by w_{col} and w_{pos}.

Yoon and Kweon [YK06] justify their choice of w_{col} with the perceptual difference between two colors,

$$D(c_{\mathbf{p}}, c_{\mathbf{q}}) = 1 - \exp\Big(-\frac{\Delta c_{\mathbf{pq}}}{14}\Big), \tag{3.5}$$

which is defined in the CIE Lab color space, and where $\Delta c_{\mathbf{pq}}$ is the Euclidean distance. Based on this, they define the similarity strength as

$$w_{\text{col}}(\Delta c_{\mathbf{pq}}) = \exp\Big(-\frac{\Delta c_{\mathbf{pq}}}{\gamma_{\text{col}}}\Big), \tag{3.6}$$

depending on a fixed positive parameter γ_{col}.

In our implementation we used colors in the RGB space and the L^1 distance

$$\Delta c_{\mathbf{pq}} = \frac{1}{3}\|I(\mathbf{p}) - I(\mathbf{q})\|_1 = \frac{1}{3}\sum_{c\in\{r,g,b\}} |I^c(\mathbf{p}) - I^c(\mathbf{q})|. \tag{3.7}$$

Working in the CIE Lab space implies extra computations and, in addition, our tests yield better results using RGB and the L^1 norm. The computation advantage of using the L^1 norm comes from the fact that it is easy to tabulate: if color channels are integers between 0

and 255, the L^1 norm is an integer between 0 and $3 \cdot 255$, whose exponential values can be tabulated. Our tests exhibit a huge impact on computation time when tabulating: tabulation divides the running time by a factor 13.

Following the same strategy, the proximity strength is defined as

$$w_{\text{pos}}(\Delta g_{\mathbf{pq}}) = \exp\Big(-\frac{\Delta g_{\mathbf{pq}}}{\gamma_{\text{pos}}}\Big), \tag{3.8}$$

where γ_{pos} should be the patch radius and Δg_{pq} is the Euclidean distance

$$\Delta g_{\mathbf{pq}} = \|\mathbf{p} - \mathbf{q}\|_2 = \sqrt{(x - x')^2 + (y - y')^2} \tag{3.9}$$

where $\mathbf{p} = (x, y)$ and $\mathbf{q} = (x', y')$. Putting (3.4), (3.6) and (3.8) together we have

$$w(\mathbf{p}, \mathbf{q}) = w_{\text{col}}(\mathbf{p}, \mathbf{q}) \cdot w_{\text{pos}}(\mathbf{p}, \mathbf{q}) = \exp\Big(-\Big(\frac{\Delta c_{\mathbf{pq}}}{\gamma_{\text{col}}} + \frac{\Delta g_{\mathbf{pq}}}{\gamma_{\text{pos}}}\Big)\Big). \tag{3.10}$$

In Figure 3.5 some examples of weighted patches are shown for the *Tsukuba* image.

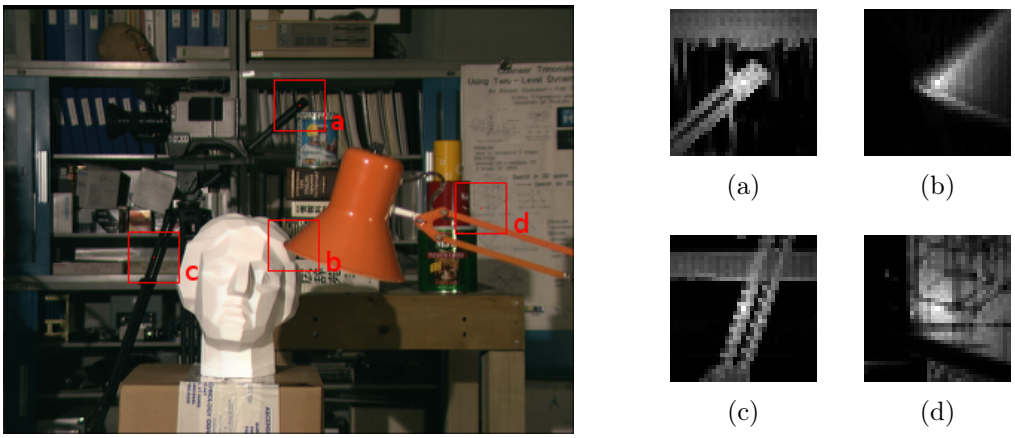


Figure 3.5 – Resulting weights for four different 35×35 patches in the *Tsukuba* left image, with $\gamma_{\text{col}} = 12$ and $\gamma_{\text{pos}} = 17$. The weights are represented as an intensity, white for high weights, black for low weights.

3.2.2 Dissimilarity Computation and Disparity Selection

For the dissimilarity computation step of the method we have to take into account the weights that we have just defined. In the computation of the matching cost between pixels, a raw matching cost e based on the color similarity is combined with the corresponding support-weights in both reference and target patches.

The dissimilarity between two pixels $\mathbf{p}_1$ (in the reference image) and ${\mathbf{p}_2}^d$ (in the target image with disparity d) is defined as

$$E(\mathbf{p}_1, {\mathbf{p}_2}^d) = \frac{\displaystyle\sum_{\mathbf{q}_1 \in N_1,\, {\mathbf{q}_2}^d \in N_2} \text{comb}(w(\mathbf{p}_1, \mathbf{q}_1), w({\mathbf{p}_2}^d, {\mathbf{q}_2}^d))\, e(\mathbf{q}_1, {\mathbf{q}_2}^d)}{\displaystyle\sum_{\mathbf{q}_1 \in N_1,\, {\mathbf{q}_2}^d \in N_2} \text{comb}(w(\mathbf{p}_1, \mathbf{q}_1), w({\mathbf{p}_2}^d, {\mathbf{q}_2}^d))} \tag{3.11}$$

where $N_1 = N_{\mathbf{p}_1} \cap I_1$ and $N_2 = N_{\mathbf{p}_2{}^d} \cap I_2$.

The function comb : $\mathbb{R}^2 \to \mathbb{R}$ combines the weights of tentative corresponding pixels $\mathbf{q}$ and $\mathbf{q}_2{}^d$. In the original article [YK06], comb is the product of its arguments, but we consider variants later on. A faster algorithm is the asymmetric case, where $\text{comb}(w_1, w_2) = w_1$ because the weights in the target image need not be computed. Notice that the former combination leads to a spatial weighting by $\exp(-\|\mathbf{p}-\mathbf{q}\|_2/\gamma_{\text{pos}})^2 = \exp(-2\|\mathbf{p}-\mathbf{q}\|_2/\gamma_{\text{pos}})$, while the latter gives a weight $\exp(-\|\mathbf{p}-\mathbf{q}\|_2/\gamma_{\text{pos}})$. To have consistent values in all cases, we modified slightly the definition by pulling the proximity term outside the combination function

$$E(\mathbf{p}_1, \mathbf{p}_2{}^d) = \frac{\displaystyle\sum_{\mathbf{q}_1 \in N_1,\, \mathbf{q}_2{}^d \in N_2} w_{\text{pos}}(\mathbf{p}_1, \mathbf{q}_1)^2 \; \text{comb}(w_{\text{col}}(\mathbf{p}_1, \mathbf{q}_1), w_{\text{col}}(\mathbf{p}_2{}^d, \mathbf{q}_2{}^d))\, e(\mathbf{q}_1, \mathbf{q}_2{}^d)}{\displaystyle\sum_{\mathbf{q}_1 \in N_1,\, \mathbf{q}_2{}^d \in N_2} w_{\text{pos}}(\mathbf{p}_1, \mathbf{q}_1)^2 \; \text{comb}(w_{\text{col}}(\mathbf{p}_1, \mathbf{q}_1), w_{\text{col}}(\mathbf{p}_2{}^d, \mathbf{q}_2{}^d))}. \tag{3.12}$$

In (3.12), the raw matching cost e between pixels $\mathbf{q}_1$ and $\mathbf{q}_2{}^d$ is defined as the truncated (by a positive parameter τ_{col}) absolute difference of color

$$e(\mathbf{q}_1, \mathbf{q}_2{}^d) = e_c(\mathbf{q}_1, \mathbf{q}_2{}^d) = \min\Big\{\frac{1}{3} \sum_{c \in \{r,g,b\}} |I_1^c(\mathbf{q}_1) - I_2^c(\mathbf{q}_2{}^d)| \;,\; \tau_{\text{col}}\Big\}. \tag{3.13}$$

However, results improve by introducing another term in the raw matching cost taking into account the similarity of the x-derivative value between both pixels [TM14][2]. Analogously to e_c, define

$$e_g(\mathbf{q}_1, \mathbf{q}_2{}^d) = \min\big\{|\nabla_x I_1(\mathbf{q}_1) - \nabla_x I_2(\mathbf{q}_2{}^d)| \;,\; \tau_{\text{grad}}\big\}, \tag{3.14}$$

where $\nabla_x I$ is the x-derivative computed in the gray-level image I^G as

$$\nabla_x I^G(x, y) = \frac{I^G(x+1, y) - I^G(x-1, y)}{2} \tag{3.15}$$

and τ_{grad} is a threshold for the gradient difference. Hence, we redefine the total raw matching cost as a linear combination of e_c and e_g,

$$e(\mathbf{q}_1, \mathbf{q}_2{}^d) = (1-\alpha) \cdot e_c(\mathbf{q}_1, \mathbf{q}_2{}^d) + \alpha \cdot e_g(\mathbf{q}_1, \mathbf{q}_2{}^d) \tag{3.16}$$

with $\alpha \in [0, 1]$.

Several choices for the combination function comb (some symmetric, one asymmetric) have been tested (see Section 3.7), but the choice for the implementation and general experiments is the multiplication of its arguments.

Finally, a pixel's final disparity is selected by the winner-takes-all method,

$$d_{\mathbf{p}_1} = \underset{d \in S_d}{\text{argmin}}\; E(\mathbf{p}_1, \mathbf{p}_2{}^d), \tag{3.17}$$

where $S_d = \{d_{\min}, \ldots, d_{\max}\}$ is the set of all possible disparities, which is assumed known.

[2]The use of the x-derivative, among other choices, is due to the horizontal apparent motion, since a horizontal edge (with large y-derivative) is not discriminative because of the aperture problem.

3.3 Implementation

The C++ implementation of this method is available as an article and demo in IPOL.

The pseudo-code for the disparity map computation using adaptive weights is presented in Algorithm 3. The key to a fast algorithm is to compute only once the support weight patches in the target image I_2, based on the trivial observation

$$(x+1) + S_d = x + (S_d \cup \{d_{\max}+1\} \setminus \{d_{\min}\})\,. \tag{3.18}$$

In other words, the patches in I_2 that are compared to pixel $(x+1, y) \in I_1$ are the same as for pixel $(x, y) \in I_1$, with the addition of the new patch centered on $(x+1+d_{\max}, y) \in I_2$ and the subtraction of the one centered on $(x+d_{\min}, y)$. If the patches of I_2 centered at pixels $\{(x+d, y) : d \in S_d\}$ are stored in an array `weights`, we can simply update the array to pixel $x+1$ by storing at index $x+d_{\min}$, whose patch centered on $(x+d_{\min}, y)$ is no longer useful, the new patch centered on $(x+1+d_{\max}, y)$. This is why in Algorithm 3, the lines 7-8 initialize the patches centered on pixels $\{(0+d, y) : d \in S_d\}$ (except for $d_{\max}$, computed in the x loop) that will be used for the pixel $(0, y)$ of I_1. Then, for each new x, only the patch of I_2 centered on $(x+d_{\max}, 0)$ needs be computed at line 11. Of course, this storage is useless in the asymmetric case, when only the weights of I_1 are used.

Another factor to accelerate the implementation is the tabulation of the similarity and proximity strengths in order to accelerate the computation of weights in function `support` (Algorithm 4). As observed above, this is possible only when we use the L^1 color distance. The same code is able to use different combinations of weights in the dissimilarity computation[3], the function `costCombined` applies Equation (3.12) for the chosen combination. Also, the raw matching costs are computed only once for every pair of pixels. For every possible disparity value the matching costs of pixels in the image are precomputed and stored in a new image, as explained in Algorithm 5. These images form the layers of the cost volume, defined as

$$cost[d](\mathbf{p}_1) = e(\mathbf{p}_1, {\mathbf{p}_2}^d). \tag{3.19}$$

In the original article [YK06], the authors do not specify whether they use a post-processing method to detect and fill occlusions and smooth the resulting disparity map, but it seems that they do. In our implementation we have used the post-processing presented and implemented by Tan and Monasse [TM14]. It proceeds by first detecting inconsistent pixels in left-right disparity maps, replacing their disparity with a simple scan line based filling, and finally applying a median filter with weights taken from the original image.

[3]This is a compile-time option, the dynamic use of a function pointer resulting in slower code.

Algorithm 3: Disparity map computation, function `disparityAW`

input : Color images I_1 and I_2, disparity range $S_d = \{d_{\min}, \dots, d_{\max}\}$, patch radius r, parameters γ_{pos}, γ_{pol}.
output : Disparity maps $d_1(\cdot)$, $d_2(\cdot)$ for images I_1 and I_2 respectively.

Tabulate color similarity strengths $distC(\cdot)$, array of size $3 \cdot 255 + 1$. // Eq.(3.6),(3.7)
Tabulate proximity strengths $distP(\cdot)$, array of size $(2r+1)^2$. // Eq.(3.8),(3.9)
Compute array *cost* of $\#S_d$ cost images for all disparities. // Alg. 5
$E_1(\cdot) \leftarrow +\infty \quad E_2(\cdot) \leftarrow +\infty$
foreach $0 \le y < height$ **do**
    Create array *weights* of $\#S_d$ patches.
    foreach $d \in [d_{min}, d_{max} - 1]$ **do** // Weighted target patches
        $weights[d - d_{\min}] \leftarrow$ `support`$(I_2, d, y, distC)$. // Alg. 4
    foreach $0 \le x < width$ **do**
        $W_1 \leftarrow$ `support`$(I_1, x, y, distC)$. // Alg. 4
        $weights[x + d_{\max} - d_{\min} \pmod{\#S_d}] \leftarrow$ `support`$(I_2, x + d_{\max}, y, distC)$. // Alg. 4
        foreach $d \in S_d$ **do**
            $W_2 \leftarrow weights[x + d - d_{\min} \pmod{\#S_d}]$
            $c \leftarrow cost[d - d_{\min}]$
            Compute dissimilarity E combining W_1, W_2, c, and $distP$. // Eq. (3.12)
            if $E < E_1(x, y)$ **then**
                $E_1(x, y) \leftarrow E$
                $d_1(x, y) \leftarrow d$
            if $E < E_2(x + d, y)$ **then**
                $E_2(x + d, y) \leftarrow E$
                $d_2(x + d, y) \leftarrow -d$

// The blue lines are not executed in the asymmetric combination of weights.

Algorithm 4: Weighted patch computation, function `support`

input : Color image I_1, central pixel p, patch radius r, color similarity strength table $distC$.
output : $(2r+1) \times (2r+1)$ image of color-based weights *weight* centered on p in I_1.

$weight(.) \leftarrow 0$
foreach *pixel* $q \in N_p \cap I_1$ **do**
    Compute $\delta \leftarrow \|I_1(p) - I_1(q)\|_1$
    $weight(q) = distC[\delta]$

Algorithm 5: Cost volume computation

input : Color images I_1 and I_2, disparity range $S_d = \{d_{\min}, \ldots, d_{\max}\}$, parameters α, τ_{col}, τ_{grad}.
output : Cost image vector *cost*.

Compute x-derivatives $\nabla_x I_1$ and $\nabla_x I_2$.
foreach $d \in S_d$ **do**
  Create cost image $cost[d](\cdot)$
  foreach $p \in I_1$ **do**
    $cost_c \leftarrow \tau_{\text{col}}$ $cost_g \leftarrow \tau_{\text{grad}}$
    if ${\mathbf{p}_2}^d \in I_2$ **then**
      $cost_c \leftarrow \min(\frac{1}{3}\Delta c_{p{\mathbf{p}_2}^d}, \tau_{\text{col}})$ // Eq. (3.13)
      $cost_g \leftarrow \min(|\nabla_x I_1(p) - \nabla_x I_2({\mathbf{p}_2}^d)|, \tau_{\text{grad}})$ // Eq. (3.14)
    $cost[d](p) \leftarrow (1-\alpha)cost_c + \alpha\, cost_g$ // Eq. (3.16)

3.4 Study of Parameters

The parameters of the method are summed up in Table 3.1, where the default values used in our experiments are also specified. The values that we have used are not the same as the default parameters of Yoon and Kweon [YK06] since they had to be adapted to a different color space and norm and the new matching cost formula. We chose the values that gave the better results for the Middlebury benchmark images. Moreover, tuning the parameters for each image individually can lead to better results in the final disparity map.

Parameter	Value	Description
α	0.9	Value in $[0, 1]$ for the combination of the color and the gradient similarity. The bigger α, the more important the gradient term is with respect to the color term.
γ_{col}	12	Positive parameter for the color similarity strength.
γ_{pos}	17.5	Positive parameter for the proximity strength.
r	17	Radius of the patch considered in the correspondence search.
τ_{col}	30	Color threshold for the matching cost computation.
τ_{grad}	2	Gradient threshold for the matching cost computation.

Table 3.1 – Default parameters of the algorithm.

We tested our implementation varying the parameters (one at a time) and studied the error in the resulting disparity map. We compared the resulting disparity maps for the Middlebury benchmark with their ground truth and computed the percentage of "bad" pixels (pixels whose absolute disparity error is greater than 1) for all the pixels in the image, the pixels in non-occluded areas and pixels near depth discontinuities.

Seeing that varying some parameters had a greater effect on the error near depth discontinuities than in other pixels, we show in Figure 3.6 the variation of this error for the four Middlebury benchmark images when varying γ_{pos}, γ_{col} and α. For γ_{pos}, we varied γ_{pos} alongside the patch radius r to see the error evolution when we changed the size of the patch. The error stabilizes for $\gamma_{\text{pos}} > 15.5$ ($r = 15$) so we preserved the default value of Yoon and Kweon [YK06] $\gamma_{\text{pos}} = 17.5$ ($r = 17$) and patch size 35×35.

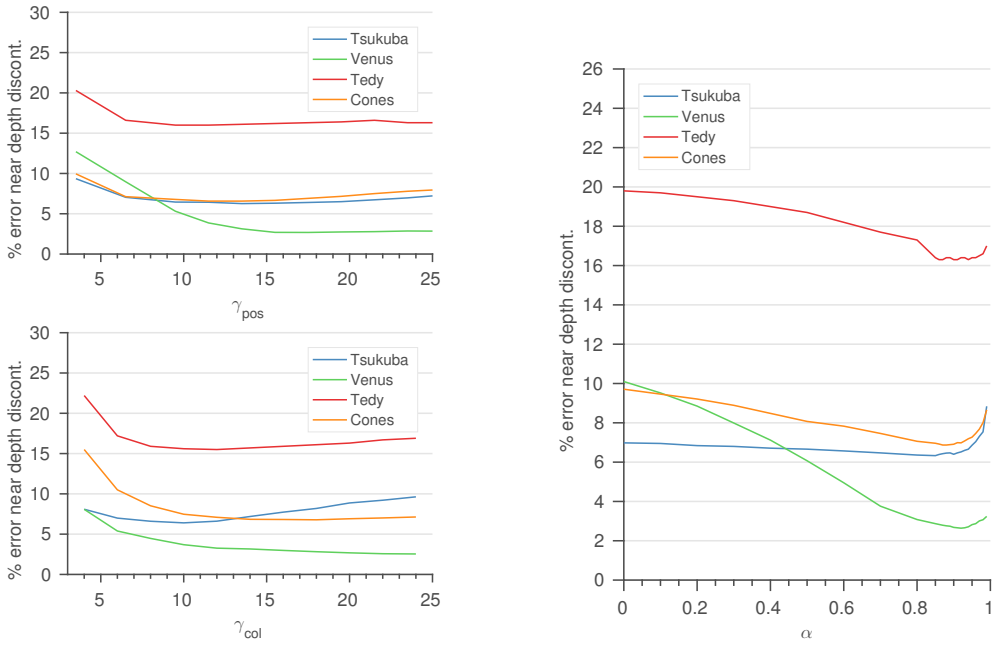


Figure 3.6 – Influence of the parameters γ_{pos}, γ_{col} and α on the error percentage near depth discontinuities for the four reference images.

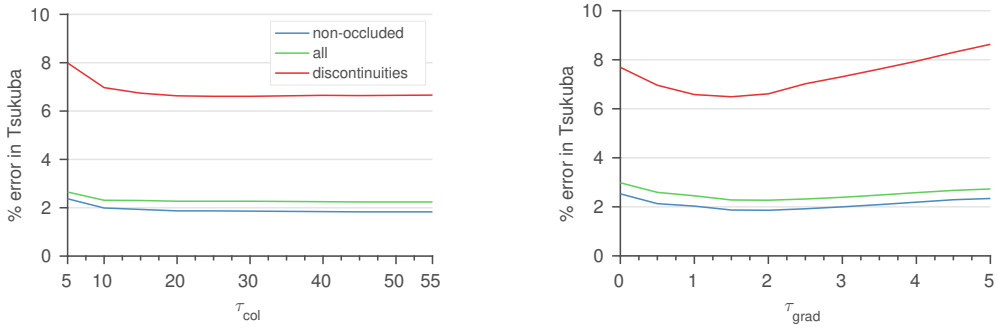
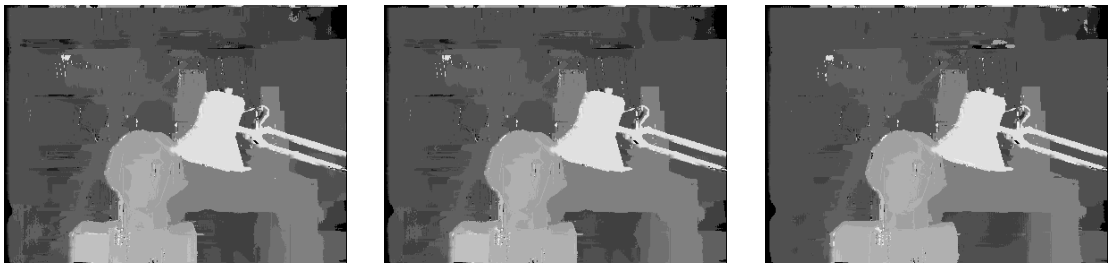


Figure 3.7 – Influence of the thresholds τ_{col} and τ_{grad} on the error percentage in *Tsukuba* image.

The essence of the method lies in a good choice for the parameter γ_{col}. As we increase γ_{col}, we take less into account the color distance w.r.t. the central pixel, so we take more pixels in the window into consideration for the comparison. This translates to the results in a first improvement of the accuracy when γ_{col} is increased, but the error grows eventually when too many pixels with different disparities are taken into account for high γ_{col}. As γ_{col} gets very large, we are using regular block matching. To find a compromise between all the images we choose $\gamma_{\text{col}} = 12$.

The addition of the gradient term proves to be very beneficial and choosing $\alpha = 0.9$ we get the best results. Notice that although this term is highly helpful, we should not disregard the color term, since the error increases sharply after $\alpha = 0.95$. Figure 3.8 shows the influence of α in the resulting disparity map. We can see that increasing α we erase some of the errors in the background and recover the edges a lot better, with less fattening effect.

Regarding the color and the gradient thresholds, their effect on the error is similar for all four images, so we chose to illustrate their influence by showing in Figure 3.7 the results for the *Tsukuba* image at all pixels, pixels near depth discontinuities and non-occluded pixels. It is clear that while the error decreases when τ_{col} increases and stabilizes after $\tau_{\text{col}} = 20$, the influence of τ_{grad} is greater and its optimal value is $\tau_{\text{grad}} = 2$.



(a) $\alpha = 0$, $e_{\text{total}} = 5.18\%$. (b) $\alpha = 0.5$, $e_{\text{total}} = 4.82\%$. (c) $\alpha = 0.9$, $e_{\text{total}} = 4.46\%$.

Figure 3.8 – Disparity maps for *Tsukuba* for different values of α without the post-processing step and the percentage of total error, e_{total}.

3.5 Middlebury Benchmark Disparity Maps

The results for the Middlebury benchmark are presented and compared to those presented in the original article in Figure 3.9 and Table 3.2. The parameters values used in all cases are the default ones specified in Table 3.1. The disparity maps have been improved and smoothed by a final post-processing step as in the article by Tan and Monasse [TM14].

We notice that whereas we get a few additional isolated errors compared to the original method and we are not able to compute the disparity in some small regions, our implementation recovers the depth discontinuities much better and with less fattening effect. As a result, our implementation achieves better results in general as it is shown in Table 3.2.

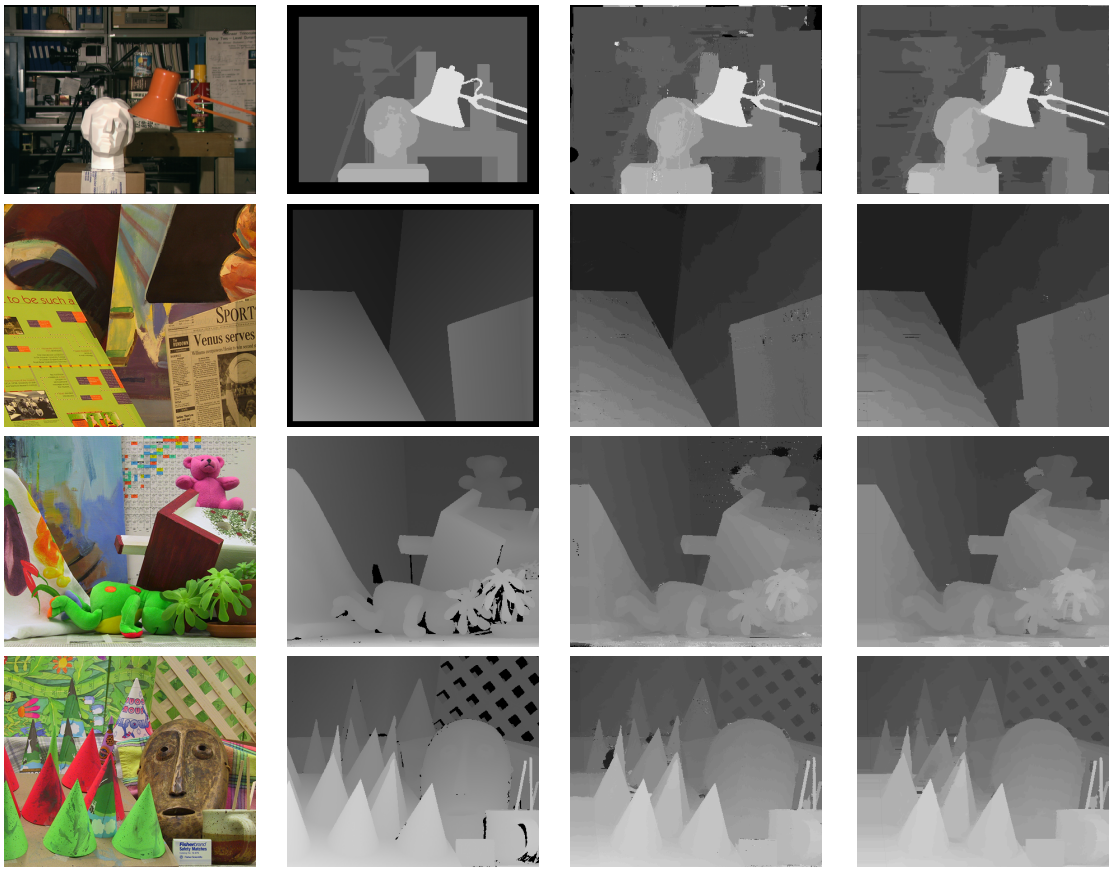


Figure 3.9 – Disparity maps and comparison with the results of Yoon and Kweon [YK06].

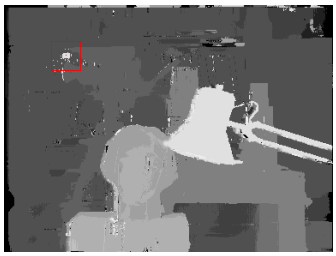
		Tsukuba			Venus			Teddy			Cones		
	Rank	non	all	disc	non	all	disc	non	all	disc	non	all	disc
Original art. [YK06]	80.2	**1.38**	**1.85**	6.90	0.71	1.19	6.13	7.88	**13.3**	18.6	3.97	9.79	8.26
Ours w/o post.-proc.	90.4	2.50	4.46	7.25	1.29	2.86	4.61	7.60	17.0	17.0	3.13	13.8	8.29
Ours with post.-proc.	**63.7**	1.86	2.27	**6.61**	**0.65**	**1.02**	**3.15**	**6.56**	14.4	**15.5**	**2.48**	**8.81**	**6.91**

Table 3.2 – Error percentages on Middlebury stereo benchmark (with tolerance of ±1 pixel for disparity). The rank is the algorithm average rank in the Middlebury website. In **bold** are the best (i.e., lowest) values of the three rows for each column. Our results overcome the ones from [YK06] in most of the cases. The results are significantly improved by the post-processing (p.p.).

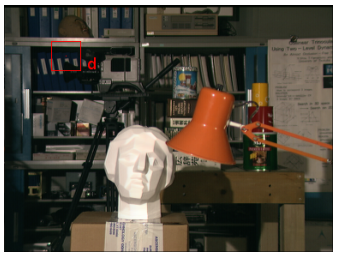
3.6 Weights at Error Pixels

The adaptive weights method combined with the use of the gradient gives good results and it has a good performance at disparity discontinuities. Nevertheless, it is not able to overcome some of the usual problems of local methods. Indeed, most of the errors produced by this method are found in repetitive areas, homogeneous regions or isolated points with a very different color from their neighbors. Examples of these errors are presented in Figures 3.10, 3.11 and 3.12, and the weights around the erroneous pixels are shown in order to understand their cause.

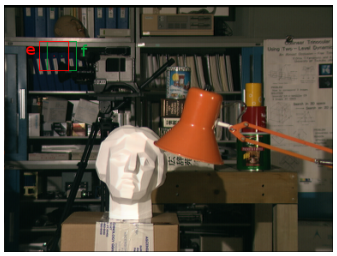
Figure 3.10 is an example of repetition (stroboscopic effect). The binders in the shelf are organized one after another and they look all similar. The method fails to find the good correspondence in the case of some pixels in that region since it ignores the pixels that are not part of the repetition. In fact, the pixels are assigned to the previous binder.



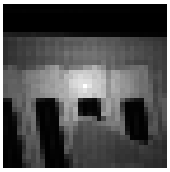
(a) Computed disparity map.



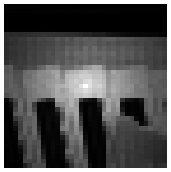
(b) Left image.



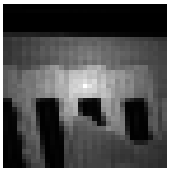
(c) Right image.



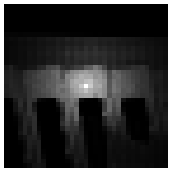
(d) $w(p_d, .)$



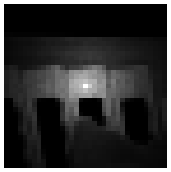
(e) $w(p_e, .)$



(f) $w(p_f, .)$



(g) $w(p_d, .)w(p_e, .)$



(h) $w(p_d, .)w(p_f, .)$

Figure 3.10 – Example of repetition. The patch `d` of center p_d in (b) is matched to patch `e` of center p_e in (c), but the true correspondence is in fact patch `f` of center p_f. Images (d), (e) and (f) show the weights in the patches. (g) is the combinations of the weights (d) and (e). (h) is the combination of (d) and (f). The two combined weights are similar and since the color distances in both correspondences are also similar because of the repetition, the assignation fails to find the correct disparity.

Figure 3.11 is an example of isolated point. Below the bust in *Tsukuba* image there are some letters on the box that get a bad disparity. The problem in this case is the presence of some points in that area that have a color really different from their neighbors so the weights in the reference image (3.11a) are scattered. Consequently, the possible combinations of weights are also scattered and the similarity measure takes into account few pixels. Ultimately, the correspondence is assigned to a combination of weights that consists in almost all the weight concentrated in the central pixel. This effect was observed in a survey paper [HBG13].

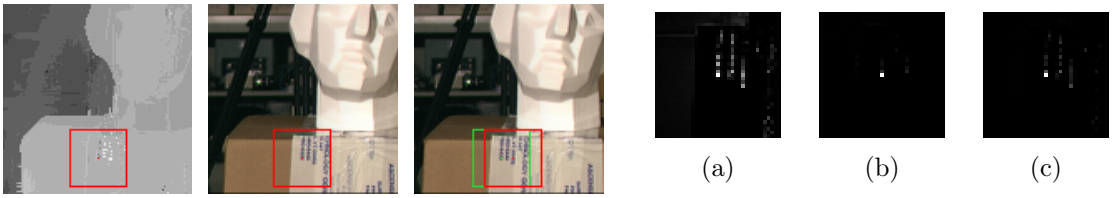


Figure 3.11 – Example of isolated point. (a) Weighted patch of the reference pixel. (b) Combined weights of the detected correspondence. (c) Combined weights of the true correspondence.

Figure 3.12 shows that the method, even with its large 35×35 windows, is not immune to the aperture problem, that is, the window is not large enough to contain significant horizontal gradient. In the background of the image there is a bookcase. The method assigns significant weights only to those pixels on the shelf (3.12a), which is a homogeneous region of the image. Because of that, the method fails to find the correct correspondence.

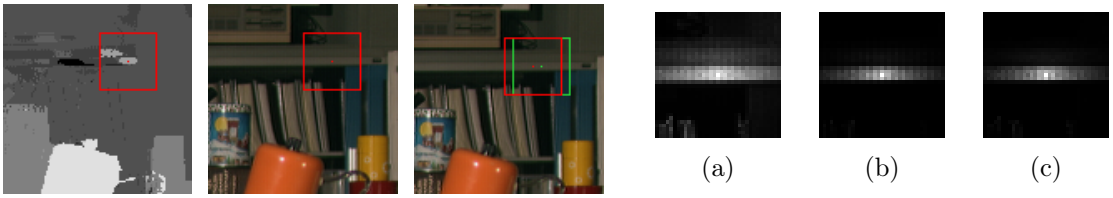


Figure 3.12 – Example of aperture problem, where the patch is not discriminative enough, because it has no significant non-horizontal edge. (a) Weighted patch of the reference pixel. (b) Combined weights of the detected correspondence. (c) Combined weights of the true correspondence.

3.7 Variants of the Method

In our implementation we have modified the original color space used in the similarity strength (3.6), but we can study how the method works in the original space. One could also study other ways to combine the weights of the two patches when we compute the dissimilarity (3.12) for a certain pair of pixels. These variants of the method are studied in the following sections.

3.7.1 CIE Lab Space

Using the CIE Lab color space takes more computations due to the conversion of the values originally in RGB. Although we have not used this space in our implementation, some results using the CIE Lab space are presented below. For the implementation of the method in this space we used the following conversion:[4] if we have a color $c = (r, g, b)$ in RGB coordinates, first we have to convert it to XYZ coordinates,

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.950 \end{pmatrix} \begin{pmatrix} r/2.55 \\ g/2.55 \\ b/2.55 \end{pmatrix} \tag{3.20}$$

and then to CIE Lab,

$$\begin{cases} L &= 116 \cdot f(Y/Y_n) - 16 \\ a &= 500 \cdot (f(X/X_n) - f(Y/Y_n)) \\ b &= 200 \cdot (f(Y/Y_n) - f(Z/Z_n)) \end{cases} \quad \text{with } f(t) = \begin{cases} t^{\frac{1}{3}} & \text{if } t > (\frac{6}{29})^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{otherwise.} \end{cases} \tag{3.21}$$

where $(X_n, Y_n, Z_n) = (95.047, 100.00, 108.883)$ is the white point. The parameters used are $\gamma_{\text{col}} = 3.5$, $\tau_{\text{col}} = 10$, $\gamma_{\text{pos}} = 17.5$ and $\alpha = 0.9$. We can observe in Figure 3.13 that the results near discontinuities are a little worse than using RGB and we get inaccurate edges. Apart from this, the numerical results are quite close to the ones using the RGB space (Table 3.3), some of them being even better (non-occluded pixels of *Venus* image).

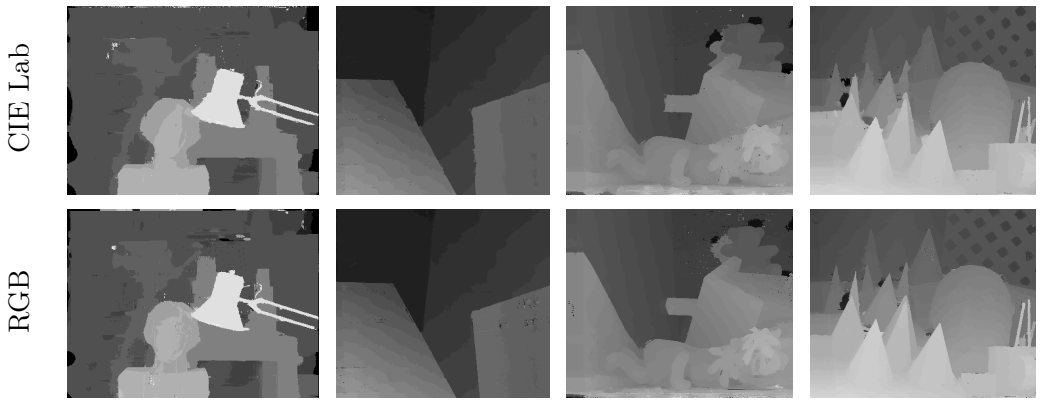


Figure 3.13 – Comparison of color spaces for the Middlebury benchmark images.

	Tsukuba			**Venus**			**Teddy**			**Cones**		
	non	all	disc	non	all	disc	non	all	disc	non	all	disc
RGB space	1.86	2.27	6.61	0.65	1.02	3.15	6.56	14.4	15.5	2.48	8.81	6.91
CIE Lab space	1.85	2.60	7.94	0.50	1.25	5.33	7.53	15.1	18.3	3.35	9.83	9.58

Table 3.3 – Error comparison between our implementation using CIE Lab and RGB color spaces.

[4]Formula taken from `http://www.cs.rit.edu/~ncs/color/t_convert.html`

3.7.2 Combination of Weights

To compute each patch pair matching cost, the presented method computes the product of weights of the target and reference patches, see Equation (3.12). Other possibilities are explored below. The simplest approach is to consider only the weights of one patch (asymmetric weights), for example, those of the reference patch, which amounts to $w(\mathbf{p}_2{}^d, \mathbf{q}_2{}^d) = 1$ in Equation (3.12). Another approach, similar to taking the product is to take the sum of weights

$$E(\mathbf{p}_1, \mathbf{p}_2{}^d) = \frac{\sum_{\mathbf{q}_1 \in N_1,\, \mathbf{q}_2{}^d \in N_2} w_{\text{pos}}(\mathbf{p}_1, \mathbf{q}_1)^2 \big(w_{\text{col}}(\mathbf{p}_1, \mathbf{q}_1) + w_{\text{col}}(\mathbf{p}_2{}^d, \mathbf{q}_2{}^d)\big)\; e(\mathbf{q}_1, \mathbf{q}_2{}^d)}{\sum_{\mathbf{q}_1 \in N_1,\, \mathbf{q}_2{}^d \in N_2} w_{\text{pos}}(\mathbf{p}_1, \mathbf{q}_1)^2 \big(w_{\text{col}}(\mathbf{p}_1, \mathbf{q}_1) + w_{\text{col}}(\mathbf{p}_2{}^d, \mathbf{q}_2{}^d)\big)}. \quad (3.22)$$

The parameters that give the best results for these two approaches and used in our tests are $\gamma_{\text{pos}} = 18$, $\gamma_{\text{col}} = 4$ and $\tau_{\text{col}} = 20$. Finally, in order to take the union of supports and penalize non-coincident regions, we can consider taking the maximum of weights

$$E(\mathbf{p}_1, \mathbf{p}_2{}^d) = \frac{\sum_{\mathbf{q}_1 \in N_1,\, \mathbf{q}_2{}^d \in N_2} w_{\text{pos}}(\mathbf{p}_1, \mathbf{q}_1)^2 \max\big(w_{\text{col}}(\mathbf{p}_1, \mathbf{q}_1), w_{\text{col}}(\mathbf{p}_2{}^d, \mathbf{q}_2{}^d)\big)\; e(\mathbf{q}_1, \mathbf{q}_2{}^d)}{\sum_{\mathbf{q}_1 \in N_1,\, \mathbf{q}_2{}^d \in N_2} w_{\text{pos}}(\mathbf{p}_1, \mathbf{q}_1)^2 \max\big(w_{\text{col}}(\mathbf{p}_1, \mathbf{q}_1), w_{\text{col}}(\mathbf{p}_2{}^d, \mathbf{q}_2{}^d)\big)}. \quad (3.23)$$

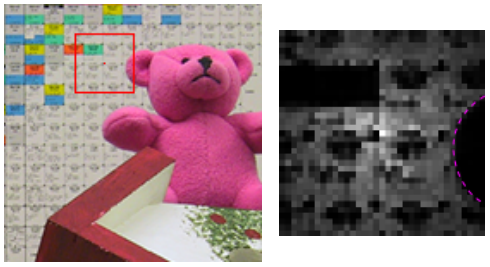
For this approach the best results are achieved by using $r = 14$, $\gamma_{\text{pos}} = 14$, $\gamma_{\text{col}} = 4$ and $\tau_{\text{col}} = 15$.

Figure 3.14 shows that near depth discontinuities, the best combination is the product: it puts a tiny weight on the region occupied by the ear of the teddy bear in left or right image. A program included in the source code archive, `show_weights`, permits to output the weighted patches with different combinations.

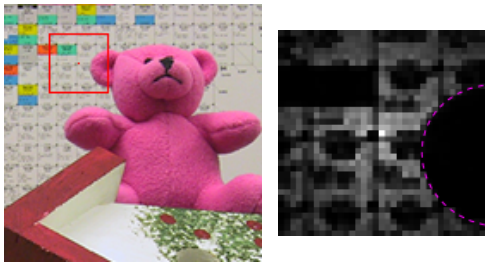
Visually, the results for the other alternatives are really similar to the ones given by the product (Figure 3.15). However, when considering the error (Table 3.4), the asymmetric and maximum combinations are a little worse in the discontinuities than the product and the sum (except for *Tsukuba*). Surprisingly, using asymmetric weights does not reduce the accuracy as much as we could expect and it proves to be a good substitute if we need to speed up computations. Nevertheless, the maximum has a worse performance than the other two variants, giving many errors and low accuracy in depth discontinuities.

	Tsukuba			**Venus**			**Teddy**			**Cones**		
	non	all	disc	non	all	disc	non	all	disc	non	all	disc
Product	1.86	2.27	6.61	0.65	1.02	3.15	6.56	14.4	15.5	2.48	8.81	6.91
Asymmetric	1.95	2.41	7.99	0.74	1.42	8.12	6.90	14.6	17.0	3.21	9.90	9.03
Sum	2.28	2.68	8.56	0.65	1.11	5.03	6.82	14.6	16.8	3.01	9.65	8.49
Maximum	2.74	3.14	10.8	1.62	2.23	11.7	7.72	15.2	18.6	3.38	9.72	9.35

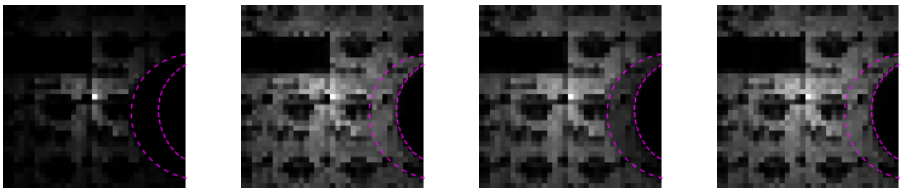
Table 3.4 – Error percentages on Middlebury benchmark for different weight combination functions.



Patch and weights centered at pixel (307,44) in *Teddy* left image.

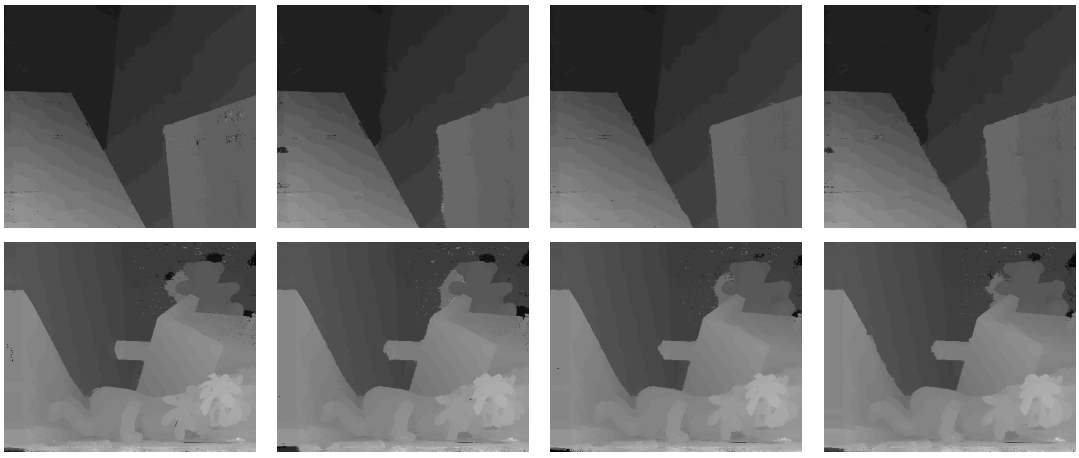


Patch and weights centered at the corresponding matching pixel (292,44) in *Teddy* right image.



(a) Product. (b) Asymmetric. (c) Sum. (d) Maximum.

Figure 3.14 – Example of the resulting weights for the matching cost computation of two patches using different combinations of weights.



(a) Product. (b) Asymmetric. (c) Sum. (d) Maximum.

Figure 3.15 – Disparity maps (with post-processing) comparison using diverse weights combinations for *Venus* and *Teddy* images.

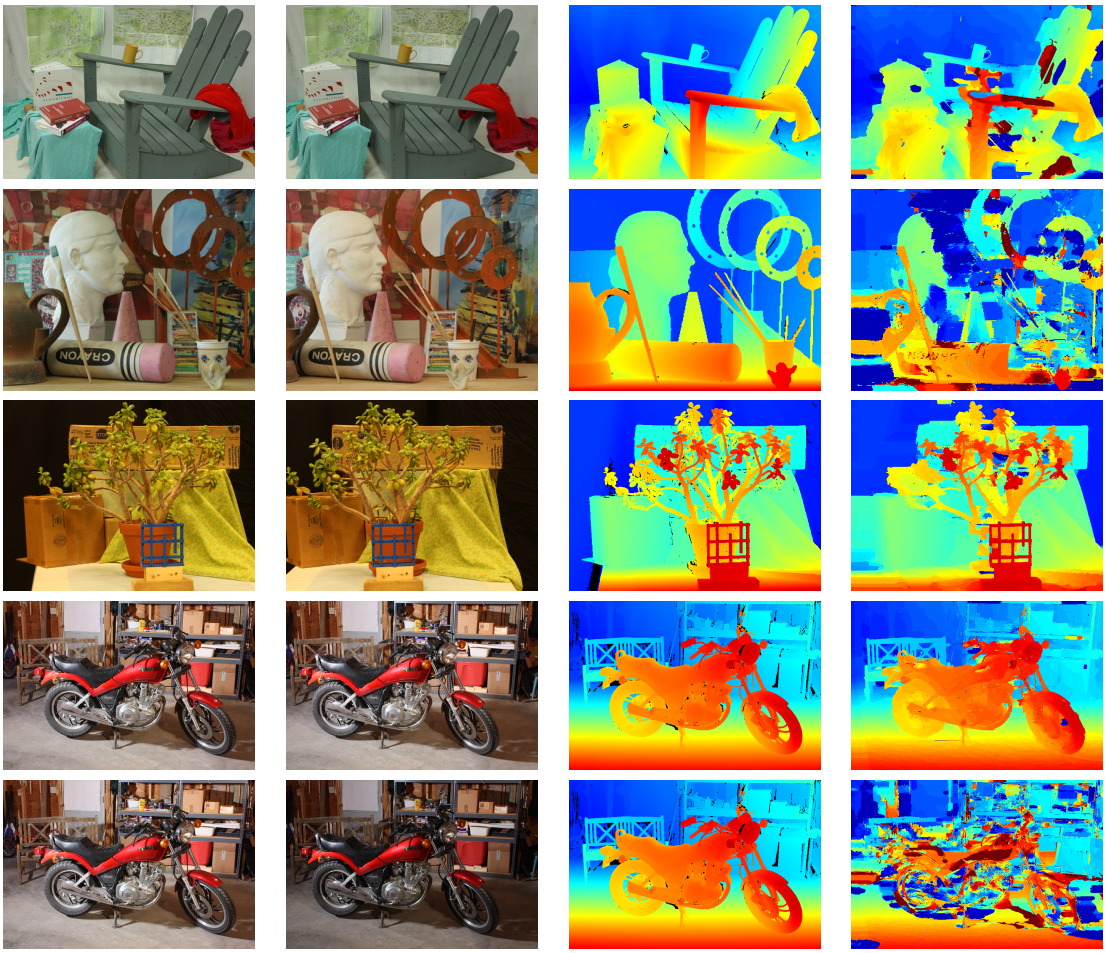
3.8 Other Examples

We have tested the algorithm using the default parameters with images from other Middlebury datasets and obtained satisfying results. Some examples for the images of Middlebury 2005 and 2006 datasets [SP07] are presented in Figure 3.16. The method is able to recover fine structures, for instance, most of the bars, brushes and holes in *Art* image are respected. In addition, homogeneous and textured regions are not a problem in most of the cases (*Flowerpots* and *Baby3* images). Nonetheless, we get some errors in the *Laundry* image in the region of the laundry basket.

Left image Ground truth Our result

Figure 3.16 – Resulting disparity maps for other images (from top to bottom: Art, Laundry from Middlebury 2005; Flowerpots, Baby3 from Middlebury 2006).

In Figures 3.17, 3.18 and 3.19 the resulting disparity maps for all training images of the new Middlebury 2014 datasets [SHK+14] are shown. Both left and right images are presented to visualize some changes and the resulting disparity map is shown as a colorized depth map since it is the chosen representation for the provided ground truths. Most image pairs of the dataset are not perfectly rectified (except *PlaytableP*) which makes the resulting disparity maps have some errors. What is more, some of the images have changes in lighting and exposure (*ArtL, MotorcycleE, PianoL*) which our algorithm does not account for. In consequence the results are completely erroneous in those cases. For the rest of the images, where no challenging light change is present, our algorithm is able to give a good estimation of the disparity map. The different results obtained by *Piano* and *PianoL* or *Motorcycle* and *MotorcycleE* exemplify how the change of lighting and exposure affects our algorithm success.



Left image | Right image | Ground truth | Our result

Figure 3.17 – Resulting disparity maps for new datasets (from top to bottom: Adirondack, ArtL, Jadeplant, Motorcycle and MotorcycleE).

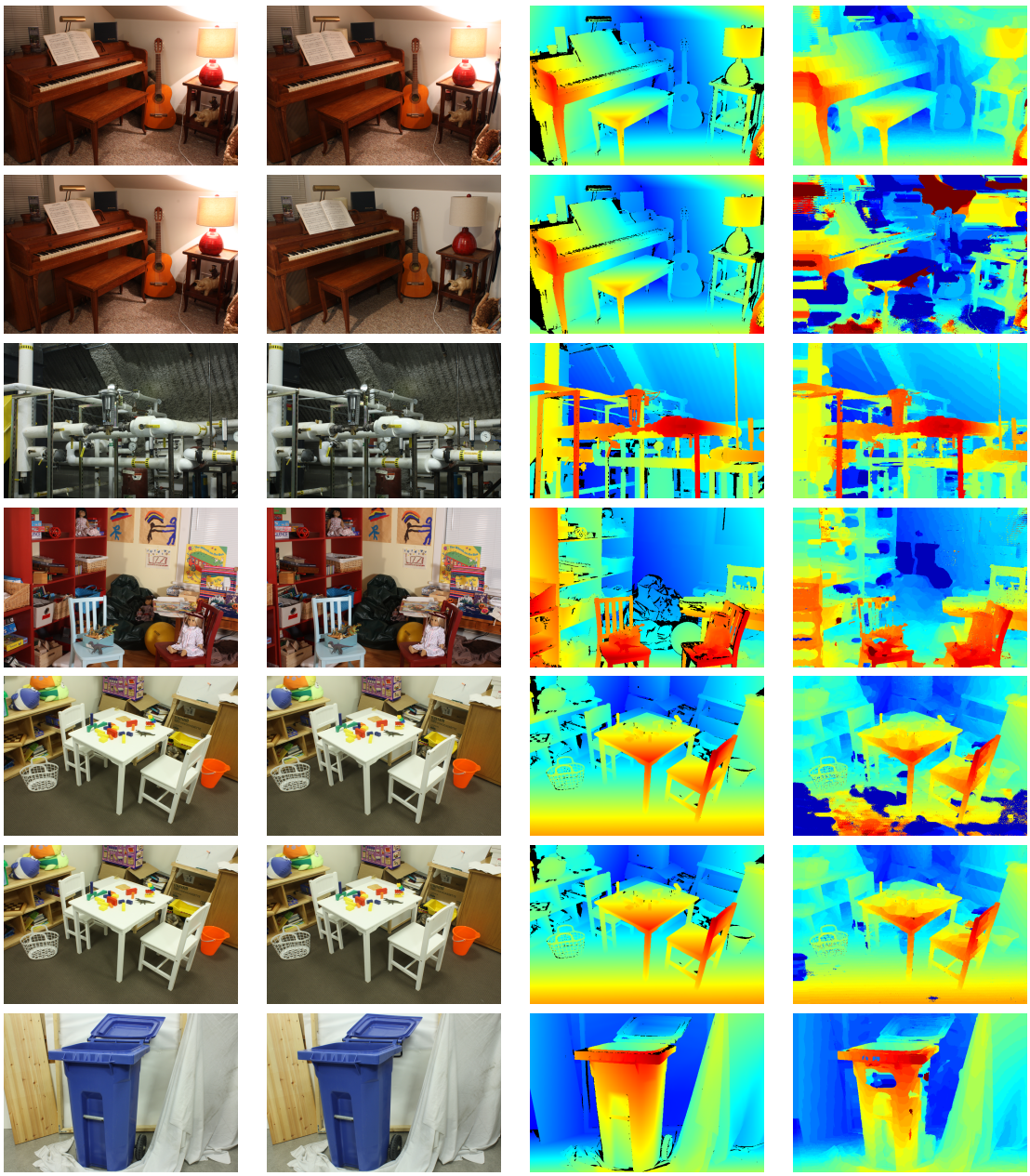
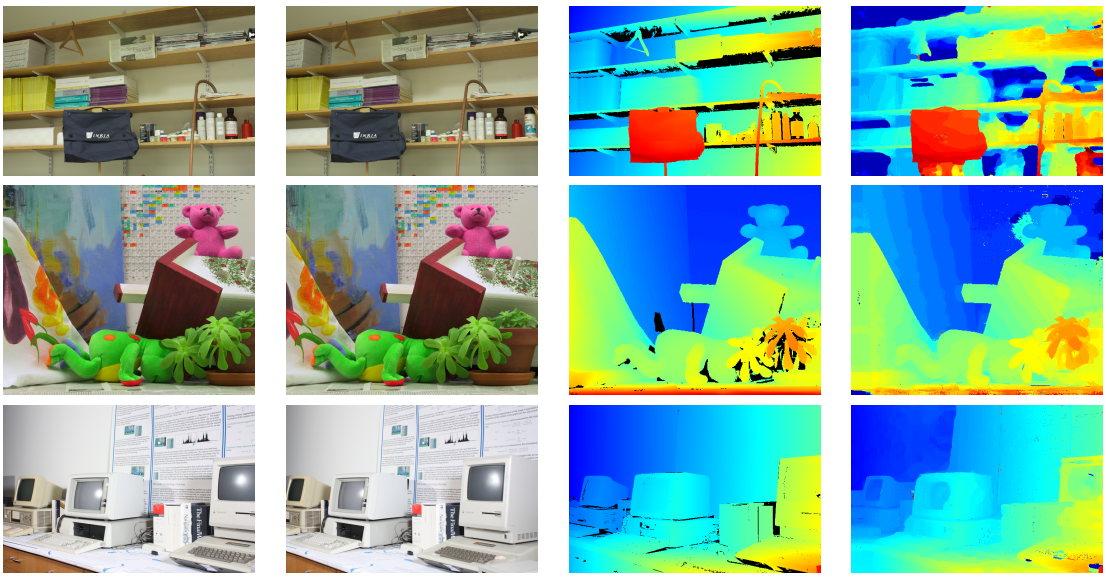


Figure 3.18 – Resulting disparity maps for new datasets (from top to bottom: Piano, PianoL, Pipes, Playroom, Playtable, PlaytableP and Recycle).



Left image | Right image | Ground truth | Our result

Figure 3.19 – Resulting disparity maps for new datasets (from top to bottom: Shelves, Teddy and Vintage).

3.9 Other Adaptive Windows for Stereo Matching

The good results of the bilateral weights inspired us to explore other adaptive windows and weights for local stereo matching, with the goal of improving the accuracy or computational time of Yoon and Kweon's algorithm [YK06]. Several approaches were considered either to define different weights from the ones in equation (3.4) or replace the taken neighborhood altogether. On a first step, we turned to General Adaptive Neighborhoods as defined by Debayle and Pinoli in [DP06]. The second approach was using the Tree of Shapes developed by Ballester, Caselles and Monasse [BCM03] to define new distance between pixels in the same image in order to describe new weights.

The weak General Adaptive Neighborhood (GAN) of a point $\mathbf{p} \in I$ of a gray image is the path-connected set of points $\mathbf{q} \in I$ verifying $|I(\mathbf{p}) - I(\mathbf{q})| < m$ for a tolerance $m \in [0, 255]$, we note it $V_m^I(\mathbf{p})$ (a similar definition is given for color images). Three different strategies using GANs were tested: GANs as an adaptive window, GAN-based distance to compute new weights and GAN shape matching as dissimilarity measure. The second one was the most successful obtaining similar accuracy results than the Bilateral Weights (refer to Table 3.5 and Figure 3.20), but the computation time was multiplied by a factor of 10. The new weights are defined using the weak GAN metric [PD11]

$$dV^I(\mathbf{p}, \mathbf{q}) = \inf_{m \in [0,255]} \{m \mid \mathbf{p} \in V_m^I(\mathbf{q})\} \tag{3.24}$$

which replaces the color distance $\Delta c_{\mathbf{pq}}$ in the similarity strength weight (3.6),

$$w(\mathbf{p}, \mathbf{q}) = \exp\Big(- \Big(\frac{dV^I(\mathbf{p}, \mathbf{q})}{\gamma_{\text{col}}} + \frac{\Delta g_{\mathbf{pq}}}{\gamma_{\text{pos}}} \Big) \Big) \ . \tag{3.25}$$

The Tree of Shapes of a gray image I is a graph representing the shapes of the image and its relationships of inclusion. The nodes are the *shapes*, the connected components of inferior or superior gray level sets, filling interior "holes". The edges connect each shape with the shapes that it contains in the image (*children*) and the shapes in which is contained (*parent*). A distance $d_{\text{tree}}(S_1, S_2)$ between shapes can be defined by counting the number of edges of the shortest path between the shapes in graph. With this distance, it is straightforward to define a distance between points in the same image as the distance between the smallest shapes containing each point, $d_{\text{tree}}(\mathbf{p}, \mathbf{q}) = d_{\text{tree}}(S(\mathbf{p}), S(\mathbf{q}))$. We used this distance (a variation combining the distances in the three color channels) to substitute the color distance $\Delta c_{\mathbf{pq}}$ in the similarity strength weight (3.6) for the disparity computation using support-weights. The final weights are

$$w(\mathbf{p}, \mathbf{q}) = \exp\Big(- \Big(\frac{d_{\text{tree}}(\mathbf{p}, \mathbf{q})}{\gamma_{\text{col}}} + \frac{\Delta g_{\mathbf{pq}}}{\gamma_{\text{pos}}} \Big) \Big) \ . \tag{3.26}$$

The accuracy results on the Middlebury images using this weights were slightly better, specially in depth discontinuities, than the Bilateral Weighted patches (see Table 3.5 and Figure 3.20). Moreover, the computation time of our implementation was comparable to the implementation of the bilateral weights.

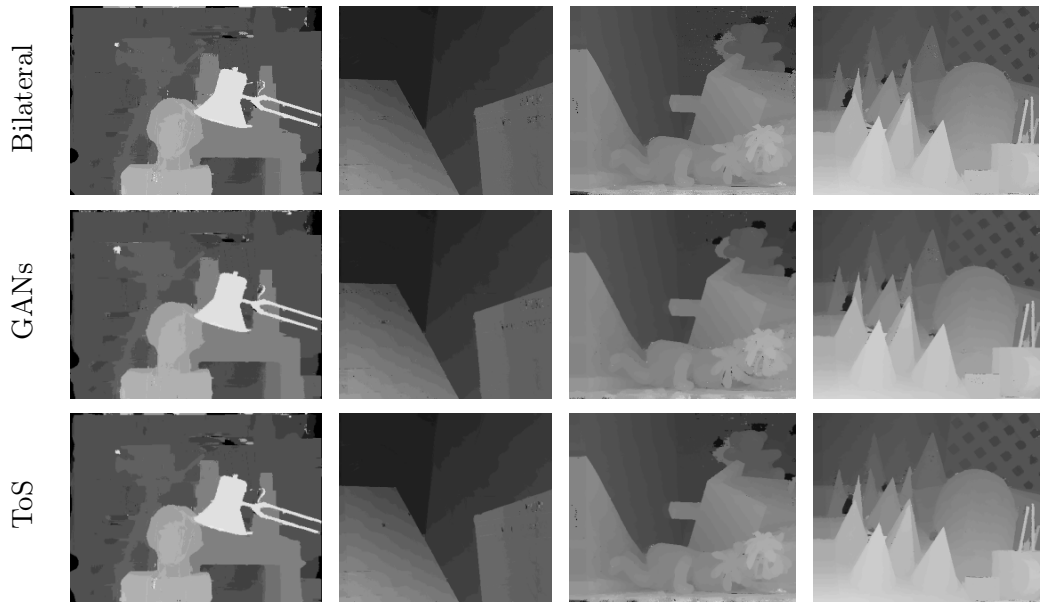


Figure 3.20 – Comparison of different weights for the adaptive-support disparity computation for the Middlebury benchmark images. Bilateral weights, weights based on weak GAN metric (GANs) and weights based on the Tree of Shapes distance (ToS).

	Tsukuba			**Venus**			**Teddy**			**Cones**		
	non	all	disc	non	all	disc	non	all	disc	non	all	disc
Bilateral Weights	1.86	2.27	6.61	0.650	1.020	3.15	6.56	14.4	15.5	2.48	8.81	6.91
GAN weights	1.82	2.18	6.79	0.580	1.040	3.30	6.50	14.2	15.9	**2.26**	8.71	**6.26**
Tree of Shapes	**1.72**	**2.11**	**5.66**	**0.465**	**0.769**	**2.68**	**6.02**	**13.8**	**14.6**	2.28	**8.57**	6.54

Table 3.5 – Error percentages on Middlebury benchmark for other adaptive weights methods we have tried.

3.10 Conclusion

We have presented a study and implementation of Yoon and Kweon's method [YK06] for disparity map computation. This adaptive support-weight approach was the first published block-matching method that put weights in the blocks. Others have followed, but a survey [HBG13] found that the method of Yoon and Kweon [YK06] yields the best results. Very similar results, but with much faster computation, can be obtained using a guided filter of the cost volume [TM14], which can also be tested in IPOL [TM14]. Hosni et al. [HBG10] propose to use an explicit segmentation of the image in order to speed up computations. Moreover, one of the best performing stereo matching methods [MSZ$^+$11] uses a faster approximation of the presented algorithm that selects cross-shaped neighborhoods. However, it remains to be seen in what measure the better results are due to the adaptive neighborhoods on their own.

Chapter 4

Trifocal Tensor for 3-View Pose Estimation

The natural extension after the study of the geometry of two views is to consider three views and analyze the constraints between points to find a similar operator to the fundamental matrix, the solution is the trifocal tensor. In this chapter we explore the advantages offered by the trifocal tensor in the pose estimation of a triplet of cameras as opposed to computing the relative poses pair by pair with the fundamental matrix. Theoretically, the trilinearities characterize uniquely three corresponding image points in a tighter way than the three epipolar equations and this translates in an increasing accuracy. However, we show that this initial improvement is not enough to have a remarkable impact on the pose estimation after bundle adjustment, and the use of the fundamental matrix with image triplets remains relevant.

Contents

4.1 Introduction

As seen in Section 2.4, the perspective projection induced by pinhole cameras gives the constraints between the space points and their projections onto the images. Taking two images, the fundamental matrix is an algebraic operator encoding the relation between corresponding image points, which gives a way to infer the relative orientations and positions of a pair of camera viewpoints.

When taking three views of the same scene, there exists an equivalent operator to the fundamental matrix describing the geometric constraints linking the three views: the trifocal tensor; the algebraic constraints relating three corresponding image points are known as trilinearities. It was shown that a general multi-view matrix can be found for n views, but that the relations given by these n views depend only on the constraints involving two or three views at a time [MHV$^+$04]. Theoretically, no extra geometric information about three views comes from considering additional views at once. Therefore, multi-view structure from motion pipelines always rely on initial view pairs [MMM12b, SF16, Sna10] or triplets [HTKP09, MMM13b].

The conventional wisdom advocates the use of the trifocal tensor with a triplet of views rather than taking pairs and the fundamental matrix. We question this assumption with a study of the trifocal tensor and its performance against the fundamental matrix. Moreover, the main possible parameterizations of the tensor are described, analyzed and tested to find the advantages and disadvantages offered by each one.

4.2 Definition of The Trifocal Tensor

The **Trifocal Tensor** (written TFT for short) associated to three views is a $3 \times 3 \times 3$ tensor $\mathbf{T} = [T_1, T_2, T_3]$ usually defined for three cameras in canonical[1] configuration with projection matrices $P_1 = (Id_3|0)$, $P_2 = (A|a_4)$, $P_3 = (B|b_4)$ with each slice T_i the 3×3 matrix

$$T_i = a_i b_4^\top - a_4 b_i^\top, \tag{4.1}$$

where a_i and b_i are the columns of A and B. A more general definition for non canonical cameras can be found in [HZ04].

From a trifocal tensor $\mathbf{T}$ we can extract the epipoles: the epipole e_{31} can be computed as the common intersection of the lines represented by the right null-vectors of T_1, T_2 and T_3 and, analogously, epipole e_{21} can be computed as the common intersection of the lines represented by the left null-vectors of T_1, T_2 and T_3.

The TFT has 27 parameters, is unique up-to-scale for any 3-view configuration and invariant by projectivity. Still, the degrees of freedom of a set of three projective cameras up-to-projectivity is 18 [HZ04]. Hence, the parameters of the trifocal tensor must satisfy some constraints reducing the 8 remaining degrees of freedom of the trifocal tensor. However, the missing constraints are not obvious nor easily derivable. Section 4.3 focuses on presenting and analyzing several minimal parameterizations and constraints developed over the years.

[1] Any triplet of projection matrices can be brought to canonical configuration by projectivity, that is a transformation of the projective space.

Trilinearities

At its origin, the TFT is derived from the relation between the projections of the same 3D line in the three images. Other incidence relations can be found for this tensor, in particular, the following equation for triplets of corresponding image points $\bar{\mathbf{x}}_1$, $\bar{\mathbf{x}}_2$, $\bar{\mathbf{x}}_3$ is satisfied:

$$[\bar{\mathbf{x}}_2]_\times \Big(\sum_{i=1}^{3} (\bar{\mathbf{x}}_1)_i T_i \Big) [\bar{\mathbf{x}}_3]_\times = 0_{3\times 3} \ . \tag{4.2}$$

Among the 9 scalar equations in (4.2), only 4 are linearly independent. They are linear on the trifocal tensor parameters and trilinear on the image coordinates.

Remark 1. *Considering the views pairwise, the incidence relations given by the fundamental matrices for the same corresponding triplet $\bar{\mathbf{x}}_1$, $\bar{\mathbf{x}}_2$, $\bar{\mathbf{x}}_3$ are the set of 3 epipolar equations* (2.8)*, linear on the fundamental matrices parameters and bilinear on the image points*

$$\bar{\mathbf{x}}_2^\top F_{21} \bar{\mathbf{x}}_1 = 0, \qquad \bar{\mathbf{x}}_3^\top F_{31} \bar{\mathbf{x}}_1 = 0, \qquad \bar{\mathbf{x}}_3^\top F_{32} \bar{\mathbf{x}}_2 = 0 \ . \tag{4.3}$$

The involved fundamental matrices are $F_{21} = [a_4]_\times A$, $F_{31} = [b_4]_\times B$, $F_{32} = [b_4 - BA^{-1}a_4]_\times BA^{-1}$.

4.3 Minimal Parameterizations and Constraints

Many possible minimal characterizations for the trifocal tensor have been proposed in the literature [Can00, FP98, Nor09, PF98, PH14, Res02, TZ97]. We chose to focus on four representative ones that can be efficiently implemented in the pose estimation process.

4.3.1 Correlation Slices - C. Ressl

The minimal parameterization of the trifocal tensor proposed by Ressl in his thesis [Res02] is based on algebraic constraints of the correlation slices. It involves 20 parameters and 2 constraints. With this parameterization it is possible to completely characterize the trifocal tensor for three views. The three matrices of the trifocal tensor T_i can be parameterized in the following minimal form:

$$T_i = \begin{bmatrix} s_i, & v s_i + m_i e_{31}, & w s_i + n_i e_{31} \end{bmatrix}^\top \quad i = 1,2,3 \tag{4.4}$$

where $s_i \in \mathbb{R}^3$ are such that $\left\| \begin{pmatrix} s_1 & s_2 & s_3 \end{pmatrix} \right\| = 1$, $e_{31} \in \mathbb{R}^3$ with $\|e_{31}\| = 1$, and $v, w, m_i, n_i \in \mathbb{R}$.

This parameterization is directly related to the epipoles since $e_{31} = b_4$ corresponds to the epipole, projection of the first camera center in the third image, and the epipole in the second image $e_{21} = a_4$ is proportional to $(1, v, w)^\top$. It is also related to an equivalent parameterization of three canonical projective matrices.

From a trifocal tensor $\mathbf{T} = [T_1, T_2, T_3]$ we can find the parameters of Ressl by extracting the epipoles e_{21} and e_{31} (with norm 1), then v and w are found by dividing e_{21} by its first coordinate $(1, v, w)^\top = \frac{e_{21}}{(e_{21})_1}$. The vectors s_i and the scalars m_i, n_i can be computed

$$s_i^\top = \frac{T_i^{1\cdot}}{\lambda}, \qquad m_i = (\frac{T_i^{2\cdot}}{\lambda} - v\, s_i^\top)\, e_{31}, \qquad n_i = (\frac{T_i^{3\cdot}}{\lambda} - w\, s_i^\top)\, e_{31}, \qquad i = 1,2,3 \tag{4.5}$$

where $\lambda = \|(T_1^{1\cdot}, T_2^{1\cdot}, T_3^{1\cdot})\|$ and $T_i^{j\cdot}$ is the j-th row of the i-th correlation slice.

4.3.2 Orthogonal Matrices - K. Nordberg

The trifocal tensor can also be parameterized by three 3×3 orthogonal matrices U, V and W that transform the original tensor into a sparse one, $\tilde{\mathbf{T}}$, with only 10 non-zero parameters up-to-scale [Nor09]:

$$\tilde{\mathbf{T}} = \mathbf{T}(U \otimes V \otimes W) \quad \Rightarrow \quad \mathbf{T} = \tilde{\mathbf{T}}(U^\top \otimes V^\top \otimes W^\top) \tag{4.6}$$

where the tensor operation corresponds to the matrix operation on the slices $\tilde{T}_i = V^\top(\sum_m U_{m,i} T_m)W$. The scale can be fixed by imposing $\|\tilde{\mathbf{T}}\| = 1$. For canonical cameras, such orthogonal matrices can be computed as:

$$U_0 = (A^{-1}a_4,\ [A^{-1}a_4]_\times{}^2 B^{-1}b_4,\ [A^{-1}a_4]_\times B^{-1}b_4), \qquad U = U_0(U_0^\top U_0)^{-\frac{1}{2}} \tag{4.7}$$

$$V_0 = (a_4,\ [a_4]_\times AB^{-1}b_4,\ [a_4]_\times{}^2 AB^{-1}b_4), \qquad V = V_0(V_0^\top V_0)^{-\frac{1}{2}} \tag{4.8}$$

$$W_0 = (b_4,\ [b_4]_\times BA^{-1}a_4,\ [b_4]_\times{}^2 BA^{-1}a_4), \qquad W = W_0(W_0^\top W_0)^{-\frac{1}{2}} \tag{4.9}$$

and each one can be parameterized by 3 parameters. Therefore, the trifocal tensor $\mathbf{T}$ is parameterized in this case by a total of 19 parameters and one constraint fixing the scale of $\tilde{\mathbf{T}}$.

A main disadvantage of this specific parameterization is that the matrices U_0, V_0 and W_0 become singular when the three camera centers are collinear and, therefore, no orthogonal matrix can be computed from them. It is then a parameterization only valid for non-collinear centers.

4.3.3 Determinants on the Slices - O. D. Faugeras and T. Papadopoulo

In [FP98] a set of 12 algebraic equations are presented as sufficient constraints to characterize a trifocal tensor. It consists of 3 constraints of degree 3 corresponding to the determinant of the slices being zero, $|T_i| = 0$ for $i \in \{1, 2, 3\}$, and 9 more constraints of degree 6 combining several determinants of the elements of $\mathbf{T}$, for $j_1, j_2, k_1, k_2 \in \{1, 2, 3\}$ with $j_1 \neq j_2,\ k_1 \neq k_2$

$$\begin{aligned} &|t_\cdot^{j_1 k_1}\ t_\cdot^{j_1 k_2}\ t_\cdot^{j_2 k_2}|\,|t_\cdot^{j_1 k_1}\ t_\cdot^{j_2 k_1}\ t_\cdot^{j_2 k_2}| - \\ &\quad |t_\cdot^{j_2 k_1}\ t_\cdot^{j_1 k_2}\ t_\cdot^{j_2 k_2}|\,|t_\cdot^{j_1 k_1}\ t_\cdot^{j_2 k_2}\ t_\cdot^{j_1 k_2}| = 0 \end{aligned} \tag{4.10}$$

where $t_\cdot^{jk}$ represents the vector $\left(T_1^{jk}, T_2^{jk}, T_3^{jk}\right)^\top$.

This set is not minimal since only 9 constraints should be enough for the characterization of a valid trifocal tensor. The authors give an outline of how to obtain a minimal parameterization using the constraints that requires to solve a polynomial of degree 2, thus giving two possible tensors. We considered best to use the minimization of the constraints instead of the minimal parameters for a more straightforward implementation.

4.3.4 Π matrices - J. Ponce and M. Hebert

A completely different approach to characterize the 3-view model has been explored in [PH14]. Through the study on the incidence of three lines on space, a set of three matrices (related to the principal lines) that give constraints on the correspondence of three image points can be defined. These matrices have a total of 27 parameters and play a role similar to the TFT.

Similarly to the parameterization of the trifocal tensor by Nordberg, the main drawback of the Π matrices is that they are only valid for non-collinear camera centers. However, Ponce and Hebert[PH14] also proposed equivalent matrices with one extra trilinear constraint for collinear camera centers. We describe both cases below working in the projective space.

Non-Collinear Cameras

Given three cameras P_1, P_2, P_3, with non-collinear centers we can consider, without loss of generality, that the camera centers are positioned on the fundamental points of the projective space, $\bar{\mathbf{C}}_1 = (1,0,0,0)^\top$, $\bar{\mathbf{C}}_2 = (0,1,0,0)^\top$, $\bar{\mathbf{C}}_3 = (0,0,1,0)$. We can define the following three 4×3 matrices up-to-scale

$$\Pi_1 = \begin{pmatrix} \diamondsuit_1^\top \\ \pi_{21}^\top \\ \pi_{31}^\top \\ \pi_{41}^\top \end{pmatrix} \quad \Pi_2 = \begin{pmatrix} \pi_{12}^\top \\ \diamondsuit_2^\top \\ \pi_{32}^\top \\ \pi_{42}^\top \end{pmatrix} \quad \Pi_3 = \begin{pmatrix} \pi_{13}^\top \\ \pi_{23}^\top \\ \diamondsuit_3^\top \\ \pi_{43}^\top \end{pmatrix} \quad \text{such that } \; P_i\Pi_i \sim \text{Id } \forall i \tag{4.11}$$

where $\diamondsuit_i \in \mathbb{R}^3$ represent undetermined vectors and can all be fixed to $(0,0,0)^\top$.

These matrices uniquely characterize the triplets of corresponding image points. For three image points $\bar{\mathbf{x}}_1$, $\bar{\mathbf{x}}_2$, $\bar{\mathbf{x}}_3$, the following equations are verified if, and only if, they form a triplet of corresponding points

$$\bar{\mathbf{x}}_1^\top(\pi_{41}\pi_{32}^\top - \pi_{31}\pi_{42}^\top)\bar{\mathbf{x}}_2 = 0 \tag{4.12}$$

$$\bar{\mathbf{x}}_1^\top(\pi_{41}\pi_{23}^\top - \pi_{21}\pi_{43}^\top)\bar{\mathbf{x}}_3 = 0 \tag{4.13}$$

$$\bar{\mathbf{x}}_2^\top(\pi_{42}\pi_{13}^\top - \pi_{12}\pi_{43}^\top)\bar{\mathbf{x}}_3 = 0 \tag{4.14}$$

$$(\pi_{21}^\top\bar{\mathbf{x}}_1)(\pi_{32}^\top\bar{\mathbf{x}}_2)(\pi_{13}^\top\bar{\mathbf{x}}_3) = (\pi_{31}^\top\bar{\mathbf{x}}_1)(\pi_{12}^\top\bar{\mathbf{x}}_2)(\pi_{23}^\top\bar{\mathbf{x}}_3) \; . \tag{4.15}$$

To reduce the parameters of the Π matrices, Ponce and Hebert propose the 6 homogeneous constraints:

$$\pi_{21}^1 \; = \; \pi_{32}^2 \; = \; \pi_{13}^3 = 0, \quad \pi_{31}^2 = \pi_{41}^3, \quad \pi_{12}^3 = \pi_{42}^1, \quad \pi_{23}^1 = \pi_{43}^2 \tag{4.16}$$

that can be achieved by a projective transformation of the space. This reduces the parameters to 21 and with 3 norm constraints on the matrices, $\|\Pi_i\| = 1$, the minimal representation is attained.

However, the homogeneous constrains (4.16) cannot be achieved by all configurations of non-collinear cameras. In fact, in [TPH16]-Lemma C1 a subset of 3-view configurations of non-collinear cameras denoted as *general configurations* is defined to exclude all configurations that cannot achieve the minimal parameterization. A configuration is general when

none of the principal lines of each camera (three lines per camera, that go through the center $\mathbf{C}_i$ with direction $\vec{i}_i$, $\vec{j}_i$ and $\vec{k}_i$) intersect (in the projective space) with any principal line from another camera.

The scenes excluded from this group are many and not unusual in the pose estimation task. For instance, all three cameras pointing at the same point in space or a baseline perpendicular to one of the camera planes are configurations not fulfilling the conditions for the minimal parameterization of the Π matrices. For that reason we propose new constraints to achieve a more inclusive minimal parameterization.

Proposition 1. *The following 9 quadratic constraints are sufficient and restrict the 27 parameters of the Π_i matrices to the theoretical 18 degrees of freedom, since they also fix the scale of each Π_i matrix:*

$$\|\pi_{41}\| = 1 \quad \|\pi_{42}\| = 1 \quad \|\pi_{43}\| = 1 \tag{4.17}$$

$$\|\pi_{21}\| = 1 \quad \pi_{21}^\top \cdot \pi_{41} = 0 \tag{4.18}$$

$$\|\pi_{32}\| = 1 \quad \pi_{32}^\top \cdot \pi_{42} = 0 \tag{4.19}$$

$$\|\pi_{13}\| = 1 \quad \pi_{13}^\top \cdot \pi_{43} = 0 \tag{4.20}$$

Proof. Let Π_i be three matrices as defined in (4.11). To prove Proposition 1 we need to find a projectivity Q transforming the Π_i matrices so that they verify the constraints. Such a projectivity has to leave the camera centers invariants, given that they have to coincide with the three first fundamental points for the definition of the Π_i matrices to be still valid. Any projectivity Q leaving the camera centers fixed on the fundamental points will have the following form

$$Q = \begin{pmatrix} \alpha & 0 & 0 & \kappa \\ 0 & \beta & 0 & \lambda \\ 0 & 0 & \gamma & \mu \\ 0 & 0 & 0 & \nu \end{pmatrix} . \tag{4.21}$$

The transformed matrices $Q \cdot \Pi_i$,

$$Q \cdot \Pi_1 = \begin{pmatrix} X_1^\top \\ \beta\pi_{21} + \lambda\pi_{41}^\top \\ \gamma\pi_{31}^\top + \mu\pi_{41}^\top \\ \nu\pi_{41}^\top \end{pmatrix} \quad Q \cdot \Pi_2 = \begin{pmatrix} \alpha\pi_{12}^\top + \kappa\pi_{42}^\top \\ X_2^\top \\ \gamma\pi_{32}^\top + \mu\pi_{42}^\top \\ \nu\pi_{42}^\top \end{pmatrix} \quad Q \cdot \Pi_3 = \begin{pmatrix} \alpha\pi_{13}^\top + \kappa\pi_{43}^\top \\ \beta\pi_{23}^\top + \lambda\pi_{44}^\top \\ X_3^\top \\ \nu\pi_{43}^\top \end{pmatrix} \tag{4.22}$$

will be valid matrices verifying the definition (4.11) for the new camera matrices $P_i \cdot Q^{-1}$.

We can first fix the scale of these matrices by scaling each one of them so that (4.17) is verified. In addition, the scale of Q can be fixed by letting $\nu = 1$ so that the norm of vectors π_{4i} does not change. Then, β and λ can be determined by imposing $\|\beta\pi_{21} + \lambda\pi_{41}\| = 1$ and $(\beta\pi_{21} + \lambda\pi_{41})^\top \pi_{41} = 0$ so that $\Pi_1' = Q\Pi_1$ verifies (4.18). γ and μ can be determined by imposing $\|\gamma\pi_{32} + \mu\pi_{42}\| = 1$ and $(\gamma\pi_{32} + \mu\pi_{42})^\top \pi_{42} = 0$, so that $\Pi_2' = Q\Pi_2$ verifies (4.19). And lastly, α and κ can be determined by imposing $\|\alpha\pi_{13} + \kappa\pi_{43}\| = 1$ and $(\alpha\pi_{13} + \kappa\pi_{43})^\top \pi_{43} = 0$ so that $\Pi_3' = Q\Pi_3$ verifies (4.20). □

Just like with the trilinearities (4.2) in the trifocal tensor case, these parameters give 4 equations describing the incidence relation for image points. Here, (4.12) to (4.14) are

bilinear on the points and completely equivalent to the epipolar equations given by the fundamental matrices. Equation (4.15) is trilinear on the image points and it is key to the characterization of the correspondence of three points, which the fundamental matrices fail to achieve when one of the points lies on the line joining two epipoles. This is precisely the geometric contribution of taking three views instead of individual pairs to the characterization of matches.

Collinear Cameras

For three collinear cameras we can no longer position the centers onto the fundamental points. In this case we transform the projective space so that two camera centers are positioned in the two first fundamental points and the third one in the same line, $\bar{\mathbf{C}}_1 = (1,0,0,0)^\top$, $\bar{\mathbf{C}}_2 = (0,1,0,0)^\top$, $\bar{\mathbf{C}}_3 = (1,1,0,0)^\top$. This fixes 7 of the 15 degrees of freedom of the projective framework.

In such a projective space, we can define three 4×3 matrices up-to-scale

$$\Pi_1 = \begin{pmatrix} \lozenge_1^\top \\ \pi_{21}^\top \\ \pi_{31}^\top \\ \pi_{41}^\top \end{pmatrix} \quad \Pi_2 = \begin{pmatrix} \pi_{12}^\top \\ \lozenge_2^\top \\ \pi_{32}^\top \\ \pi_{42}^\top \end{pmatrix} \quad \Pi_3 = \begin{pmatrix} \lozenge_3^\top \\ \lozenge_3^\top + \omega_3^\top \\ \pi_{33}^\top \\ \pi_{43}^\top \end{pmatrix} \quad \text{such that } P_i\Pi_i \sim \mathrm{Id}\ \forall i \tag{4.23}$$

where $\lozenge_i \in \mathbb{R}^3$ are undetermined vectors that can all be fixed to $(0,0,0)^\top$.

These matrices uniquely characterize the triplets of corresponding image points. For three image points $\bar{\mathbf{x}}_1$, $\bar{\mathbf{x}}_2$, $\bar{\mathbf{x}}_3$, the following equations are verified if, and only if, they form a triplet of corresponding points

$$\bar{\mathbf{x}}_1^\top \left(\pi_{41}\pi_{32}^\top - \pi_{31}\pi_{42}^\top\right) \bar{\mathbf{x}}_2 = 0 \tag{4.24}$$

$$\bar{\mathbf{x}}_1^\top \left(\pi_{41}\pi_{33}^\top - \pi_{31}\pi_{43}^\top\right) \bar{\mathbf{x}}_3 = 0 \tag{4.25}$$

$$\bar{\mathbf{x}}_2^\top \left(\pi_{42}\pi_{33}^\top - \pi_{32}\pi_{43}^\top\right) \bar{\mathbf{x}}_3 = 0 \tag{4.26}$$

$$(\pi_{31}^\top\bar{\mathbf{x}}_1)(\pi_{32}^\top\bar{\mathbf{x}}_2)(\omega_3^\top\bar{\mathbf{x}}_3) + (\pi_{33}^\top\bar{\mathbf{x}}_3)[(\pi_{31}^\top\bar{\mathbf{x}}_1)(\pi_{12}^\top\bar{\mathbf{x}}_2) - (\pi_{21}^\top\bar{\mathbf{x}}_1)(\pi_{32}^\top\bar{\mathbf{x}}_2)] = 0 \tag{4.27}$$

$$(\pi_{41}^\top\bar{\mathbf{x}}_1)(\pi_{42}^\top\bar{\mathbf{x}}_2)(\omega_3^\top\bar{\mathbf{x}}_3) + (\pi_{43}^\top\bar{\mathbf{x}}_3)[(\pi_{41}^\top\bar{\mathbf{x}}_1)(\pi_{12}^\top\bar{\mathbf{x}}_2) - (\pi_{21}^\top\bar{\mathbf{x}}_1)(\pi_{42}^\top\bar{\mathbf{x}}_2)] = 0\ . \tag{4.28}$$

To reduce the parameters of the Π matrices, Ponce and Hebert propose the 8 homogeneous constraints:

$$\pi_{21}^1 = \pi_{31}^2 = \pi_{12}^1 = \pi_{42}^2 = 0, \quad \pi_{31}^3 = \pi_{21}^3, \quad \pi_{32}^3 = \pi_{42}^3, \quad \omega_3^1 = \omega_3^2 = \omega_3^3 \tag{4.29}$$

that can be achieved by a projective transformation of the space. This reduces the parameters to a total of 19, and along with the three norm constraints, $\|\Pi_i\| = 1$, the minimal representation of 16 d.o.f. is attained.

Just like in the non-collinear case, a subset of collinear configurations is defined in [TPH16]-Lemma D1 in order to exclude the configurations that cannot achieve the minimal parameterization. In the same way, this definition leaves out many configurations as common as two cameras pointing at the same point in space or parallel axes in two image planes. We propose new constraints to achieve a more inclusive minimal parameterization.

Proposition 2. *The following 11 quadratic constraints are sufficient and restrict the 27 parameters of the Π_i matrices to the theoretical 16 degrees of freedom, since they also fix the scale of each matrix:*

$$\|w\| = 1 \quad \|\pi_{33}\| = 1 \quad \|\pi_{43}\| = 1 \tag{4.30}$$

$$\|\pi_{33}\| = 1 \quad \|\pi_{43}\| = 1 \tag{4.31}$$

$$\pi_{21}^\top \cdot \pi_{31} = \pi_{21}^\top \cdot \pi_{41} = 0 \tag{4.32}$$

$$\pi_{12}^\top \cdot \pi_{32} = \pi_{12}^\top \cdot \pi_{42} = 0 \tag{4.33}$$

$$\pi_{31}^\top \cdot \pi_{41} = \pi_{32}^\top \cdot \pi_{42} = 0 \tag{4.34}$$

Proof. Let Π_i be three matrices as defined in (4.23). Similarly to the proof of Proposition 1 we need to find a projectivity leaving the camera centers invariant and transform the matrices Π_i to verify the constraints of the minimal parameterization. Any projectivity Q leaving the camera centers fixed will have the following form

$$Q = \begin{pmatrix} \alpha & 0 & \beta & \gamma \\ 0 & \alpha & \kappa & \lambda \\ 0 & 0 & \mu & \nu' \\ 0 & 0 & \delta' & \tau \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & \tau \end{pmatrix} \begin{pmatrix} \alpha & 0 & \beta & \gamma \\ 0 & \alpha & \kappa & \lambda \\ 0 & 0 & 1 & \nu \\ 0 & 0 & \delta & 1 \end{pmatrix} \quad . \tag{4.35}$$

All the parameters of the projectivity Q can be found by imposing the constraints of the proposition to the matrices $Q\,\Pi_i$ $i = 1, 2, 3$. First, the scale of Q can be fixed by imposing $\alpha = 1$. Then, imposing (4.32) is equivalent to imposing $(\pi_{21} + \kappa\pi_{31} + \lambda\pi_{41})^\top \pi_{31} = 0$ and $(\pi_{21} + \kappa\pi_{31} + \lambda\pi_{41})^\top \pi_{41} = 0$ which is enough to specify κ and λ. Analogously, imposing (4.33) is equivalent to imposing $(\pi_{12}+\beta\pi_{32}+\gamma\pi_{42})^\top \pi_{32} = 0$ and $(\pi_{12}+\beta\pi_{32}+\gamma\pi_{42})^\top \pi_{42} = 0$ which gives the values of β and γ. The parameters δ and ν are determined by imposing (4.34), that give equations $(\pi_{31}+\nu\pi_{41})^\top(\delta\pi_{31}+\pi_{41}) = 0$ and $(\pi_{32}+\nu\pi_{42})^\top(\delta\pi_{32}+\pi_{42}) = 0$. Finally, after fixing the scale of the matrices $Q\,\Pi_i$ $i = 1, 2, 3$ by imposing (4.30) the final parameters μ and τ can be determined by imposing (4.31). □

4.4 Pose Estimation of 3 views

From a trifocal tensor $\mathbf{T}$ we can extract the epipoles e_{31} and e_{21}, and then two fundamental matrices of the triplet can be computed:

$$F_{21} = [e_{21}]_\times [T_1 e_{31}, T_2 e_{31}, T_3 e_{31}] \ , \quad F_{31} = [e_{31}]_\times [T_1^\top e_{21}, T_2^\top e_{21}, T_3^\top e_{21}] \ . \tag{4.36}$$

From F_{21}, F_{31} and the calibration matrices K_1, K_2, K_3 the two essential matrices can be computed from which the relative orientations (R_{21}, t_{21}) and (R_{31}, t_{31}) (see Figure 4.1) can be retrieved by the singular value decomposition. Each translation vector is up to unknown scale. The overall scale is fixed by setting $\|t_{21}\| = 1$ and the relative scale λ of t_{31} can be computed by using a triangulation of the space points $\{X^n\}_n$ from the projections in the first two cameras and minimizing the algebraic error with respect to the third image:

$$\arg\min_{\lambda \in \mathbb{R}} \sum_{n=1}^{N} \left\| \bar{\mathbf{x}}_3^n \times \left(K_3 (R_{31} X^n + \lambda \frac{t_{31}}{\|t_{31}\|}) \right) \right\|^2 , \tag{4.37}$$

which admits a closed form solution.

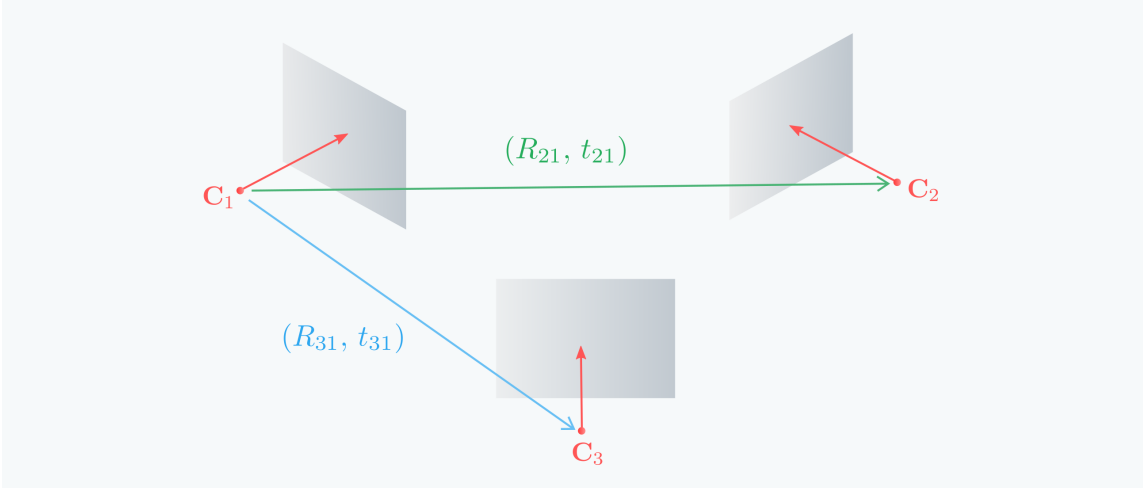


Figure 4.1 – Three views of the same scene and the relative motions of two pairs.

4.4.1 Linear Estimation of the Trifocal Tensor

The TFT can be estimated from a linear system given by the trilinearities of (4.2). From each triplet we get 9 equations linear on the parameters of the tensor, from which only 4 are linearly independent. At least 7 correspondences are needed to solve the linear system if we also impose $\|\mathbf{T}\| = 1$. If more triplets are available, a solution minimizing the algebraic error can be found by SVD. The resulting trifocal tensor will not necessarily be a valid tensor, i.e. it may not verify the internal constraints, not corresponding in consequence to any projective triplet of views. A valid trifocal tensor can be computed from the one found in the following way: extract the epipoles e_{21} and e_{31}, find matrices A and B that minimize (4.1) using the found tensor and epipoles (resulting in linear systems) and finally compute the valid trifocal tensor from the triplet of canonical projection matrices using the definition (4.1).

4.4.2 Optimization with Minimal Parameterization

Section 4.3 detailed four ways to parameterize minimally the 3-view model. All parameterizations involve non-linear constraints, so to be able to estimate the parameters an initialization is necessary. The linear solution from Section 4.4.1 can be used as an initial guess to estimate the different initial minimal parameterizations. Once the correct parameters of the initial model have been found they can be optimized by reinforcing the constraints and minimizing the Gold standard error (maximum likelihood estimator) with the Gauss-Helmert algorithm [Nei10]. The constrained least square problem to which we find a local optimum is

$$\underset{x,p}{\text{minimize}} \; \|x - x_0\|^2 \qquad \text{subject to} \qquad f(x,p) = 0, \qquad g(p) = 0 \tag{4.38}$$

where the vector p contains the parameters to be optimized (parameters of the trifocal tensor or fundamental matrix) and g are the constraints specific for each parameterization. The vector of observations x_0 corresponds to the triplets or pairs of matching points (concatenated) and the main constraints f are the trilinearities or epipolar equations where

both the observation and parameters p are involved. In Table 4.1 the parameters and constraints to use for each minimal parameterization are summarized, as well as the ones to use to optimize a fundamental matrix. A detailed description of the iterative method to solve the Gauss-Helmert model can be found in Appendix B.

parameterization	p	#	f	g	#
Ressl	s_i, m_i, n_i e_{31}, v, w	20	(4.2)	$\|(s_1, s_2, s_3)\| = 1$, $\|e_{31}\| = 1$	2
Nordberg	$\tilde{\mathbf{T}}, U, V, W$	19	(4.2)	$\|\tilde{\mathbf{T}}\| = 1$	1
Faug.-Papad.	$\mathbf{T}$	27	(4.2)	$\|T_i\| = 0$, (4.10)	12
Ponce-Hebert	Π_i	21	(4.12)–(4.15)	$\|\Pi_i\| = 1$	3
Fundamental	F_{21}	9	(4.3)	$\|F_{21}\| = 1$, $\|F_{21}\| = 0$	2

Table 4.1 – Parameters and constraints to use in the Gauss-Helmert algorithm for the different minimal parameterizations of the 3-view model and the 2-view model.

4.4.3 Optimization with Bundle Adjustment

To refine the estimated orientations given by the trifocal tensor, the common last step of Bundle Adjustment can be applied. It minimizes the square reprojection error over the possible cameras orientations and space points (see Section 2.6 for more details). The optimization can be carried out by the Levenberg-Marquardt algorithm [Lev44].

4.5 Experiments and Discussion

We implemented and evaluated the results of the pose estimation for synthetic and real data using the trifocal tensor and also using the fundamental matrix. In the first case, we compute the tensor linearly (TFT-L) and applying a Gauss-Helmert optimization with the minimal parameterizations of Ressl (TFT-R), Nordberg (TFT-N), Faugeras and Papadopoulo (TFT-FP) and Ponce and Hebert (TFT-PH). For the fundamental matrix we compute it linearly (F-L), with the 8-point algorithm [LH81], and with a Gauss-Helmert optimization (F-O). One last result is represented for the minimum found by the bundle adjustment (BA) initialized by any of the other methods. Indeed, we found that all the initializations gave the same final pose after the minimization in almost all our experiments, an important observation of our tests that we discuss later.

A MATLAB implementation for all these pose estimation methods has been developed and is available at the GitHub repository `https://github.com/LauraFJulia/TFT_vs_Fund.git` as well as the code to reproduce the experiments presented below.

4.5.1 Synthetic Scene

Pose estimation using the trifocal tensor and the fundamental matrix have been tested on synthetic data for different configurations. The standard scene for our experiments is composed of a set of space points contained in a cube of side 400 mm centered at the world's origin (see Figure 4.2). Points are projected onto three views and Gaussian noise is added

to the image points with $\sigma = 1$ pixel, if not stated otherwise. A sample of 12 points is used for the computations of the different models. The image size is 1800×1200 pixels, corresponding to a 36 mm $\times$ 24 mm sensor and the focal length is set to 50 mm. The cameras all point at the origin and have centers $C_1 = (0, -1400, 400)$, $C_2 = (-400, -1000, 0)$, $C_3 = (600, -800, -200)$. Results are averaged over 20 simulations of data.

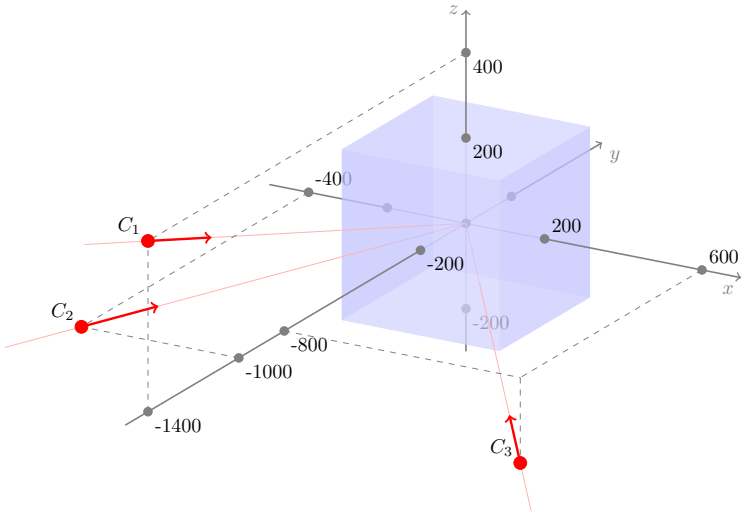


Figure 4.2 – Synthetic data.

Varying Noise

The angular error in the estimated rotations and translation directions, as well as the reprojection error are shown in Figure4.3 against Gaussian noise level added to the data points. The experiments reveal that the pose estimation based on the trifocal tensor is consistently more accurate than the fundamental matrix pose estimation. All different methods optimizing the trifocal tensor with a minimal parameterization manage to improve the initial linear solution and end up in the same minimum. In the same way, the optimization of the fundamental matrix decreases the error of the linear solution. All these improvements, while clear, have no consequence on the minimum found by the bundle adjustment, which is reached even when initialized by the simplest method (F-L). Also in Figure 4.3, a plot of the computational time spent on each initial estimation is shown.[2] As expected, linear methods (TFT-L, F-L) are faster than methods involving optimization, since the former are prerequisites for initialization of the latter. However, from the latter group, the fastest one is F-O, which involves two consecutive optimizations for two fundamental matrices.

Number of Correspondences

Figure 4.4 tests the effect of changing the number of corresponding points used for the pose estimation. It shows how the fundamental matrix is much more affected by using a minimal set of correspondences than any trifocal tensor estimators but TFT-FP. The

[2]based on the MATLAB code run on an Intel Xeon E5-2643 CPU at 3.3 GHz with 192 GB of RAM.

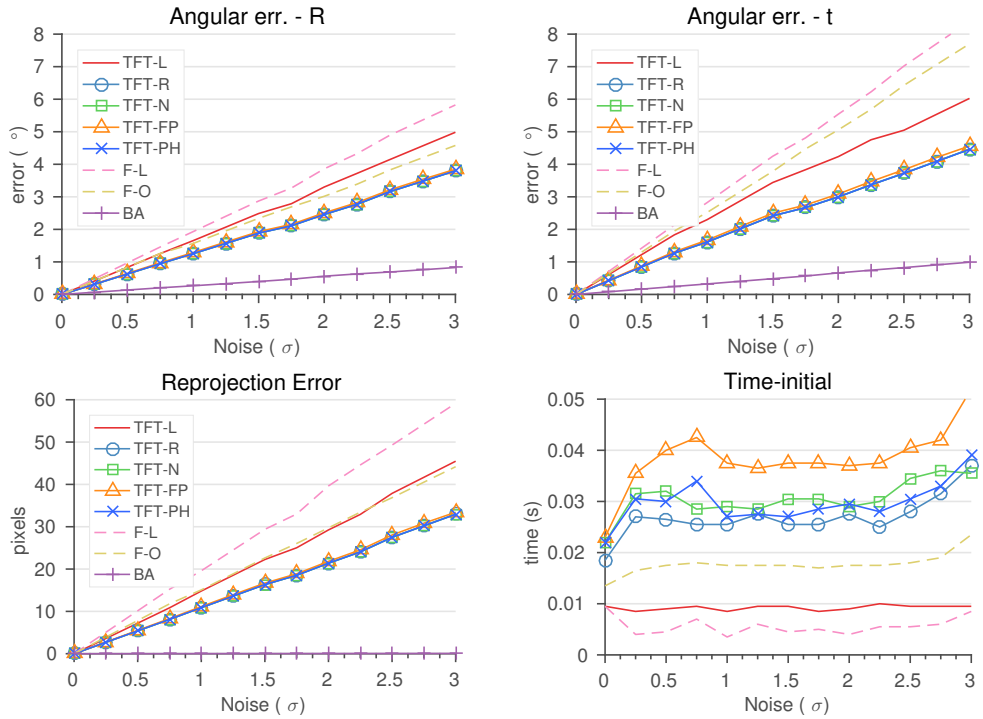


Figure 4.3 – Average errors (for rotations, -R, and translation directions, -t), reprojection error and computational time, when varying the Gaussian noise added to the image points.

Faugeras-Papadopoulo minimal parameterization not only fails to improve the pose given by the linear estimation of the tensor for the minimal set of 7 correspondences but it returns a much worse estimation. For initial sets of more than 7 triplets, however, it performs as well as the other TFT methods. For sets with more than 15 triplets, all models start to stabilize. On the time plot in Figure 4.4 we can see that linear methods maintain a constant computation time while optimization methods increase linearly with the number of initial points used.

Focal Length

Long focal lengths are known to make difficult the camera pose estimation with the fundamental matrix [YF14]. We studied the effect of increasing the focal length of our synthetic scene (while also proportionally getting the cameras farther away from the point cloud and from each other). Figure 4.5 shows that even if all methods get worse results in a similar way, the methods based on the fundamental matrix have an unstable higher increase of iterations for the bundle adjustment to converge after $f = 200$ mm. Still, the computational time seems not to be affected by the iterations increase and the final estimation remains the same, whatever the initialization method.

Collinear Cameras

In all these experiments, all TFT-based methods generally give the exact same results, showing the equivalence of all parameterizations. However, there is a degenerate case

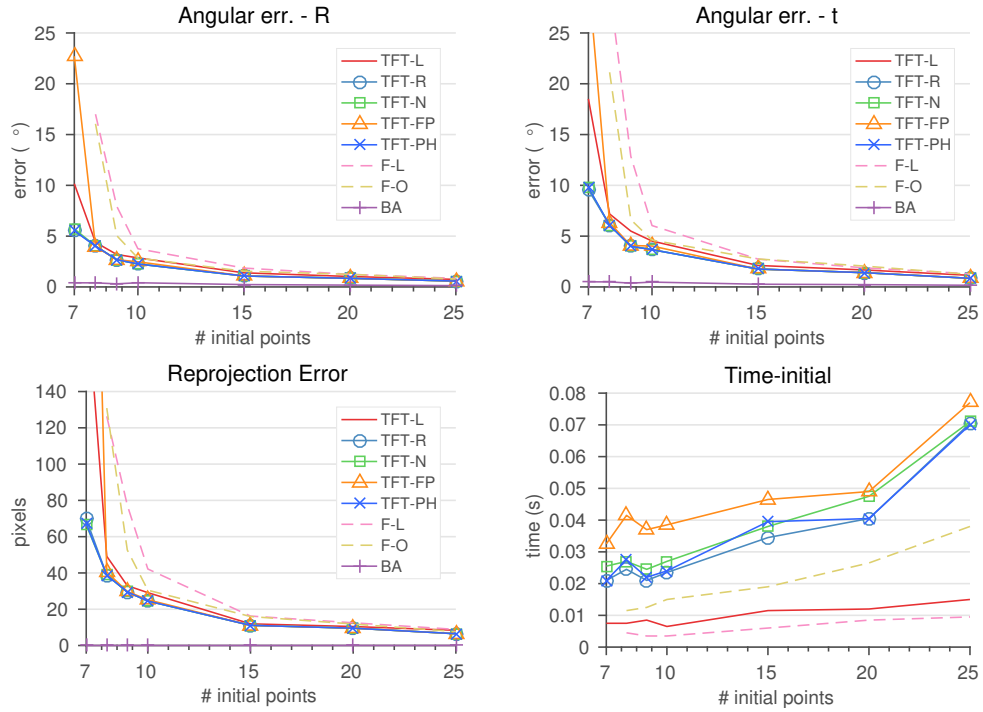


Figure 4.4 – Average errors (for rotations, -R, and translation directions, -t), reprojection error and computational time, when the number of corresponding points is varied.

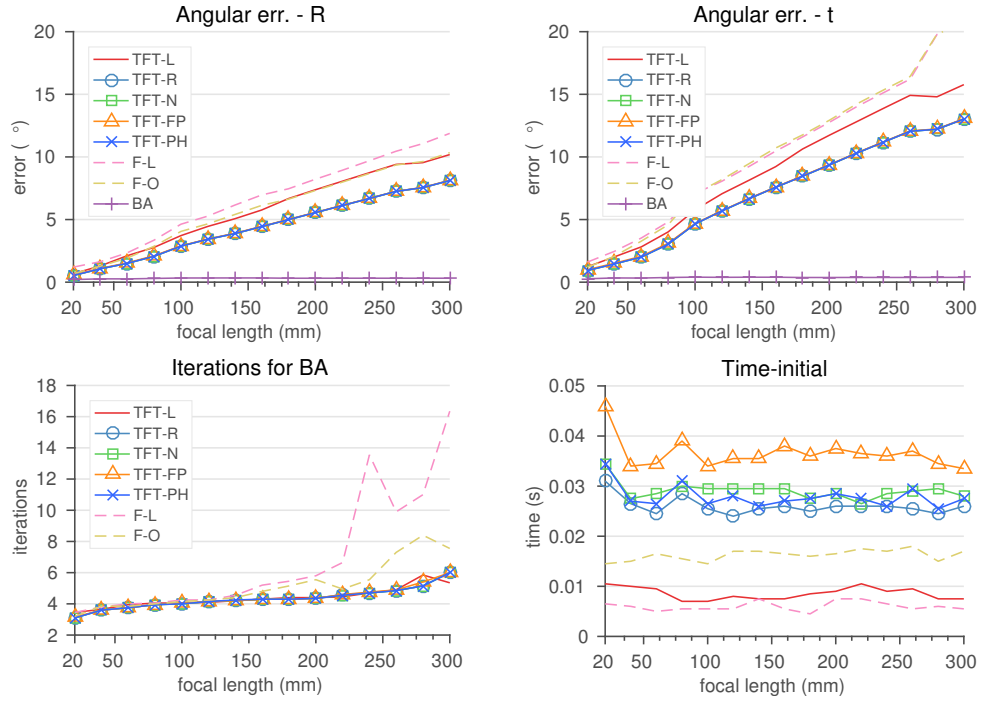


Figure 4.5 – For different focal lengths, the average angular errors (for rotations, -R, and translation directions, -t), the average number of iterations needed in bundle adjustment to reach a minimum and the computational time.

specific for the Ponce-Hebert and Nordberg parameterizations: collinear camera centers. Alongside all the previously presented methods, we implemented and tested the collinear parameterization of the Π_i matrices given by Ponce and Hebert in Section 4.3.4 (TFT-PH(Col)). We tested all methods gradually moving the camera centers of the scene in order to make them align. The measure of collinearity is the angle $\widehat{C_2C_1C_3}$ (180° when collinear). Figure 4.6 shows the reprojection error with the estimated poses, for 100 points not used in the estimation. The results show an increasing accuracy on the collinear method, starting to be comparable to the others at 176°, the same point where the non-collinear parameterizations suffer a jump on the error, much greater for TFT-N than for TFT-PH. After 178° the initial poses given by the non-collinear parameterizations are no longer able to find the right minimum through bundle adjustment.

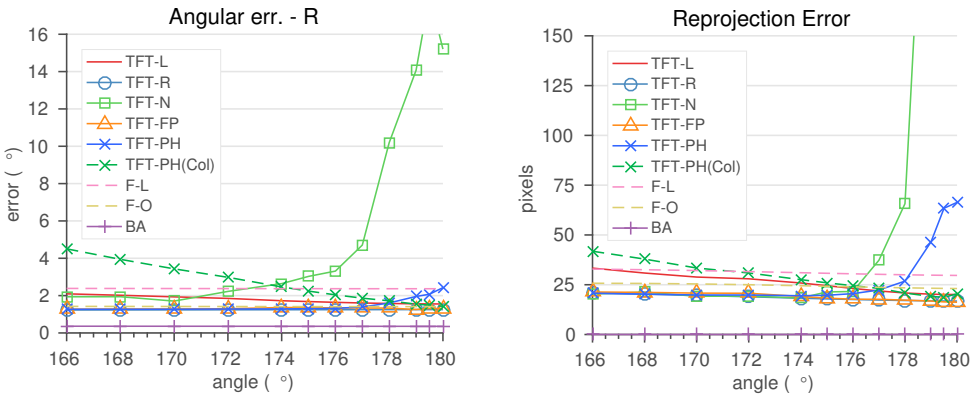


Figure 4.6 – On the left, average angular errors for rotations (-R) and on the right, average reprojection error of the pose estimation, both when making camera centers collinear.

Based on these results and the instability of the Faugeras-Papadopoulo parameterization with a minimal set of initial points, the trifocal tensor parameterization of Ressl seems to be the most robust to degenerate scenes and the most recommended for pose estimation using the TFT.

4.5.2 Real Datasets

AC-RANSAC

In order tot test this methods to real images, we implemented a variant of AC-Ransac based on the linear estimation of the trifocal tensor. For each triplet of real images to test, we can apply AC-RANSAC to the triplets of correspondences found by a matching algorithm in order to discard outliers. The parameters for the algorithm 2 in this case will be

E	linear estimation of the trifocal tensor.
n_E	7.
n_{out}	1.
$e(d, M)$	distance from point to transferred point using the TFT and the two points in the other images.
d	2 (dimension of point-to-point distance).
α_0	$\frac{\pi}{A}$ where A is the area of the image (upper bound).

The error $e(d, M)$ for an observation d, a triplet of corresponding points $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, and the estimated model M, a trifocal tensor $\mathbf{T}$, is defined in this case by the point transfer using the trifocal tensor. From the incidence relationships between corresponding points and lines given by the trifocal tensor we have the two following equations:

$$\bar{\mathbf{x}}_3 \sim \left(\sum_i (\bar{\mathbf{x}}_1)_i \mathbf{T}_i^\top \right) l_2 \qquad \bar{\mathbf{x}}_2 \sim \left(\sum_i (\bar{\mathbf{x}}_1)_i \mathbf{T}_i \right) l_3 \tag{4.39}$$

where l_2 and l_3 are lines, represented as 3-vectors s.t. $l^\top \bar{\mathbf{x}}$ for all $\mathbf{x} \in l$, in the second and third images respectively such that $\mathbf{x}_2 \in l_2$ and $\mathbf{x}_3 \in l_3$. So given a point in the first image and a corresponding line on the second or third image we can estimate a corresponding point in the third or second image respectively. However, if the line l_2 or l_3 is chosen so it happens to be the epipolar line corresponding to $\mathbf{x}_1$ the transferred point will be undefined (the right side of both equations in (4.39) will be 0). For this reason a safe approach is to use several lines, for example the three defined by the columns in $[\bar{\mathbf{x}}_2]_\times$ and in $[\bar{\mathbf{x}}_3]_\times$, compute the transferred points using all of them and choosing the result with largest norm.

Let us note $\mathbf{x}_2^{\text{transf}}$ and $\mathbf{x}_3^{\text{transf}}$ the two transferred points following this procedure. The error is defined as the maximum of the two distances to the transferred points

$$e(\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \mathbf{T}) = \max \left\{ \|\mathbf{x}_2 - \mathbf{x}_2^{\text{transf}}\|, \ \|\mathbf{x}_3 - \mathbf{x}_3^{\text{transf}}\| \right\} . \tag{4.40}$$

Application to the EPFL datasets

To evaluate the performance of the 3-view methods in real settings, we chose to use two scenes from the EPFL dense multi-view stereo test image datasets [SvHG+08] that come with a reliable ground truth. These datasets consist of images of size 3072×2048 pixels taken with a 35 mm equivalent focal length. For each triplet of images and method tested, the pose estimation is computed from a set of $N_{\text{init}} = 100$ triplets of correspondences chosen randomly from the total N correspondences computed with the SIFT implementation in [ROD14]. The bundle adjustment optimization is carried out using a subset of $N_{\text{BA}} = 50$ correspondences from the initial set. The reprojection error is evaluated on all N inliers.

The first scene is the fountain-P11 dataset which has 11 images (see Figure 4.7). We tested 50 of the possible image triplets and the averages of the results are shown in Table 4.2. In this scene the inlier tracks are selected from the $N_{\text{init}} = 100$ triplets of the initial set using AC-RANSAC adapted to each method (trifocal tensor and fundamental matrix). In the case of the fundamental matrix, the AC-RANSAC method is applied to each pair of views separately and the common inliers in the three pairs are selected.

The second scene is the Herz-Jesu-P8 dataset which consists of 8 images (see Figure 4.8), from which we tested 50 possible image triplets. The averages of the obtained errors are shown in Table 4.3. For this scene, the initial set of $N_{\text{init}} = 100$ tracks are selected from the original SIFT matches after discarding the ones with reprojection error (using ground truth poses) higher than 1 pixel.

On the one hand, the results confirm that Ressl's parameterization is the most robust and better performing of all TFT-based methods getting the smallest error in all metrics. On the other hand, Nordberg's parameterization fails to improve the linear estimation since it gets a higher angular error in rotation. This might be due to the near-collinearity of some

Figure 4.7 – Triplet of images of the EPFL fountain-P11 dataset.

	$e_{\mathbf{repr}}$ **(px)**	$e_{\mathbf{rot}}$ **(°)**	$e_{\mathbf{trans}}$ **(°)**	**init. time (s)**	**iter. BA**
TFT-L	6.605	0.306	1.720	0.057	4.12
TFT-R	5.132	**0.269**	1.504	0.716	4.12
TFT-N	6.556	0.846	1.589	0.579	4.34
TFT-FP	6.547	0.312	1.733	0.642	4.12
TFT-PH	**5.070**	0.270	**1.468**	0.564	4.12
F-L	6.083	**0.269**	1.726	0.038	4.08
F-O	5.918	0.273	1.650	0.314	4.06
BA	0.280	0.061	0.086		

Table 4.2 – Average results over 50 triplets of images from the EPFL fountain-P11 dataset. In **bold** are the best results (lowest average error) over all the pose estimation methods before bundle adjustment. The last row shows the final errors after bundle adjustment, which are the same no matter the initial method used for pose estimation.

Figure 4.8 – Triplet of images of the EPFL Herz-Jesu-P8 dataset.

	$e_{\mathbf{repr}}$ **(px)**	$e_{\mathbf{rot}}$ **(°)**	$e_{\mathbf{trans}}$ **(°)**	**init. time (s)**	**iter. BA**
TFT-L	4.806	0.459	0.871	0.062	4.06
TFT-R	**3.479**	**0.397**	**0.668**	1.591	4.00
TFT-N	4.093	0.540	0.692	1.480	4.04
TFT-FP	4.506	0.446	0.833	1.887	4.06
TFT-PH	4.306	0.421	0.672	1.249	4.00
F-L	3.762	0.414	0.772	0.040	4.00
F-O	3.650	0.420	0.765	0.858	4.02
BA	0.372	0.063	0.068		

Table 4.3 – Average results over 50 triplets of images from the EPFL Herz-Jesu-P8 dataset. In **bold** are the best results (lowest average error) over all the pose estimation methods before bundle adjustment. The last row shows the final errors after bundle adjustment, which are the same no matter the initial method used for pose estimation.

triplets (2 triplets in fountain-P11 and 4 in Herz-Jesu-P8 have a maximum angle between camera centers greater than 175°) which can cause great instability in the pose estimation of this methods as seen in the synthetic experiments (Figure 4.6).

We also notice how the fundamental-based methods get comparable results or even outperform the TFT-based methods in both datasets. What is more, they achieve it with less initial computation time and a similar average number of iterations to converge to the minimum in the bundle adjustment (two last columns of Tables 4.2 and 4.3).

In fact, all methods manage to reach the same minimum in the bundle adjustment optimization with around 4 iterations on average. The difference between the errors corresponding to the optimum reached and the errors from any method is much greater than the difference in the errors of the optimization-based methods and the linear methods. Therefore, one can conclude that the advantage of using an optimization to reinforce the constraints or minimal parameterization of the model before carrying out a bundle adjustment is negligible. The other lesson is that the bundle adjustment, even if performed with a small subsets of points for reduced computation time, is highly beneficial according to all error metrics.

Although not all known parameterizations of the trifocal tensor were covered by our tests, they all involve non-linear constraints admitting no closed form solution. As a consequence, they require also an initialization phase through the linear estimation of Sect. 4.4.1 and the possible initial benefits in terms of reduced error are likely to be erased by the bundle adjustment; the extra computation time would not make them advantageous alternatives to the standard fundamental matrix computation.

4.6 Conclusion

We reviewed methods of estimation of trifocal tensor and of the pose of three views. Compared with the pose estimation obtained by the fundamental matrices from the pairs of views, our experiments show that the trifocal tensor does not offer enough improvement to be considered the preferred choice. By its simplicity and lower computation time, the recommended option is to consider only pairwise constraints through the fundamental matrix, provided some bundle adjustment is used at the end (which is also highly recommended, as it can routinely decrease the error by a significant factor). In other words, the only usage of points viewed in image triplets, in the initialization phase of that approach, is to determine the relative scales of translations. Still, it would be interesting to study whether the use of the trifocal tensor improves results when $M > 3$ views are considered. However, in such a multi-view stereo pipeline, the way the image pairs and triplets are integrated is likely to have a preponderant importance.

Chapter 5

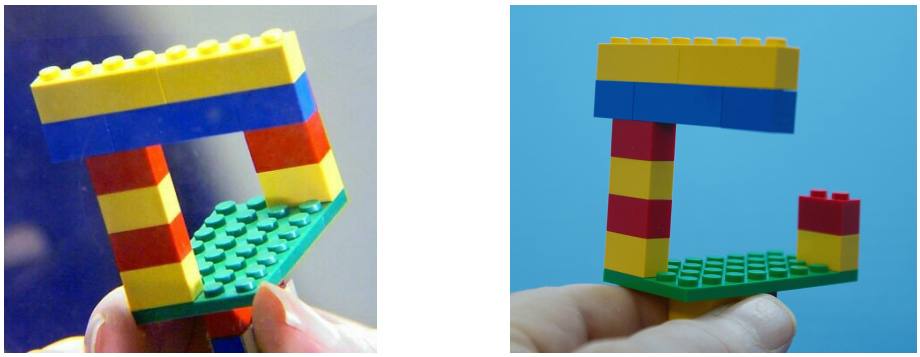
The Orthographic Projection for Long Focal Images

Most stereovision and SfM methods rely on the pinhole camera model based on perspective projection, from which the fundamental matrix and the epipolar constraints are derived. In this chapter we present a method based on the matrix factorization due to Tomasi and Kanade that relies on a simpler camera model, resulting in orthographic projection. This method can be used for the pose estimation of perspective cameras in configurations where other methods fail, in particular, when using cameras with long focal length lenses. We show this projection is an approximation of the pinhole camera model when the camera is far away from the scene. The performance of our implementation of this pose estimation method is compared to that given by the perspective-based methods for several configurations using both synthetic and real data. We show through some examples and experiments that the accuracy achieved and the robustness of this method make it worth considering in any SfM procedure.

Contents

5.1 Introduction

The epipolar geometry relies on the pinhole camera model, which describes the generation of photographic images as a perspective projection, but many other geometric projection models exist that are of interest for their applications. For instance, the orthographic model has been used in technical domains for its lack of perspective deformation so that measurements can be taken from the orthographic images, proportional to the 3D object dimensions. Real orthographic images (Figure 5.1a) can be obtained with a Telecentric Lens system, a camera recreating the orthographic projection.



(a) orthographic image (b) perspective image

Figure 5.1 – Two photographs of the same object: one taken with telecentric lens system (a), creating a visual illusion, and the other (b) taken with a normal lens. Note the lack of perspective deformation on the left, where parallel lines remain parallel and the object depth does not decrease its size, as opposed to the perspective image. Images from Donald Simanek's Pages (`https://lockhaven.edu/~dsimanek/`).

However, there are other cameras that can be modeled after the orthographic projection and our interest resides on seeing the orthographic camera as an approximation of a perspective camera when the scene is far away with respect to its size. It is precisely in such situations, generally generated by the use of long focal lengths, that the standard pose estimation approach based on the perspective model is badly suited and shows great errors in the estimated rotations and translations [YF14]. For this reason, the use of other projection models that are more robust in this kind of scenes becomes of interest and the orthographic projection has been explored in several works in the literature [TK92, SF11, OH02].

The orientation of orthographic views has been discussed as early as 1962 [Art62] and it is solvable when at least three views are considered. The factorization method by Tomasi and Kanade [TK92] provides a simple method for pose estimation of orthographic image streams and it can also be used to calibrate perspective images.

In this chapter we show the advantages of using the orthographic projection model for pose estimation of perspective images. We investigate the types of scenes where the orthographic model outperforms a perspective approach and test the robustness of the method for degenerate scenes.

5.2 The Orthographic Model

The orthographic model consists of a projection of the space points onto a plane along its orthogonal direction, the plane's normal. Rotation, scaling and translation can be applied to the points in the plane after the projection. We can parameterize the projection by the plane's axes, orthonormal vectors $\vec{i}$ and $\vec{j}$ that will also characterize the rotation, the origin of coordinates (a, b) in the plane and a scaling factor $s > 0$. The direction of the projection is $\vec{k} = \vec{i} \times \vec{j}$. The orthographic projection can be defined by the affine function ρ_{O}:

$$\begin{array}{rccl} \rho_{\mathrm{O}}: & \mathbb{R}^3 & \longrightarrow & \mathbb{R}^2 \\ & \mathbf{X} & \longmapsto & \mathbf{x} = s \begin{pmatrix} \vec{i}^{\top} \\ \vec{j}^{\top} \end{pmatrix} \mathbf{X} + \begin{pmatrix} a \\ b \end{pmatrix} \quad . \end{array} \tag{5.1}$$

5.2.1 Pinhole Camera at Infinity

It can be seen that the orthographic model is the result of taking a pinhole model and placing the camera center at an infinite distance of the scene. In order to draw the connection between the orthographic model and the standard pinhole camera, we take the projection equation (2.5), $\bar{\mathbf{x}} \sim K[R, \vec{t}]\bar{\mathbf{X}}$, and rewrite it in Cartesian coordinates with the non-linear function ρ_{P}:

$$\begin{array}{rccl} \rho_{\mathrm{P}}: & \mathbb{R}^3 & \longrightarrow & \mathbb{R}^2 \\ & \mathbf{X} & \longmapsto & \mathbf{x} = \dfrac{f}{\vec{k}^{\top}\mathbf{X} + t^z} \left[\begin{pmatrix} \vec{i}^{\top} \\ \vec{j}^{\top} \end{pmatrix} \mathbf{X} + \begin{pmatrix} t^x \\ t^y \end{pmatrix} \right] + \begin{pmatrix} c^x \\ c^y \end{pmatrix} \end{array} \tag{5.2}$$

where $R = (\vec{i}, \vec{j}, \vec{k})^{\top}$, $\vec{t} = (t^x, t^y, t^z)^{\top}$, f is the focal length (in pixels) and (c^x, c^y) the principal point.

We study then the effect of increasing the distance from the camera center $\mathbf{C}$ to the center of the scene $\mathbf{O}$ (concept defined later) in the direction orthogonal to the image plane, $d := \vec{k}^{\top}(\mathbf{O} - \mathbf{C}) = \vec{k}^{\top}O + t^z$. In order to get the camera further away from the scene while maintaining the projection of the scene inside the image, we fix the ratio $\alpha = f/d$, and we put the focal length as a function of distance to the scene $f = \alpha d$. With this we can compute the limit of the projection $\mathbf{x} = \rho_{\mathrm{P}}(\mathbf{X})$ for $X \in \mathbb{R}^3$ when d approaches infinity:

$$\begin{aligned} \lim_{d \to +\infty} \mathbf{x} &= \lim_{d \to +\infty} \frac{\alpha d}{\vec{k}^{\top}(\mathbf{X} - \mathbf{O}) + d} \left[\begin{pmatrix} \vec{i}^{\top} \\ \vec{j}^{\top} \end{pmatrix} \mathbf{X} + \begin{pmatrix} t^x \\ t^y \end{pmatrix} \right] + \begin{pmatrix} c^x \\ c^y \end{pmatrix} \\ &= \alpha \begin{pmatrix} \vec{i}^{\top} \\ \vec{j}^{\top} \end{pmatrix} \mathbf{X} + \alpha \begin{pmatrix} t^x \\ t^y \end{pmatrix} + \begin{pmatrix} c^x \\ c^y \end{pmatrix} \quad . \end{aligned} \tag{5.3}$$

This result indicates that the perspective projection with camera center at an infinite distance from the scene is an orthographic projection with scale $\alpha = \frac{f}{d}$, axes $\vec{i}$, $\vec{j}$ and translation $\alpha(t^x,\ t^y)^{\top} + (c^x,\ c^y)^{\top}$. This happens because as the camera gets further away from the scene, the rays arriving at the image plane become more and more parallel to each other, almost orthogonal to the image plane (see Figure 5.2).

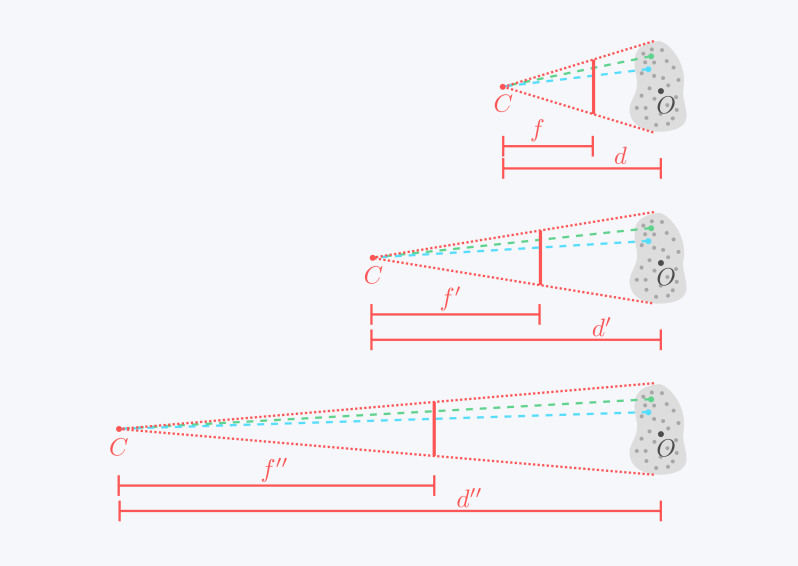


Figure 5.2 – For the same scene, we can see how the projection lines of two points (the ones in blue and green) get more and more parallel as we get the camera away from the scene. In the image, we increased the distance d along with the focal length f so that $\frac{f}{d} = \frac{f'}{d'} = \frac{f''}{d''}$.

5.2.2 The Scaled-Orthographic Model

In Poelman and Kanade [PK97] a specific orthographic projection is defined to emulate the pinhole camera at infinity: the Scaled-Orthographic Model, also known as the weak-perspective camera. This model is defined by fixing the scaling factor to the ratio between the focal length and the distance to the scene of the emulated pinhole camera, $s = \alpha$. Also, we will write the translation vector in terms of the camera parameters to match those of the limit at infinity. In this case, the projection function will be a linear function ρ_{SO}:

$$\begin{array}{rccl} \rho_{\text{SO}}: & \mathbb{R}^3 & \longrightarrow & \mathbb{R}^2 \\ & \mathbf{X} & \longmapsto & \mathbf{x} = \dfrac{f}{\vec{k}^\top \mathbf{O} + t^z} \left[\begin{pmatrix} \vec{i}^\top \\ \vec{j}^\top \end{pmatrix} \mathbf{X} + \begin{pmatrix} t^x \\ t^y \end{pmatrix} \right] + \begin{pmatrix} c^x \\ c^y \end{pmatrix} \end{array} \tag{5.4}$$

Notice that this definition is the same as ρ_{P} except that the point $\mathbf{X}$ in the denominator has been replaced by the center of the scene $\mathbf{O}$ so that it coincides with the perspective model at infinity. The new parameter $\mathbf{O}$ will be defined as the centroid of all the observed points of the scene, i.e. $\mathbf{O} = \frac{1}{N}\sum_j \mathbf{X}^j$. For that, we make the assumption that only a finite number N of space points is observed.

The fact that the perspective model is an orthographic projection at infinity is a good indication that the scaled orthographic projection of a scene will be a good approximation of the perspective projection for scenes far away from the camera. To further show this, we can study the distance between the two projections $\rho_{\text{P}}(\mathbf{X})$ and $\rho_{\text{SO}}(\mathbf{X})$. Let the scene be a set of space points $\{\mathbf{X}^j\}$ and the coordinates be centered on the centroid $\mathbf{O} = \frac{1}{N}\sum_j \mathbf{X}^j = \mathbf{0}$. We define a camera with center of projection $\mathbf{C}$, axes $\vec{i}$, $\vec{j}$ and $\vec{k}$, focal length f and principal point $(c^x,\ c^y) = (0,\ 0)$, to simplify. Using these parameters also for the scaled-orthographic model we can compute both a perspective projection $\mathbf{x}_{\text{P}} = \rho_{\text{P}}(\mathbf{X})$ and a scaled orthographic

projection $\mathbf{x}_{\mathrm{SO}} = \rho_{\mathrm{SO}}(\mathbf{X})$ for any point $\mathbf{X} = (X, Y, Z)^\top$. The distance between these two projections is

$$d(\mathbf{x}_{\mathrm{P}}, \mathbf{x}_{\mathrm{SO}}) = \|\rho_{\mathrm{P}}(\mathbf{X}) - \rho_{\mathrm{SO}}(\mathbf{X})\| = \left|\frac{f}{Z+d} - \frac{f}{d}\right| \left\| \begin{pmatrix} \vec{i}^\top \\ \vec{j}^\top \end{pmatrix} \mathbf{X} + \begin{pmatrix} t^x \\ t^y \end{pmatrix} \right\| = \left|\frac{Z}{Z+d}\right| \|\mathbf{x}_{\mathrm{SO}}\| \ . \tag{5.5}$$

Therefore, the difference between the orthographic and perspective images is proportional to the image point distance to the "center" of the image and the ratio $\frac{Z}{Z+d}$. This term, is the ratio between the relative depth of the space point to the center of the scene, Z, and the mean depth of the scene with respect to the camera center, d, plus Z. When this ratio decreases, so does the difference between the two projections. This proves that the scaled orthographic model can be a good approximation of the perspective model in situations where $\frac{Z}{Z+d}$ is small. This is generally the case for long focal length images, where the scene observed (and the matched points) are far away in comparison to the relative depth of the scene. It is also the case for close-to-planar scenes, but we will see that those are not easy to handle with the scaled-orthographic model.

5.3 Tomasi-Kanade Factorization of the Orthographic Model

Tomasi and Kanade [TK92] presented a method to factorize the matrix built from the image measurements into two matrices representing shape and motion under the orthographic model. Later, Poelman and Kanade [PK97] extended this work to the scaled-orthographic model. We present the latter below.

Suppose we have a set of M scaled-orthographic cameras with parameters $\mathbf{C}_i$, $\vec{i}_i$, $\vec{j}_i$, $\vec{k}_i$, focal length f_i and principal point (0, 0). Let the scene be a set of N space points $\{\mathbf{X}^j\}$ and the origin of the world coordinates the centroid $\mathbf{O} = \frac{1}{N}\sum_j \mathbf{X}^j$. Then, the projection of the points $\{\mathbf{X}^j\}$ by each camera onto the image points $\{\mathbf{x}_i^j\}$ is described by the following equations:

$$\mathbf{x}_i^j = \begin{pmatrix} \vec{m}_i^\top \\ \vec{n}_i^\top \end{pmatrix} \mathbf{X}^j + \begin{pmatrix} a_i \\ b_i \end{pmatrix} \qquad \text{for} \quad i = 1, \ldots, M \quad j = 1, \ldots, N \tag{5.6}$$

where,

$$\begin{pmatrix} \vec{m}_i^\top \\ \vec{n}_i^\top \end{pmatrix} = \frac{f_i}{t_i^z} \begin{pmatrix} \vec{i}_i^\top \\ \vec{j}_i^\top \end{pmatrix} , \qquad \begin{pmatrix} a_i \\ b_i \end{pmatrix} = \frac{f_i}{t_i^z} \begin{pmatrix} t^x \\ t^y \end{pmatrix} \ . \tag{5.7}$$

We can gather all image points in a $2M \times N$ measurement matrix $\mathcal{W}$, all the scaled axes on the $2M \times 3$ motion matrix $\mathcal{R}$, all 3D points in the $3 \times N$ shape matrix $\mathcal{S}$ and the translation vectors in a global translation vector $\mathcal{T}$ of length $2M$. The projection equations can be written then in matrix form:

$$\underbrace{\begin{pmatrix} \mathbf{x}_1^1 & \mathbf{x}_1^2 & \cdots & \mathbf{x}_1^N \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_M^1 & \mathbf{x}_M^2 & \cdots & \mathbf{x}_M^N \end{pmatrix}}_{\mathcal{W}} = \underbrace{\begin{pmatrix} \vec{m}_1^\top \\ \vec{n}_1^\top \\ \vdots \\ \vec{m}_M^\top \\ \vec{n}_M^\top \end{pmatrix}}_{\mathcal{R}} \underbrace{\begin{pmatrix} \mathbf{X}^1 \ldots \mathbf{X}^N \end{pmatrix}}_{\mathcal{S}} + \underbrace{\begin{pmatrix} a_1 \\ b_1 \\ \vdots \\ a_M \\ b_M \end{pmatrix}}_{\mathcal{T}} \begin{pmatrix} 1 \overset{N}{\ldots} 1 \end{pmatrix} \tag{5.8}$$

Notice that the global translation vector $\mathcal{T}$ can be computed from the mean of image points thanks to the centroid being placed at the origin of space coordinates: if we multiply (5.8) on the right by the vector $\frac{1}{N}(1 \dots 1)^\top$ of size N,

$$\frac{1}{N}\sum_{j=1}^{N}\begin{pmatrix}\mathbf{x}_1^j\\ \vdots \\ \mathbf{x}_M^j\end{pmatrix} = \mathcal{R}\left(\frac{1}{N}\sum_{j=1}^{N}\mathbf{X}^j\right) + \mathcal{T} = \mathcal{R}\,\mathbf{O} + \mathcal{T} = \mathcal{T}\ . \tag{5.9}$$

Therefore, we define the matrix $\mathcal{W}^* := \mathcal{W} - \mathcal{T}(1\ \dots\ 1)$ that can be computed from image data only. This matrix will have, when no noise is present, at most rank three due to the fact that $\mathcal{W}^* = \mathcal{R}\mathcal{S}$. This decomposition of $\mathcal{W}^*$ is a rank factorization and it is the key to Tomasi and Kanade [TK92] pose estimation method.

Remark 2. *The rank factorization of any matrix A of rank $r > 0$ is not unique. However, if we have two rank factorizations of the same matrix, $A = A_1A_2 = A_1'A_2'$, there always exists an $r \times r$ invertible matrix Π s.t. $A_1 = A_1'\Pi$ and $A_2 = \Pi^{-1}A_2'$.*

The matrix $\mathcal{R}$ from the factorization verifies several constraints given the fact that it is the motion matrix. These constraints are about the norm and scalar products of its rows, these are:

$$\|\vec{m}_i\| = \|\vec{n}_i\| = \left|\frac{f_i}{t_i^z}\right| \quad \text{and} \quad \vec{m}_i^\top\vec{n}_i = 0 \quad \forall i = 1, \dots, M \tag{5.10}$$

In addition, there is a rotation and scaling ambiguity. For any rotation matrix R and scalar s, the matrices $s\mathcal{R}R^\top$ and $\frac{1}{s}R\mathcal{S}$ give equally valid motion and shape matrices respectively.

Minimal M For $M \leq 2$ there is no unique reconstruction. The case $M = 1$ is obvious. In the case of two views, $M = 2$, given a possible reconstruction, any rotation of one of the views around the axis perpendicular to the projection direction of both views will produce another valid reconstruction with different motion and shape. This means that we need at least three views to proceed to unambiguous pose estimation.

Minimal N We can see by studying the rank of $\mathcal{W}^*$ with respect to N that the minimal number of correspondences is 4. We know that $\text{rank}(\mathcal{W}) \leq 3$ and the number of correspondences N should not limit in any way this rank, otherwise we would not be able to compute $\mathcal{R}$ and $\mathcal{S}$ from a degenerate $\mathcal{W}^*$. If we write the measurement matrix like $\mathcal{W} = (w_1, \dots, w_N)$ and $\mathcal{W}^* = (w_1^*, \dots, w_N^*)$ we will have

$$w_j^* = w_j - \frac{1}{N}\sum_{j'=1}^{N} w_{j'} \quad \Rightarrow \quad \sum_{j=1}^{N} w_j^* = 0 \tag{5.11}$$

and then $\text{rank}(\mathcal{W}^*) < N$ since its N columns are linearly dependent. For $N \leq 3$, the matrix $\mathcal{W}^*$ would have rank lower than 3, so we need $N \geq 4$ in order to have a non-deficient $\mathcal{W}^*$ from which we can compute the motion and shape matrices.

5.3.1 Planar Scene

If all the 3D points lie on a plane Π, the matrix $\mathcal{S}$ will no longer have full rank and $\mathcal{RS}$ will no longer be a rank factorization of $\mathcal{W}$. For this reason we consider this case a degenerate scene for the orthographic pose estimation.

In the perspective case, it has been studied that when a planar scene is observed there exists an homography transforming the projected points from one perspective image to the another. It is through this homographic transformation that the relative pose between two cameras can be extracted.

When orthographic cameras are considered, however, we show that there is no equivalent procedure to uniquely extract the poses. This is because given an initial scene with M orthographic cameras and N points in a plane, we are able to find other scenes (not equivalent by global translation, rotation or scaling) with the same projection points but different poses and/or structure. Therefore, the poses can not be uniquely identified when the 3D points all lie in a plane.

Scene

Let us have M orthographic cameras $\{s_i, R_i, \vec{t}_i\}_{i=1,\dots,M}$, defined by a scale s_i, rotation $R_i^\top = (\vec{i}_i, \vec{j}_i, \vec{k}_i)$ and translation $\vec{t}_i^\top = (a_i, b_i)$. Let $\{\mathbf{X}^j\}_{j=1,\dots,N}$ be N 3D points lying on a plane Π. Without loss of generality we can fix $\Pi = \{Z = 0\}$, so that $\mathbf{X}^j = (X_1^j, X_2^j, 0)^\top$ for all i. The projection equations will be

$$\mathbf{x}_i^j = s_i R_i' X^j + \vec{t}_i \qquad \forall j = 1, \dots, N,\ \forall i = 1, \dots, M \tag{5.12}$$

where $R_i'^\top = (\vec{i}_i, \vec{j}_i)$.

Les us suppose that there exists another scene with M orthographic cameras $\{r_i, Q_i, \vec{l}_i\}_{i=1,\dots,M}$ and N points $\{\mathbf{Y}^j\}_{j=1,\dots,N}$ lying on a plane that has the same projections

$$\mathbf{x}_i^j = r_i Q_i' Y^j + \vec{l}_i \qquad \forall j = 1, \dots, N,\ \forall i = 1, \dots, M\ . \tag{5.13}$$

We can fix the plane to Π in the same way and without loss of generality so that $\mathbf{Y}^j = (Y_1^j, Y_2^j, 0)^\top$.

Remark 3. *We will assume that no camera has a projection direction $\vec{k}_i$ parallel to the plane Π, otherwise the projections to this camera would lie on the same line.*

Like before, we fix the world origin as the centroid of the 3D points in both scenes, $\mathbf{O} = \frac{1}{N}\sum_{j=1}^N \mathbf{X}^j = \frac{1}{N}\sum_{j=1}^N \mathbf{Y}^j$, from which follows (analogously to eq. (5.9)) that the translation vectors in both scenes have to coincide

$$\vec{t}_i = \frac{1}{N}\sum_{j=1}^N \mathbf{x}_i^j = \vec{l}_i \quad \forall i = 1, \dots, M\ .$$

We define the "centralized" image points as $\mathbf{x}^{*j}_i = \mathbf{x}_i^j - \vec{t}_i$. Replacing this in the projection equations (5.12) and (5.13),

$$\mathbf{x}^{*j}_i = s_i R_i' X^j = r_i Q_i' Y^j \tag{5.14}$$

Let us suppose that there exist 3×3 invertible matrices $\{A_i\}_{i=1,\dots,M}$ such that for each $i = 1, \dots, M$ we have

$$(s_i R'_i) A_i^{-1} = r_i Q'_i \quad \text{and} \quad A_i X^j = Y^j \quad \forall j = 1, \dots, N \tag{5.15}$$

so that equation (5.14) is verified. From these equalities we can deduce three conditions that the matrices A_i have to satisfy

1. $A_i\ \mathbf{X}^j \in \Pi \quad \forall i, j$
2. $A_i\ \mathbf{X}^j = A_{i'}\ \mathbf{X}^j \quad \forall j, i \neq i'$
3. $A_i^{-\top} \vec{i}_i$ and $A_i^{-\top} \vec{j}_i$ are perpendicular vectors with the same norm for all i.

Determining A_i

From condition 1. we get that the last row of A_i must be proportional to (0 0 1). From condition 2. we get that the upper-left 2×2 sub-matrix of each A_i should be equal to the others. The same is true for A_i^{-1} and we can establish the following notation, for $i = 1, \dots, M$

$$A_i^{-1} = \begin{pmatrix} \lambda_1 & \lambda_2 & u_i \\ \lambda_3 & \lambda_4 & v_i \\ 0 & 0 & w_i \end{pmatrix} \quad \text{with} \quad \begin{array}{l} w_i \neq 0, \\ \lambda_1\lambda_4 - \lambda_2\lambda_3 \neq 0\ . \end{array} \tag{5.16}$$

Notice that the inverse of the matrix $\Lambda = \begin{pmatrix} \lambda_1 & \lambda_2 \\ \lambda_3 & \lambda_4 \end{pmatrix}$ describes the transformation on the plane applied to the points $\mathbf{X}^j$ when multiplied by any A_i. For each camera $i = 1, \dots, M$, the condition 3. translates to

$$\left\{ \begin{array}{rcl} (\vec{i}_i^{\top} A_i^{-1})\,(A_i^{-\top} \vec{i}_i) & = & r_i^2/s_i^2 \\ (\vec{j}_i^{\top} A_i^{-1})\,(A_i^{-\top} \vec{j}_i) & = & r_i^2/s_i^2 \\ (\vec{i}_i^{\top} A_i^{-1})\,(A_i^{-\top} \vec{j}_i) & = & 0 \end{array} \right. \Rightarrow \left\{ \begin{array}{rcl} (\vec{i}'_i)^{\top} \Lambda\, \Lambda^{\top} \vec{i}'_i + \vec{i}_i^{\top}\, \vec{u}_i\, \vec{u}_i^{\top}\, \vec{i}_i & = & r_i^2/s_i^2 \\ (\vec{j}'_i)^{\top} \Lambda\, \Lambda^{\top} \vec{j}'_i + \vec{j}_i^{\top}\, \vec{u}_i\, \vec{u}_i^{\top}\, \vec{j}_i & = & r_i^2/s_i^2 \\ (\vec{i}'_i)^{\top} \Lambda\, \Lambda^{\top} \vec{j}'_i + \vec{i}_i^{\top}\, \vec{u}_i\, \vec{u}_i^{\top}\, \vec{j}_i & = & 0 \end{array} \right. \tag{5.17}$$

where $\vec{u}_i = (u_i, v_i, w_i)^{\top}$ and $\vec{i}'_i = (i_i^1, i_i^2)^{\top}$, $\vec{j}'_i = (j_i^1, j_i^2)^{\top}$. Written in matrix form,

$$\underbrace{\begin{pmatrix} (i_i^1)^2 & (i_i^2)^2 & 2i_i^1 i_i^2 \\ (j_i^1)^2 & (j_i^2)^2 & 2j_i^1 j_i^2 \\ i_i^1 j_i^1 & i_i^2 j_i^2 & i_i^1 j_i^2 + i_i^2 j_i^1 \end{pmatrix}}_{I_i} \underbrace{\begin{pmatrix} \lambda_1^2 + \lambda_2^2 \\ \lambda_3^2 + \lambda_4^2 \\ \lambda_1\lambda_3 + \lambda_2\lambda_4 \end{pmatrix}}_{\Lambda'} + \underbrace{\begin{pmatrix} (\vec{u}_i^{\top}\vec{i}_i)^2 \\ (\vec{u}_i^{\top}\vec{j}_i)^2 \\ (\vec{u}_i^{\top}\vec{i}_i)(\vec{u}_i^{\top}\vec{j}_i) \end{pmatrix}}_{U_i} = \underbrace{\frac{r_i^2}{s_i^2} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}}_{S_i} \tag{5.18}$$

from which we can deduce that $\Lambda' = (I_i)^{-1}(S_i - U_i)$. (I'_i) is invertible as a consequence of remark 3. After computations and using $\vec{k}_i = \vec{i}_i \times \vec{j}_i$ we obtain the equations

$$\left\{ \begin{array}{rcll} \lambda_1^2 + \lambda_2^2 & = & \frac{r_i^2}{s_i^2}\big(1 + \big(\frac{k_i^1}{k_i^3}\big)^2\big) - \bar{u}_i^2 & \\ \lambda_3^2 + \lambda_4^2 & = & \frac{r_i^2}{s_i^2}\big(1 + \big(\frac{k_i^2}{k_i^3}\big)^2\big) - \bar{v}_i^2 & \text{for } i = 1, \dots, M \\ \lambda_1\lambda_3 + \lambda_2\lambda_4 & = & -\frac{r_i^2}{s_i^2}\big(\frac{k_i^1}{k_i^3}\big)\big(\frac{k_i^2}{k_i^3}\big) - \bar{u}_i\bar{v}_i & \end{array} \right. \tag{5.19}$$

where $\bar{u}_i = u_i - \frac{k_i^1}{k_i^3} w_i$ and $\bar{v}_i = v_i - \frac{k_i^2}{k_i^3} w_i$.

The four unknowns of Λ do not depend on the camera i and the three equations (5.19) only constrain its parameters in three degrees of freedom. If we consider (λ_1, λ_2) and (λ_3, λ_4) as two vectors, we see that only their norms and relative angle are constrained. We can rewrite Λ as

$$\Lambda = \begin{pmatrix} m & 0 \\ n\cos\gamma & n\sin\gamma \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{5.20}$$

with $m = \sqrt{\lambda_1^2 + \lambda_2^2}$, $n = \sqrt{\lambda_3^2 + \lambda_4^2}$ and $\gamma = \arccos(\frac{\lambda_1\lambda_3 + \lambda_2\lambda_4}{mn})$. The rotation determined by θ is applied globally to all camera axes and inversely to the points in the plane, a transformation that gives a completely equivalent configuration. Since we are only interested in non-equivalent configurations we fix it to $\theta = 0$. We get equations,

$$\begin{cases} m^2 &= \frac{r_i^2}{s_i^2}\left(1 + \left(\frac{k_i^1}{k_i^3}\right)^2\right) - \bar{u}_i^2 \\ n^2 &= \frac{r_i^2}{s_i^2}\left(1 + \left(\frac{k_i^2}{k_i^3}\right)^2\right) - \bar{v}_i^2 \qquad \text{for } i = 1, \ldots, M \ . \\ mn\cos\gamma &= -\frac{r_i^2}{s_i^2}\left(\frac{k_i^1}{k_i^3}\right)\left(\frac{k_i^2}{k_i^3}\right) - \bar{u}_i\bar{v}_i \end{cases} \tag{5.21}$$

The system has a total of $3M$ quadratic equations and $3 + 3M$ unknowns. Therefore, the three unknowns of the planar transformation, m, n and γ, can be fixed to any value giving $3M$ independent systems of equations that will determine the remaining $3M$ parameters, r_i, $\bar{u}_i$ and $\bar{v}_i$ for $i = 1, \ldots, M$. Then, the vectors $\vec{u}_i$ will not be completely determined. In fact, if we rewrite $\vec{u}_i$ as

$$\vec{u}_i = \begin{pmatrix} \bar{u}_i \\ \bar{v}_i \\ 0 \end{pmatrix} + \frac{w_i}{k_i^3}\vec{k}_i$$

we notice that parameter w_i is free and its value does not affect the transformations described in (5.15), so we can fix $w_i = 1$ without loss of generality.

So we have seen that for any transformation Λ^{-1} of the plane Π, where the $\{\mathbf{X}^j\}_j$ lie, we can find a transformation A_i for each scale-orthographic camera, giving a new configuration of the scene, but with exactly the same projections in each image. As a result, we proved that it is not possible to recover the camera poses of scaled-orthographic cameras only from the images when all the projected points of the scene lie on the same plane. It is a result independent of the number of cameras M and the number of points N.

Example. *As a simple example we can take an orthographic camera positioned so that its projection axis is perpendicular to the plane* Π*, with rotation matrix* $R = Id$ *and scaling* $s = 1$*. We choose the plane transformation described to be applied to the points* $\mathbf{X}^j$ *by the matrix*

$$\Lambda^{-1} = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 2 \end{pmatrix}$$

so then, $\gamma = \frac{3\pi}{4}$*,* $m = 2$*,* $n = \frac{\sqrt{2}}{2}$*. Solving the system* (5.21) *and rejecting imaginary solutions and negative* r*, we get two valid solutions which are equivalent by "depth ambiguity" (this equivalence is described in the following Section, 5.3.2, with matrix in eq.* (5.27)*). Therefore, we only study the resulting camera from the first solution,*

$$r = \frac{\sqrt{10} + \sqrt{26}}{4} = 2.0653 \quad \bar{u} = \frac{(\sqrt{65} - 7)\sqrt{\sqrt{65} + 7}}{8} = 0.5153 \quad \bar{v} = \frac{\sqrt{\sqrt{65} + 7}}{2} = 1.9405 \ . \tag{5.22}$$

The scaling of the new camera, r, is already computed. The solution for the new camera rotation axes is,

$$Q' = \frac{s}{r} Id' A^{-1} = \frac{1}{2.0653} \begin{pmatrix} 0.5 & 0 & 0.5153 \\ 0.5 & 2 & 1.9405 \end{pmatrix} = \begin{pmatrix} 0.9684 & 0 & 0.2495 \\ -0.2421 & 0.2421 & 0.9396 \end{pmatrix} . \quad (5.23)$$

This solution can be visualized in Figure 5.3.

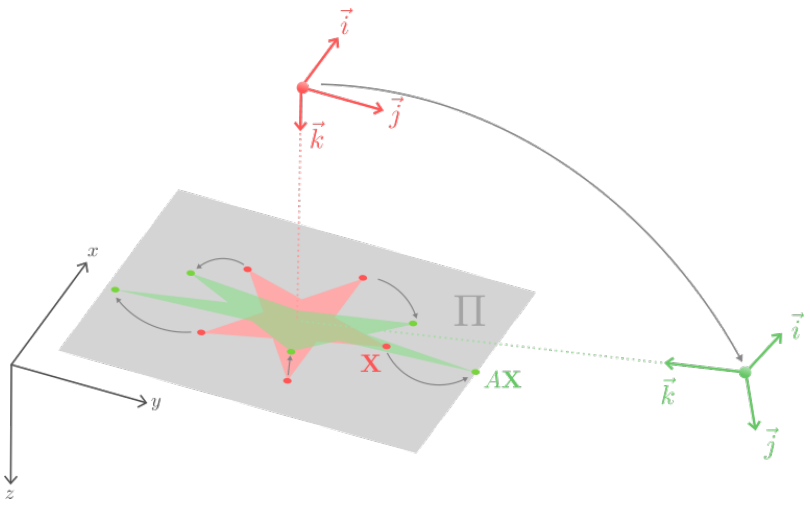


Figure 5.3 – Visualization of the example presented for a planar scene. The initial scene corresponds to the red star formed by the 3D points $\mathbf{X}$ that lie onto the plane Π and the camera represented by the red axes $\vec{i}$, $\vec{j}$, $\vec{k}$. The plane transformation Λ^{-1} brings the red points to the green points $A\mathbf{X}$, the star being deformed into the green one. Then, the green axes $\vec{i}$, $\vec{j}$, $\vec{k}$ represent the initial camera transformed by A^{-1} which will obtain an orthographic image that will match exactly the one produce by the initial configuration.

5.3.2 The Pose Estimation Method

The factorization leads to the following pose estimation method. Let us have $N \geq 4$ corresponding image points throughout $M \geq 3$ images $\{\mathbf{x}_i^j\}$ and the (estimated) focal length of each view $\{f_i\}$.

1. From the image data we can compute $\tilde{\mathcal{W}}^* = \tilde{\mathcal{W}} - \tilde{\mathcal{T}}(1 \ \ldots \ 1)$ (the tilde indicates that noise is present in the measurements).

2. $\tilde{\mathcal{W}}^*$ might have rank higher than three due to error and noise. We impose the rank deficiency by using the singular value decomposition. For $\tilde{\mathcal{W}}^* = U\Sigma V^\top$, we define $\mathcal{W}^* = U'\Sigma'(V')^\top$, where U' and V' are the first 3 columns of U and V respectively and Σ' is the upper-left 3×3 sub-matrix of Σ, formed by the three largest singular values.

3. A first rank factorization $\mathcal{W}^* = \hat{\mathcal{R}}\hat{\mathcal{S}}$ is given by $\hat{\mathcal{R}} = U'(\Sigma')^{\frac{1}{2}}$ and $\hat{\mathcal{S}} = (\Sigma')^{\frac{1}{2}}(V')^\top$.

4. We search for a 3×3 invertible matrix Q such that the new rank factorization $\mathcal{W}^* = (\hat{\mathcal{R}}Q)(Q^{-1}\hat{\mathcal{S}})$ is a valid motion-shape decomposition. Hence, $\mathcal{R} = \hat{\mathcal{R}}Q$ should verify the constraints in (5.10). They translate to:

$$\hat{m}_i^\top QQ^\top \hat{m}_i - \hat{n}_i^\top QQ^\top \hat{n}_i = 0 \quad \text{and} \quad \hat{m}_i^\top QQ^\top \hat{n}_i = 0 \qquad \forall i = 1, \ldots, M \tag{5.24}$$

   The numerical value of $\|\hat{m}_i\|$ is unknown since we do not have t_i^z. In consequence, only the equality of norms $\|\hat{m}_i\| = \|\hat{n}_i\|$ can be imposed and we have a linear system of $2M$ homogeneous equations on the coefficients of the matrix $\Pi = QQ^\top$. Since the matrix Π is positive semi-definite, its coefficients are reduced to 6 unknowns and the $2M \geq 6$ equations are enough to determine the coefficients up-to-scale by finding the kernel of the matrix of the homogeneous system (the matrix Π could also be estimated imposing its positive semi-definiteness using convex optimization). Afterwards, we recover Q through the Cholesky factorization of Π.

5. The final factorization of $\mathcal{W}^*$ is $\mathcal{R} = \hat{\mathcal{R}}Q$ and $\mathcal{S} = Q^{-1}\hat{\mathcal{S}}$. From it we can compute all camera parameters. For the axes,

$$\vec{i}_i = \frac{\vec{m}_i}{\|\vec{m}_i\|}, \quad \vec{j}_i = \frac{\vec{n}_i}{\|\vec{n}_i\|}, \quad \vec{k}_i = \vec{i}_i \times \vec{j}_i \tag{5.25}$$

   and for the translation vector we estimate the value of f_i/t_i^z using the mean of the norms of both $\vec{m}_i$ and $\vec{n}_i$ (since they might not be exactly equal),

$$\vec{t}_i = \frac{2}{\|\vec{m}_i\| + \|\vec{n}_i\|}(a_i,\ b_i,\ f_i)^\top \ . \tag{5.26}$$

Rotation ambiguity Any rotation can be applied to the final result. The usual approach is to bring $\vec{m}_1$ and $\vec{n}_1$ to $(1, 0, 0)^\top$ and $(0, 1, 0)^\top$.

Depth ambiguity The final factorization obtained in step 4 can be transformed into a new valid factorization $\mathcal{W}^* = (\mathcal{R}A)(A\mathcal{S})$ by using the matrix

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \ . \tag{5.27}$$

This transformation gives an equivalent 3D reconstruction but the views are placed "at the back" of the scene. For orthographic data this ambiguity cannot be solved. In Figure 5.4 there is an example of two possible configurations of the same reconstruction.

5.3.3 Application to Perspective Cameras

As seen in Section 5.2.2 the scaled-orthographic model can be seen as an estimation of the perspective model in some particular scenes like the images acquired with long focal lengths. It turns out that the pose estimation of narrow-angle images is a difficult task for perspective-based methods like the fundamental matrix and, therefore, we propose to use the scaled-orthographic model which proves to be more robust.

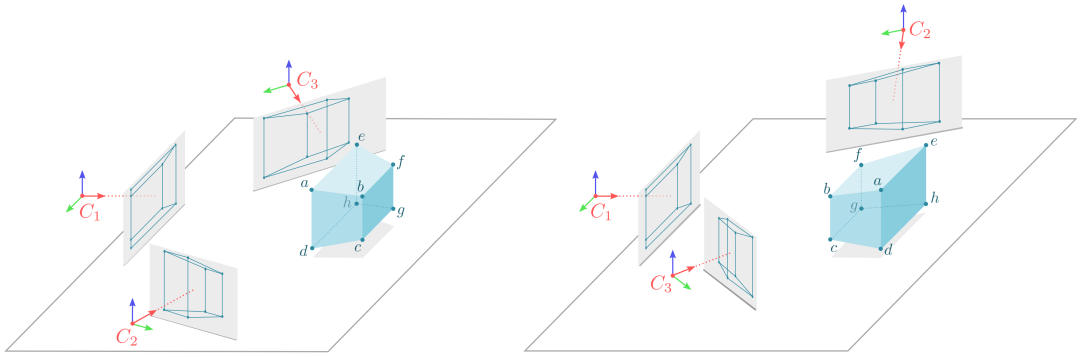


Figure 5.4 – Two possible configurations for the same image points in three orthographic views. Notice that for each view the relative depth of the observed points swaps from one configuration to the other.

To apply the method of the previous section to pose estimation of perspective cameras we can use the image data as if it was produced by a scaled orthographic camera. Once we estimate the camera parameters we can reinterpret them as perspective cameras instead of orthographic and proceed with a bundle adjustment refinement if desired. The depth ambiguity can be solved by two different approaches. We can manually choose between the two possible solutions by identifying one image point from the correspondences in the images that is closer to the camera (on the first plane of the scene). Otherwise, if no manual help is intended, there is no other solution than to keep both solutions and to choose the one that gives smaller reprojection error after the bundle adjustment step.

5.4 Implementation

The implementation of the pose estimation of perspective cameras based on the scaled-orthographic method is in the Matlab function `OrthographicPoseEstimation`. For this method, the input data are the matching points throughout $M \geq 3$ views and the calibration parameters for each camera (calibration matrix K with the focal length and principal point). The output are the two possible orientations in variables `Sol1` and `Sol2` that contain a rotation matrix R and translation vector $\vec{t}$ for each view and also matrices $\mathcal{R}$, $\mathcal{S}$, $\mathcal{T}$ from the Tomasi and Kanade factorization ($\mathcal{S}$ contains the 3D reconstruction of the matching points passed as input). The general pipeline for pose estimation from images to a refined solution would be the following:

Matching between pairs: For each pair of images the matching features should be found. This step can be done with any available software and it is not implemented in our code.

Extract tracks: From the couples of matching points of each pair of images, the consistent tracks between three views are extracted by the function `matches2tracks`. At least 4 tracks are needed to follow the procedure.

AC-RANSAC: A set of inliers from the tracks between the three views can be chosen by Random Sample Consensus. For our implementation we use the a contrario approach

(adapted from [MMM13a]) with the scaled-orthographic model for three views. The AC-RANSAC is programmed in function `AC_RANSAC_Orthographic`, where the set of inliers is computed using the orthographic model with a minimal sample of 4 tracks. The output is the set of maximal inliers. The parameters for the algorithm 2 in this case are

E	the Tomasi-Kanade factorization for the scaled-orthographic model (to compute $\mathcal{R}$ and $\mathcal{T}$).
n_E	4.
n_{out}	2 due to the depth ambiguity.
$e(d, M)$	distance from point to transferred point computed by the 3D reconstruction given by the other two scaled-orthographic cameras and the two corresponding image points.
d	2 (dimension of point-to-point distance).
α_0	$\frac{\pi}{A}$ where A is the area of the image (upper bound).

The error $e(d, M)$ for an observation d, a triplet of corresponding points $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, and the estimated model M, the scaled-orthographic cameras given by $\mathcal{R}, \mathcal{T}$, is defined in this case by the point transfer using the the scaled-orthographic 3D reconstruction. For two cameras i_1 and i_2, an estimation of the 3D points $\tilde{S}_{i_1 i_2}$ can be computed using equation (5.8)

$$\tilde{\mathcal{S}}_{i_1 i_2} = \mathcal{R}^{+}_{i_1 i_2} \left(\mathcal{W}_{i_1 i_2} - \mathcal{T}_{i_1 i_2} \left(1 \overset{N}{\ldots} 1 \right) \right) \tag{5.28}$$

where $\mathcal{R}_{i_1 i_2}$, $\mathcal{T}_{i_1 i_2}$ and $\mathcal{W}_{i_1 i_2}$ are the matrices composed only by the rows corresponding to cameras i_1 and i_2 of $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{W}$ respectively. Then, the reconstructed points can be projected onto the third image

$$\tilde{\mathcal{W}}_{i_3} = \mathcal{R}_{i_3} \tilde{\mathcal{S}}_{i_1 i_2} + \mathcal{T}_{i_3} \left(1 \overset{N}{\ldots} 1 \right) \tag{5.29}$$

The error for an observation $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ is defined as the maximum of the three possible distances

$$e(\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathcal{R}, \mathcal{T}\}) = \max \left\{ \|\mathbf{x}_1 - \tilde{\mathbf{x}}_1\|, \ \|\mathbf{x}_2 - \tilde{\mathbf{x}}_2\|, \ \|\mathbf{x}_3 - \tilde{\mathbf{x}}_3\| \right\} . \tag{5.30}$$

Initial pose estimation: Using all the inliers the scaled orthographic method is applied to get a first estimation of the pose for each camera. Calling the function `OrthographicPoseEstimation` we will get two possible configurations of the cameras. As seen in the description of this method in Section 5.3.2, the operations needed in the pose estimation are computationally simple and they only involve the SVD and the Cholesky decomposition.

Bundle Adjustment: The configurations obtained in the last step are used as two possible initializations for the orientations of perspective cameras in a bundle adjustment. This method consists in minimizing the reprojection error over the possible cameras orientations and space points:

$$\min_{\{R_i, \vec{t}_i\}_i, \, \{\mathbf{X}^j\}_j} \sum_{j=1}^{N} \sum_{i=1}^{M} \|\mathbf{x}_i^j - \rho_{\mathrm{P}i}(\mathbf{X}^j)\|^2 \tag{5.31}$$

where $\rho_{\mathrm{P}i}$ is the perspective projection as in (5.2) associated to camera i. The optimization is carried out by the function `BundleAdjustment` using the Levenberg-Marquardt algorithm [Lev44] (already implemented in Matlab). After using the two possible initializations, the solution with lower reprojection error at the end of the optimization is chosen as the correct final pose estimation.

In our implementation, the script `pipeline_script.m` carries out the four last steps of the pipeline described above for a given set of matching points between the pairs of a triplet of images and specified focal length value. For the experiments, the matches have been computed by the SIFT implementation available in the IPOL publication [ROD14].

5.5 Experiments

To study the robustness and performance of this method we have evaluated its results when applied to several and different scenes while comparing its results to the performance of other algorithms based on the perspective model. Synthetic data has been used to test the sensibility to different parameters such as noise, focal length and configuration of the space points and cameras. Moreover, we also applied the method to real images taken with long focal lengths to test its performance in real situations to evaluate its usefulness. Since the minimum number of views needed in the orthographic pose estimation is three, we have evaluated the results for scenes composed of three views. The errors used for evaluation are the reprojection error (2.14) and the angular error in rotations and translations (2.15) applied to a 3-view configuration.

For comparison with a perspective-based method, we evaluate the results given by the pose estimation using the fundamental matrix on the same scenes. More specifically, the two fundamental matrices F_{21} and F_{31} are computed from their correspondences by the 8-point algorithm [Har97]. Then, the relative orientations $[R_{21},\ \vec{t}_{21}]$ and $[R_{31},\ \vec{t}_{31}]$ are extracted by singular value decomposition of the essential matrices [HZ04]. Finally, we choose the global poses [Id 0], $[R_{21},\ \vec{t}_{21}]$, $[R_{31},\ \lambda\vec{t}_{31}]$ with λ the solution of

$$\arg\min_{\lambda\in\mathbb{R}} \sum_{j=1}^{N} \|\bar{\mathbf{x}}_3^j \times \big(K_3(R_{31}\mathbf{X}^j + \lambda\vec{t}_{31})\big)\|^2 \tag{5.32}$$

where the space points $\{\mathbf{X}^j\}_{j=1,\dots,N}$ are reconstructed using only the image points in the first two images and the camera matrices $K_1[\mathrm{Id}\ 0]$, $K_2[R_2,\ \vec{t}_2]$ and the notation $\bar{\mathbf{x}}$ is used to indicate the homogeneous coordinates of $\mathbf{x}$. This is not the minimization of a geometric error but an algebraic one. The solution has the closed form:

$$\lambda = \frac{\sum_{j=1}^{N} \big(\bar{\mathbf{x}}_3^j \times (K_3R_{31}\mathbf{X}^j)\big)^\top \big(\bar{\mathbf{x}}_3^j \times (K_3\vec{t}_{31})\big)}{\sum_{j=1}^{N} \|\bar{\mathbf{x}}_3^j \times (K_3\vec{t}_{31})\|} \ . \tag{5.33}$$

We also compare the results to the final pose given after a bundle adjustment based on the perspective model with the initial pose obtained by one of the previous methods. The optimization is carried out by the Levenberg-Marquardt algorithm. In most of our experiments, the same minimum is reached regardless of the method used as initialization, so the results will be represented as one.

5.5.1 Synthetic Data

For our experiments we take the same synthetic scene used in the previous Chapter in Section 4.5. The standard scene (Figure 5.5) is composed of a set of space points contained in a cube of 400 mm × 400 mm centered in the world's origin. The points are projected onto three cameras and Gaussian noise is added to the image points with $\sigma = 1$ pixel, if not stated otherwise. A sample of 20 points is used for the computations of the different models and a test set of 100 to evaluate the results. The image size is 1800×1200 pixels that correspond to a 36 mm × 24 mm sensor. The cameras all point at the origin and their placement depends on the chosen focal length, $\mathbf{C}_1 = f \cdot (0, -28, 8)$, $\mathbf{C}_2 = f \cdot (-8, -20, 0)$ and $\mathbf{C}_3 = f \cdot (12, -16, -4)$, so that the ratio α is fixed, see Figure 5.5. Results are averaged over 20 simulations of data.

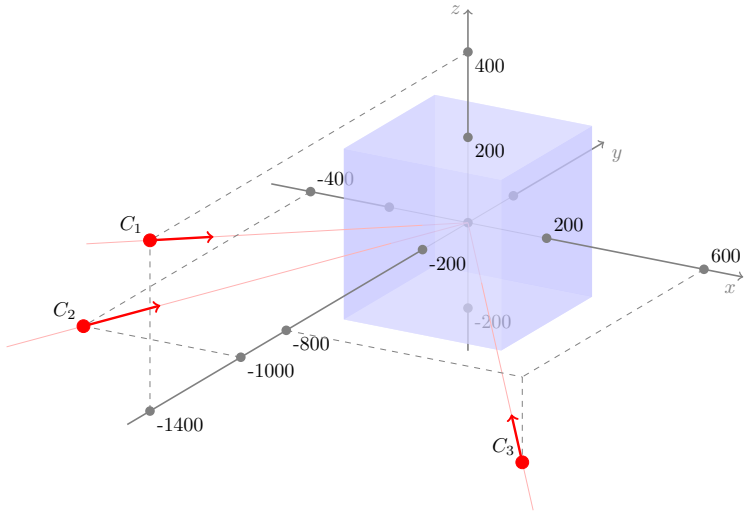


Figure 5.5 – Illustration of the configuration of the synthetic data scene for f = 50 mm. When the focal length is increased, the camera centers move along the red lines a proportional distance from the origin.

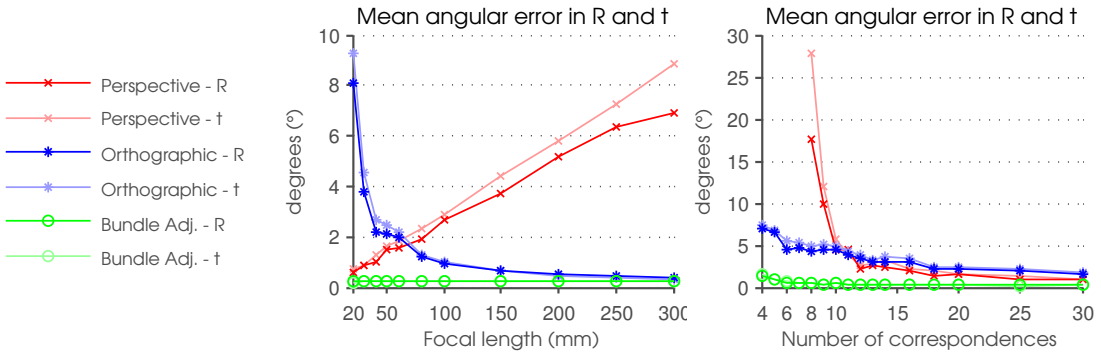
Focal Length

In a first approach, we look at the angular error for different focal lengths, f varying in $[20, 300]$ mm. This changes not only f but also the position of the cameras as stated previously. We can see in Figure 5.6a that the initial pose given by the perspective-based estimation loses accuracy linearly with the increase of the focal length. On the other hand, the orthographic method has bad results for short lengths but rapidly gains accuracy and gets better results than the perspective-based method for lengths starting at $f = 60$ mm. For $f \geq 200$ mm the orthographic solution is really close to the final solution given by the bundle adjustment, both solutions getting less than 0.5° error.

Number of Correspondences

The orthographic method shows a good stability also in the number of correspondences used for the pose estimation. Here we vary the cardinal M of the sample of image points used

in the computation of the estimated pose for all methods. The fundamental matrix can be computed with the 8-point algorithm for $M \geq 8$ while the orthographic method only needs $M \geq 4$. Figure 5.6b shows how the fundamental matrix gets bad results when a minimal or close to minimal set of correspondences is used and how the orthographic solution is not so affected by M. This also means that when $4 \leq M \leq 7$ and the fundamental matrix cannot be computed, the orthographic model can give a first pose estimation not so far from the ground truth. The stability of the factorization method even with few correspondences can be attributed to the strong rank-3 constraint, which prevents overfitting to the observations.



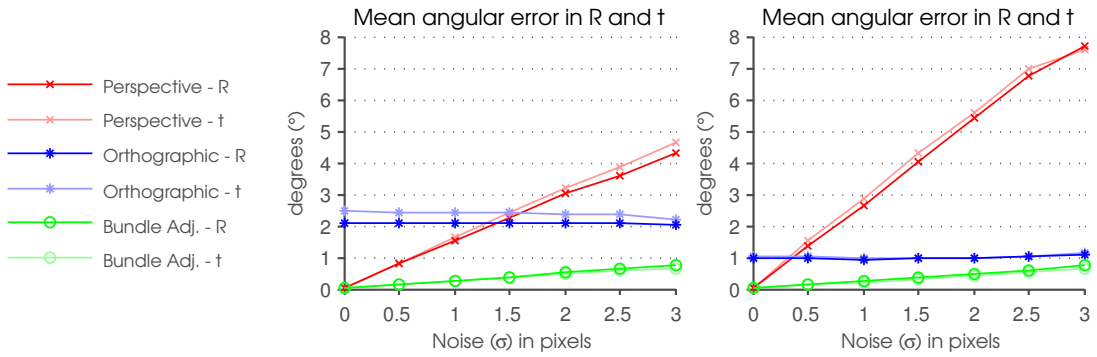
(a) (b)

Figure 5.6 – Angular errors for each method changing different parameters. Two lines are drawn for each method, one for the mean angular error in rotations (-R) and another for the mean error in translations (-t). In (a) the focal length is varied and in (b) the focal is fixed at $f = 50$ mm while the number of correspondences is varied.

Robustness to Noise

Varying the Gaussian noise added to the image measurements with $\sigma \in [0, 3]$ pixels, the orthographic method proves to be much more robust to noise than a perspective-based method. The angular error of rotation and translation stays almost constant in comparison to the increasing error proportional to σ of the initial pose given by the perspective-based method. Testing with $f = 100$ mm (Figure 5.7b) we see that the orthographic solution has a constant error smaller than the perspective solution for $\sigma \geq 0.5$ pixel. However, with a shorter length $f = 50$ mm (Figure 5.7a) the performance of the orthographic method is not as good, though still almost constant, and it is surpassed by the perspective solution when noise is not high.

When we look at the reprojection error, for example with $f = 50$ mm and varying the noise in Figure 5.8a, we can see that the error in the perspective solution is extremely affected by increasing the noise while the orthographic solution has a smaller constant error for $\sigma \geq 0.5$ pixel. On the other hand, we have seen that the angular error for this focal length is smaller in the perspective solution for noise with $\sigma \leq 1.5$ pixels.



(a) $f = 50$ mm (b) $f = 100$ mm

Figure 5.7 – The angular errors in rotations and translations for varying Gaussian noise are shown for two different fixed focal lengths.

Planar Case

There is a particular case where the orthographic method fails: the case of quasi planar or planar scenes. We evaluated the failures of the orthographic and perspective methods when the scene becomes planar. The planarity of a scene is measured with the percentage

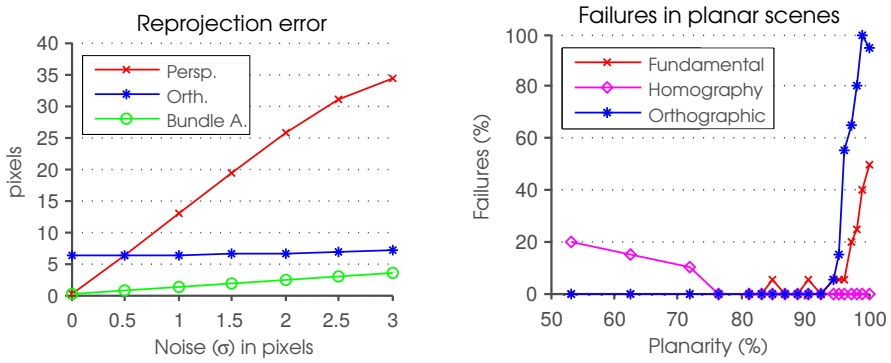
$$\text{planarity} = 100 \cdot \left(1 - \frac{v_3}{v_1}\right) \tag{5.34}$$

where v_1 and v_3 are respectively the first and third eigenvalues of the matrix $X^\top X$ formed from the point cloud[1].

We consider that a pose estimation is a **valid solution** when after applying bundle adjustment it gets small angular errors, $e_{\text{rot}} \leq 5°$ and $e_{\text{trans}} \leq 10°$. In Figure 5.8b it is clear how the failures in the orthographic model rapidly rise for a planarity greater than 95%. The method based on the fundamental matrix is not robust to planarity either, but it manages to give at least 50% of valid solutions. It is known that for a perspective camera model observing a planar scene the homography between images should be used to compute the pose [Zha96], so we also included the results with this perspective method.

The failures of the orthographic model for the planar scenes can be explained by the fact that the situation is similar to having only $N = 3$ points. If all the points projected onto the three orthographic cameras lie on the same 3D plane, the matrix $\mathcal{S}$ in (5.8) has rank 2 which makes $\text{rank}(\mathcal{W}^*) \leq 2$. Therefore, the factorization $\mathcal{W}^* = \mathcal{R}\mathcal{S}$ is no longer a rank decomposition and there is no guarantee that the first factorization given by SVD in step 3 of the pose estimation method will lead to the correct solution. This usually translates into obtaining a non positive semi-definite matrix Q in step 4 or simply getting a wrong pose estimation. Even if the planar hypothesis is assumed, the new pose estimation problem is not solvable as seen in Section 5.3.1.

[1] $X^\top X$ is the estimation of the covariance matrix of the point cloud.



(a) (b)

Figure 5.8 – In (a) the reprojection error in pixels for varying Gaussian noise for a short focal length, $f = 50$ mm, is shown. In (b) the percentage of failures is shown for the orthographic model and for two perspective methods (fundamental matrix and homography) in the case of the space points lying on a quasi-planar surface, the focal length is fixed at 100 mm.

Parallel Camera Axes

The projection rays in the orthographic model are parallel to each other, no matter the depth or position of the space points. For this reason, having M orthographic cameras with different position but same direction of projection ($\vec{k}_i$) will, in fact, create M equivalent orthographic images, i.e. related only by scaling, translation and rotation on the plane. This translates into not having enough information for the depth recovery of the space points since in Equation (5.8) the matrix $\mathcal{R}$ will have only rank 2. Even if we are working with perspective data, this configuration generates ambiguous data for the proposed pose estimation method.

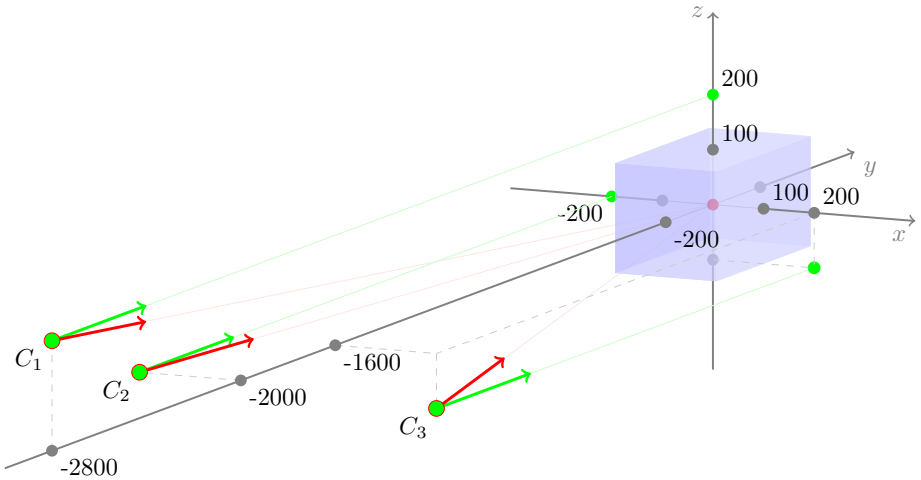
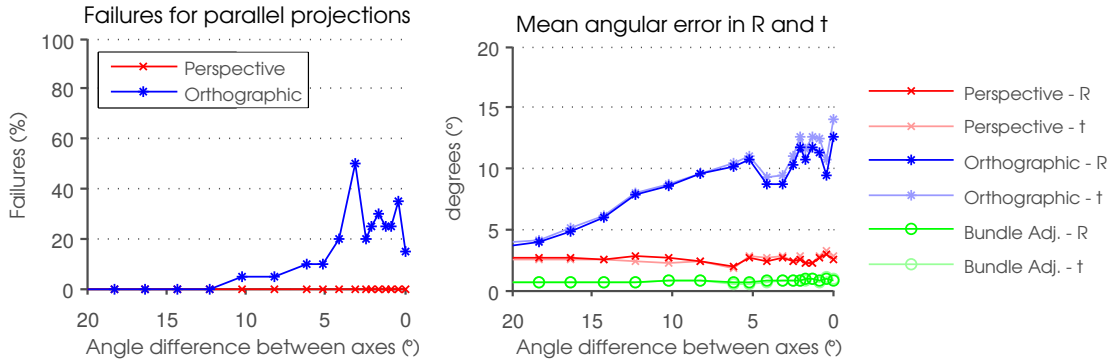


Figure 5.9 – The synthetic data scene to study parallel camera axes. The cameras orientations are varied from the configuration in red (all cameras pointing at the origin) to configuration in green, where all three camera axes are parallel and the cameras are directed toward the green points.

We analyzed the potential instability of the orthographic pose estimation on these type of scenes by modifying the synthetic data used. In Figure 5.9 we can see the modified scene where the cameras positions are fixed but the orientation varies from all cameras pointing at the origin to three parallel camera axes. The experiments showed that the method starts to fail as the camera axes get close to parallel (Figure 5.10a) and the angular error of the obtained pose increases above 10° (Figure 5.10b). However, the pose estimation based on the fundamental matrix is not at all affected by this special configuration.



(a) (b)

Figure 5.10 – The case of quasi-parallel camera axes is studied in (a) with the percentage of failures and in (b) with the angular errors.

5.5.2 Real Data

We have evaluated the orthographic method along with the perspective method based on the fundamental matrix on several real databases. Again, we test the two methods on triplets of images and compare the errors and the failures of each method. We also show the results for the solutions after applying a bundle adjustment (BA) step to both methods. In these real experiments, we chose to use a small sample of 50 correspondences on the bundle adjustment in order to improve the speed of the optimization.

To select the inlier tracks of each evaluated triplet, we use an a contrario RANSAC adapted to each method. In the case of the fundamental matrix, the RANSAC method is applied to each pair of views separately and the common inliers in the three pairs are selected.

In Tables 5.1–5.4 we show, for each dataset, several statistics averaged over all evaluated triplets: the resulting reprojection error (e_{repr}), rotation and translation errors (e_{rot} and e_{trans}), the percentage of correspondences selected as inliers, the percentage of valid solutions (VS), as described in the Planar Case of Section 5.5.1, and the number of iterations of the bundle adjustment.

Reims This dataset is made out of a total of 55 images all acquired with a long focal length ($f = 400$ mm) and it covers the main interior wall of the cathedral in Reims, France (see Figure 5.11). We tested the 200 triplets of images with the most correspondences and we compared the results to an "artificial" ground truth, the solution given by the SfM

realized by MicMac [PD07] with not only the long focal length images but also with 16 other images with shorter focal length (7 with $f = 100$ mm and 9 with $f = 50$ mm) to stabilize the process. The internal calibration and distortion on the images was estimated by the same MicMac process and used in our computations. For this particular dataset, most of the scenes are quasi planar and the cameras are aligned. We can see in Table 5.1 that while both methods have problems to estimate the translations, the orthographic model gives a better estimate according to the lower number of mean iterations needed to reach the minimum and the higher percentage of pose estimation success.

Figure 5.11 – Some images of the Reims dataset. The first four on the left were taken with a focal length of 400 mm and used in our experiments. The last one on the right was taken with a focal length of 100 mm and used in the computations of the MicMac SfM algorithm.

	e_{repr}	e_{rot}	e_{trans}	inliers	VS	# iter
Perspective	240.995	1.554	42.70	88.6%	75.5%	68.9
after BA	0.410	0.461	1.52			
Orthographic	7.926	2.660	36.78	84.9%	82.0%	41.5
after BA	0.413	0.457	1.56			

Table 5.1 – Statistics for 200 triplets of Reims dataset.

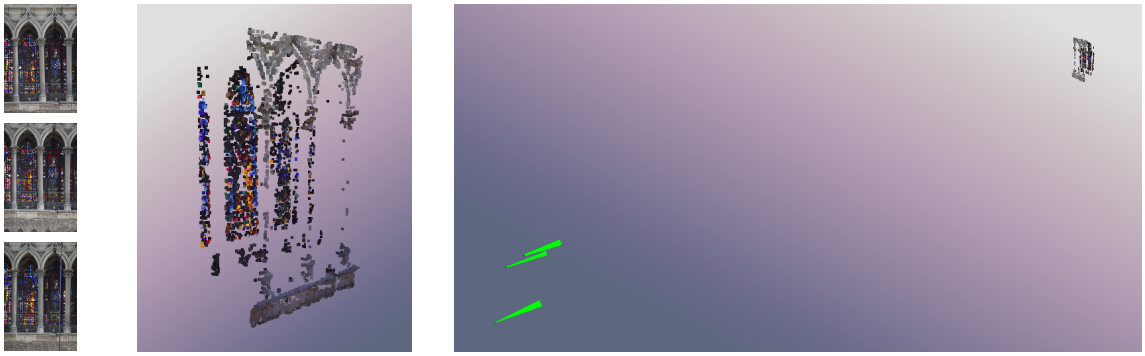


Figure 5.12 – Example of the reconstruction obtained by the orthographic model for one triplet of the Reims dataset. On the left, a close up of the point cloud recovered. On the right, a representation with the estimated camera poses (in green), so that the long distance between cameras and photographed object becomes obvious.

EPFL fountain-P11 [SvHG+08] In juxtaposition to the previous case, we tested our algorithm in a short focal length dataset to evaluate its robustness. The fountain-P11 dataset is made out of a total of 10 images taken with a 35 mm equivalent focal length of 32.5 mm and has an available ground truth. We tested 70 of the possible triplets and the results are shown in Table 5.2. The orthographic method does not manage to get similar results to the perspective method; moreover, it has much higher reprojection and angular errors. However, the orthographic model is able to give as much as 91.43% of valid results, meaning that, even if the perspective method is the obvious choice for this kind of scenes, the orthographic method would still be a viable option to get a first pose estimation to proceed to bundle adjustment.

	e_{repr}	e_{rot}	e_{trans}	inliers	VS	# iter
Perspective	13.084	0.49	1.87	95.9%	100.0%	4.5
after BA	0.366	0.16	0.17			
Orthographic	76.219	29.38	26.50	48.0%	91.4%	36.2
after BA	0.369	0.16	0.18			

Table 5.2 – Statistics for 70 triplets of the EPFL fountain-P11 dataset.

Figure 5.13 – Example of the reconstruction obtained by the orthographic model for one triplet of the EPFL fountain-P11 dataset. We can see the short distance between the cameras and photographed object.

Statue To show the accuracy of the orthographic method for long focal lengths, we acquired a new set of images with a focal of length approximately 1000 mm. The images cover the face of a statue (Figure 5.14) positioned in the middle of one of the fountains of Château de Champs-sur-Marne, France. The dataset consists of a set of 58 images and we tested 150 possible triplets. Since there is no available ground truth only the reprojection error can be computed. We can see in Table 5.3 how the mean reprojection error for the perspective-based method is extremely high while for the orthographic model it is below 1 pixel in the initial pose estimation being very close to the final reprojection error achieved by the bundle adjustment minimization. Looking at the number of iterations needed to

reach the minimum it becomes clear that the orthographic method is more suited for the long focal length case.

Figure 5.14 – A sample of 48 images of the Statue dataset.

	e_{repr}	e_{repr}-BA	inliers	# iter
Perspective	4977.959	0.648	98.8%	74.1
Orthographic	0.735	0.576	97.7%	11.2

Table 5.3 – Statistics for 150 triplets of Statue dataset.

Example. *As an example, we show the results for one triplet of images from the Statue dataset. In Figure 5.15 the images are shown along with the 1177 tracks between them. The RANSAC based on the orthographic model gives a big enough set of inliers for the pose estimation but leaves out some true tracks.*

Figure 5.15 – A triplet of images from a statue in Château de Champs. The matches are drawn in green if they are considered inliers by the orthographic AC-RANSAC and red otherwise.

For this triplet of images, the solution found by the orthographic method is much closer to the final refined solution than the initial pose given by the perspective method. We can see

this quantitatively, in the number of iterations needed to reach the minimum in Table 5.4, where the iterations taken by the perspective solution are much higher. Also qualitatively in Figure 5.16, where we can see how the initial pose given by the orthographic method is practically the same as the final refined solution, while the perspective method gives a worse initial guess, which translates in incredibly high reprojection error. Looking at the orthogonal views of the estimated poses in Figure 5.16, it becomes clear that the error of the perspective-based method is mainly in the direction of the projection, which is related to a bad estimation in the depth of the object with respect to the camera. In Figure 5.17 we show the resulting 3D reconstruction and the estimated camera poses after the bundle adjustment.

	e_{repr}	e_{repr}-BA	inliers	# iter
Perspective	449.4004	0.1900	1116	29(+3)
Orthographic	0.3984	0.1900	1008	7(+3)

Table 5.4 – Results for the calibration of the triplet of images in Figure 5.15. The final solution of the bundle adjustment was produced with an extra minimization step involving all inliers that took 4 iterations in both cases. That is why it appears (+3) in the iterations column.

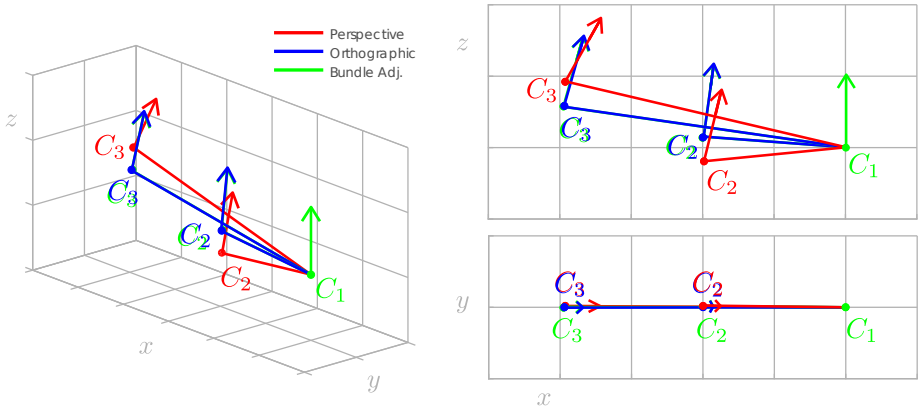


Figure 5.16 – Pose estimation of the triplet of images in Figure 5.15 given by the orthographic and perspective methods along with the solution of the bundle adjustment. Different views are shown for the configuration of the poses: a general view on the left and two orthogonal views on the right; one perpendicular to the y-axis (top) and one perpendicular to the z-axis (bottom).

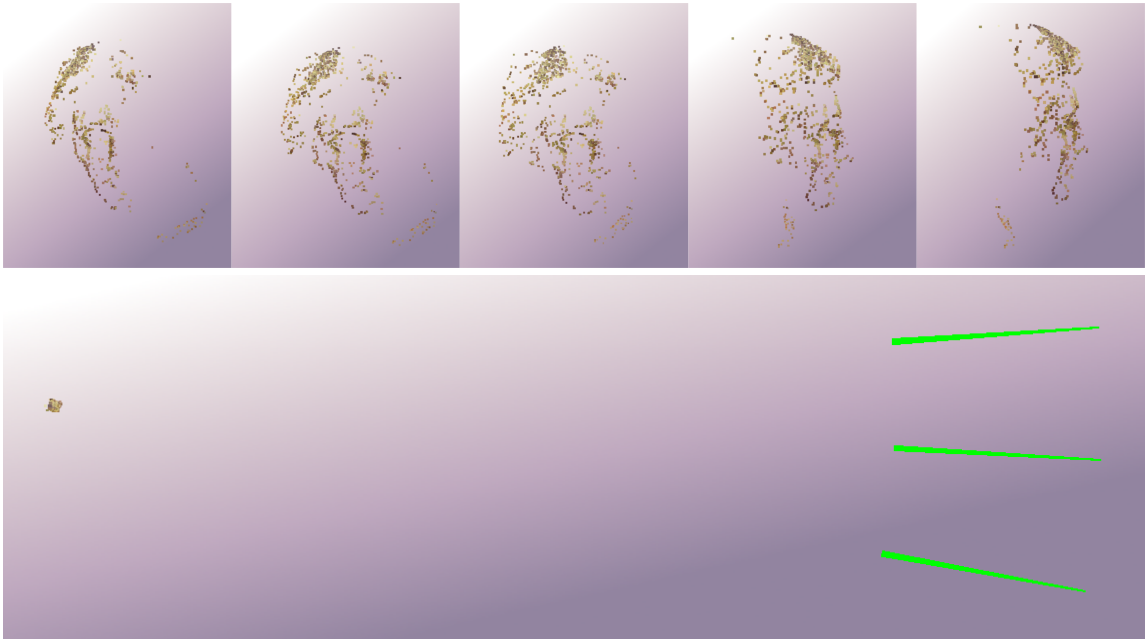


Figure 5.17 – Several views of the final reconstruction, after bundle adjustment, for the inliers of triplet of images shown in Figure 5.15.

5.6 Conclusion

In this chapter we have described a non-perspective model for pose estimation of perspective cameras and we have provided our own implementation. The model is based on the orthographic projection that we have theoretically shown to model a pinhole camera with center at infinity. With the experiments we have proven that this translates into a good performance in the pose estimation task in the case of cameras with long focal length lenses. Moreover, the method robustness to noise is outstanding. Using this estimation as an initial guess for a bundle adjustment procedure we have seen that the minimization converges faster. We conclude that this method based on the orthographic model is well suited for the external calibration of perspective cameras with relatively long focal lengths outperforming the perspective-based pose estimation and being comparable in computational complexity to the linear computation of the fundamental matrix.

Chapter 6

Conclusion and Perspectives

6.1 Conclusion

In this thesis we aimed at improving and reviewing some of the methods involved in 3D estimation and structure from motion. With accuracy in mind, we have analyzed different models and methods in order to provide reliable conclusions about their usefulness, robustness and advantages over others. Our research has yield to some improvements with small modifications, theoretical analysis and new models. During the thesis, we have considered essential to make publicly available the code we have developed with two objectives in mind: transparency, since we think it is important to be able to reproduce the experiments in any research work, and to contribute to the community with useful tools for the different steps of the SfM process where we have worked.

In the first place, we have presented a study and implementation of Yoon and Kweon's method [YK06] for disparity map computation. Our implementation is a useful and efficient tool for stereo vision. The study of this adaptive support method along with some variants and other approaches has shown the importance of adaptive weights in block-matching methods and its prevalent significance.

In the second place, we reviewed methods of estimation of trifocal tensor and of the pose of three views. Compared with the pose estimation obtained by the fundamental matrices from the pairs of views, our experiments demonstrate that the trifocal tensor does not offer enough improvement to be considered the preferred choice. By its simplicity and lower computation time, the recommended option is to consider only pairwise constraints through the fundamental matrix, provided some bundle adjustment is used at the end (which is also highly recommended, as it can routinely decrease the error by a significant factor).

Finally, we have described a non-perspective model for pose estimation of perspective cameras. The model is based on the orthographic projection that we have theoretically shown to model a pinhole camera with center at infinity. Our experiments have proven that this method has good performance in the pose estimation task in the case of cameras with long focal length lenses with great robustness to noise. We have shown that when using this estimation as an initial guess for a bundle adjustment procedure, the minimization converges faster. We conclude that this method based on the orthographic model is well suited for the external calibration of perspective cameras with relatively long focal lengths outperforming

the perspective-based pose estimation and being comparable in computational complexity to the linear computation of the fundamental matrix.

6.2 Future Work

Adaptive Supports for Stereo Matching

In the first part of the thesis we have looked at block-matching methods for disparity estimation and we quickly explored other possible methods with adaptive weights. While the results suggest that the accuracy limit for these methods might have been reached, the positive results using the Tree of Shapes-based distance, lets us believe that there is still room for improvement by testing other adaptive support weights. There are many ways to try to define a "probability" with which two pixels in the same patch might be at the same depth. We have seen that both spacial distance and color/intensity similarity are important factors to consider. Yet we have seen that taking into account some kind of "connectivity" can also be beneficial. We think that this is a lead worth exploring by investigating other distances that take into consideration the topology of the image.

Bundle Adjustment Optimization

The bundle adjustment optimization has not been discussed in depth in this thesis. However it is an essential step in any 3D reconstruction and pose estimation process. The research we have carried out in the pose estimation has brought to our attention a very interesting issue. We have observed both in Chapter 4 and Chapter 5 that the bundle adjustment optimization is able to reach the correct minimum, even when starting from a far initial solution. This motivates us to study in future research the possible extended local convexity of the minimized energy.

Simplified Point Clouds for Bundle Adjustment

During this thesis we have not approached the pose estimation for $M > 3$ or the lasts steps of the structure from motion, that being the application of a global bundle adjustment optimization with many views and partial correspondences (points seen in only two or three images). In the SfM process the bundle adjustment is applied multiple times in situations where the relative poses for the pairs and/or triplets have already been computed and optimized. We have become more and more interested in getting involved in this step whose main issues are their high computational cost and time, since the bundle adjustment optimization process increases in complexity with the number of views and 3D points involved.

Our intuition is that we can reduce the complexity of the minimization by reducing the parameters of the 3D reconstruction. This would be achieved by compacting all the pose information held by the point cloud of a pair or triplet of views by simplifying the cloud itself. The main geometric characteristics of a point cloud can be described by its principal components or moments. The idea is that by representing the cloud with an equivalent reduced set of 3D points and their projections onto the images, these could be used for a global estimation of all the poses. We believe this minimization should be equivalent to the

one taking all original correspondences with a small loss of information and would vastly improve the speed of the optimization algorithm. A last bundle adjustment should still be applied using the computed poses, for the original 3D points to be estimated.

We have started some experiments on this subject but realized it would be a much bigger project than anticipated. It remains a very interesting approach that we plan to continue studying in future research.

Appendix A

Pixel Error on Calibration

In the pose estimation task the error of reference to evaluate the quality of any method is the reprojection error. When ground truth is available we have more reliable options, the angular errors in rotations and translations. To better understand the behavior of these errors we address the question of how these two kinds of errors are connected. In order to find an answer we study the propagation of an error in calibration to the projected points, in terms of the displacement in pixels.

A.1 Perspective Camera

Let us take projection equation for the pinhole camera (2.5) and rewrite it in Cartesian coordinates,

$$\mathbf{x} = \gamma\big(K(R\mathbf{X} + \vec{t})\big) \qquad \mathbf{X} \in \mathbb{R}^3 \tag{A.1}$$

where γ is a function transforming homogeneous vectors into Cartesian coordinates, $\gamma(\vec{v}) = (\frac{v_1}{v_3}, \frac{v_2}{v_3})^\top$, for $\vec{v} \in \mathbb{R}^3$ with $v_3 \neq 0$. We can parametrize the rotation matrix R by its three Euler angles,

$$R(\beta_x, \beta_y, \beta_z) = \begin{pmatrix} \cos\beta_z & -\sin\beta_z & 0 \\ \sin\beta_z & \cos\beta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta_y & 0 & \sin\beta_y \\ 0 & 1 & 0 \\ -\sin\beta_y & 0 & \cos\beta_y \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\beta_x & -\sin\beta_x \\ 0 & \sin\beta_x & \cos\beta_x \end{pmatrix} . \tag{A.2}$$

For a fixed $\mathbf{X} \in \mathbb{R}^3$, we can define a function depending on all calibration parameters

$$\mathbf{x} = \zeta_{\mathrm{P}}(\omega) := \gamma\big(K(R\mathbf{X} + \vec{t})\big) \tag{A.3}$$

with $\omega = (f, c, \beta, \vec{t})$, where $\beta = (\beta_x, \beta_y, \beta_z)$.

The effect of a small variation $\delta\omega$ in the calibration parameters can be approximated by the linear Taylor expansion around a point ω_0,

$$\begin{aligned} \zeta_{\mathrm{P}}(\omega_0 + \delta\omega) \approx\ & \zeta_{\mathrm{P}}(\omega_0) + J(\zeta_{\mathrm{P}})(\omega_0) \cdot \delta\omega & \text{(A.4)} \\ \approx\ & \zeta_{\mathrm{P}}(\omega_0) + J_f(\zeta_{\mathrm{P}})(\omega_0) \cdot \delta f + J_c(\zeta_{\mathrm{P}})(\omega_0) \cdot \delta c & \text{(A.5)} \\ & + J_\beta(\zeta_{\mathrm{P}})(\omega_0) \cdot \delta\beta + J_{\vec{t}}(\zeta_{\mathrm{P}})(\omega_0) \cdot \delta\vec{t} & \text{(A.6)} \end{aligned}$$

where the Jacobian of ζ_{P}, $J(\zeta_{\mathrm{P}})$, has been separated in the second line into smaller matrices, the Jacobians with respect to the different parameters. In this way, we can see that for a small change in only one set of parameters, only the value in the initial point and the corresponding Jacobian will intervene in the error.

Let ω_0 be such that the camera center is at the origin of coordinates, $\vec{t_0} = (0,0,0)$ and $R_0 = \mathrm{Id}$, which makes $\beta_0 = (0,0,0)$. We also center the image coordinates so that $c_0 = (0,0)$. The initial focal length is f_0. Then, the Jacobian matrix evaluated in ω_0 is as follows

$$J(\zeta_{\mathrm{P}})(\omega_0) = \begin{pmatrix} \frac{X}{Z} & 1 & 0 & -\frac{XY}{Z^2}f_0 & \frac{X^2+Z^2}{Z^2}f_0 & -\frac{Y}{Z}f_0 & \frac{1}{Z}f_0 & 0 & -\frac{X}{Z^2}f_0 \\ \frac{Y}{Z} & 0 & 1 & -\frac{Y^2+Z^2}{Z^2}f_0 & \frac{XY}{Z^2}f_0 & \frac{X}{Z}f_0 & 0 & \frac{1}{Z}f_0 & -\frac{Y}{Z^2}f_0 \end{pmatrix} . \quad \text{(A.7)}$$

We can compute the approximate displacement of the original image point, $\mathbf{x}_0 = \zeta_{\mathrm{P}}(\omega)$, for each kind of increment,

Increment in...	**Pixel displacement $\delta\mathbf{x}$**	
focal length, δf	$\delta\mathbf{x} \approx \frac{\delta f}{f_0}\mathbf{x}_0$	(A.8)
camera center, δc	$\delta\mathbf{x} \approx \delta c$	(A.9)
rotation angles, $\delta\beta$	$\delta\mathbf{x} \approx \left(\frac{1}{f_0}\mathbf{x}_0\mathbf{x}_0^\top + f_0\mathrm{Id}_2\right)\begin{pmatrix}\delta\beta_y \\ -\delta\beta_x\end{pmatrix} + \begin{pmatrix}0 & -1 \\ 1 & 0\end{pmatrix}\mathbf{x}_0\delta\beta_z$	(A.10)
translation vector, $\delta\vec{t}$	$\delta\mathbf{x} \approx \frac{f_0}{Z}\begin{pmatrix}\delta t_x \\ \delta t_y\end{pmatrix} - \frac{\delta t_z}{Z}\mathbf{x}_0$	(A.11)

However, in order to be able to simulate an increment in the angular error in translations another parametrization for the translation vector $\vec{t}$ is needed. We can use spheric coordinates to parametrize the angles and norm of the vector with respect to the origin of coordinates,

$$\vec{t}(r,\theta,\phi) = \begin{pmatrix} r\cos\theta\sin\phi \\ r\sin\theta\sin\phi \\ r\cos\phi \end{pmatrix} \qquad \text{with } \; r \in [0,\infty],\ \theta \in [0,2\pi),\ \phi \in [0,\pi] \ . \quad \text{(A.12)}$$

We redefine our original function in terms of the new parameterization, setting $\omega = (f, c, \beta, \vec{t}(r,\theta,\phi))$ and the initial point as before but not fixing the camera center at the origin, instead we put $\vec{t_0} = \vec{t}(r_0,\theta_0,\phi_0)$. Then, for a variation of the translation parameters $(\delta r, \delta\theta, \delta\phi)$, the pixel displacement can be approximated as

$$\begin{aligned}\delta\mathbf{x} \approx & \left[\frac{f_0}{Z'}\sin\phi_0\begin{pmatrix}\cos\theta_0 \\ \sin\theta_0\end{pmatrix} - \mathbf{x}_0\cos\phi_0\right]\delta r \\ & + r_0\left[\frac{f_0}{Z'}\cos\phi_0\begin{pmatrix}\cos\theta_0 \\ \sin\theta_0\end{pmatrix} + \mathbf{x}_0\sin\phi_0\right]\delta\phi + r_0\left[\frac{f_0}{Z'}\sin\phi_0\begin{pmatrix}-\sin\theta_0 \\ \cos\theta_0\end{pmatrix}\right]\delta\theta \quad \text{(A.13)}\end{aligned}$$

where $Z' = Z + r_0\cos\theta_0$ is the depth of $\mathbf{X}$, relative to the camera.

We are interested in the effects of variations mainly in the angles of rotation and translation since we use them for evaluating the accuracy of an estimated pose. When the rotation angles are varied the pixel error only depends on the angle variations, the original point $\mathbf{x}_0$ and the focal length f_0 (Equation (A.10)). On the other hand, note that the error of an increment in the angles of translation $\delta\phi$ and $\delta\theta$ (Equation (A.13)) will depend on the original angles, the ratio f_0/Z', and also will be proportional to the translation's norm r_0. The latter is specially relevant when evaluating the pose estimation of several images taken with the same camera of the same scene. In such a case, and supposing that all cameras are at a similar distance of the scene, all the initial parameters involved in the the increment $\delta\mathbf{x}$ due to errors in rotation and translation angles, as we have computed, will be similar but the translation norm r_0. This means that, even though we might get very similar reprojection errors for all images with the retrieved relative poses, the angular errors in translations will not be proportional to those. In fact, they will be inversely proportional to the norm of the translations which will depend entirely on the chosen origin (usually one of the camera centers). We can better understand this effect with an example.

Application to Real Dataset

We take EPFL fountain-P11 dataset [SvHG$^+$08] to study pixel displacement in a specific case. The size of the images of this dataset is 3072×2048 and the principal point is situated at $c = (1520.7, 1006.8)$. The focal length is $f \approx 2761.8$ pixels.

Let us take a test point in the middle of the first quadrant of each image, $\mathbf{x}_0 = (\frac{3}{4}3072, \frac{3}{4}2048)$, that is $(783, 529)$ when the coordinates are centered at the principal point. We can already compute the approximation for some errors:

- Varying principal point has direct effect, $\delta\mathbf{x} = \delta c$.
- Varying the focal length in δf pixels,

$$\delta\mathbf{x} = \frac{\delta f}{2761.8}\begin{pmatrix} 783 \\ 529 \end{pmatrix} = \begin{pmatrix} 0.2781 \\ 0.1854 \end{pmatrix}\delta f \ .$$

- Varying the rotation angles in $\delta\beta_x$, $\delta\beta_y$, $\delta\beta_z$ degrees,

$$\delta\mathbf{x} = \begin{pmatrix} -2.4849 \\ -49.8595 \end{pmatrix}\delta\beta_x + \begin{pmatrix} 51.9302 \\ 2.4849 \end{pmatrix}\delta\beta_y + \begin{pmatrix} -8.9361 \\ 13.4041 \end{pmatrix}\delta\beta_z \ .$$

For the variation on the translation vector we need an approximation of the depth Z of the projected point $\mathbf{X}$. For each image we can take the average depth of the point cloud with respect to the camera. Also, we fix the origin of the space coordinates to the first camera center. All the computations depend on the chosen camera, for this reason we present a table with the approximated errors for each image. The camera positions are shown in Figure A.1. In Table A.1 the pixel displacement corresponding to each increment of the translation parameters (for both parameterizations) for the 11 images. Notice that the error produced by an increment in the spherical angles of the translation, highly increases for the cameras that are further away from the first camera.

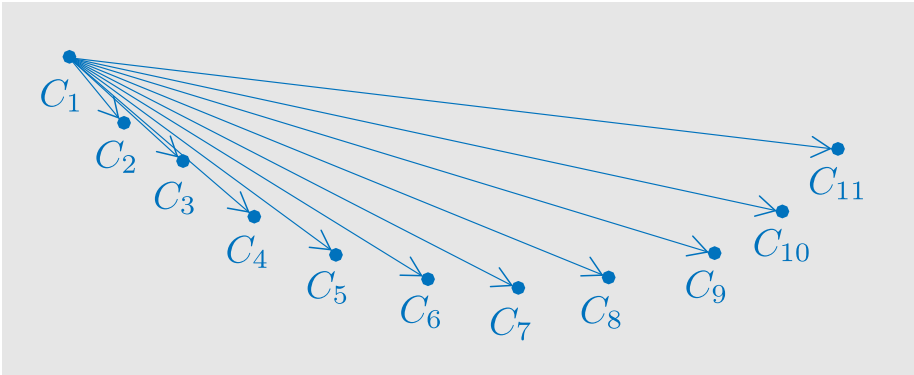


Figure A.1 – Representation of the camera centers positions in the fountain-P11 dataset [SvHG+08] and the translation vectors relative to the first camera.

		Increments in translation					
Camera		**vector parameters**			**spherical parameters**		
		δt_x	δt_y	δt_z	δr	$\delta\phi$	$\delta\theta$
1	δx	0.2729	0	-0.0774			
	δy	0	0.2729	-0.0523			
2	δx	0.2912	0	-0.0826	0.2961	1.7792	-0.1547
	δy	0	0.2912	-0.0558	0.0092	1.5715	8.2550
3	δx	0.3141	0	-0.0891	0.3214	2.9569	0.1843
	δy	0	0.3141	-0.0602	0.0024	3.1111	16.1391
4	δx	0.3353	0	-0.0951	0.3292	9.2710	-0.7514
	δy	0	0.3353	-0.0642	0.0056	5.2611	27.2240
5	δx	0.3613	0	-0.1025	0.3401	17.6485	-0.9753
	δy	0	0.3613	-0.0692	-0.0023	7.7406	39.5592
6	δx	0.3871	0	-0.1098	0.3417	29.9688	-1.3294
	δy	0	0.3871	-0.0742	-0.0110	10.4494	52.5282
7	δx	0.4081	0	-0.1157	0.3375	43.3294	-1.8215
	δy	0	0.4081	-0.0782	-0.0177	13.0053	64.1219
8	δx	0.4310	0	-0.1223	0.3255	60.0013	-2.2544
	δy	0	0.4310	-0.0826	-0.0266	15.4596	74.5133
9	δx	0.4361	0	-0.1237	0.2634	83.0613	-2.4649
	δy	0	0.4361	-0.0836	-0.0412	16.6814	76.7481
10	δx	0.4440	0	-0.1259	0.2257	98.1693	-3.6548
	δy	0	0.4440	-0.0851	-0.0450	18.3652	76.7928
11	δx	0.4610	0	-0.1307	0.1828	114.2365	-5.3383
	δy	0	0.4610	-0.0883	-0.0486	20.9220	73.7881

Increments in pixel coordinates

Table A.1 – Pixel displacement $\delta\mathbf{x} = (\delta x, \delta y)$ in the image point $\mathbf{x}_0$ for an increment of one unit (degrees for angles, mm for distance) in the translation parameters. All the values are shown for each one of the images in the fountain-P11 dataset [SvHG+08] but the first, since its center is the origin of coordinates and the increment with spheric coordinates cannot be computed.

A.2 Orthographic Projection

As a comparison, we have also studied the pixel displacement for the orthographic projection. Let us take the scaled-orthographic projection equation (5.4) for a space point $\mathbf{X} \in \mathbb{R}^3$ and rewrite it in the following way,

$$\mathbf{x} = \alpha \left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R\,\mathbf{X} + \begin{pmatrix} t^x \\ t^y \end{pmatrix} \right) + \begin{pmatrix} c^x \\ c^y \end{pmatrix} \qquad \text{with} \quad \alpha = \frac{f}{\vec{k}^\top \mathbf{O} + t^z} \tag{A.14}$$

where we will call α the focal ratio.

We can parametrize the rotation R by its three Euler angles as in eq. (A.2). Then, for a fixed $\mathbf{X} \in \mathbb{R}^3$, we can define a function depending on all calibration parameters

$$\mathbf{x} = \zeta_{\mathrm{SO}}(\omega) := \alpha \left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} R\,\mathbf{X} + \begin{pmatrix} t^x \\ t^y \end{pmatrix} \right) + \begin{pmatrix} c^x \\ c^y \end{pmatrix} \quad . \tag{A.15}$$

with $\omega = (\alpha, c, \beta, t')$, where $\beta = (\beta_x, \beta_y, \beta_z)$ and $t' = (t^x, t^y)$.

In the same way as in the perspective projection, the effect of a small variation $\delta\omega$ in the calibration parameters can be approximated by the linear Taylor expansion around point ω_0, separating the Jacobians as before,

$$\begin{aligned} \zeta_{\mathrm{SO}}(\omega_0 + \delta\omega) \approx &\ \zeta_{\mathrm{SO}}(\omega_0) + J(\zeta_{\mathrm{SO}})(\omega_0) \cdot \delta\omega && \text{(A.16)} \\ \approx &\ \zeta_{\mathrm{SO}}(\omega_0) + J_\alpha(\zeta_{\mathrm{SO}})(\omega_0) \cdot \delta\alpha + J_c(\zeta_{\mathrm{SO}})(\omega_0) \cdot \delta c && \text{(A.17)} \\ &+ J_\beta(\zeta_{\mathrm{SO}})(\omega_0) \cdot \delta\beta + J_{t'}(\zeta_{\mathrm{SO}})(\omega_0) \cdot \delta t' && \text{(A.18)} \end{aligned}$$

where we have also separated the Jacobian of ζ_{SO} in smaller matrices.

Setting ω_0 so that the center of the camera is the origin of the space points, $R_0 = \mathrm{Id}$, the initial focal ratio is α_0 and the image coordinates are centered on the principal point of the camera, we get the following Jacobian,

$$J(\zeta_{\mathrm{SO}})(\omega_0) = \begin{pmatrix} X & 1 & 0 & 0 & Z\alpha_0 & -Y\alpha_0 & \alpha_0 & 0 \\ Y & 0 & 1 & -Z\alpha_0 & 0 & X\alpha_0 & 0 & \alpha_0 \end{pmatrix} \tag{A.19}$$

We can compute the approximate displacement of the original image point, $\mathbf{x}_0 = \zeta_{\mathrm{SO}}(\omega)$, for each kind of increment,

Increment in...	**Pixel displacement $\delta\mathbf{x}$**	
focal ratio, $\delta\alpha$	$\delta\mathbf{x} \approx \dfrac{\delta\alpha}{\alpha_0}\mathbf{x}_0$	(A.20)
camera center, δc	$\delta\mathbf{x} \approx \delta c$	(A.21)
rotation angles, $\delta\beta$	$\delta\mathbf{x} \approx \alpha_0 Z \begin{pmatrix} \delta\beta_y \\ -\delta\beta_x \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{x}_0 \delta\beta_z$	(A.22)
translation vector, $\delta t'$	$\delta\mathbf{x} \approx \alpha_0 \begin{pmatrix} \delta t_x \\ \delta t_y \end{pmatrix}$	(A.23)

Notice that this increments depend only on the parameter variations, the original point $\mathbf{x}_0$, the ration α and the depth of the point, Z. Therefore, if we compare the pixel errors for two equal sets of parameters ω_0 but one with higher focal length and mean depth of the points, $\mathbf{O}$, (so that the ratio α is the same, as we do in our experiments), in particular Z will be higher, then the increment $\delta\mathbf{x}$ will be the same for the variation of all parameters but for $\delta\beta_x$ and $\delta\beta_y$, that will be proportional to Z. This means that for a pose estimation method for scaled-orthographic views, the angular errors of the rotation may decrease with respect to the reprojection errors when getting far away from the scene with fixed α.

Using spheric coordinates (A.12) for the translation vector as before we can compute the approximate increment for a variation of the translation parameters $(\delta r, \delta\theta, \delta\phi)$,

$$\delta\mathbf{x} \approx \alpha_0 \sin\phi_0 \begin{pmatrix}\cos\theta_0\\ \sin\theta_0\end{pmatrix} \delta r + r_0\alpha_0 \cos\phi_0 \begin{pmatrix}\cos\theta_0\\ \sin\theta_0\end{pmatrix} \delta\phi + r_0\alpha_0 \sin\phi_0 \begin{pmatrix}-\sin\theta_0\\ \cos\theta_0\end{pmatrix} \delta\theta \ . \quad \text{(A.24)}$$

We an see that the same effect as in the perspective projection appears, a variation in the translation angle will make the pixel error increase proportionally to the translation norm. Also, distancing the camera from the scene while keeping α fixed has no effect in the pixel error due to variations in the translations.

Appendix B

Gauss-Helmert model

The Gauss-Helmert model [Nei10], also known as *General Case of Least Squares Adjustment*, describes a functional model made up of condition equations $f(x, p) = 0 \in \mathbb{R}^{c_1}$ describing the relations between the model parameters $p \in \mathbb{R}^u$ and the observations $x \in \mathbb{R}^m$, an also additional constraints $g(p, q) = 0 \in \mathbb{R}^{c_2}$, where $q \in \mathbb{R}^s$ are unknown additional parameters which do not take part in the condition equations. Then for given noisy observations $x_0 \in \mathbb{R}^m$ it proposes to solve the least-squares problem

$$\underset{x,\,p,\,q}{\text{minimize}} \; \|x - x_0\|^2 \qquad \text{subject to} \qquad f(x, p) = 0, \qquad g(p, q) = 0 \tag{B.1}$$

using an iterative approach to find a local optimum by linearizing at each iteration the constraints f and g using Taylor. For x_k, p_k and q_k the estimated parameters at iteration k, the Taylor series to first order about the point (x_k, p_k, q_k) provides an approximation of the constraints,

$$\begin{aligned} f(x, p) &\approx f(x_k, p_k) + J_x(f)(x_k, p_k)\,(x - x_k) + J_p(f)(x_k, p_k)\,(p - p_k) && \text{(B.2)}\\ &= A\,\Delta p + B\,v - w && \text{(B.3)} \end{aligned}$$

with $w = -f(x_k, p_k) - B\,(x_0 - x_k)$. The matrices $A = J_p(f)$ and $B = J_x(f)$ are the Jacobian matrices of function f w.r.t. p and x respectively, $\Delta p = p - p_k$ and $v = x - x_0$ are the residuals.

$$\begin{aligned} g(p, q) &\approx g(p_k, q_k) + J_p(g)(p_k, q_k)\,(p - p_k) + J_q(f)(p_k, q_k)\,(q - q_k) && \text{(B.4)}\\ &= C\,\Delta p + D\,\Delta q - t && \text{(B.5)} \end{aligned}$$

with $t = -g(p_k, q_k)$. The matrices $C = J_p(g)$ and $D = J_q(g)$ are the Jacobian matrices of function g w.r.t. p and q and $\Delta q = q - q_k$.

With these linearized constraints, the original minimization problem (B.1) can be approximated by

$$\underset{x,\,p,\,q}{\text{minimize}} \; v^\top v \qquad \text{subject to} \qquad \begin{aligned} A\,\Delta p + B\,v - w &= 0,\\ C\,\Delta p + D\,\Delta q - t &= 0 \end{aligned} \;. \tag{B.6}$$

Using Lagrange multipliers $\lambda \in \mathbb{R}^{c_1}$ and $\mu \in \mathbb{R}^{c_2}$ we get the equivalent problem,

$$\underset{v,\,\Delta p,\,\Delta q,\,\lambda,\,\mu}{\text{minimize}} \; \Omega = v^\top v + 2\lambda^\top (A\,\Delta p + B\,v - w) + 2\mu^\top (C\,\Delta p + D\,\Delta q - t) \;. \tag{B.7}$$

In order to find a local minimum of Ω we differentiate with respect to v, Δp, Δq, λ and μ and set to zero the derivatives, obtaining the following linear system,

$$\begin{pmatrix} A^\top W A & 0 & C^\top \\ 0 & 0 & D^\top \\ C & D & 0 \end{pmatrix} \begin{pmatrix} \Delta p \\ \Delta q \\ \mu \end{pmatrix} = \begin{pmatrix} A^\top W w \\ 0 \\ t \end{pmatrix} \quad \text{then} \quad \begin{array}{rl} \lambda = & W(A\Delta p - w) \\ v = & -B^\top \lambda \end{array} \tag{B.8}$$

where $W = (BB^\top)^{-1}$. Therefore, by solving this system, the next iteration point can be computed $x_{k+1} = x_0 + v$, $p_{k+1} = p_k + \Delta p$, $q_{k+1} = q_k + \Delta q$. As any other iterative method, an initial estimated point close enough to the minimum is necessary for initialization.

An example of the Gauss-Helmert model applied to the fundamental matrix optimization is detailed below.

Application to Fundamental Matrix

In 2-view and 3-view models, as seen in Chapter 4, the extra parameters q are not necessary in the minimization problem (B.1). Now we study the optimization of the fundamental matrix parameters for the 2-view model. Let us have N pairs of matching points in two images, $\{(\mathbf{x}_1^i, \mathbf{x}_2^i)\}_{i=1,\dots,N}$, we suppose that there is some noise in the measurements. These will form the vector of observations,

$$x_0 = (\mathbf{x}_1^1, \mathbf{x}_2^1, \dots, \mathbf{x}_1^N, \mathbf{x}_2^N) \ \in \mathbb{R}^{2N} \ . \tag{B.9}$$

The parameters to optimize p are the 9 parameters of the fundamental matrix,

$$p = (f_1, f_2, \dots, f_9) \quad \text{where} \quad F_{21} = \begin{pmatrix} f_1 & f_4 & f_7 \\ f_2 & f_5 & f_8 \\ f_3 & f_6 & f_9 \end{pmatrix} \ . \tag{B.10}$$

The constraints f will be the epipolar constraints (4.3) for each pair of matching points and the two internal constraints of F_{21} (null determinant and fixed scale) will from g,

$$f(x_0, p) = \left(\ (\bar{\mathbf{x}}_2^1)^\top F_{21}\, \bar{\mathbf{x}}_1^1, \ (\bar{\mathbf{x}}_2^2)^\top F_{21}\, \bar{\mathbf{x}}_1^2, \ \dots \ , \ (\bar{\mathbf{x}}_2^N)^\top F_{21}\, \bar{\mathbf{x}}_1^N \ \right) \tag{B.11}$$

$$g(p) = \left(\ |F_{21}|, \ \|F_{21}\|^2 - 1 \ \right) \tag{B.12}$$

The Jacobian matrices are,

$$A = \begin{pmatrix} x_1^1 x_2^1 & x_1^1 y_2^1 & x_1^1 & x_2^1 y_1^1 & y_1^1 y_2^1 & y_1^1 & x_2^1 & y_2^1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^N x_2^N & x_1^N y_2^N & x_1^N & x_2^N y_1^N & y_1^N y_2^N & y_1^N & x_2^N & y_2^N & 1 \end{pmatrix} \tag{B.13}$$

$$B = \begin{pmatrix} f_3 + f_1 x_2^1 + f_2 y_2^1 & f_6 + f_4 x_2^1 + f_5 y_2^1 & f_7 + f_1 x_1^1 + f_4 y_1^1 & f_8 + f2\, x_1^1 + f_5 y_1^1 \\ \vdots & \vdots & \vdots & \vdots \\ f_3 + f_1 x_2^N + f_2 y_2^N & f_6 + f_4 x_2^N + f_5 y_2^N & f_7 + f_1 x_1^N + f_4 y_1^N & f_8 + f2\, x_1^N + f_5 y_1^N \end{pmatrix} \tag{B.14}$$

$$C = \begin{pmatrix} f_5 f_9 - f_6 f_8 & f_6 f_7 - f_4 f_9 & f_4 f_8 - f_5 f_7 & f_3 f_8 - f_2 f_9 & f_1 f_9 - f_3 f_7 & \dots \\ 2\, f_1 & 2\, f_2 & 2\, f_3 & 2\, f_4 & 2\, f_5 & \dots \\ \dots & f_2 f_7 - f_1 f_8 & f_2 f_6 - f_3 f_5 & f_3 f_4 - f_1 f_6 & f_1 f_5 - f_2 f_4 & \\ \dots & 2\, f_6 & 2\, f_7 & 2\, f_8 & 2\, f_9 & \end{pmatrix} \tag{B.15}$$

The matrix D does not exist since there are no extra parameters q.

The iterative process starts from an initial point, that would be the measured observations x_0, and an initial estimation of the parameters of the fundamental matrix p_0 verifying the internal constraints $g(p_0) = 0$. Then, from the point (x_k, p_k) at iteration k the matrices A_k, B_k and C_k are computed, as well as the vector $w_k = -f(x_k, p_k) - B_k\,(x_0 - x_k)$, matrix $W_k = (B_k {B_k}^\top)^{-1}$ and $t_k = -g(p_k)$. The system in eq. (B.8) is simplified due to the absence of matrix D,

$$\begin{pmatrix} {A_k}^\top W_k A_k & {C_k}^\top \\ C_k & 0 \end{pmatrix} \begin{pmatrix} \Delta p_k \\ \mu_k \end{pmatrix} = \begin{pmatrix} {A_k}^\top W_k w_k \\ t_k \end{pmatrix} \tag{B.16}$$

and can be solved to find Δp_k and μ_k, with which we compute $\lambda_k = W_k(A_k \Delta p_k - w_k)$ and $v_k = -{B_k}^\top \lambda_k$. With that, the next iteration point can be computed,

$$x_{k+1} = x_0 + v_k \qquad p_{k+1} = p_k + \Delta p_k \ . \tag{B.17}$$

Standard stopping criteria can be used to end the iterative algorithm. For instance, we consider a threshold on the norm of the increments Δp_k and $x_{k+1} - x_k$.

Bibliography

[Art62] D. W. G. Arthur. Model formation with narrow-angle photography. *The Photogrammetric Record*, 4(19):49–53, 1962. http://dx.doi.org/10.1111/j.1477-9730.1962.tb00325.x.

[BCM03] Ballester, Coloma, Caselles, Vicent, and Monasse, P. The tree of shapes of an image. *ESAIM: COCV*, 9:1–18, 2003.

[Can00] Nikos Canterakis. *A Minimal Set of Constraints for the Trifocal Tensor*, pages 84–99. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[DMM08] Agnés Desolneux, Lionel Moisan, and Jean-Michel Morel. *From Gestalt Theory to Image Analysis*. Springer New York, 2008.

[DP06] Johan Debayle and Jean-Charles Pinoli. General adaptive neighborhood image processing. *J. Math. Imaging Vis.*, 25(2):267–284, sep 2006.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[FP98] Olivier D. Faugeras and Théodore Papadopoulo. A nonlinear method for estimating the projective geometry of three views. In *ICCV*, 1998.

[Har97] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, Jun 1997. http://dx.doi.org/10.1109/34.601246.

[HBG10] A. Hosni, M. Bleyer, and M. Gelautz. Near real-time stereo with adaptive support weight approaches. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 1–8, 2010. http://publik.tuwien.ac.at/files/PubDat_190923.pdf.

[HBG13] A. Hosni, M. Bleyer, and M. Gelautz. Secrets of adaptive support weight techniques for local stereo matching. *Computer Vision and Image Understanding*, 117(6):620–632, 2013. http://dx.doi.org/10.1016/j.cviu.2013.01.007.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[HTKP09] M. Havlena, A. Torii, J. Knopp, and T. Pajdla. Randomized structure from motion based on atomic 3d models from camera triplets. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2874–2881, June 2009.

[HZ04] R. I. Hartley and A Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. ISBN 0521540518.

[Lev44] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[LH81] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981. `http://dx.doi.org/10.1038/293133a0`.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[LZ99] C. Loop and Zhengyou Zhang. Computing rectifying homographies for stereo vision. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 125–131 Vol. 1, June 1999.

[Met36] W. Metzger. *Gesetze des Sehens*. W. Kramer Frankfurt am Main, 1936. ISBN 3782910478.

[MHV+04] Yi Ma, Kun Huang, René Vidal, Jana Košecká, and Shankar Sastry. Rank conditions on the multiple-view matrix. *International Journal of Computer Vision*, 59(2):115–137, 2004.

[MMM12a] Lionel Moisan, Pierre Moulon, and Pascal Monasse. Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers. *Image Processing On Line*, 2:56–73, 2012.

[MMM12b] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Adaptive structure from motion with a contrario model estimation. In *Asian Conference on Computer Vision*, pages 257–270. Springer, 2012.

[MMM13a] Pierre Moulon, Pascal Monasse, and Renaud Marlet. *Adaptive Structure from Motion with a Contrario Model Estimation*, pages 257–270. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. `http://doi.org/10.1007/978-3-642-37447-0_20`.

[MMM13b] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3248–3255, 2013.

[MMM16] Lionel Moisan, Pierre Moulon, and Pascal Monasse. Fundamental Matrix of a Stereo Pair, with A Contrario Elimination of Outliers. *Image Processing On Line*, 6:89–113, 2016. `http://doi.org/10.5201/ipol.2016.147`.

[MS01] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision (ICCV '01)*, volume 1, pages 525–531, Vancouver, Canada, July 2001. IEEE Computer society.

[MS04] Lionel Moisan and Bérenger Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57(3):201–218, may 2004.

[MSZ+11] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 467–474, 2011. `http://dx.doi.org/10.1109/ICCVW.2011.6130280`.

[Nei10] Frank Neitzel. Generalization of total least-squares on example of unweighted and weighted 2D similarity transformation. *Journal of Geodesy*, 84(12):751–762, 2010.

[Nor09] K. Nordberg. A minimal parameterization of the trifocal tensor. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1224–1230, June 2009.

[OH02] Tetsu Ono and Susumu Hattori. Fundamental principle of image orientation using orthogonal projection model. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34:194–199, 2002.

[PD07] M. P. Pierrot-Deseilligny. MicMac, un logiciel pour la mise en correspondance automatique d'images dans le contexte géographique. *Bulletin d'Information Scientifique et Technique de l'IGN n*, 77:1, 2007.

[PD11] J. Pinoli and J. Debayle. General adaptive distance transforms on gray tone images: Application to image segmentation. In *2011 18th IEEE International Conference on Image Processing*, pages 2845–2848, Sept 2011.

[PF98] Théodore Papadopoulo and Olivier Faugeras. *A new characterization of the trifocal tensor*, pages 109–123. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[PH14] J. Ponce and M. Hebert. Trinocular geometry revisited. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, June 2014.

[PK97] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, Mar 1997. `http://doi.org/10.1109/34.584098`.

[RDGM10] Julien Rabin, Julie Delon, Yann Gousseau, and Lionel Moisan. Macransac: a robust algorithm for the recognition of multiple objects. In *in Proceedings of 3DPTV 2010*, page 051, 2010.

[Res02] C. Ressl. A minimal set of constraints and a minimal parameterization for the trifocal tensor. In *In The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV, Part 3A, ISPRS-Comm. III Symposium, Graz, 9*, page 13, 2002.

[ROD14] Ives Rey Otero and Mauricio Delbracio. Anatomy of the SIFT Method. *Image Processing On Line*, 4:370–396, 2014. https://doi.org/10.5201/ipol.2014.82.

[SF11] Christos Stamatopoulos and Clive S. Fraser. Calibration of long focal length cameras in close range photogrammetry. *The Photogrammetric Record*, 26(135):339–360, 2011. http://dx.doi.org/10.1111/j.1477-9730.2011.00648.x.

[SF16] Johannes L. Schönberger and Jan Michael Frahm. *Structure-from-motion revisited*, volume 2016-January, pages 4104–4113. IEEE Computer Society, United States, 2016.

[SHK+14] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, volume 8753, pages 31–42, 09 2014.

[Sna10] Noah Snavely. Bundler: Structure from motion for unordered image collections, 2010. https://www.cs.cornell.edu/~snavely/bundler/.

[SP07] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[SS03] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I, June 2003.

[SSZ01] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, Dec 2001.

[SvHG+08] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[TK92] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, Nov 1992. http://doi.org/10.1007/BF00129684.

[TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *IEEE Sixth International Conference on Computer Vision*, pages 839–846, 1998. http://dx.doi.org/10.1109/ICCV.1998.710815.

[TM14] P. Tan and P. Monasse. Stereo Disparity through Cost Aggregation with Guided Filter. *Image Processing On Line*, 4:252–275, 2014. `http://dx.doi.org/10.5201/ipol.2014.78`.

[TMHF00] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment — a modern synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, pages 298–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[TPH16] Matthew Trager, Jean Ponce, and Martial Hebert. Trinocular Geometry Revisited. *International Journal on Computer Vision (IJCV)*, 2016.

[TZ97] P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, 1997.

[YF14] Xieliu Yang and Suping Fang. Effect of field of view on the accuracy of camera calibration. *Optik - International Journal for Light and Electron Optics*, 125(2):844 – 849, 2014. `http://doi.org/10.1016/j.ijleo.2013.07.089`.

[YK06] K.-J. Yoon and I.S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, 2006. `http://dx.doi.org/10.1109/TPAMI.2006.70`.

[Zha96] Z. Zhang. 3d reconstruction based on homography mapping. *Proc. ARPA96*, pages 1007–1012, 1996. `http://ci.nii.ac.jp/naid/10030410351/en/`.