



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Joint Estimation of Correspondence and Motion Using Global Rigidity and Local Descriptors

Ricardo da Silveira Cabral

Dissertação para a obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

Júri:

Presidente:	Professor Carlos Jorge Ferreira Silvestre
Orientador:	Professor Pedro Manuel Quintas Aguiar
Co-Orientador:	Professor João Manuel de Freitas Xavier
Vogal:	Professor José Manuel Bioucas Dias

Julho de 2009

Agradecimentos

Neste culminar de uma etapa, ainda que apenas um máximo local na função da aprendizagem, surge nesta secção um espaço de pausa. Muito embora sabendo que a minha parca eloquência não fará jus à tarefa, devo um sincero agradecimento por escrito a um conjunto de pessoas sem as quais aqui não teria chegado.

Em primeiro lugar, aos meus orientadores, Professor Pedro Aguiar e Professor João Xavier, pela disponibilidade e incansável apoio. O seu constante método e rigor ajudaram-me a estabelecer *upper e lower bounds* que garantiram a convergência deste trabalho.

Ao longo destes últimos anos, nos quais tive como segunda casa o Instituto Superior Técnico e o Instituto de Sistemas e Robótica, tive o prazer de conhecer Professores que contribuíram significativamente para o meu percurso académico e pessoal, de entre os quais realço o Professor João Paulo Costeira, o Professor João Luís Sobrinho, o Professor Luís Borges de Almeida e o Professor Victor Barroso. Tanto os meus orientadores como estas individualidades, como disse Randy Pausch na sua última aula, “já se esqueceram de mais do que alguma vez saberei”, pelo que foi um verdadeiro privilégio poder beneficiar da sua companhia.

Aos meus colegas e amigos, em especial à Daniela, à Carla, ao Tiago e ao Nuno, por me lembrarem ao longo deste percurso que os cafés e a cerveja ainda existem, e pelas discussões, quer altamente técnicas ou de índole filosófica (da de trazer por casa).

A toda a minha família, por todo o apoio e carinho.

À minha irmã, por todas as técnicas de argumentação que me fez aprender para poder ter razão, que certamente me serão úteis em contextos mais sérios.

Aos meus pais, porque nunca é demais agradecer o terem toda a paciência do mundo e por, ainda hoje, se preocuparem em saber se eu ando a comer fruta.

À Maria, por me ensinar a amar, por tomar conta de mim (mesmo nos meus sonhos) e por acrescentar toda uma dimensão de pensamentos que não seria capaz de ter sozinho.

Resumo

Nesta tese, apresenta-se uma nova formulação para o problema das correspondências no caso de objectos rígidos e lambertianos, no contexto de problemas de “Structure from Motion”. O objectivo trata-se, portanto, de estimar as correspondências e o movimento da câmara de uma forma integrada, sem nenhuma informação outra que um par de imagens.

A solução aqui apresentada toma como entradas conjuntos de coordenadas de pontos, ou mais especificamente um conjunto de pontos na primeira imagem, a cada um dos quais corresponde um conjunto de coordenadas de potenciais correspondências na segunda, obtidas como tendo o mesmo valor de intensidade. Com estes dados, estima-se o movimento relativo entre ambas as câmaras e selecciona-se um número de candidatos mais prováveis para cada ponto na primeira imagem (o que, em particular, pode tomar o caso de correspondência 1-para-1).

Para cumprir este objectivo, codifica-se a informação imbuída na intensidade dos pontos na imagem com um modelo que restringe as trajectórias destes segundo um modelo rígido e dependente das profundidades do objecto num único problema de optimização. Nestas condições, é possível descrever-se o conjunto de todos os movimentos possíveis como uma variedade de Stiefel, cujas propriedades permitem a resolução deste problema através de uma minimização sobre uma grelha de pesquisa em dois parâmetros. O algoritmo resultante — Correspondence Estimation using Rigidity and local Descriptors, ou CERD — permite, portanto, uma resolução com complexidade polinomial de um problema de natureza tipicamente combinatória, exibindo propriedades óptimas no caso em que os parâmetros da câmara são conhecidos.

As experiências efectuadas com dados sintéticos e reais mostram que o CERD consegue estimar correctamente as correspondências e movimento, quando sujeito a um grande número de *outliers* e oclusão leve. Para além disso, a vantagem de poder ser conjugado com qualquer algoritmo de extração de *features* (e.g., SIFT) providencia um nível adicional de versatilidade, alargando a sua aplicabilidade a uma vasta gama de possíveis cenários.

Palavras-chave: Visão estéreo; Correspondência entre imagens; Problema das correspondências; Permutações; estimação de forma a partir do movimento; Visão por Computador; Percepção de profundidade; Brilho do pixel; Intensidade do pixel; Objecto rígido; Objecto Lambertiano.

Abstract

In this thesis we present a new approach to the modeling of the correspondence problem for rigid and Lambertian objects within a Structure from Motion framework. Our goal is to solve for correspondences and camera motion altogether, using only a pair of images with no additional information.

The approach presented takes as input sets of point coordinates, specifically a set of points in the first image, each with a set of correspondence candidates in the second image presenting the same intensity value. With this data, we estimate motion between both cameras and select a number of most likely candidates for each point in the first image (which, in particular, can be the case of 1-to-1 correspondence).

To fulfill this goal, our work merges information from brightness cues available in image points extracted using local descriptor algorithms with a model that constrains point trajectories due to the rigidity of the object. The use of both these constraints allows us to formulate the estimation of correspondences, motion and depth within a single optimization problem. We describe the set of possible motions as a Stiefel manifold, whose properties allow us to solve the optimization problem by minimizing a cost function over a two-parameter search grid. The resulting algorithm — Correspondence Estimation using Rigidity and local Descriptors, or CERD — has polynomial complexity, therefore bypassing the combinatorial explosion typically associated with the correspondence problem, and has optimal properties when the camera parameters are known.

Our experiments with synthetic and real data show that CERD is able to accurately estimate correspondence and motion, being able to cope with a large number of outliers and light occlusion. Additionally, its ability to be coupled with any feature extraction algorithm (e.g., SIFT) allows for extra versatility and broadens its applicability to a wide range of scenarios.

Keywords: Stereo Vision; Image Matching; Correspondence problem; Permutations; Structure from Motion; Computer Vision; Depth Perception; Pixel brightness; Pixel intensity; Rigid object; Lambertian object.

Contents

1	Introduction	2
1.1	Context and motivation	2
1.2	State of the art	4
1.3	Proposed approach	5
1.4	Summary of original contributions	5
1.5	Thesis outline	6
2	Problem formulation	7
2.1	Motion	7
2.2	Cameras	8
2.3	Structure from Motion	8
2.4	SfM without correspondences as an optimization problem	9
3	CERD — Correspondence Estimation using Rigidity and local Descriptors	11
3.1	Solving for translation	13
3.2	Solving for rotation	15
3.3	Implementation	16
4	Experiments	18
4.1	Methodology and setup	18
4.2	Known θ, ϕ	18
4.3	Fully automatic SfM	23
4.4	Real images	26
4.5	Coupling with existing feature extraction methods	33
5	Conclusions and future work	36
A	Set distance	37
B	Parameterization of the Stiefel manifold $O(2,3)$	39
C	Depth as a byproduct of CERD	41

List of Figures

1.1	Panorama obtained from 2 images. The stitching is made by overlapping the images according to features common to both frames.	3
1.2	Partial reconstruction of a cube from 3 images. The process can be seen as a generalization of image stitching (see Fig. 1.1) into 3D space, as the correspondence information embeds the information on the camera positions and the 3D coordinates of the object. <i>N.B.</i> : without knowing the distance of the camera to the object, the real scale of the object is not known.	3
2.1	Orientation of the x and y axis in an image.	7
2.2	Image motion decomposed in three sub-transformations: the one applied by the operator \mathbf{A} (left), translation along the vector \mathbf{b} according to each point depth z_i (center) and translation \mathbf{t} applied commonly to all points (right).	9
3.1	Ambiguity of correspondence for known \mathbf{H} and unknown object depths \mathbf{z} . The dashed lines represent the variation of object depth \mathbf{z} , making each intersection of the lines passing along $\mathbf{A}\mathbf{u}_i$ with direction given by the vector \mathbf{b} and level curves \mathcal{S}_i a valid correspondence.	12
3.2	RGB color space. The axis represent channels Red, Green and Blue, with integers varying between the values 0 and 255.	17
4.1	Points in a pair of synthesized images for $N = 5$ and $N_c = 10$. The colors represent points with equal intensity value: (a) input data; (b) results obtained.	19
4.2	Behavior of $C(\gamma_{m,n})$ for $N = 5$ and $N_c = 10$	19
4.3	Image points after projection in \mathbf{b}_\perp and subtraction of component γ for $N = 5$ and $N_c = 10$. The colors represent points with equal intensity value.	20
4.4	Point and candidate distribution throughout the images for a typical scenario of $N = 20$ points and $N_c = 40$ match candidates. The existence of a large number of candidate matches makes the correspondence estimation hard, even for a human.	21
4.5	Variation of $C(\gamma_{m,n})$ with AWGN of standard deviation σ averaged over 6000 tests.	22
4.6	Variation of average point disparity with AWGN of standard deviation σ averaged over 6000 tests.	22
4.7	Variation of $C(\gamma_{m,n})$ for known camera parameters for $N = 5$ and $N_c = 10$ for noise with standard deviation of 1×10^{-3}	22
4.8	Variation of cost function $\Phi(\mathbf{A}, \mathbf{b})$, average point displacement and camera parameters θ and ϕ estimation errors with the number N of points and number N_c of candidate matches for a grid resolution of 2500 points: (a) variation of $\Phi(\mathbf{A}, \mathbf{b})$; (b) variation of average point disparity; (c) variation of estimation error for θ ; and, (d) variation of estimation error for ϕ	23

4.9	Variation of cost function $\Phi(\mathbf{A}, \mathbf{b})$ with camera parameters θ and ϕ for $N = 5$ points and $N_c = 10$ candidates for a grid resolution of 2500 points (the cross in the contours represents parameters θ, ϕ used to generate data): (a) surface obtained with multiple candidates; (b) surface obtained with ground truth; (c) contour obtained with multiple candidates; and, (d) contour obtained with ground truth.	24
4.10	Variation of $\Phi(\mathbf{A}, \mathbf{b})$ with grid resolution and number of candidates N_c for $N = 10$ points, averaged over 200 tests.	25
4.11	Variation of average point disparity with grid resolution and number of candidates N_c for a set of $N = 10$ points, averaged over 200 tests.	25
4.12	Error on the estimation of θ with variation of grid resolution and number of candidates N_c for $N = 10$ points, averaged over 200 tests.	25
4.13	Error on the estimation of ϕ with variation of grid resolution and number of candidates N_c for $N = 10$ points, averaged over 200 tests.	26
4.14	Variation of $\Phi(\mathbf{A}, \mathbf{b})$ with number of occlusions N_o and number of candidates N_c for $N = 30$ points averaged over 200 tests.	27
4.15	Variation of average point disparity with number of occlusions N_o and number of candidates N_c for $N = 30$ points averaged over 200 tests.	27
4.16	Error on the estimation of θ with variation of number of occlusions N_o and number of candidates N_c for $N = 30$ points averaged over 200 tests.	27
4.17	Error on the estimation of ϕ with variation of number of occlusions N_o and number of candidates N_c for $N = 30$ points averaged over 200 tests.	28
4.18	Lego blocks pair, INRIA © copyright.	29
4.19	Corners detected by the algorithm in Lego blocks pair for use in correspondence estimation.	29
4.20	Matching results for Lego blocks pair. Red lines represent mismatches.	30
4.21	Matching results for Lego blocks pair with occluded points. Blue line represents correspondence obtained for occluded point. Red lines represent mismatches.	30
4.22	Rubik's cube pair.	31
4.23	Corners detected by the algorithm in Rubik's cube pair for use in correspondence estimation.	31
4.24	Matching results for Rubik's cube pair. Red lines represent mismatches.	32
4.25	Results for best 5 matches in Rubik's cube pair. Green lines represent ground truth explanation.	32
4.26	Original Rubik's cube pair.	33
4.27	Corners detected by the algorithm in original Rubik's cube pair for use in correspondence estimation.	33
4.28	Correspondence results for original Rubik's cube pair using corners as features for a grid resolution of 2500 points.	34
4.29	Features detected by SIFT in the original Rubik's cube pair for use in correspondence estimation.	34
4.30	Correspondence results for original Rubik's cube pair using SIFT features for a grid resolution of 2500 points.	35
A.1	Visual explanation of distance operator $d(\mathbf{p} + \mathbf{a}k, \mathcal{S})$	38
C.1	Object in three-dimensional (3D) coordinate system.	42
C.2	Relative position of the cameras w.r.t. the world coordinate system. Left: Camera 1. Right: Camera 2.	42
C.3	Image obtained by camera 1. The remainder of the points is invisible due having overlapping projection coordinates with the points represented.	43

C.4	Image obtained by camera 2. The remainder of the points is invisible due to having overlapping projected coordinates with the points represented.	43
C.5	Partial reconstruction of the cube (points A, B, E, F) according to the coordinates obtained in (C) using different values for the parameter r . Left: $r = 0$. Right: $r = 0.9$. All the other points are plotted with the original coordinates for comparison purposes.	44

List of Tables

4.1	Camera calibration parameters for Lego blocks pair.	28
4.2	Error between calibration parameters obtained and ground truth for Lego blocks pair.	28
C.1	Coordinates of the points in Fig. C.1 w.r.t. the world coordinate system.	42
C.2	Coordinates of the points in Fig. C.3 w.r.t. camera 1's coordinate system.	43
C.3	Coordinates of the points in Fig. C.4 w.r.t. camera 2's coordinate system.	43

Acronyms

2D two-dimensional

3D three-dimensional

AWGN Additive White Gaussian Noise

CERD Correspondence Estimation using Rigidity and local Descriptors

OF Optical Flow

SfM Structure from Motion

SVD Singular Value Decomposition

Notation

\mathbb{R}	Set of Real Numbers.
$\{a, \dots, z\} \in \mathbb{R}$	Scalars.
$\{\mathbf{a}, \dots, \mathbf{z}\} \in \mathbb{R}^n$	n -dimension vectors with real entries.
$\mathbf{1}_n$	n -dimension vector of ones. When dimension can be easily inferred from the context, the n is dropped.
$\{\mathbf{A}, \dots, \mathbf{Z}\} \in \mathbb{R}^{n \times m}$	$n \times m$ matrices with real entries.
\mathbf{I}_n	n -by- n Identity matrix. When dimension can be easily inferred from the context, the n is dropped.
$\mathbf{0}_n$	n -by- n matrix of zeros. When dimension can be easily inferred from the context, the n is dropped.
$\ \cdot\ _1$	ℓ_1 Norm.
$\ \cdot\ _2$	ℓ_2 Norm.

Chapter 1

Introduction

1.1 Context and motivation

Symmetry presents itself in almost every aspect of nature, but one remarkable example of it is the fact that natural selection has geared the majority of animals towards having no less than two eyes [1].

The evolution of the eyes or, in a more abstract sense, light sensing organs, is of utter importance for survival, as it allows both predators and preys to perceive their surroundings, detect motion and ultimately identify its origin as a friend or foe; the redundancy of the eyes not only helps secure this purpose, but allows the gathering of additional information, extending the field of view or pinpointing positions more accurately in a 3D space.

Many problems in the area of computer vision intend to endow electronic systems with such kind of information extraction mechanisms and are, alike, solved by using redundancy. Whether obtained from multiple cameras or from a single camera over time and in different positions, systems rely on the existence of multiple images (*frames*) to infer additional information: depth — referred to in the literature as Structure from Motion (SfM) [2, 3, 4] — panoramic vision (or *image stitching*) [5], resolution enhancement [5], object tracking [6] or face recognition [7, 8] are all examples of what can be done today using a computer no different than the ones already present in a modern household (for a review of the various existing mechanisms for the referred applications, we refer the reader to [9, 10]).

However, most of the solutions for these problems require the position of a set of points to be known in various images. The answer as to why this information is needed as an input for these methods can be inferred from simple visual explanations: in Fig. 1.1, where an example of image stitching is depicted, the presence of a number of points in more than one projection acts, similarly to the edges in puzzle pieces, as cues on how both images are positioned in the plane relative to each other; for the case of SfM, which we exemplify in Fig. 1.2, the puzzle analogy still holds, with the exception that now the pieces are put together in a 3D space and each piece correspondence bears information about the way the remainder of the pieces exhibiting the same points are positioned in space.

The task of finding the coordinates (or *trajectories*) of a set of points in various images, known as the *correspondence problem*, is often relegated to manual labor due to its intrinsic combinatoric nature: to each point in the first image corresponds a high cardinality set of respective candidates in subsequent images, caused by the scarcity of criteria (*features*) available in the images — typically, intensity value and position — to distinguish between candidates, when no additional information is present.

The lack of features, together with the existence of noise in captured images — *e.g.*, from the image acquisition system itself or changes of lighting — and the possible occlusion of points in some frames makes it challenging to obtain correct matches automatically, using a computer, in an efficient manner.

In this thesis, we tackle the correspondence problem for the case where the depicted objects are Lambertian

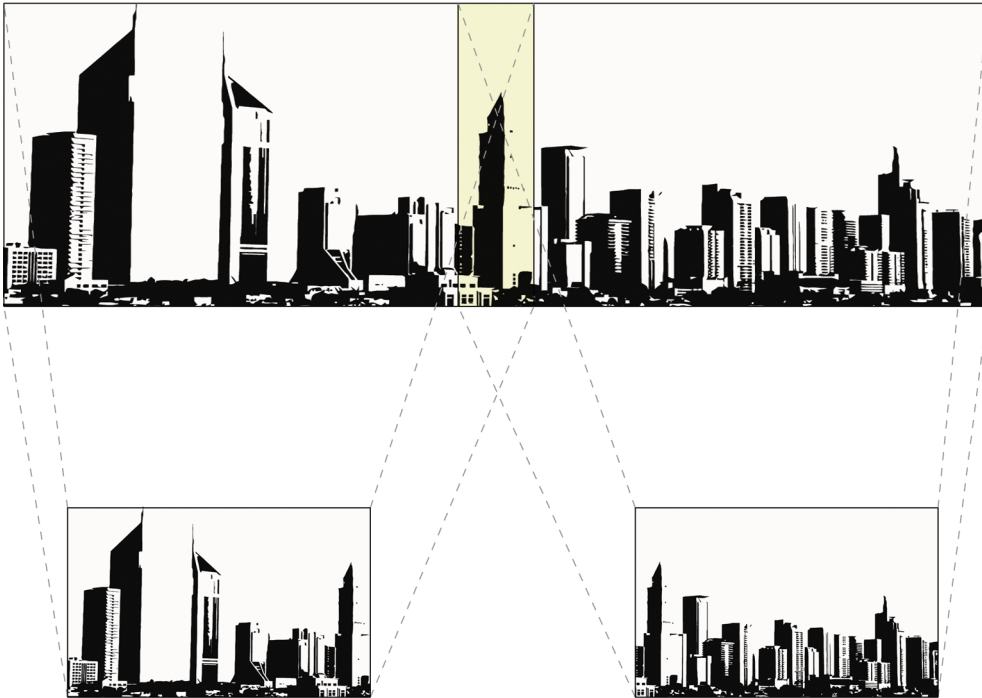


Figure 1.1: Panorama obtained from 2 images. The stitching is made by overlapping the images according to features common to both frames.

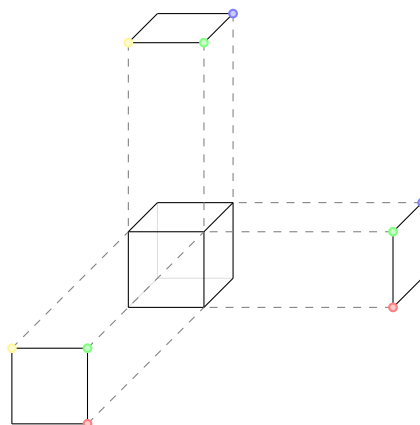


Figure 1.2: Partial reconstruction of a cube from 3 images. The process can be seen as a generalization of image stitching (see Fig. 1.1) into 3D space, as the correspondence information embeds the information on the camera positions and the 3D coordinates of the object. *N.B.*: without knowing the distance of the camera to the object, the real scale of the object is not known.

(such that the camera's observed brightness is the same regardless of its point of view) and rigid, *i.e.*, a body with no deformable surfaces; we also assume an orthographic projection model is valid, which is the case, for instance, of a picture taken from a considerable distance (such that the variation of the object depths is very small when compared to the distance between the camera and the object) with a large focal length. While these assumptions may seem restrictive, they cover a significant amount of real life cases and we tolerate small deviations from the Lambertian and orthogonality conditions. This method is therefore of interest to a number of application domains ranging from architecture — where it may facilitate the acquisition of existing buildings into 3D models for *e.g.*, planning future interventions — to industrial applications — such as the rapid reproduction of a prototype design — and 3D mapping.

1.2 State of the art

In this thesis, we look at the correspondence information from a holistic point of view, as it is not our ultimate goal but rather a mere part of a system that also comprises motion and shape. Our work, therefore, lies at the intersection of two fertile research fields in computer vision — the correspondence problem and the reconstruction of 3D scene geometry and camera motion — obviating the need for a detailed description of both frameworks.

A plethora of methods are available for reconstruction, and can be categorized by the key assumptions they make [11]: **1) Known Camera** methods [12], where we wish to infer the shape of an object based on calibrated images from known camera viewpoints; **2) Known Shape** methods [13], which aim to determine camera viewpoints given the 3D model and one or more images of it; **3) Known Correspondence** methods [2, 14, 15], which solve, as mentioned in Sec. 1.1, for camera motion and 3D object shape simultaneously, assuming as known a set of image point trajectories over various images.

Methods that obtain point trajectories in a set of images [16], on the other hand, range from dense to sparse and feature-based. Within this last category, the majority of the existing methods [5, 17, 18, 19, 20] tackle the problem by searching for candidates along a direction specified by the Optical Flow (OF), a measure of apparent movement of objects (see Sec. 2.1). However, due to its non-linearity, these methods use, for simplicity purposes, a first (sometimes second) order approximation of OF rather than the quantity itself; since this approximation is built around the origin point, methods based on OF are thus plagued by the *aperture problem*, the inadequacy in handling large motions. This poses as somewhat of a paradox for the purpose of SfM algorithms insofar as while small motions allow for the reliable determination of feature positions between frames, known correspondence methods require motion to be considerable between frames in order for the estimation to be robust, due to its estimation being based on a Singular Value Decomposition (SVD). Moreover, these methods, with the exception of [5, 20], disregard the information available in the object's rigidity or on the camera model, which reduce the number of possible matches as each point is not able to move independently from the others.

Recently, methods [21, 22] were discovered that extract local descriptors in images which not only possess interesting properties — such as invariance to scale and rotation — but ease the correspondence of these features in a pair of images, formulating it as a nearest neighbor problem. The success of these techniques within the frameworks in which they are currently used — *e.g.*, object recognition — motivates its use in a reconstruction setting.

To the best of the author's knowledge, little work has been done on solving for correspondences simultaneously with motion and shape: Mota *et al.* [23] propose optimal algorithms for solving the permutation case, where one has N points in every image and only their correct assignment is not known, assuming that the camera orientation is either known or easy to infer from camera calibration techniques; Oliveira *et al.* [24], under the same assumption, formulate a more general case where a multitude of candidates (outliers) are available for each of the points in the first image by reducing it to a linear optimization problem. Dellaert *et*

al. [11] bypass the correspondence problem entirely by introducing *virtual measurements* and then formulating SfM as a maximum likelihood problem which is solved using the EM Algorithm, due to the intractability of the likelihood function. Also worthy of note is [25, 26], who formulate structure from motion as a Bayesian inference problem, which results in iterative methods that may converge to local minima depending on the initialization.

1.3 Proposed approach

The goal in this thesis is to solve for correspondences, camera motion and object depth altogether, using only a pair of images with no additional information. The approach presented takes as input sets of point coordinates, specifically a set of points in the first image, each with a bag of correspondence candidates in the second image presenting the same intensity value. With this data, we estimate motion between both cameras and select a number of most likely candidates for each point in the first image according to a threshold defined by the user (which, in particular, can be the case of 1-to-1 correspondence).

To fulfill this goal, our work proposes a unified approach, one that merges information from brightness cues available in the images with a model that constrains point trajectories due to the rigidity of the object.

For rigid objects, the influential Factorization approach [2] asserts that the matrix obtained by stacking point trajectories along multiple frames can also be obtained as a product of a matrix comprising cameras responsible for each frame by a matrix enclosing the depicted object's 3D coordinates. The rank deficiency of this matrix allows us to obtain an estimate of the camera and object matrices as the result of its SVD decomposition, a solution which provides the optimal approximation of a given rank matrix in the least squares sense.

We build up on this result, in the sense that we exploit this formulation (albeit not making use of its rank deficiency) to further constrain and select candidates found using feature extraction algorithms such as the ones described in Sec. 1.2. Using constraints on the structure of each frame's camera matrix, we obtain a subspace in which motion is independent from the object depths. Hence, our problem is split in two phases: first, the estimation of correspondence and motion parameters within this subspace; then, the estimation of object depths.

Since this subspace is entirely specified by two parameters, we travel through all subspace possibilities by performing a grid search on the parameters mentioned. In each case, we look for the explanation that best concurs with the available candidates according to a specific cost function. By forcing a given physical motion model in the candidates, we are able to check which points are coherent with each other and dismiss other candidates in a natural way, therefore bypassing the combinatorial explosion inherent to this problem.

Note that this method, unlike many others, is able to cope with large motions and can be used in a combined effort with any feature extraction technique. Specifically, it enables us to merge cues from brightness and color without being prone to the aperture problem.

1.4 Summary of original contributions

In this section, we emphasize the contributions introduced by this work:

New formulation of the correspondence problem for rigid objects We reduce a problem that is plagued by a combinatorial explosion to an equivalent optimization problem by relying on the rank deficient formulation of SfM problems to constrain the orbits for each point. This formulation couples the estimation of correspondences, motion and depth within a single optimization problem.

CERD — Correspondence Estimation using Rigidity and local Descriptors We describe the set of possible motions as a Stiefel manifold, whose properties allow us to solve the optimization problem by minimizing a cost function over a search grid in two parameters. The resulting algorithm — CERD — has polynomial complexity and when the camera parameters are known, it degenerates in an algorithm that is able to find the optimal solution, with a $\mathcal{O}(N^2 N_i^2)$ complexity, where N_i is the number of correspondence candidates in the second image for each of the N points in the first image.

The use of color cues for finding correspondence candidates We propose a simple technique to endow tracking systems to use color as a discriminative feature. This information allows an extra degree of distinction between features, which results in a reduction of the possible correspondence feature set.

1.5 Thesis outline

This thesis is organized as follows. Chapter 2 formulates our goal within an optimization framework, after presenting preliminary notions necessary to understand the problem. Chapter 3 describes the steps taken in order to derive the solution to the problem described in Chapter 2, presents the algorithms derived and discusses some technical aspects concerning the implementation on a computer system using `matlab`. Chapter 4 describes results of the experiments performed to assess the performance of the algorithms devised. Finally, Chapter 5 presents closing comments and provides possible future work directions.

Chapter 2

Problem formulation

We start by considering an image as an application

$$I : \mathbb{R}^2 \mapsto \mathbb{R}^3$$

that maps pixel coordinates (within a bound given by the image length and width) into a certain intensity value in three color components: red, green and blue. We distinguish the frame f from an image sequence of the same object by using the notation I_f , where f corresponds to the frame index in the sequence. We also refer to the i -th feature point present in the frame f by the coordinate vector \mathbf{u}_i and to its intensity on frame f by $I_f(\mathbf{u}_i)$.

2.1 Motion

Provided points have the same intensity value in different frames, *i.e.*, the object is Lambertian, the motion of the i -th pixel between two frames I_1 and I_2 can be defined as a space shift of the image function along both of its axis as

$$I_1(x_i, y_i) = I_2(x_i + \Delta x_i, y_i + \Delta y_i),$$

in which coefficients Δx and Δy respectively account for movement along the x and y axis, as represented in Fig. 2.1.

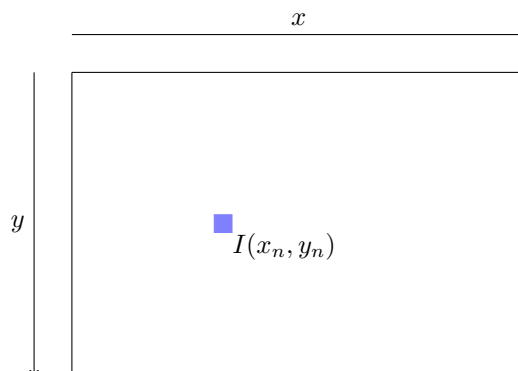


Figure 2.1: Orientation of the x and y axis in an image.

2.2 Cameras

A camera image can be viewed as a two-dimensional (2D) representation of a 3D world. Thus, we model cameras as projection operators that transform points in 3D space onto an image. While various models exist, we consider in this thesis that images are obtained according to an *orthographic* camera model. This particular model is valid when the variation of the object depths is very small when compared to the distance between the camera and the object. This happens, for instance, if we consider the camera's focal length at infinity [9]. Mathematically, it can be written as

$$\mathbf{u} = \mathbf{H} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (2.1)$$

where $\mathbf{u} \in \mathbb{R}^2$ corresponds to the point coordinates in the *image coordinate system*, (x, y, z) corresponds to the coordinates of the same point in the *object coordinate system* and $\mathbf{H} \in \mathbb{R}^{2 \times 4}$ is the camera matrix that maps points from the latter into the former.

In this model, motions are obtained by the composition of a matrix $\mathbf{R} \in \mathbb{R}^{2 \times 3}$ and a translation vector \mathbf{t} about the object coordinate system, comprising a total of 5 degrees of freedom, as

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}. \quad (2.2)$$

With \mathbf{R} being an Euclidean coordinate change, it follows that its rows are orthogonal and have unit norm. Therefore, all possible matrices \mathbf{R} constitute a Stiefel Manifold $O(2, 3)$ [27], the set of $\mathbb{R}^{2 \times 3}$ matrices whose rows are orthonormal:

$$O(2, 3) = \{\mathbf{R} \in \mathbb{R}^{2 \times 3} : \mathbf{R}\mathbf{R}^\top = \mathbf{I}_2\}. \quad (2.3)$$

Due to the fact that both vectors have unit norm and are linearly independent, we assert every matrix in this manifold as having rank 2. Moreover, since the singular values are the square roots of the eigenvalues of $\mathbf{R}\mathbf{R}^\top$, each of the singular values of \mathbf{R} is equal to 1 [28].

2.3 Structure from Motion

In SfM problems, the intent is to estimate the shape of an unknown 3D object from 2D images depicting different perspectives.

Using the camera definition on (2.1), we write the position of N feature points along F image frames as the product of a matrix $\mathbf{M} \in \mathbb{R}^{2F \times 4}$ (obtained by stacking all camera matrices \mathbf{H} responsible for each frame) by a matrix $\mathbf{S}^\top \in \mathbb{R}^{4 \times N}$, containing the 3D coordinates of the N points tracked in the object coordinate system, as

$$\mathbf{W} = \mathbf{M}\mathbf{S}^\top. \quad (2.4)$$

Equation (2.4) strongly constrains the rank of \mathbf{W} : $\text{rank } \mathbf{W} \leq 4$. The factorization method [2] exploits this fact to obtain the camera motions \mathbf{M} and shape \mathbf{S} from an assumed known matrix \mathbf{W} by obtaining its SVD and dismissing all but the first four singular values — the best rank 4 approximation of the matrix in the least squares sense — and subsequent normalization of the obtained matrices by imposing the the structure given by (2.2) on each of camera matrices in \mathbf{M} .

We consider the case of two sets \mathbf{W}_1 and \mathbf{W}_2 , containing the coordinates of N points on a pair of images ($F = 2$). Assuming the first camera coordinate system is aligned with the object coordinate system and

perpendicular to the z axis [14], we can write \mathbf{M} as

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 \\ \mathbf{R} & \mathbf{t} \end{bmatrix},$$

where $\mathbf{R} \in \mathbb{R}^{2 \times 3}$ models the orientation of camera 2 relative to the object coordinate system and $\mathbf{t} \in \mathbb{R}^2$ describes the translation between the camera's and object's points of origin.

By splitting \mathbf{R} in the matrix $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ and vector $\mathbf{b} \in \mathbb{R}^2$ as

$$\mathbf{R} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}, \quad (2.5)$$

the transformation between images \mathbf{W}_1 and \mathbf{W}_2 can be decomposed as

$$\mathbf{W}_2 = \mathbf{A}\mathbf{W}_1 + \mathbf{b}\mathbf{z}^\top + \mathbf{t}\mathbf{1}^\top, \quad (2.6)$$

allowing the interpretation of a transformation comprised by three steps as in Fig. 2.2: the one applied by the operator \mathbf{A} (that performs a 2D rotation followed by an asymmetrical scaling of the axis and another rotation) and the addition of components along the axis \mathbf{b} (according to the respective point depths \mathbf{z}) and \mathbf{t} (common to all points).

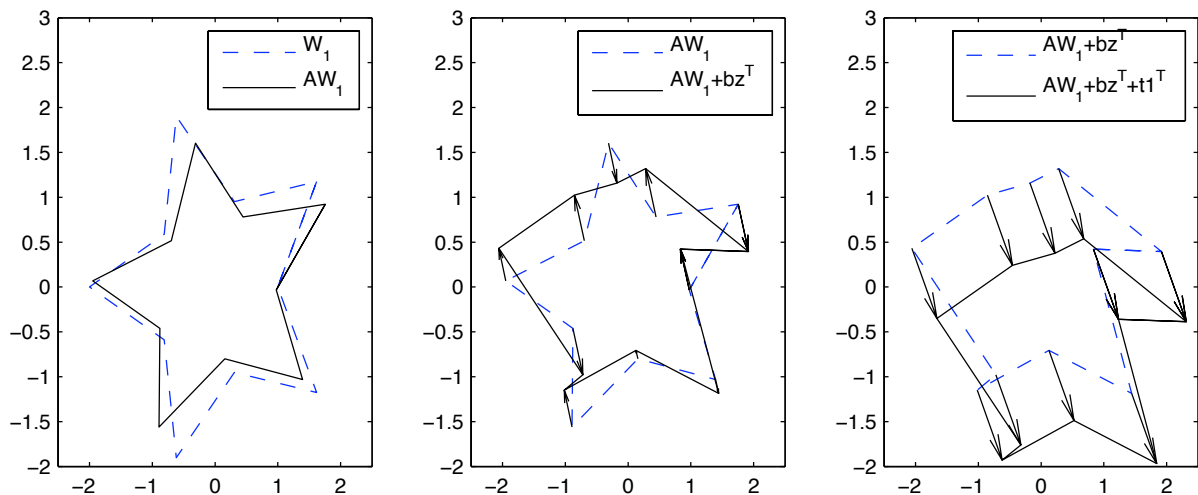


Figure 2.2: Image motion decomposed in three sub-transformations: the one applied by the operator \mathbf{A} (left), translation along the vector \mathbf{b} according to each point depth z_i (center) and translation \mathbf{t} applied commonly to all points (right).

According to this formulation, in the SfM with correspondences case, one wishes to find, given a set of point trajectories along two frames, the unknowns \mathbf{A} , \mathbf{b} and \mathbf{t} that define the camera motion, as well as the unknown object depths \mathbf{z} , such that (2.6) holds.

2.4 SfM without correspondences as an optimization problem

However, equations (2.4) and (2.6) are only valid for the case where correspondences are known. The problem we formulate, on the other hand, makes no such assumption, and is thus plagued with a large number of outliers in the form of candidate matches. This fact alone is what constitutes the main challenge of this problem, due to the **combinatoric explosion** arising from the existence of several possible matches.

We are, however, able to surpass this curse of dimensionality and **achieve an accurate solution with polynomial complexity**, by making use of the fact that the intensity is independent from the object's point of view (we assume a Lambertian object) and, as such, point correspondences between images must display

the same intensity value. For the aforementioned case of two image frames \mathbf{W}_1 and \mathbf{W}_2 , this yields

$$I_2(\mathbf{A}\mathbf{W}_1 + \mathbf{b}\mathbf{z}^\top + \mathbf{t}\mathbf{1}^\top) = I_1(\mathbf{W}_1),$$

a formulation that restricts the possible destinations of each of the $i \in 1, \dots, N$ points present in the image can take by unifying rigidity and intensity constraints. We now have everything set to formulate our goal of joint correspondence, motion and shape estimation in more detail.

Problem Statement Given $I_2, I_1, \mathbf{W}_1 = [\mathbf{u}_1 \dots \mathbf{u}_N]$, find $\mathbf{A}, \mathbf{b}, \mathbf{t}, \mathbf{z}$ such that

$$I_2(\mathbf{A}\mathbf{u}_i + \mathbf{b}z_i + \mathbf{t}) = I_1(\mathbf{u}_i), \quad \forall i \in 1, \dots, N. \quad (2.7)$$

Each constraint specified in (2.7) can be interpreted as a restriction on the point trajectory to candidates that match the point intensity specified in image I_1 as

$$I_2(\mathbf{A}\mathbf{u}_i + \mathbf{b}z_i + \mathbf{t}) \in \mathcal{S}_i, \quad \forall i \in 1, \dots, N,$$

where \mathcal{S}_i represents the level curve of I_2 with value given by the point \mathbf{u}_i 's intensity in image I_1 . In practice, we have small variations due to illumination and shading, so we consider all points whose intensities are in the small polyhedron centered in $I_1(\mathbf{u}_i)$, as

$$\mathcal{S}_i = \{\mathbf{v} \in \mathbb{R}^2 : \|I_1(\mathbf{u}_i) - I_2(\mathbf{v})\|_1 \leq \xi\}. \quad (2.8)$$

This formulation suggests the optimization problem of finding correspondences such that the distance d of each point $\mathbf{A}\mathbf{u}_i + \mathbf{b}z_i + \mathbf{t}$ to its respective intensity level set \mathcal{S}_i is minimized, or

$$\begin{aligned} & \text{minimize} && \Phi(\mathbf{A}, \mathbf{b}) \\ & \text{subject to} && \mathbf{A}\mathbf{A}^\top + \mathbf{b}\mathbf{b}^\top = \mathbf{I}_2, \end{aligned} \quad (2.9)$$

where

$$\Phi(\mathbf{A}, \mathbf{b}) = \inf_{\mathbf{t} \in \mathbb{R}^2, \mathbf{z} \in \mathbb{R}^N} \sum_{i=1}^N d(\mathbf{A}\mathbf{u}_i + \mathbf{b}z_i + \mathbf{t}, \mathcal{S}_i), \quad (2.10)$$

the set distance $d(\cdot, \cdot)$ is defined as

$$d(\mathbf{x}_i, \mathcal{S}_i) = \min_{\mathbf{v} \in \mathcal{S}_i} \|\mathbf{x}_i - \mathbf{v}\|_2$$

and the restriction $\mathbf{A}\mathbf{A}^\top + \mathbf{b}\mathbf{b}^\top = \mathbf{I}_2$ follows from (2.3) and (2.5).

In words, we wish to find the transformation of type (2.6) that best repositions points \mathbf{u}_i obtained using any given feature extraction method in the first image with their respective level sets \mathcal{S}_i in the second image. By doing this, we are not only able to assess the goodness of a camera model given image intensity data, but obtain from the level sets \mathcal{S}_i the correspondence candidates that best fit that model.

Chapter 3

CERD — Correspondence Estimation using Rigidity and local Descriptors

So far we have formulated the problem of finding correspondences, motion and depth within a unified optimization problem. In this chapter, we deepen our formulation to a level where we can propose a solution (Sec. 3.1 to Sec. 3.2) and describe the resulting implementation (Sec. 3.3).

An important remark should be made at this point: for each given \mathbf{A}, \mathbf{b} , the value of the cost function in (2.10) is itself an optimization problem. Thus, the question naturally following is as to how this value should then be calculated, given these matrices.

Let us start by decomposing \mathbf{t} in two components

$$\mathbf{t} = \alpha \mathbf{b} + \gamma \mathbf{b}_\perp, \quad \|\mathbf{b}_\perp\|_2 = 1,$$

where \mathbf{b}_\perp stands for the orthogonal complement of \mathbf{b} . In doing this, we implicitly assume that $\mathbf{b} \neq \mathbf{0}$, a condition equivalent to not having a pure planar motion, as the influence of object depths in motion is confined within that subspace. Provided this condition is met, we are able to rewrite (2.10) as

$$\Phi(\mathbf{A}, \mathbf{b}) = \inf_{\alpha \in \mathbb{R}, \gamma \in \mathbb{R}, \mathbf{z} \in \mathbb{R}^N} \sum_{i=1}^N d(\mathbf{A}\mathbf{u}_i + \mathbf{b}(z_i + \alpha) + \gamma \mathbf{b}_\perp, \mathcal{S}_i). \quad (3.1)$$

There is an intrinsic ambiguity to finding the value of the factors $z_i + \alpha$ from its sum alone. Thus, we collapse these two variables in z_i (*i.e.*, we assume $\alpha = 0$ for simplicity), rendering (3.1) as the optimization problem

$$\Phi(\mathbf{A}, \mathbf{b}) = \inf_{\gamma \in \mathbb{R}, \mathbf{z} \in \mathbb{R}^N} \sum_{i=1}^N d(\mathbf{A}\mathbf{u}_i + \mathbf{b}z_i + \gamma \mathbf{b}_\perp, \mathcal{S}_i),$$

which we further simplify by using the fact that the infimum over two variables can be decoupled as

$$\Phi(\mathbf{A}, \mathbf{b}) = \inf_{\gamma \in \mathbb{R}} \left[\inf_{\mathbf{z} \in \mathbb{R}^N} \sum_{i=1}^N d(\mathbf{A}\mathbf{u}_i + \mathbf{b}z_i + \gamma \mathbf{b}_\perp, \mathcal{S}_i) \right]. \quad (3.2)$$

When $\mathbf{A}, \mathbf{b}, \gamma$ are known, we have a clear geometric interpretation of the problem formulated in (3.2). As can be seen in Fig. 3.1, the effect of not knowing the object depths \mathbf{z}_i is that every point in I_2 belonging to the line with direction given by the vector \mathbf{b} that intersects the point $\mathbf{A}\mathbf{u}_i + \mathbf{t}$ is considered valid by the motion model. All points in this line intersecting the level curve \mathcal{S}_i are, therefore, considered correct correspondences. Having said this, we are interested in minimizing the distance of the level curve \mathcal{S}_i to the line passing through

the point $\mathbf{A}\mathbf{u}_i + \gamma\mathbf{b}_\perp$ with direction \mathbf{b} . In Appendix A, we show that the cost function in (3.2) becomes

$$\Phi(\mathbf{A}, \mathbf{b}) = \inf_{\gamma \in \mathbb{R}} \sum_{i=1}^N d((\mathbf{b}_\perp)^\top \mathbf{A}\mathbf{u}_i + \gamma, (\mathbf{b}_\perp)^\top \mathcal{S}_i).$$

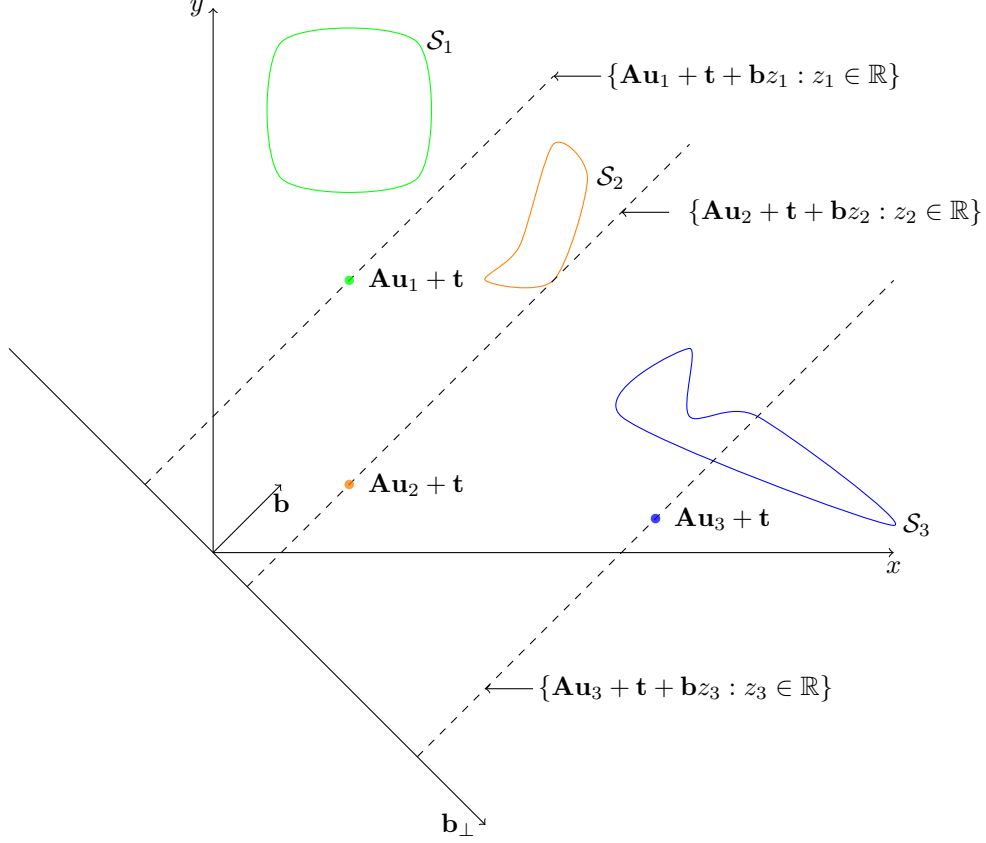


Figure 3.1: Ambiguity of correspondence for known \mathbf{H} and unknown object depths \mathbf{z} . The dashed lines represent the variation of object depth \mathbf{z} , making each intersection of the lines passing along $\mathbf{A}\mathbf{u}_i$ with direction given by the vector \mathbf{b} and level curves \mathcal{S}_i a valid correspondence.

This new measure corresponds to projecting every point in \mathcal{S}_i and $\mathbf{A}\mathbf{u}_i$ onto the subspace defined by \mathbf{b}_\perp , the orthogonal complement of \mathbf{b} . In this subspace, the entire line passing through the point $\mathbf{A}\mathbf{u}_i$ with direction \mathbf{b} collapses in a single point, allowing a representation that is independent of the object depth z_i .

In practice, since the image has a finite resolution, and therefore a finite number of points, the level sets \mathcal{S}_i degenerate into discrete sets

$$\mathcal{S}_i = \{\mathbf{v}_{i,1}, \dots, \mathbf{v}_{i,N_i}\},$$

where N_i represents the number of points in the second image in the level set \mathcal{S}_i . Hence, the set distance operator $d(\cdot, \cdot)$ is naturally defined as the minimum distance between the point $(\mathbf{b}_\perp)^\top \mathbf{A}\mathbf{u}_i + \gamma$ and every element in the set $(\mathbf{b}_\perp)^\top \mathcal{S}_i$

$$d((\mathbf{b}_\perp)^\top \mathbf{A}\mathbf{u}_i + \gamma, (\mathbf{b}_\perp)^\top \mathcal{S}_i) = \min_{k \in \{1, \dots, N_i\}} |(\mathbf{b}_\perp)^\top \mathbf{A}\mathbf{u}_i + \gamma - (\mathbf{b}_\perp)^\top \mathbf{v}_{i,k}|. \quad (3.3)$$

Since \mathbf{R} is in the $O(2,3)$ manifold, we are able to fully parameterize its components \mathbf{A} and \mathbf{b} (see Appendix B for a detailed derivation) as

$$\mathbf{A} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} \cos \phi & \pm \sin \phi \\ \mp \sin \phi & \cos \phi \end{bmatrix}, \quad (3.4)$$

$$\mathbf{b} = \pm \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} \sqrt{1-r^2}. \quad (3.5)$$

This result allows us to span the entire manifold $O(2, 3)$ using only three parameters. It is worth noticing that the family of motion matrices with $r = 1$ corresponds to motion patterns that are entirely independent of the unknown depths \mathbf{z} of the object. This, as mentioned before, corresponds to motion matrices representing planar motions, since the object depths \mathbf{z} have no influence on point trajectories.

For the purposes of this discussion, we shall discard the signal ambiguity present in \mathbf{V}^\top , forcing it to the form

$$\mathbf{V}^\top = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix}$$

and restrict ϕ to the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$, the minimum interval necessary to span all possible matrices without repetitions.

Taking into account (3.5), we can write \mathbf{b}_\perp as

$$\mathbf{b}_\perp = \pm \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}. \quad (3.6)$$

Using the definition of \mathbf{A} in (3.4) and the expression for \mathbf{b}_\perp in (3.6), the product $(\mathbf{b}_\perp)^\top \mathbf{A}$ in (3.3) is simplified, yielding

$$d((\mathbf{b}_\perp)^\top \mathbf{A} \mathbf{u}_i + \gamma, (\mathbf{b}_\perp)^\top \mathcal{S}_i) = \min_{k \in \{1, \dots, N_i\}} \left| \mathbf{u}_i^\top \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} + \gamma - \mathbf{v}_{i,k}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \right| \quad (3.7)$$

where we have factored out the sign indetermination in \mathbf{b}_\perp .

Note that despite \mathbf{R} being defined univocally by the parameters (r, θ, ϕ) , (3.7) only depends on the last two and translation component γ . Besides that, the orthonormality constraint in (2.9) is embedded within this equation. This allows us to reformulate the optimization problem in (2.9) as

$$\text{minimize } \Phi(\mathbf{A}, \mathbf{b}) = \inf_{\gamma \in \mathbb{R}} \sum_{i=1}^N d((\mathbf{b}_\perp)^\top \mathbf{A} \mathbf{u}_i + \gamma, (\mathbf{b}_\perp)^\top \mathcal{S}_i) \quad (3.8)$$

where the set distance $d(\cdot, \cdot)$ is defined as in (3.7).

3.1 Solving for translation

Since now we have defined how to calculate the value of $\Phi(\mathbf{A}, \mathbf{b})$ given \mathbf{A}, \mathbf{b} , let us assume that the camera parameters are available, such as in the case where camera calibration techniques are easy to apply [23]. Since the parameters θ and ϕ are known, the problem formulated in (3.8) is reduced to

$$\inf_{\gamma \in \mathbb{R}} C(\gamma) \quad (3.9)$$

where the cost function $C(\gamma)$ is

$$C(\gamma) = \sum_{i=1}^N \min_{k \in \{1, \dots, N_i\}} \left| \mathbf{u}_i^\top \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} + \gamma - \mathbf{v}_{i,k}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \right| \quad (3.10)$$

According to this formulation, we wish to obtain from a set of $i \in \{1, \dots, N\}$ points in the first image and respective sets of candidates \mathcal{S}_i (each with N_i prospective matches) in the second image, a candidate $\mathbf{v}_{i,k} \in \mathcal{S}_i$ for each point that corroborates best the motion given by the camera parameters \mathbf{A}, \mathbf{b} and the

translation component γ .

The solution we propose relies on the key observation that, in the absence of noise, γ is common to every correct correspondence, *i.e.*, it is given by the difference between the point in the first image and its correct match in the second image, when projected onto the subspace defined by \mathbf{b}_\perp . According to this rationale, we restrict γ to forced correspondences between points in the first image and every possible candidates in the second, as

$$\gamma_{m,n} = \mathbf{v}_{m,n}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} - \mathbf{u}_m^\top \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix}, \quad (3.11)$$

where \mathbf{u}_m stands for the coordinates of the m -th point ($m \in \{1, \dots, N\}$) in image \mathbf{W}_1 and $\mathbf{v}_{m,n}$ for the position of the n -th correspondence candidate for point n in the level set \mathcal{S}_m .

This results in the confinement of (3.9) to a search in the finite set of possible correspondences

$$\min_{\substack{m \in \{1, \dots, N\} \\ n \in \{1, \dots, N_m\}}} C(\gamma_{m,n}) \quad (3.12)$$

Note that this problem is equivalent to (3.9) in the absence of noise, with the set of forced matches including the optimal solution.

For the problem presented in (3.12), we offer the solution presented in Algorithm 1: For each of the possible forced correspondences $\gamma_{i,j}$, we select from the level sets \mathcal{S}_i the best match for each of remaining $N - 1$ points in image 1 as the candidate with the least distance (*i.e.*, the nearest neighbor) according to the set distance $d(\cdot, \cdot)$. We could have restricted the search to one point in the first image and its respective level set, but we chose to make use of the redundancy present in the data in order to account for scenarios that might occlude some of the points in the image.

Algorithm 1 Solution for γ .

Initialize Global Cost $C = \infty$

for all ($m \in \{1, \dots, N\}$) **do**

for all ($n \in \{1, \dots, N_m\}$) **do**

 Take $\gamma_{m,n}$ as in (3.11)

for all ($i \in \{1, \dots, N\} \setminus m$) **do**

 Select u_i 's nearest neighbor from \mathcal{S}_i according to distance (3.7)

end for

if $C(\gamma_{m,n})$ calculated as in (3.10) better than current estimate C **then**

 Update Global Cost $C = C(\gamma_{m,n})$

end if

end for

end for

In doing this, we have reduced a problem with a number of possible assignments that is combinatoric $\binom{N \times N_i}{N_i}$ to polynomial, with an algorithm that has a number of operations that approximates $\mathcal{O}(N^2 N_i^2)$. To illustrate the clear distinction between the two orders of magnitude, consider the case of $N = 10$ points in the first image, each with $N_i = 15$ candidates: the polynomial approach finds the solution with a mere 22500 operations, whereas a combinatoric approach explodes to approximately 1.1696×10^{15} (!) operations. Besides, this method is capable of dealing with outliers since for a given forced correspondence $\gamma_{m,n}$, the set distance operator $d(\cdot, \cdot)$ selects candidates by how much they match a given physical camera model given by \mathbf{A}, \mathbf{b} .

In the case where the camera parameters are known and the candidates available in the second image differ

from \mathbf{W}_2 only by a permutation matrix $\mathbf{\Pi}$, such as in [23, 24], Algorithm 1 can be modified to find the optimal solution with a complexity of $\mathcal{O}(N \log N)$, where N is the number of points considered in each image.

Theorem 1 (Optimality of Algorithm 1). *Given a set of images \mathbf{W}_1 , $\mathcal{S}_i = \mathbf{W}_2 \mathbf{\Pi}, \forall i = 1, \dots, N$ and known camera parameters (θ, ϕ) , if there exists a pair $(\gamma, \mathbf{\Pi})$ such that (2.6) holds, Algorithm 1 is able to determine it. Moreover, this solution is unique if the projections of \mathbf{W}_1 and \mathbf{W}_2 in the \mathbf{b}_\perp axis (resp., constellations 1 and 2) are injective.*

Proof. Based on (3.11), we can interpret the translation component γ to be the difference between the position of the same point in both constellations 1 and 2. For the correct values of (θ, ϕ) , which in this case are assumed known, finding a value of γ such that (2.6) holds has the geometric interpretation of finding a common translation that superimposes all points from constellation 2 with their homologous in constellation 1. In order for both constellations to overlap, the only possible correspondence is the same found by the natural order of the points along the axis as any permutation would imply either of two cases: **1**) the solution requires a γ value that is not common to equation of the system, a contradiction of (2.6), which states that the translation component is common to all points in the constellation or **2**) a common translation other than the one obtained with the natural order is used, which necessarily renders at least one equation of the system impossible, as at least one point — the upper or lower bound of the set — of the constellation 2 will not overlap with any of the points in constellation 1. \square

It should be noted that if the transformations that obtain constellations 1 and 2 from \mathbf{W}_1 and \mathbf{W}_2 are not injective, two points will have the same projection on the \mathbf{b}_\perp axis; in this case, the set order makes no distinction between the two, therefore making it impossible to distinguish between the possible correspondence solutions.

3.2 Solving for rotation

Without any additional information other than the pair of images *per se*, it may be hard to obtain a constellation differing from the correct one by a permutation matrix or prior knowledge the camera parameters. In this case, we only dispose of a finite set of possible matches for each point in \mathbf{W}_1 , obtained from features embedded in the images such as, e.g., color, brightness.

We know from the beginning of this Chapter that the entire manifold $O(2, 3)$ can be spanned by varying the parameters (θ, ϕ, r) within the cube $[0, 2\pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, 1]$. The cost function introduced in (3.10) assesses the extent to which correspondences can be explained by a rigid, orthographic motion, allowing us to select the best candidates for a motion given by a specific pair (θ, ϕ) . Hence, Algorithm 1 can be subsumed under a broader algorithm that performs a discretized search on the parameter grid (θ, ϕ) and calculates for each tuple the candidates that best agree with the given motion. This method, which we call Correspondence Estimation using Rigidity and local Descriptors (CERD), is summarized in Algorithm 2.

Algorithm 2 CERD — Correspondence Estimation using Rigidity and local Descriptors.

```
Initialize Global Cost  $C = \infty$ 
for all  $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$  do
  for all  $0 \leq \theta < 2\pi$  do
    Calculate  $\Phi(\mathbf{A}, \mathbf{b})$  for current  $(\theta, \phi)$  using Algorithm 1
    if  $\Phi(\mathbf{A}, \mathbf{b})$  better than current estimate  $C$  then
      Update Global Cost  $C = \Phi(\mathbf{A}, \mathbf{b})$ 
    end if
  end for
end for
```

Since we perform a search throughout $O(2,3)$, this method finds the global minimum of (3.12) up to a specified grid resolution, giving as a byproduct the motion model and point correspondences that generate it.

It should be noted that this algorithm can be categorized as having a polynomial complexity of $\mathcal{O}(N_p N^2 N_c^2)$, where N_p stands for the number of points the grid is discretized in. Therefore, a clear trade-off exists between the accuracy of the motion estimation (and consequently, candidate selection) and computation time.

3.3 Implementation

In the following paragraphs, we describe what we believe to be the most relevant technical aspects of the `matlab` implementation we developed of the algorithm presented in Sec. 3.2:

Candidate Retrieval Although CERD is able to take as input sets of candidates resulting from any feature extraction algorithm, we have implemented a simple algorithm that allows candidate retrieval based on pixel color. Regarding this implementation, a few comments should be made:

- When selecting both points and match candidate, we only consider corners, as large color surfaces lead to a candidate explosion. We use the corner detector in [29, 30];
- An RGB color model is assumed. In this model, a pixel color is represented by three integers representing the red, green and blue channels that take value between 0 and 255. The effect on color pixel with the variation of these values is depicted in Fig. 3.2.
- The candidates obtained need not be a perfect color match for the selected pixel in the first image. Instead, they are only required to be within an bounded cube centered around this color as in (2.8). We do this in order to account for changes of lighting or camera white balance;

Candidate and Global Cost calculation In order to optimize the calculation of each candidate and iteration cost, we resorted to the use of a tensor of order 3 to collect each bag of candidates. This representation allows us to find the best candidates and cost using `matlab`'s built-in functions `min` and `sum`. According to [31, 32], these functions contain memory optimization techniques which provide a speedup relative to a regular `for` cycle, resulting in faster overall execution. For the same reasons, we have used the function `bsxfun` to perform operations and matrix replications simultaneously.

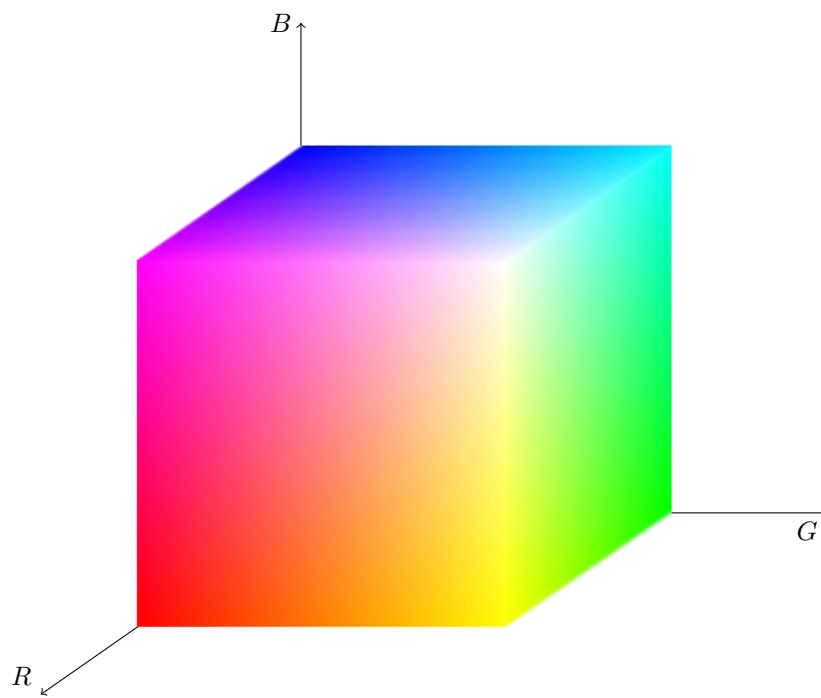


Figure 3.2: RGB color space. The axis represent channels Red, Green and Blue, with integers varying between the values 0 and 255.

Chapter 4

Experiments

In this chapter, we perform experiments in order to assess the performance of the method presented in this thesis. We have divided this chapter in sections, each describing a set of experiments exploring a specific topic: in Sec. 4.2, we evaluate the algorithms devised in Chapter 3 with synthetic data, measuring the correctness of the obtained results and the behavior of the cost function minimized by the algorithms. In Sec. 4.3, we simulate, also with synthetic data, an entirely automatic SfM scenario by making the camera parameters unknown. Sec. 4.4 extends the results obtained in the previous sections using real data. Finally, in Sec. 4.5 we couple our method with existing state of the art feature extracting algorithms, such as SIFT.

4.1 Methodology and setup

In this section, we describe the methodology used to synthesize images according to the model presented in Chapter 2.

For each test case, we obtain a pair of random images, as follows: we generate parameters t , r , θ , ϕ and the shape matrix \mathbf{S} according to a uniform distribution, the latter bound to the interval $[0, w_s]$. We then specify points in the first image (the set \mathbf{W}_1) as the first and second rows of \mathbf{S}^\top and obtain \mathbf{W}_2 according to the model in (2.6). We obtain each of the level sets \mathcal{S}_i as the union of each point in \mathbf{W}_2 with an additional N_c candidates, also generated from a uniform distribution with window size w_s . Additionally, if occlusion is to be tested, we remove from a number $0 \leq N_o \leq N$ of randomly selected level sets the original match in \mathbf{W}_2 .

A few constraints are placed on the tests made: first, we truncate the values of (θ, ϕ) to the square $[0, 2\pi \times [-\frac{\pi}{2}, \frac{\pi}{2}]]$; second, we perform a sweep of the the variable in test (grid resolution or number of occlusions) before generating a new set of points and motion. These constraints allow the comparability of the cost function and mismatches throughout the parameter variation space and ease the calculation of the error in the estimation of the parameters θ and ϕ without losing, as mentioned in Chapter 3, the generality of spanning the entire manifold.

All experiments were performed on a laptop with a 2.4 GHz Intel Core 2 Duo processor and 4 GB of RAM running Matlab R2008a.

4.2 Known θ, ϕ

In this section, we present a few illustrative examples of the behavior of Algorithm 1, namely on the optimality of its results. The experiences presented here are related to evaluating the function $\Phi(\mathbf{A}, \mathbf{b})$ for a given pair θ, ϕ , as described in Sec. 3.1.

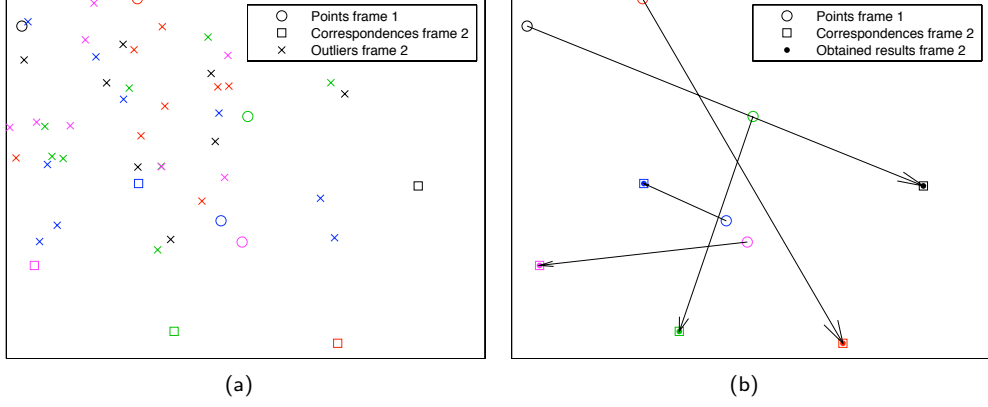


Figure 4.1: Points in a pair of synthesized images for $N = 5$ and $N_c = 10$. The colors represent points with equal intensity value: (a) input data; (b) results obtained.

Constellations on both image and on the projected axis. In this experiment, we generated a pair of images with $N = 5$ points in the first frame and N level sets in the second, each with $N_c = 10$ candidates, as described in Sec. 4.1. Fig. 4.1 shows an example of the pair of images used in this experiment and the results obtained by our method.

We feed the algorithm with points \mathbf{W}_1 and level sets \mathcal{S}_i , as well as the camera parameters θ and ϕ , leaving the correct matching between points in both images to be obtained. Fig. 4.2 shows the variation of the cost function $C(\gamma_{m,n})$ for each of the forced correspondences, with minima occurring when each of the N points is matched with its correct pair.

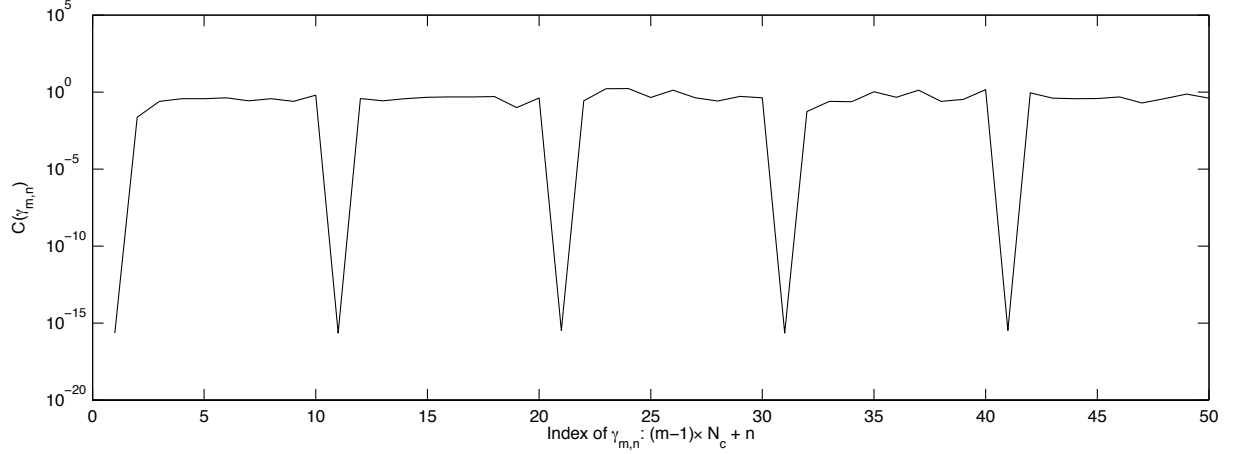


Figure 4.2: Behavior of $C(\gamma_{m,n})$ for $N = 5$ and $N_c = 10$.

It is worth noticing that, despite the considerable point distances between images 1 and 2 (Fig. 4.1), the inliers in both constellations fully overlap (Fig. 4.3) after being projected in the subspace defined by \mathbf{b}_\perp and subtracting the translation component $\gamma_{m,n}$ obtained in the minimization process. These results demonstrate the usability of this subspace not only for its dimensionality reduction but for its independence from unknown variables r and \mathbf{z} .

Robustness to outliers. In this paragraph, we assess the behavior of the algorithm with the variation of the number of points and cardinality of the level sets \mathcal{S}_i . For this purpose, we introduce the concept of average point disparity, which we define as the average of the ℓ_1 norm of the distance between ground truth (with noise) and the algorithm's output, for each of the N correspondences, or

$$\frac{1}{N \times w_s} \sum_{i=1}^N \|\hat{\mathbf{v}}_i - \mathbf{v}_i\|_1$$

where $\hat{\mathbf{v}}_i, \mathbf{v}_i$ stand respectively for each correct (ground truth) and calculated match for point \mathbf{u}_i and w_s is the normalization constant that factors in the window size bounding generated data (for simplicity sake, however we select $w_s = 1$). Note that by making this a distance measure rather than a binary (match, mismatch) decision, the average point disparity takes into account the case where the points obtained are not the original candidate but a neighboring point.

We performed 500 tests for each combination of points N and candidates N_c in the square $[1, 20] \times [1, 40]$. A typical correspondence scenario for $N = 20$ points and $N_c = 40$ match candidates is depicted in Fig. 4.4.

On every test we made, we obtained exact matches for all of the points. These results show that the knowledge of camera motion heavily constrains possible assignments and the result neither depends on the number of points N nor candidates N_c . This allows us to conjecture that the optimality conditions present in Theorem 1 may be generalized to the case where one has more candidates than the ones arising from a permutation case. **In the absence of noise, our method is therefore able to find the optimal solution while bypassing the combinatoric nature of the problem.**

Sensitivity to noise. In practice, images are subject to errors inherent not only to sensor noise but to a discretization in pixels. To illustrate the robustness to noise of our method, we now generate data with a fixed number of points of $N = 5$ and candidate matches of $N_c = 10$. We apply to the each of the point inliers in the second constellation Additive White Gaussian Noise (AWGN) with standard deviation σ as a fraction of the window size w_s . Then, we feed these points to Algorithm 1 and measure the variation of cost $C(\gamma_{m,n})$ (Fig. 4.5) and average point disparity (Fig. 4.6) with the standard deviation of noise σ .

The results present in Fig. 4.6 show that the algorithm's ability to make correct matches decreases with the presence of noise, being able to estimate correspondences within a 5% disparity bound while subject to noise levels around $\sigma = 1 \times 10^{-3}$. Translating to a real case scenario, our method is able to withstand noise levels corresponding to having a measurement error greater than 6 pixels in approximately 32 % of all points registered in a 4000×4000 pixels (a 16 Megapixel image), a value by far greater than what is obtained with today's cameras.

On the other hand, the value of $C(\gamma_{m,n})$ increases with σ , allowing us to conclude that an increase in noise levels incurs in a fading of the separation between minima and the remaining values for the cost function, making the matching process more strenuous. In fact, Fig. 4.7, which shows the variation of the cost function

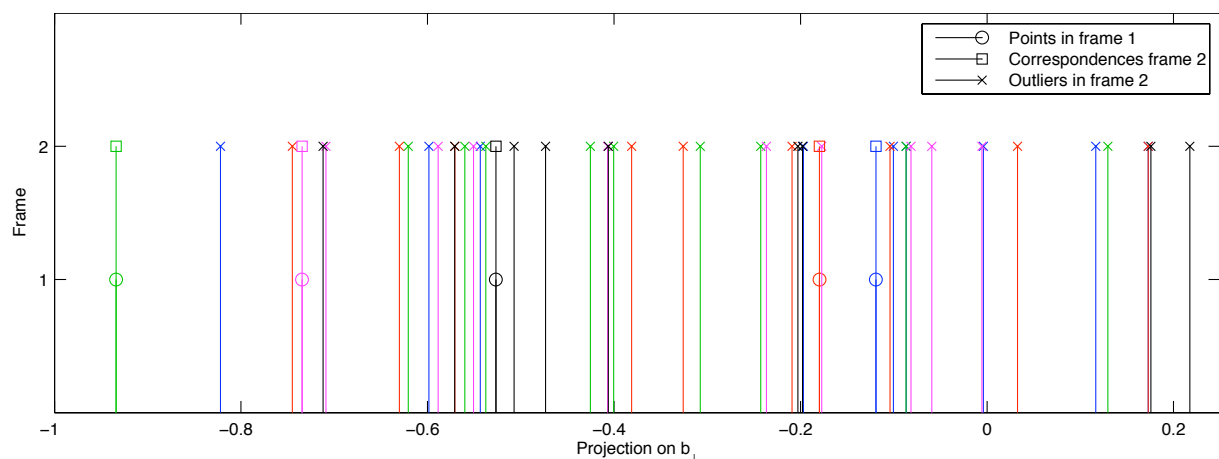


Figure 4.3: Image points after projection in \mathbf{b}_\perp and subtraction of component γ for $N = 5$ and $N_c = 10$. The colors represent points with equal intensity value.

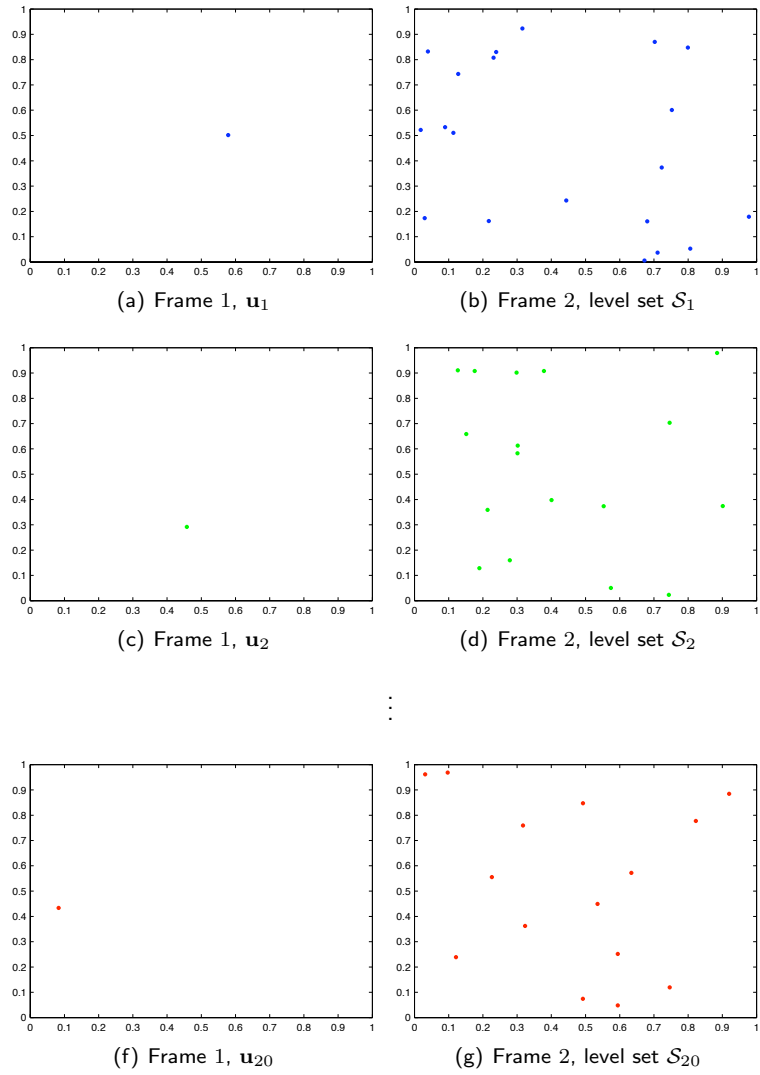


Figure 4.4: Point and candidate distribution throughout the images for a typical scenario of $N = 20$ points and $N_c = 40$ match candidates. The existence of a large number of candidate matches makes the correspondence estimation hard, even for a human.

$C(\gamma_{m,n})$ for $\sigma = 1 \times 10^{-3}$, illustrates this trend; while still showing 5 points with minimum cost, the value at these points is several orders of magnitude higher, when compared with the noiseless case presented in Fig. 4.2.

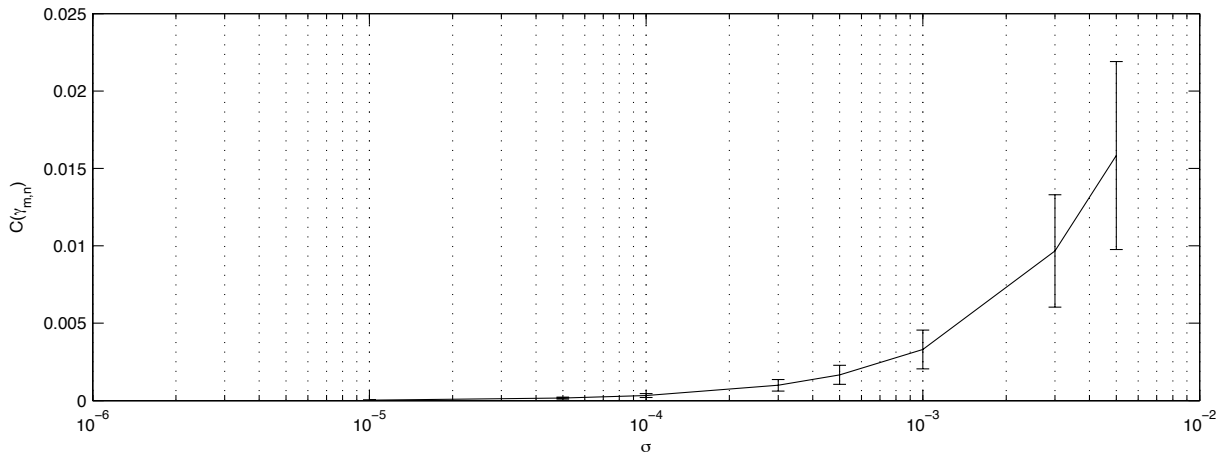


Figure 4.5: Variation of $C(\gamma_{m,n})$ with AWGN of standard deviation σ averaged over 6000 tests.

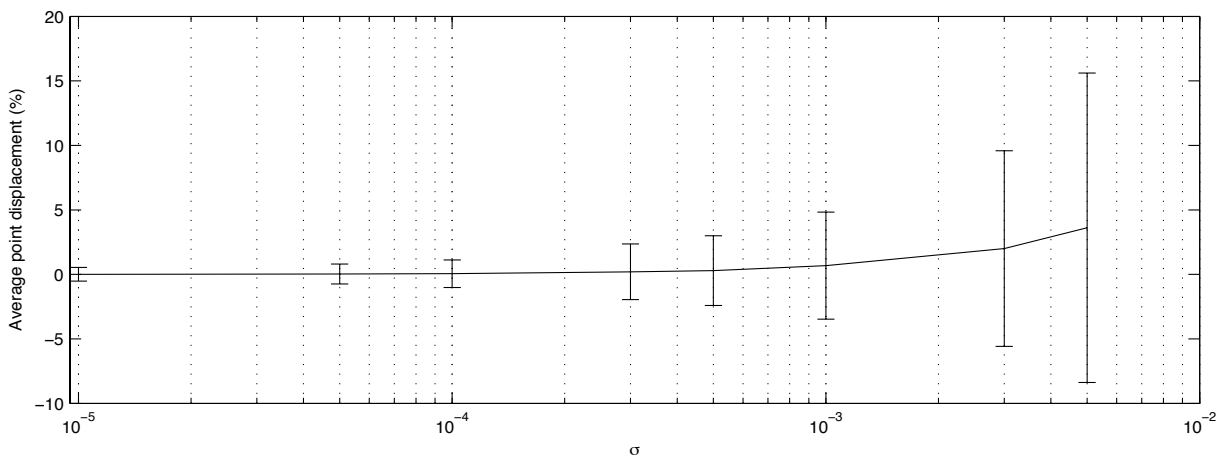


Figure 4.6: Variation of average point disparity with AWGN of standard deviation σ averaged over 6000 tests.

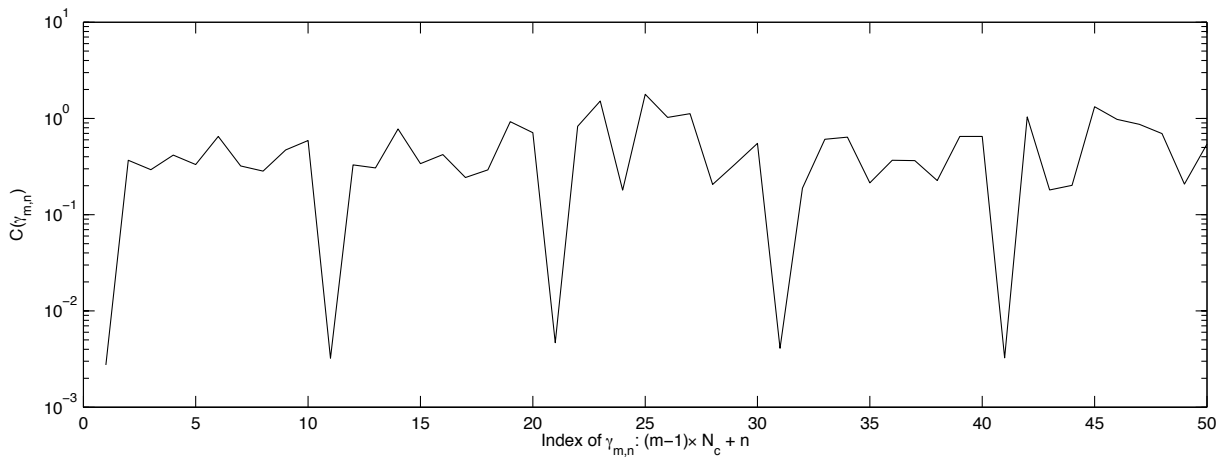


Figure 4.7: Variation of $C(\gamma_{m,n})$ for known camera parameters for $N = 5$ and $N_c = 10$ for noise with standard deviation of 1×10^{-3} .

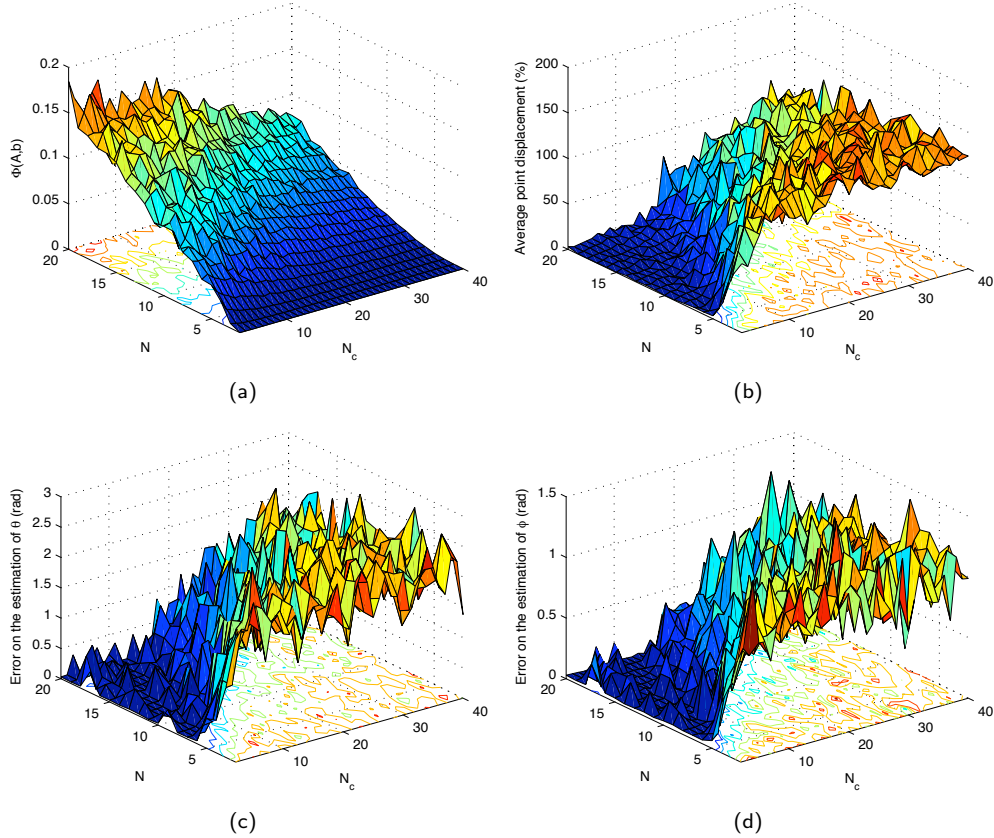


Figure 4.8: Variation of cost function $\Phi(\mathbf{A}, \mathbf{b})$, average point displacement and camera parameters θ and ϕ estimation errors with the number N of points and number N_c of candidate matches for a grid resolution of 2500 points: (a) variation of $\Phi(\mathbf{A}, \mathbf{b})$; (b) variation of average point disparity; (c) variation of estimation error for θ ; and, (d) variation of estimation error for ϕ .

4.3 Fully automatic SfM

Building up on the experiments of Sec. 4.2, we now test our algorithm as an automatic SfM recovery system. For this purpose, we only feed the algorithm the inputs \mathbf{W}_1 and level sets \mathcal{S}_i generated as before, leaving the camera parameters to be estimated.

Variation with number of points and candidates. In this experiment, we assess how the behavior of the cost function $\Phi(\mathbf{A}, \mathbf{b})$ and the average point disparity change when the algorithm is subject to a variation on the number of points N and the cardinality N_c of the level sets \mathcal{S}_i . Results are present in Fig. 4.8.

Several conclusions can be drawn from this experiment:

- The cost function $\Phi(\mathbf{A}, \mathbf{b})$ values are small, indicating a solution that respects both constraints is found for all cases;
- Although $\Phi(\mathbf{A}, \mathbf{b})$ values grow with the number of points N , this phenomena is due to the fact that the cost function is made of a larger set of residuals. This is a consequence of having grid discretization, and should therefore lose its importance as resolution is increased;
- Having more points in the first frame adds information on the shape and possibly confines motion estimation, allowing for better estimation of correspondence and camera parameters θ and ϕ ;
- On the other hand, an increase in the number of candidates N_c allows for a lower value of $\Phi(\mathbf{A}, \mathbf{b})$, since with more outliers, there are more possibilities of rearranging data into explanations other than the one given by ground truth, as can be inferred from the fact that the considerable average point disparity;

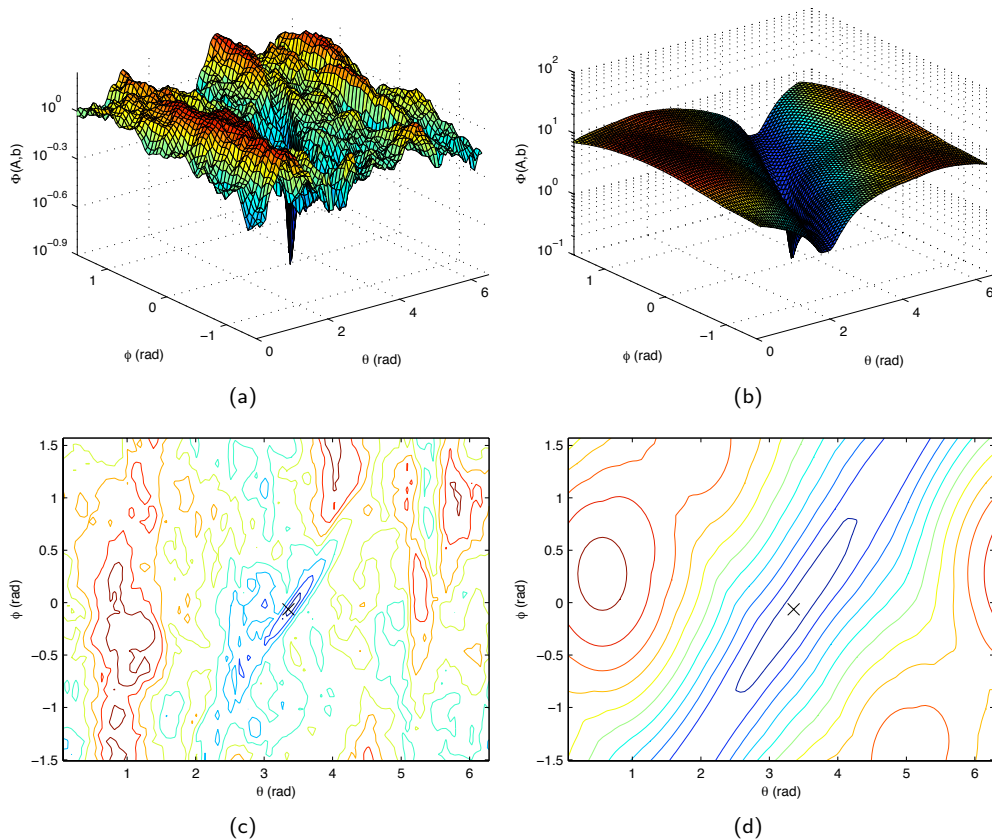


Figure 4.9: Variation of cost function $\Phi(\mathbf{A}, \mathbf{b})$ with camera parameters θ and ϕ for $N = 5$ points and $N_c = 10$ candidates for a grid resolution of 2500 points (the cross in the contours represents parameters θ, ϕ used to generate data): (a) surface obtained with multiple candidates; (b) surface obtained with ground truth; (c) contour obtained with multiple candidates; and, (d) contour obtained with ground truth.

- The fact that the estimation of motion parameters θ and ϕ worsens as the number of candidates N_c increases can be attributed to the same reasons.

These results allow us to define a region for this grid resolution of $N_c \leq N$ candidates to N points where the number of outliers is such that they are not easily coupled into an alternative valid motion model, thus encouraging the use of the proposed search on all possible motion matrices present in the Stiefel manifold. Within this region, **our method is able to successfully estimate motion and correspondences from images alone in a non-combinatoric fashion**. To get a sense of what the cost function $\Phi(\mathbf{A}, \mathbf{b})$ profile is in this region, we show in Fig. 4.9 the cost function obtained for $N = 20$ points and $N_c = 10$ candidates and the cost function obtained by feeding only ground truth to Algorithm 2. This figure shows that both cost functions exhibit their global minimum in the same neighborhood, which coincides with the values of (θ, ϕ) used to generate data.

It should be noted, in this context, that the algorithm selecting candidate matches, *i.e.*, the feature extractor, is the sole responsible for the value of N_c . The choice of local descriptors and their ability to discriminate between possible matches is, therefore, important when it comes to avoiding multiple explanations for the data.

Grid resolution variation. For each of the test cases, we now vary the resolution of the grid and measure the results of the obtained camera parameters ϕ and θ against ground truth, as well as the value of the cost function $\Phi(\mathbf{A}, \mathbf{b})$ and the average point disparity, for different values of the number of candidates N_c .

Results on the parameter estimation error for ϕ (Fig. 4.13) and θ (Fig. 4.12) restate the algorithm as a good motion estimator, even in the presence of a number of outliers that exceeds the number of points by a

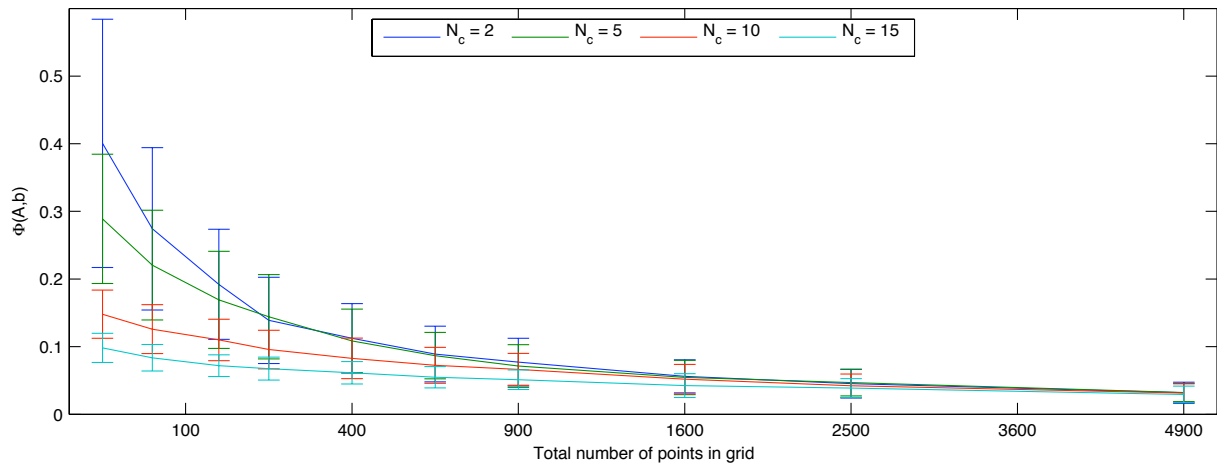


Figure 4.10: Variation of $\Phi(\mathbf{A}, \mathbf{b})$ with grid resolution and number of candidates N_c for $N = 10$ points, averaged over 200 tests.

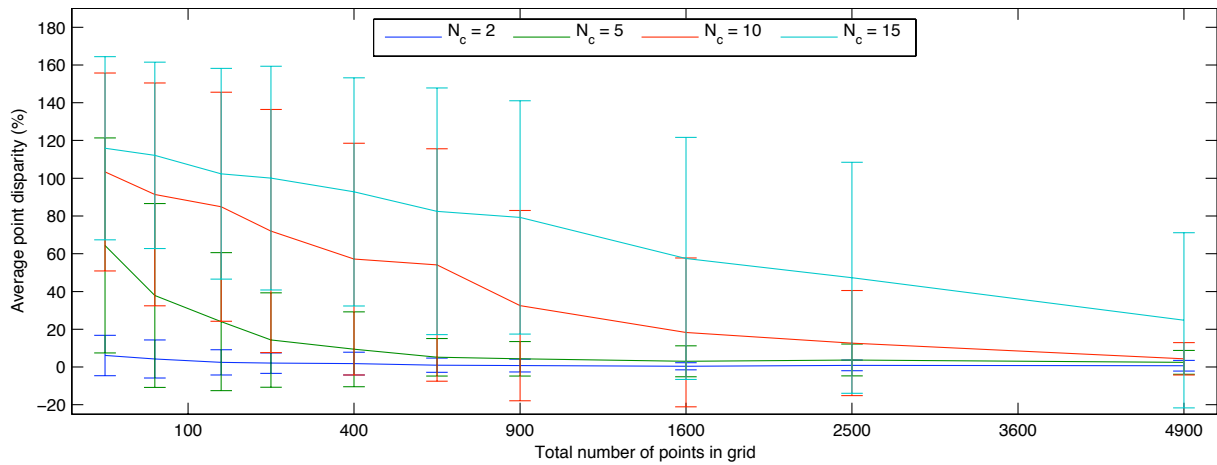


Figure 4.11: Variation of average point disparity with grid resolution and number of candidates N_c for a set of $N = 10$ points, averaged over 200 tests.

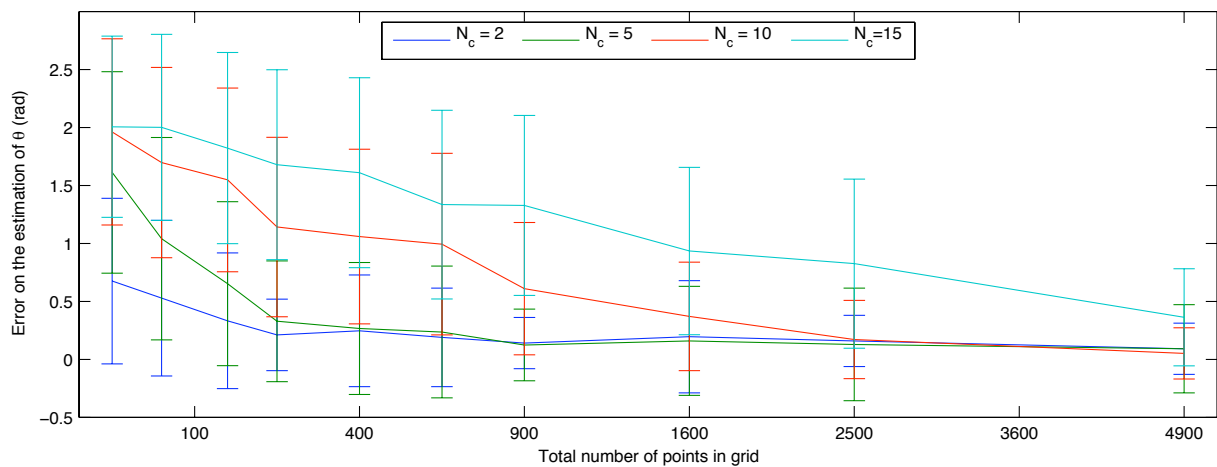


Figure 4.12: Error on the estimation of θ with variation of grid resolution and number of candidates N_c for $N = 10$ points, averaged over 200 tests.

significant margin. The estimation is enhanced by an increase of the total number of points in the parameter grid, achieving the best results for the same grid resolution when a smaller number of candidates is present.

A comparison of the cost function variation $\Phi(\mathbf{A}, \mathbf{b})$ (Fig. 4.10) with the average point disparity (Fig. 4.11) shows the existent correlation between the average point disparity and the cost value. As the number of candidates increases, however, this correlation becomes less evident for smaller scales.

From this experiment, we conclude that the grid resolution should be chosen according to the cardinality of the level sets \mathcal{S}_i , **therefore allowing the use of coarser resolutions without degrading the results of motion and correspondence estimation**, something to consider as a trade-off exists between using a finer resolution and computational effort. It should be noted, however, that while a finer scale allows a clearer distinction between some outlier arrangements and the correct match, multiple arrangements that respect rigidity constraints can still exist. In these cases, we are not able to do anything but accept all explanations within these conditions as correct.

Occlusion tests. To evaluate the susceptibility of the algorithm to occlusions, we measure the same metrics described in the previous test, only now instead of sweeping possible grid resolutions, we use a fixed grid of 2500 points and perform, for each test case, a variation of the number of occlusions in the level sets \mathcal{S}_i fed to Algorithm 2.

Results on the variation of cost $\Phi(\mathbf{A}, \mathbf{b})$ (Fig. 4.14), average point disparity (Fig. 4.15) and parameter estimation error (Fig. 4.16 and Fig. 4.17) show, in par with the ones obtained in the previous experiment, that a correlation exists between the cost function and the average point disparity. When the number of original points occluded increases, the cost function exhibits a larger value as a result of it not being able to accommodate data as well into the original motion model and not having alternative explanations.

The variation of the measures tested in this experience allows us to conclude that, for the resolution tested, **our algorithm is resilient to mild occlusion**.

4.4 Real images

In this section, we exemplify the use of our algorithm to find correspondences in real case scenarios.

Feature selection. For the system to be automatic, we have used a corner detector [29, 30] (as mentioned in Sec. 3.3) as our feature extraction algorithm to find prospective points in both images. We randomly select, according to the user input for the number of points, N points to be processed. Care is taken so as to avoid

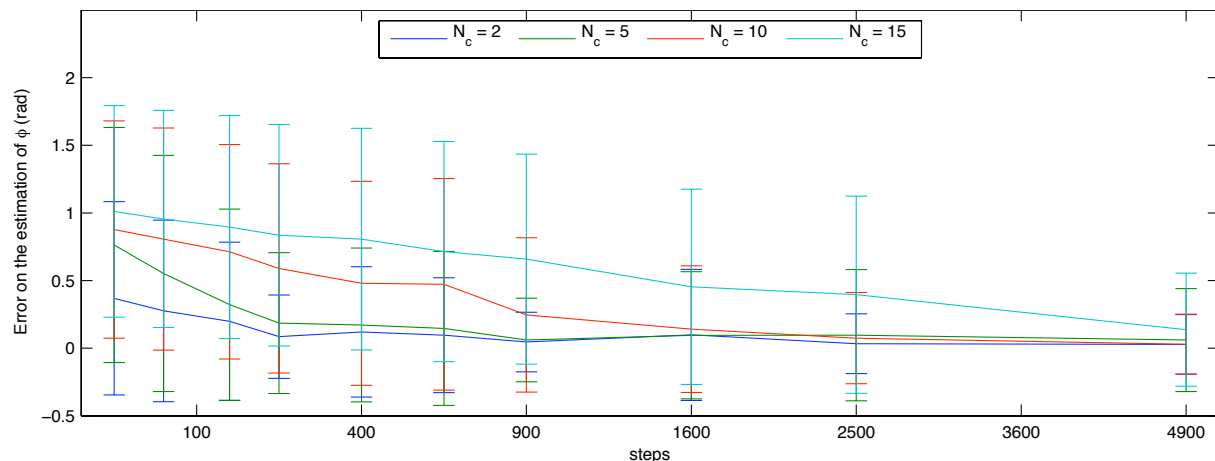


Figure 4.13: Error on the estimation of ϕ with variation of grid resolution and number of candidates N_c for $N = 10$ points, averaged over 200 tests.

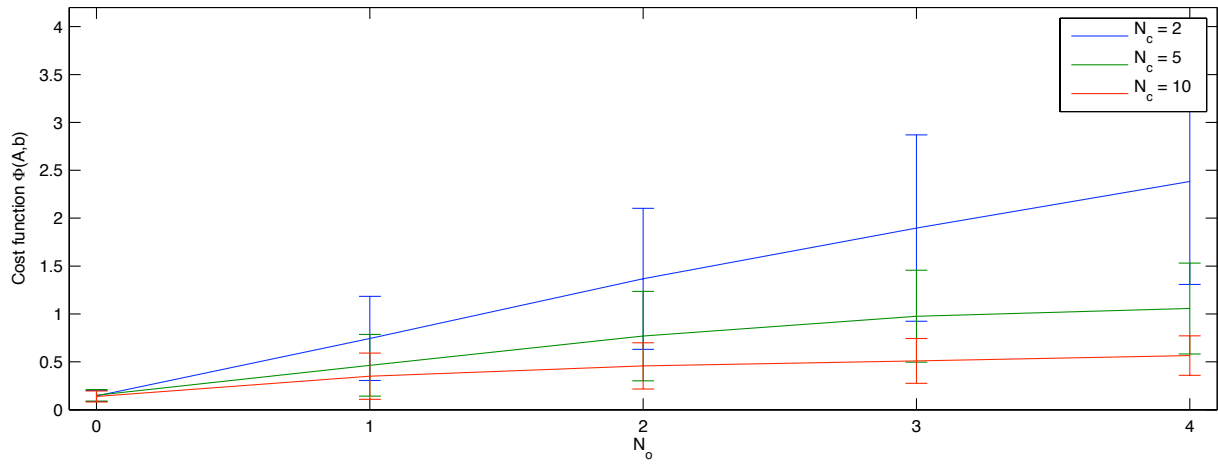


Figure 4.14: Variation of $\Phi(A, b)$ with number of occlusions N_o and number of candidates N_c for $N = 30$ points averaged over 200 tests.

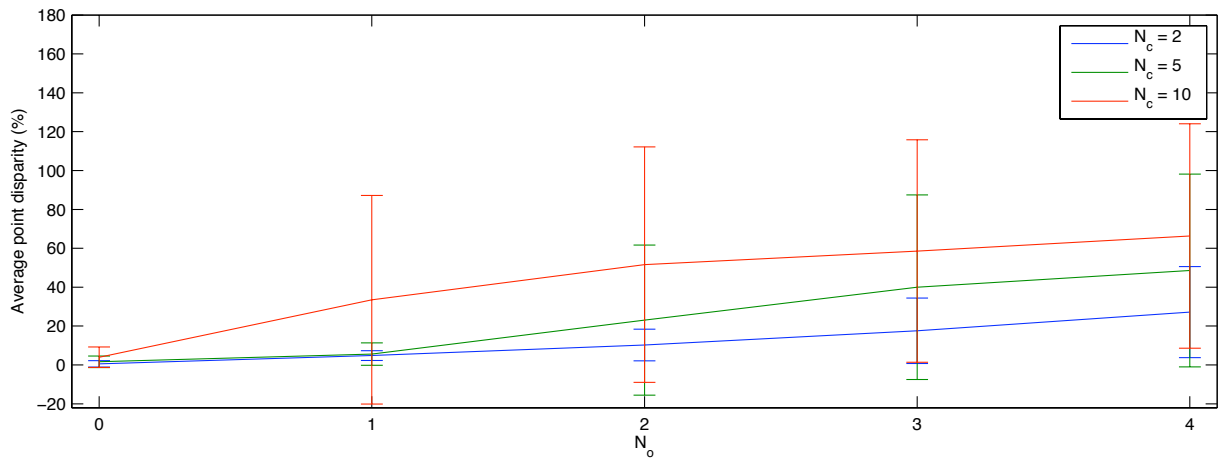


Figure 4.15: Variation of average point disparity with number of occlusions N_o and number of candidates N_c for $N = 30$ points averaged over 200 tests.

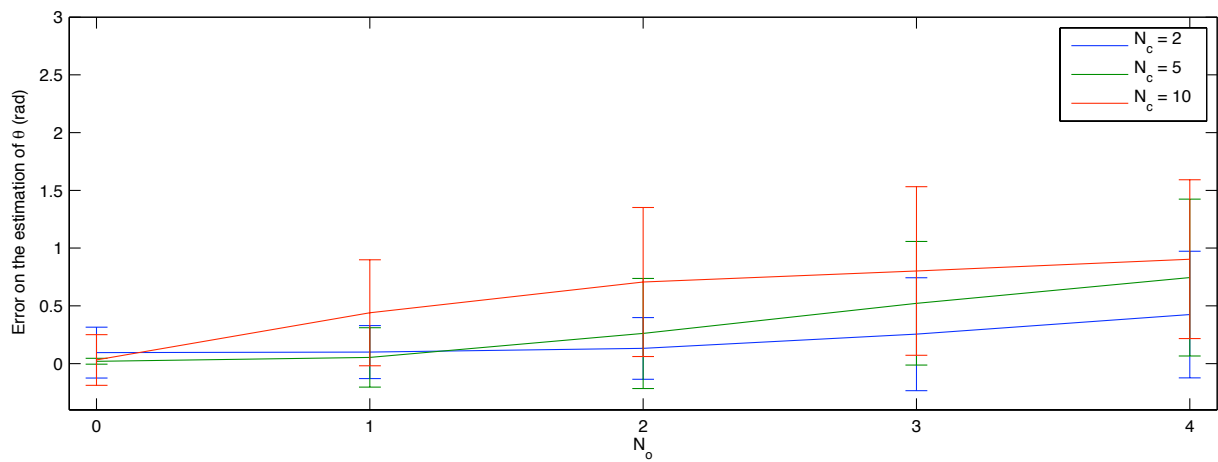


Figure 4.16: Error on the estimation of θ with variation of number of occlusions N_o and number of candidates N_c for $N = 30$ points averaged over 200 tests.

using the same point more than once, as is to discard points with a very small number of candidates in the second frame. For each of these points, we then collect the level sets \mathcal{S}_i by selecting from the detected corners in image 2 according to intensity restrictions. We restrict both points and candidate matches to corners in order to avoid the dimensionality explosion inherent to considering entire surfaces as possible matches.

Lego blocks. In this experiment, we test our method against the 768×576 pixel image pair depicting Lego blocks in different poses present in Figure 4.18. This stereo pair is a courtesy of the SYNTIM project database [33] and was captured with a calibrated setup, whose calibration data is present in Table 4.1.

Table 4.1: Camera calibration parameters for Lego blocks pair.

Parameter	Value
r	0.9753
ϕ	1.5610
θ	1.6052

For this pair, the corner detector extracted 57 and 69 points from Frames 1 and 2, respectively. The distribution of these features through the images can be seen in Fig. 4.19.

In this experiment, we used a grid with a total of 2500 points. The whole process of detecting corner features, finding candidates and solving for correspondences took 108 seconds, with approximately 51 seconds of these (47 %) being used by the correspondence estimation algorithm. The resulting camera estimated parameters and a comparison between these and ground truth is present in Tab. 4.2. It should be noted that

Table 4.2: Error between calibration parameters obtained and ground truth for Lego blocks pair.

Parameter	Value	Estimated	Error
r	0.9753	N/A	N/A
ϕ	1.5610	1.5080	0.0531
θ	1.6052	1.5708	0.0344

the error values obtained are similar to the error due to grid discretization

$$\frac{\pi}{\sqrt{2500}} = 0.0628 \text{ rad.}$$

The matching results present in Fig. 4.20 show correct matches for 23 out of a total of 24 points selected (a match rate of 95.8%). Note that our method does not resort to *downsampling* or *scale pyramids* [5],

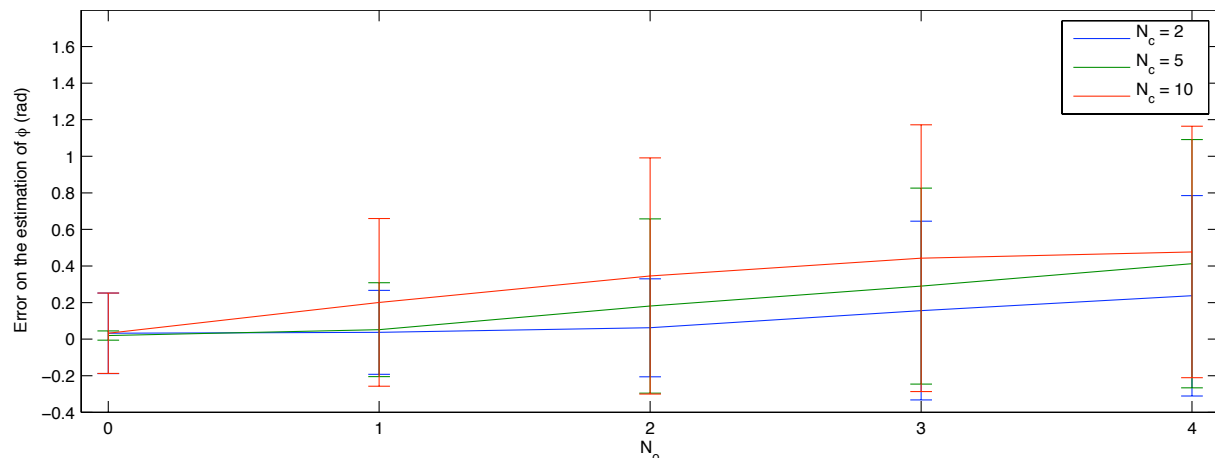
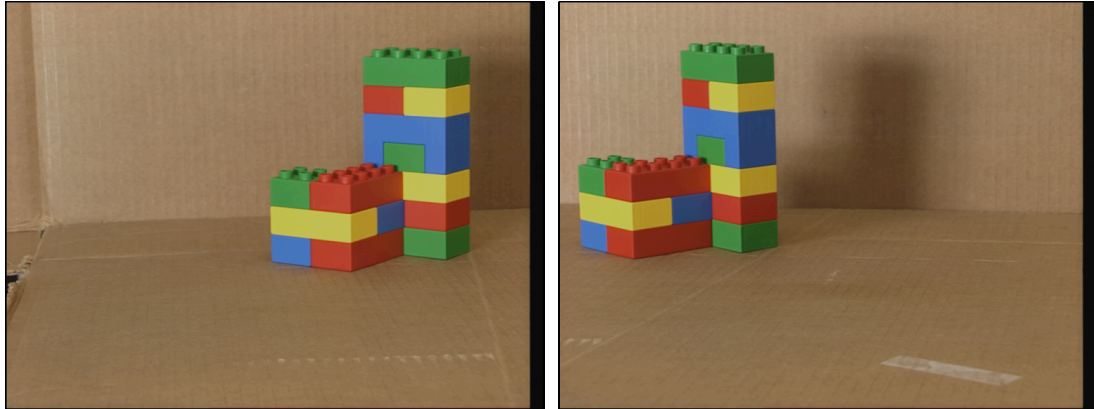


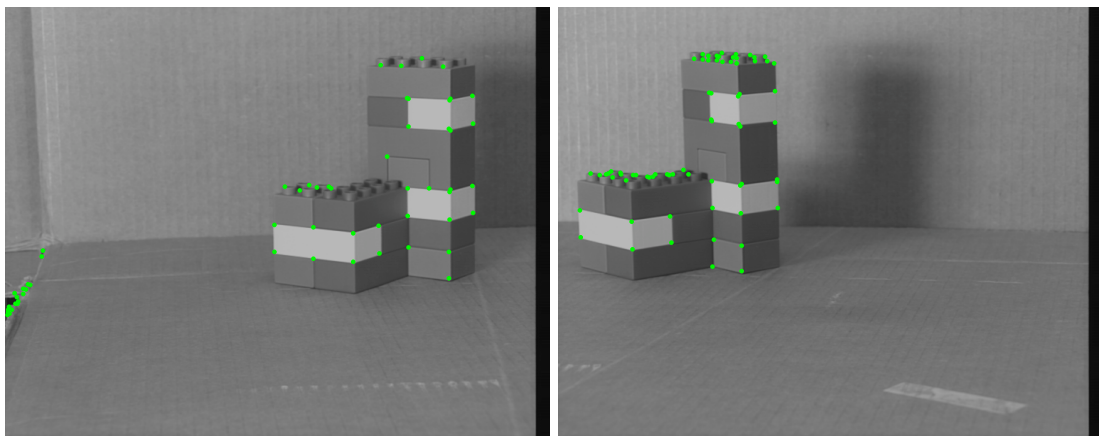
Figure 4.17: Error on the estimation of ϕ with variation of number of occlusions N_o and number of candidates N_c for $N = 30$ points averaged over 200 tests.



(a) Frame 1

(b) Frame 2

Figure 4.18: Lego blocks pair, INRIA © copyright.



(a) Frame 1

(b) Frame 2

Figure 4.19: Corners detected by the algorithm in Lego blocks pair for use in correspondence estimation.

instead handling images in their native resolution, therefore avoiding the computational price associated with these techniques.

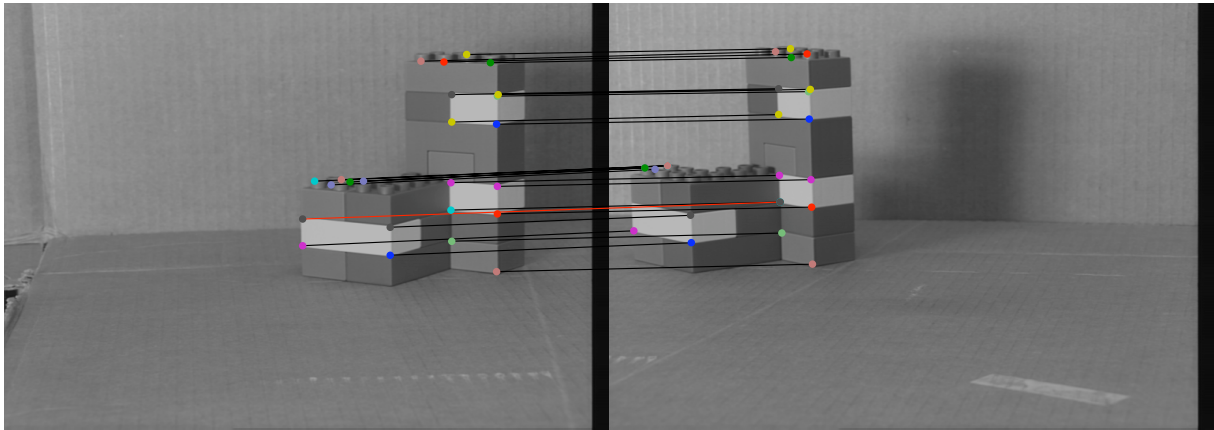


Figure 4.20: Matching results for Lego blocks pair. Red lines represent mismatches.

Lego blocks with occlusion. In this experiment, we take the points selected in the previous experiment and remove the ground truth match from level set \mathcal{S}_1 . In these circumstances, we obtain the same parameter estimation errors as before (Tab. 4.2) and attain the matching results present in Fig. 4.21. The matching

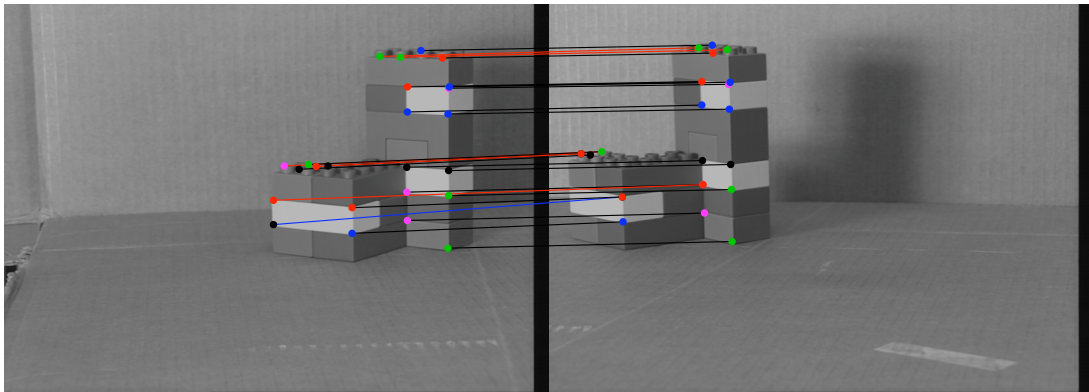
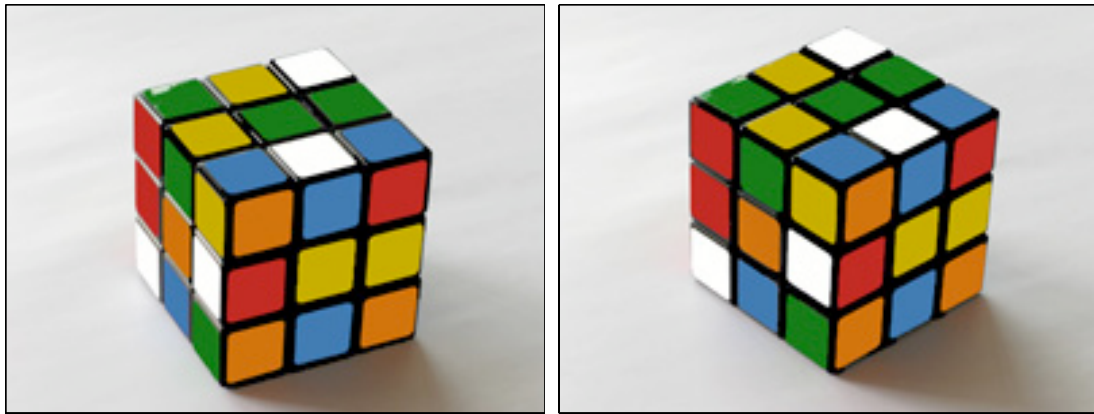


Figure 4.21: Matching results for Lego blocks pair with occluded points. Blue line represents correspondence obtained for occluded point. Red lines represent mismatches.

results still show correct matches for 18 out of a total of 24 points (a match rate of 75%). These results show that the existence of mild occlusion does not seem to perturb the matching process for the majority of remaining points, allowing us to conclude that **our method is able to estimate correspondences and motion in a robust manner while being subject to a high number of outliers and light occlusion.**

Rubik's cube. In this experiment, we test our method against the 512×384 pixel image pair present in Figure 4.22, depicting a Rubik's cube with slight perspective effects captured by the author with an uncalibrated camera across different poses. We have digitally edited the pictures to remove light reflections in the black grid of the cube and to harmonize face colors between frames. For this pair, the corner detector extracted 214 features in first frame and 210 in the second (Fig. 4.23).

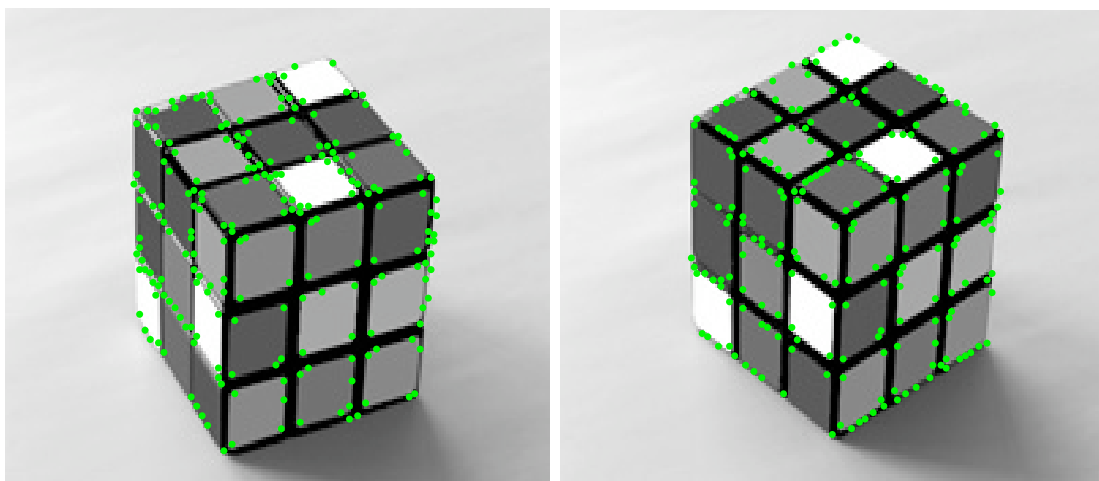
In this experiment, we selected 20 points in frame 1 and obtained level sets in frame 2 with between 5 to 19 candidate matches. Since the corner detector didn't detect some correct points in frame 2 as possible matches, we added them to their respective level sets by hand. We used a grid with a total of 2500 points.



(a) Frame 1

(b) Frame 2

Figure 4.22: Rubik's cube pair.



(a) Frame 1

(b) Frame 2

Figure 4.23: Corners detected by the algorithm in Rubik's cube pair for use in correspondence estimation.

The results are present in Fig. 4.24, and they show correct matches for 15 out of a total of 20 points selected (a match rate of 75 %). Note that in this case, we have a larger number of candidates than the ones in the previous experiment, which allows for various explanations of the data by different physical models.

If we consider the direction of \mathbf{b} defined by the parameter θ as in (3.5), we notice that the candidates chosen differ from the correct matches only by small increments.

As mentioned in Sec. 4.3, this might be solved by using a finer grid resolution, albeit incurring in a very high computational time.

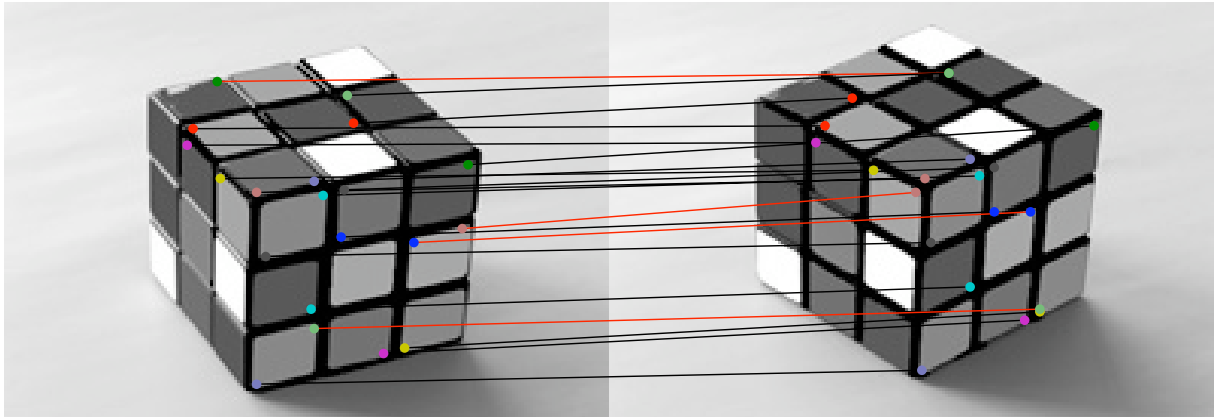


Figure 4.24: Matching results for Rubik's cube pair. Red lines represent mismatches.

Despite that fact, if we extend the results in Fig. 4.24 to include the 5 point correspondences with the least cost and consider these good correspondences, as shown in Fig. 4.25, we note that they now include the majority (4 out of the 5) of the correct matches not found previously. If we consider the best 8 matches, the remaining correct match is included.

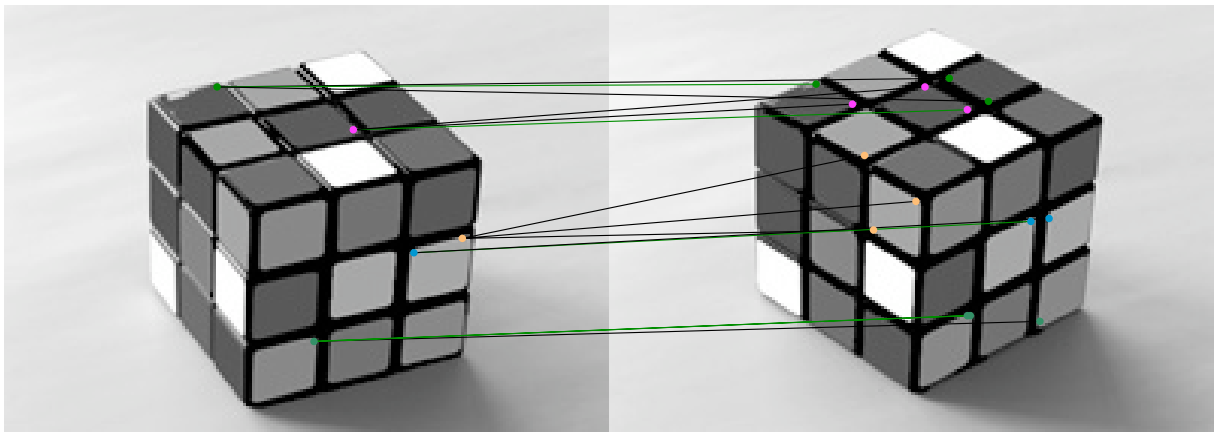


Figure 4.25: Results for best 5 matches in Rubik's cube pair. Green lines represent ground truth explanation.

This experiment comes to show that, **in cases where correspondence estimation is hard, our method is still able to drastically reduce the cardinality of the set of possible correspondence matches between frames obtained using brightness constraints.** Since all the obtained correspondence possibilities are validated by a physical model, we are not able (nor other methods, for that matter) to further disambiguate between these cases.

4.5 Coupling with existing feature extraction methods

In this section, we present a joint utilization of our method with existing state-of-the-art feature extractors. The purpose of this usage is to illustrate the versatility inherent to having our method's input consist simply of sets of point coordinates, regardless of whence they were obtained from. This characteristic lets us benefit from the advantages of each feature extraction algorithm, selecting the more appropriate method for each scenario.

For this purpose, we revisit the Rubik's cube image pair, this time in its original condition (Fig. 4.26). Note that in this case corner detection is not appropriate, since it detects features in the cube grid and faces due to lighting variations (Fig. 4.27). This leads to a large amount of features (663 for the first frame and 1590 for the second) and consequently, as mentioned in Sec. 4.3, to a poor correspondence estimation (Fig. 4.28) with no correct matches in the set of 5 least cost matches. Also, the periodic structure (chirping) of the image makes the disposition of candidates extracted with the corner detector such that they live in the lines defined by the edges of the faces. This allows for multiple explanations of the data, if the direction of the vector \mathbf{b}^\perp is aligned with any of these lines.

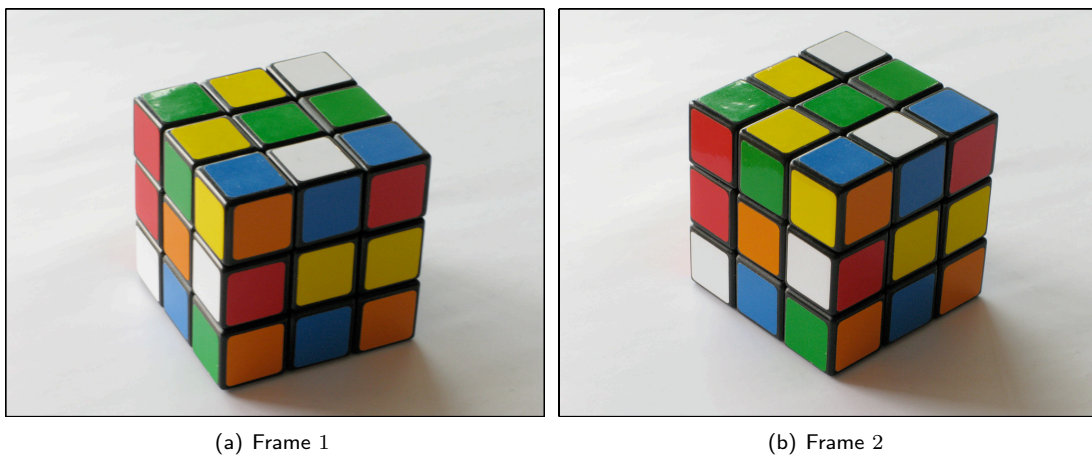


Figure 4.26: Original Rubik's cube pair.

In this experiment, we proceed as follows: we start by extracting from each of the frames image descriptors using the SIFT implementation in [34] (Fig. 4.29); then, we randomly select $N = 20$ points from the features

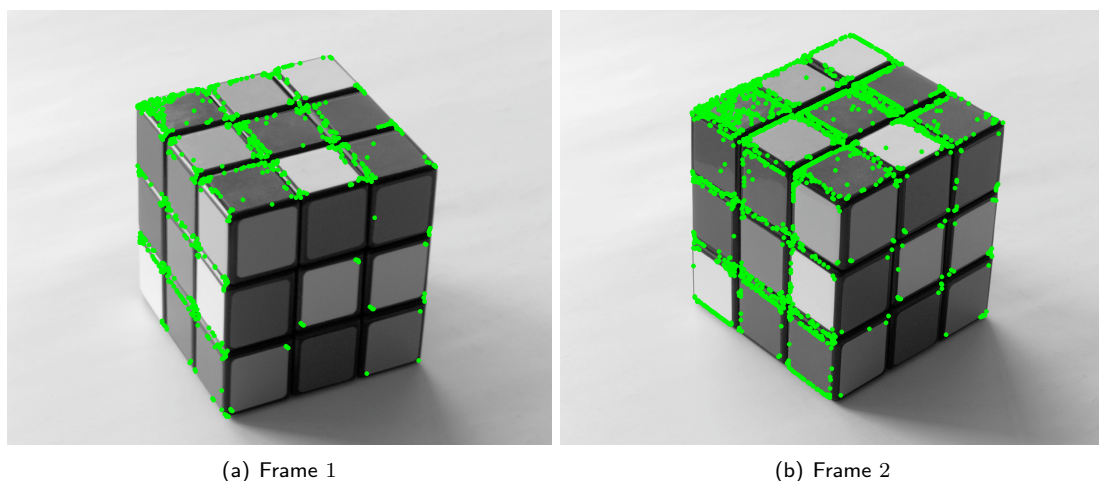


Figure 4.27: Corners detected by the algorithm in original Rubik's cube pair for use in correspondence estimation.

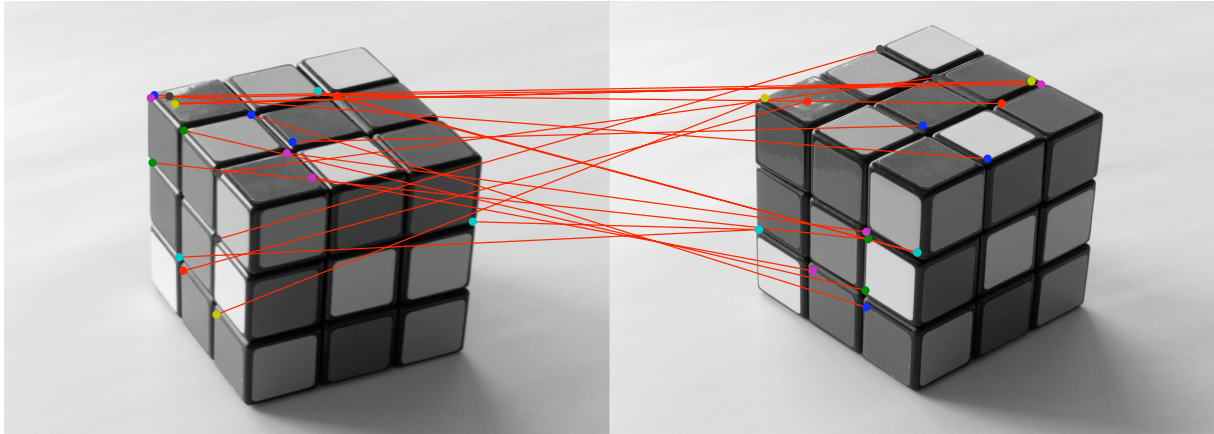


Figure 4.28: Correspondence results for original Rubik's cube pair using corners as features for a grid resolution of 2500 points.

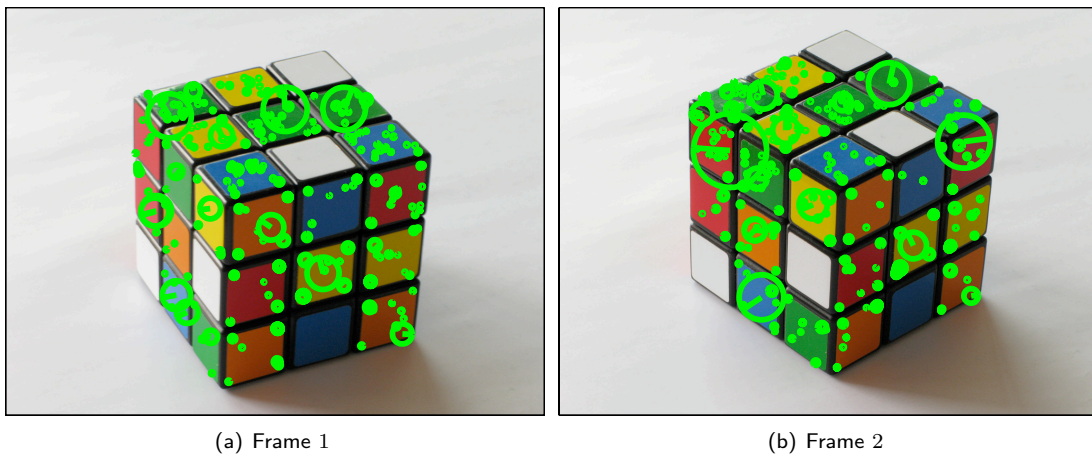


Figure 4.29: Features detected by SIFT in the original Rubik's cube pair for use in correspondence estimation.

detected in the first frame and form for each of these points their respective level sets S_i from the features detected in the second frame. Note that in this process, we have only used the features image coordinates, having dismissed the scale and orientation information, since having considerable 3D motion does not allow us to robustly match points based on these cues. Results are present in Fig. 4.30, and they show a significant improvement over the corner feature experiment. Out of a total of 20 points, 13 (63%) are included in the selection of 5 matches with least cost.

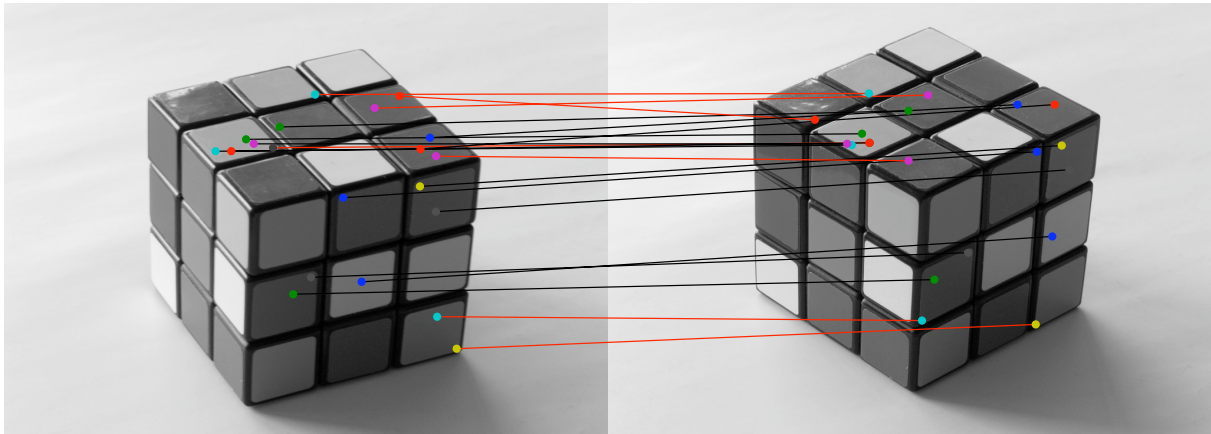


Figure 4.30: Correspondence results for original Rubik's cube pair using SIFT features for a grid resolution of 2500 points.

Chapter 5

Conclusions and future work

In this thesis, we have successfully designed and demonstrated through experimentation a method — CERD — to estimate motion and correspondence between a pair of images by imposing global rigidity constraints to points and correspondence candidates found using local descriptors. The resulting algorithm has optimal properties when the camera parameters are known. When they are not, it performs a grid search on the motion space (a Stiefel manifold) to find the globally best possible explanations for the data. The grid search nature of our method allows its computational effort to be easily distributed around multiple computers/cores, by dividing the grid. Also, extensions that provide refinement iterations around discovered minima should be straightforward to implement. This algorithm has polynomial complexity, therefore bypassing the combinatorial explosion typically associated with the correspondence problem.

The intrinsic ambiguity of having various possible explanations for the data prevents the distinction of a specific configuration as the “correct” one. This can be avoided by using more discriminative local descriptors to find points and correspondence candidates, as was shown by using SIFT features.

In our opinion, further investigation is needed in the following areas:

Polishing. Our experiments show that the algorithm presented in this thesis, despite estimating correctly the camera motion parameters, is sensitive to occlusion, due to the fact that a correspondence is forced for each point in the first image. Additional information in the model exists that may leave space for match improvement as a polishing phase after running the algorithm. For instance, the estimated $C(\gamma)$ can be checked for an existence of multiple global minima, allowing us to detect occlusion or reject motion explanations that do not have this function behavior or rejecting point classes that don't exhibit the minima at the same order of magnitude as all others. Alternatively, one could use the assumption that $\mathbf{z}^T \mathbf{1} = 0$ and compare the difference in the average of the points projected on \mathbf{b}_\perp between both frames with γ as a measure of confidence;

Extension to multi-frame. The method presented in this thesis has a linear growth on the number of frames, since we can, for each point in the grid, process every frame. Hence, multi-frame analysis poses as a natural evolution for the algorithms presented in this thesis, with which the Affine indetermination in the reconstruction can be disambiguated (see Appendix C for details). Moreover, the multi-frame scenario yields an interesting optimization problem, which could help further distinguish between the candidates obtained by this algorithm in more general settings.

Extension to scaled orthography. In this thesis, we have shown that the nature of our method's input allows for an efficient coupling with feature extraction methods, which can be used to find interesting points for correspondence estimation. Although in this thesis, we have dismissed SIFT's information on scale and orientation, one might use this to extend our method to work in a scaled orthography case.

Appendix A

Set distance

In this appendix, we define the operator

$$d(\mathbf{p} + \mathbf{a}k, \mathcal{S})$$

that calculates the distance of the line passing through p with the direction given by \mathbf{a} to a given set \mathcal{S} . Mathematically, this corresponds to

$$d(\mathbf{p} + \mathbf{a}k, \mathcal{S}) = \inf_{k \in \mathbb{R}, \mathbf{s} \in \mathcal{S}} \|\mathbf{p} + \mathbf{a}k - \mathbf{s}\|,$$

or, equivalently, to

$$d(\mathbf{p} + \mathbf{a}k, \mathcal{S}) = \inf_{\mathbf{s} \in \mathcal{S}} \inf_{k \in \mathbb{R}} \|\mathbf{p} + \mathbf{a}k - \mathbf{s}\|. \quad (\text{A.1})$$

As can be seen Fig. A.1, the solution of

$$\inf_{k \in \mathbb{R}} \|\mathbf{p} + \mathbf{a}k - \mathbf{s}\|$$

is the projection of the difference between points \mathbf{p} and \mathbf{s} in the subspace defined by \mathbf{a} ,

$$k = \frac{\langle \mathbf{p} - \mathbf{s}, \mathbf{a} \rangle}{\|\mathbf{a}\|^2}$$

or, in matrix notation

$$k = \frac{\mathbf{a}^\top}{\|\mathbf{a}\|^2} (\mathbf{p} - \mathbf{s}). \quad (\text{A.2})$$

Replacing (A.2) into (A.1) gives

$$d(\mathbf{p} + \mathbf{a}k, \mathcal{S}) = \inf_{\mathbf{s} \in \mathcal{S}} \left\| \mathbf{p} + \mathbf{a} \frac{\mathbf{a}^\top}{\|\mathbf{a}\|^2} (\mathbf{p} - \mathbf{s}) - \mathbf{s} \right\|,$$

which can be rearranged into

$$d(\mathbf{p} + \mathbf{a}k, \mathcal{S}) = \inf_{\mathbf{s} \in \mathcal{S}} \left\| \left(I - \frac{\mathbf{a}\mathbf{a}^\top}{\|\mathbf{a}\|^2} \right) (\mathbf{p} - \mathbf{s}) \right\|. \quad (\text{A.3})$$

By noting that

$$\left(I - \frac{\mathbf{a}\mathbf{a}^\top}{\|\mathbf{a}\|^2} \right) = \frac{\mathbf{a}_\perp (\mathbf{a}_\perp)^\top}{\|\mathbf{a}_\perp\|^2}$$

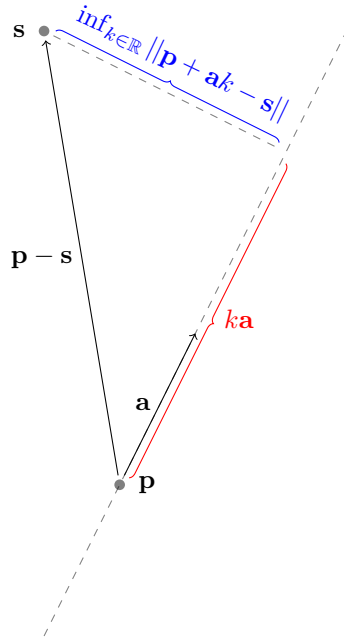


Figure A.1: Visual explanation of distance operator $d(\mathbf{p} + \mathbf{a}k, \mathcal{S})$.

where \mathbf{a}_\perp represents the vector orthogonal to \mathbf{a} , (A.3) becomes

$$d(\mathbf{p} + \mathbf{a}k, \mathcal{S}) = \inf_{\mathbf{s} \in \mathcal{S}} \left\| \frac{\mathbf{a}_\perp (\mathbf{a}_\perp)^\top}{\|\mathbf{a}_\perp\|^2} (\mathbf{p} - \mathbf{s}) \right\|$$

or the distance

$$d(\mathbf{p} + \mathbf{a}k, \mathcal{S}) = \inf_{\mathbf{s} \in \mathcal{S}} \left| \frac{(\mathbf{a}_\perp)^\top}{\|\mathbf{a}_\perp\|^2} \mathbf{p} - \frac{(\mathbf{a}_\perp)^\top}{\|\mathbf{a}_\perp\|^2} \mathbf{s} \right|,$$

which we define as

$$d\left(\frac{(\mathbf{a}_\perp)^\top}{\|\mathbf{a}_\perp\|^2} \mathbf{p}, \frac{(\mathbf{a}_\perp)^\top}{\|\mathbf{a}_\perp\|^2} \mathcal{S}\right).$$

It should be noted that, in this context, we consider the set projection $\frac{(\mathbf{a}_\perp)^\top}{\|\mathbf{a}_\perp\|^2} \mathcal{S}$ as the set obtained by projecting each of the points in the original set \mathcal{S} individually.

Appendix B

Parameterization of the Stiefel manifold

$O(2, 3)$

In this appendix, we derive a general result on the parameterization of matrices that belong to the Stiefel manifold $O(2, 3)$, the set of $\mathbb{R}^{2 \times 3}$ matrices whose rows are orthonormal:

$$O(2, 3) = \{\mathbf{R} \in \mathbb{R}^{2 \times 3} : \mathbf{R}\mathbf{R}^\top = \mathbf{I}_2\}.$$

We start by splitting \mathbf{R} in the matrix $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ and vector $\mathbf{b} \in \mathbb{R}^2$ as

$$\mathbf{R} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}.$$

Let us now consider the Singular Value Decomposition of the matrix \mathbf{A} in (2.5)

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{B.1}$$

where \mathbf{U} and \mathbf{V}^\top are unitary matrices and $\mathbf{\Sigma}$ is a diagonal matrix comprising its two singular values σ_1 and σ_2 . The matrix \mathbf{A} can be considered a compression of the matrix \mathbf{R}_0

$$\mathbf{R}_0 = \begin{bmatrix} \mathbf{R} \\ 0 \ 0 \ 0 \end{bmatrix}$$

as it can be obtained from the latter using the expression

$$\mathbf{A} = \mathbf{P}\mathbf{R}_0\mathbf{P}^\top$$

where the orthogonal projection \mathbf{P} is simply the operator that discards the last column of \mathbf{R}

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

Let $\alpha_1, \alpha_2, \alpha_3$ be the singular values of the matrix \mathbf{R}_0 . We know for a fact [35], due to the Cauchy Interlacing theorem, that

$$\begin{aligned} \alpha_1 &\leq \sigma_1 \leq \alpha_2 \\ \alpha_2 &\leq \sigma_2 \leq \alpha_3 \end{aligned} .$$

Using the fact that the matrix \mathbf{R}_0 is a concatenation of a Stiefel Manifold with a row of zeros and, as such, inherits the rank of the former, we can write $\alpha_1 = \alpha_2 = 1$ and $\alpha_3 = 0$. Also, given that \mathbf{U} and \mathbf{V}^\top are unknown rotation matrices, we can write them as

$$\mathbf{U} = \begin{bmatrix} \cos \theta & \pm \sin \theta \\ \mp \sin \theta & \cos \theta \end{bmatrix}, \quad \mathbf{V}^\top = \begin{bmatrix} \cos \phi & \pm \sin \phi \\ \mp \sin \phi & \cos \phi \end{bmatrix}.$$

This allows us to write \mathbf{A} as

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} \cos \phi & \pm \sin \phi \\ \mp \sin \phi & \cos \phi \end{bmatrix},$$

with $0 \leq r \leq 1$. It should be noted that we have dropped the sign indetermination present in the matrix \mathbf{U} due to the fact that it can be merged with the indetermination present in the matrix \mathbf{V} .

Using the definition of Stiefel Manifold presented in (2.3), we can write

$$\mathbf{R}\mathbf{R}^\top = \mathbf{A}\mathbf{A}^\top + \mathbf{b}\mathbf{b}^\top = \mathbf{I}_2. \quad (\text{B.2})$$

Replacing the matrix \mathbf{A} in (B.2) by its parameterization in (B.1), we get

$$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\mathbf{V}\boldsymbol{\Sigma}^\top\mathbf{U}^\top + \mathbf{b}\mathbf{b}^\top = \mathbf{I}_2$$

which can be further simplified, given the fact that rotation matrices are orthogonal matrices, into the form

$$\mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^\top + \mathbf{b}\mathbf{b}^\top = \mathbf{I}_2.$$

By noting that $\mathbf{U}\mathbf{I}_2\mathbf{U}^\top = \mathbf{I}_2$, we are able to write $\mathbf{b}\mathbf{b}^\top$ as

$$\mathbf{b}\mathbf{b}^\top = \mathbf{U}(\mathbf{I}_2 - \boldsymbol{\Sigma}^2)\mathbf{U}^\top$$

and, consequently, the vector \mathbf{b} as

$$\mathbf{b} = \pm \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} \sqrt{1 - r^2},$$

where the sign indetermination results from the factorization of $1 - r^2$ into two equal parts.

Appendix C

Depth as a byproduct of CERD

Since the solution to the system expressed in (2.6), repeated here for readability purposes,

$$\mathbf{W}_2 = \mathbf{A}\mathbf{W}_1 + \mathbf{b}\mathbf{z}^\top + \mathbf{t}\mathbf{1}^\top, \quad (\text{C.1})$$

is equivalent to having a solution that satisfies the systems arising from the projection of the former on a given subspace and its orthogonal complement, we shall now assume the correct correspondence and unknown parameters θ, ϕ, t have been determined and proceed to projecting (C.1) on the axis b

$$\frac{\mathbf{b}^\top \mathbf{W}_2}{\|\mathbf{b}\|} = \frac{\mathbf{b}^\top \mathbf{A}\mathbf{W}_1 + \mathbf{b}^\top \mathbf{b}\mathbf{z}^\top + \mathbf{b}^\top \mathbf{t}\mathbf{1}^\top}{\|\mathbf{b}\|}. \quad (\text{C.2})$$

Using the parameterization of the matrix \mathbf{A} in (3.4) and the form of the vector \mathbf{b} present in (3.5), we can simplify the product $\mathbf{b}^\top \mathbf{A}$, leaving (C.2) as

$$\pm \begin{bmatrix} -\sin \theta & \cos \theta \end{bmatrix} \mathbf{W}_2 = \pm \begin{bmatrix} -r \sin \phi & r \cos \phi \end{bmatrix} \mathbf{W}_1 + \frac{\mathbf{b}^\top \mathbf{b}\mathbf{z}^\top + \mathbf{b}^\top \mathbf{t}\mathbf{1}^\top}{\sqrt{1-r^2}} \quad (\text{C.3})$$

By noting that the inner product of a unit norm vector with itself is 1, transposing and rearranging (C.3), we get

$$\mathbf{W}_2^\top \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} = r \mathbf{W}_1^\top \begin{bmatrix} -\sin \phi \\ \cos \phi \end{bmatrix} \pm \mathbf{z} \sqrt{1-r^2} + \mathbf{t} \mathbf{1}^\top \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}, \quad (\text{C.4})$$

where we have used the distributive property to concentrate the sign in the term $\mathbf{z} \sqrt{1-r^2}$. From (C.4), we can conclude that, having the correspondence information and motion parameters θ and ϕ , the i -th depth component z_i is only determinable up to a transformation

$$z_i = \frac{\alpha - \beta r}{\pm \sqrt{1-r^2}}.$$

dependent on the unknown parameter r .

To understand the implications of not knowing the parameter r , we provide a concrete example. We start by considering the model present in Fig. C.1, centered in what we shall designate the *world coordinate system*, and two cameras positioned relative to the object as in Fig. C.2: the first with its principal ray perpendicular to the z axis w.r.t. (with respect to) the world coordinate system and the second as a rotation of $\frac{\pi}{4}$ radians about the y axis and no translation relative to the former. For simplicity sake, we will only consider the points in the edges of the object, whose coordinates w.r.t. the world coordinate system are specified in Table C.1.

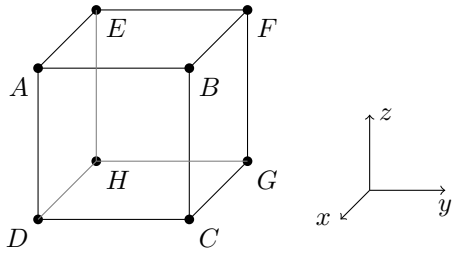


Figure C.1: Object in 3D coordinate system.

Point	(x, y, z) Coordinates
<i>A</i>	$(1, -1, 1)$
<i>B</i>	$(1, 1, 1)$
<i>C</i>	$(1, 1, -1)$
<i>D</i>	$(1, -1, -1)$
<i>E</i>	$(-1, -1, 1)$
<i>F</i>	$(-1, 1, 1)$
<i>G</i>	$(-1, 1, -1)$
<i>H</i>	$(-1, -1, -1)$

Table C.1: Coordinates of the points in Fig. C.1 w.r.t. the world coordinate system.

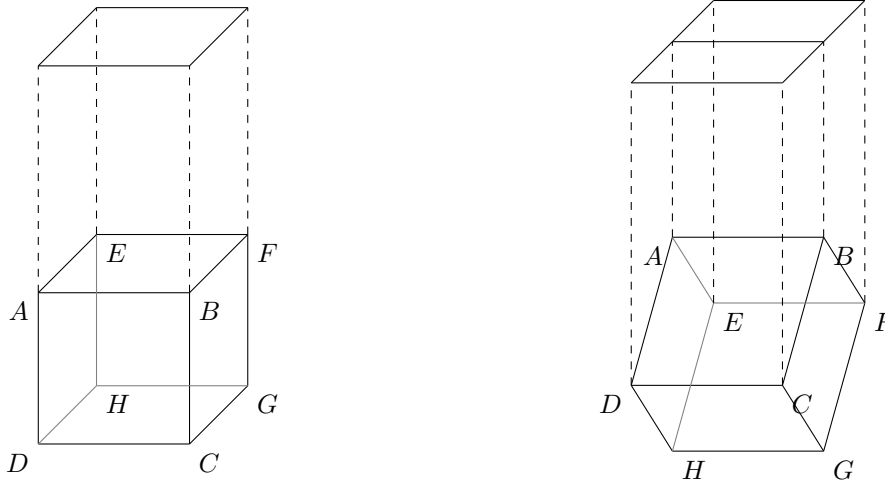


Figure C.2: Relative position of the cameras w.r.t. the world coordinate system. Left: Camera 1. Right: Camera 2.

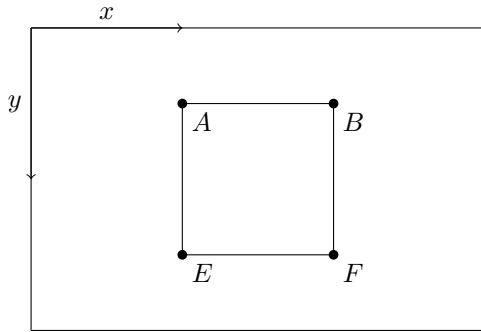
Cameras 1 and 2 correspond, therefore, to the projections

$$\begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_H \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_A & \dots & x_H \\ y_A & \dots & y_H \\ z_A & \dots & z_H \\ 1 & \dots & 1 \end{bmatrix}$$

and

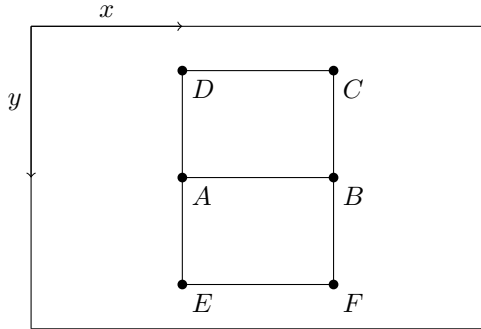
$$\begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_H \end{bmatrix} = \begin{bmatrix} \cos \frac{\pi}{4} & 0 & -\sin \frac{\pi}{4} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_A & \dots & x_H \\ y_A & \dots & y_H \\ z_A & \dots & z_H \\ 1 & \dots & 1 \end{bmatrix}$$

respectively, obtaining the images depicted in Fig. C.3, for camera 1, and Fig. C.4, for camera 2, with respective coordinates w.r.t. to each camera coordinate system listed in Table C.2 and Table C.3.



Point	(x, y, z) Coordinates
A	$(1, -1)$
B	$(1, 1)$
E	$(-1, -1)$
F	$(-1, 1)$

Figure C.3: Image obtained by camera 1. The remainder of the Table C.2: Coordinates of the points in Fig. C.3 w.r.t. camera 1's coordinate system. points is invisible due to having overlapping projection coordinates with the points represented.



Point	(x, y, z) Coordinates
A	$(0, -1)$
B	$(0, 1)$
E	$(-\sqrt{2}, -1)$
F	$(-\sqrt{2}, 1)$

Figure C.4: Image obtained by camera 2. The remainder of the Table C.3: Coordinates of the points in Fig. C.4 w.r.t. camera 2's coordinate system. points is invisible due to having overlapping projected coordinates with the points represented.

Performing a SVD on the sub-stiefel component of camera 2's projection matrix and relating it with the definition in (3.4) gives

$$\begin{aligned} \theta &= \frac{\pi}{2} \\ \phi &= \frac{\pi}{2} \\ r &= \frac{\sqrt{2}}{2}. \end{aligned}$$

Let us now take \mathbf{W}_1 and \mathbf{W}_2 as the coordinates of the points visible in both images, points A, B, E, F .

By solving (C.4) for each of the 4 points, we get equations

$$\begin{bmatrix} z_A \\ z_B \\ z_E \\ z_F \end{bmatrix} = \pm \frac{1}{\sqrt{1-r^2}} \left(\begin{bmatrix} 0 \\ 0 \\ -\sqrt{2} \\ -\sqrt{2} \end{bmatrix} - r \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \right).$$

A representation obtained using admissible values for r is present in Fig. C.5 and shows the effect of this parameter on the object reconstruction. From this image, we can conclude the model is only reconstructed up to an Affine transformation [9] and different projections are needed to disambiguate this scenario. Note that when a value of $r = 1$ is used, the depths of the object cannot be inferred from (C.4). This is coherent with the fact that transformations with such a value for this parameter are not influenced by the depths of the object the motion whatsoever, as the vector \mathbf{b} becomes the null vector.



Figure C.5: Partial reconstruction of the cube (points A, B, E, F) according to the coordinates obtained in (C) using different values for the parameter r . Left: $r = 0$. Right: $r = 0.9$. All the other points are plotted with the original coordinates for comparison purposes.

Bibliography

- [1] Richard Dawkins. *The Blind Watchmaker*. Penguin Books, 1986.
- [2] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.
- [3] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. pages 1071–1076, 1995.
- [4] T. S. Huang and A. N. Netravali. Motion and structure from feature correspondences: a review. *Proceedings of the IEEE*, 82(2):252–268, 1994.
- [5] Steve Mann and Rosalind W. Picard. Video orbits of the projective group: A simple approach to featureless estimation of parameters. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 6:1281–1295, 1997.
- [6] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *International Journal of Computer Vision*, pages 329–342, 1998.
- [7] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524, 1987.
- [8] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. pages 586–591, 1991.
- [9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [10] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, August 2002.
- [11] Chuck Thorpe Frank Dellaert, Steven Seitz and Sebastian Thrun. Structure from motion without correspondence. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, June 2000.
- [12] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.
- [13] D. G. Lowe. Fitting parameterized three-dimensional models to images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(5):441–450, 1991.
- [14] P.M.Q. Aguiar and J.M.F. Moura. Rank 1 weighted factorization for 3d structure recovery: algorithms and performance analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1134–1149, Sept. 2003.
- [15] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(3):206–218, 1997.

- [16] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, 2001.
- [17] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. Technical report, Cambridge, MA, USA, 1980.
- [18] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. pages 231–236, 1993.
- [19] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.
- [20] M. Irani. Multi-frame optical flow estimation using subspace constraints. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1:626–633 vol.1, 1999.
- [21] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [22] Alessandro Neri and Giovanni Jacovitti. Maximum likelihood localization of 2-d patterns in the gauss-laguerre transform domain: theoretic framework and preliminary results. *IEEE Transactions on Image Processing*, 13(1):72–86, 2004.
- [23] J. Mota and Aguiar. P.M.Q. Efficient methods for points matching with known camera orientation. Technical report, ISR/IST, 2008.
- [24] R. Oliveira, R. Ferreira, and J. P. Costeira. Optimal multi-frame correspondence with assignment tensors. In *ECCV (3)*, pages 490–501, 2006.
- [25] D.A. Forsyth, S. Ioffe, and J. Haddon. Bayesian structure from motion. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1:660–665 vol.1, 1999.
- [26] Gang Qian, R. Chellappa, and Qinfen Zheng. Bayesian structure from motion using inertial information. *Image Processing. 2002. Proceedings. 2002 International Conference on*, 3:III–425–III–428 vol.3, 2002.
- [27] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- [28] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, February 1990.
- [29] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision*, volume 2, pages 1508–1511, October 2005.
- [30] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (to appear)*, May 2006.
- [31] MATLAB and Simulink user community. Matlab FAQ - MATLAB wiki (formerly known as comp.soft-sys.matlab newsgroup FAQ). http://matlabwiki.mathworks.com/MATLAB_FAQ, June 2009.
- [32] P. J. Acklam. Matlab array manipulation tips and tricks. Online web resource, April 2003.
- [33] INRIA SYNTIM team. Stereograms (stereo images) of the Syntim team, INRIA. <http://perso.lpcp.fr/tarel.jean-philippe/syntim/paires.html>, 2004.
- [34] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [35] R. Oliveira, R. Ferreira, and J. P. Costeira. Reconstruction of isometrically deformable flat surfaces in 3d from multiple camera images. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 2009.