UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

Lucas Afonso Casanova de Oliveira Nogueira

A Unified Approach to Homography-Based Image Registration

Uma Abordagem Unificada para Registro de Imagens baseado em Homografia

CAMPINAS
2019

Lucas Afonso Casanova de Oliveira Nogueira

A Unified Approach to Homography-Based Image Registration

Uma Abordagem Unificada para Registro de Imagens baseado em Homografia

Dissertation presented to the School of Mechanical Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Mechanical Engineering, in the area of Mechatronics.

Dissertação apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Mecânica, na Área de Mecatrônica.

Orientador: Prof. Dr. Ely Carneiro de Paiva
Coorientador: Dr. Geraldo Figueiredo da Silveira Filho

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO LUCAS AFONSO CASANOVA DE OLIVEIRA NOGUEIRA, E ORIENTADA PELO PROF. DR. ELY CARNEIRO DE PAIVA.

ASSINATURA DO ORIENTADOR

CAMPINAS
2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

N689u

Nogueira, Lucas Afonso Casanova de Oliveira, 1991-
A unified approach to homography-based image registration / Lucas Afonso Casanova de Oliveira Nogueira. – Campinas, SP : [s.n.], 2019.

Orientador: Ely Carneiro de Paiva.
Coorientador: Geraldo Figueiredo da Silveira Filho.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Robótica. 2. Visão por computador. I. Paiva, Ely Carneiro de, 1965-. II. Silveira Filho, Geraldo Figueiredo da. III. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Uma abordagem unificada para registro de imagens baseado em homografia
Palavras-chave em inglês:
Robotics
Computer vision
Área de concentração: Mecatrônica
Titulação: Mestre em Engenharia Mecânica
Banca examinadora:
Ely Carneiro de Paiva [Orientador]
Niederauer Mastelari
Eric Rohmer
Data de defesa: 28-08-2019
Programa de Pós-Graduação: Engenharia Mecânica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-7220-2937
- Currículo Lattes do autor: http://lattes.cnpq.br/5568714185628488

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA MECÂNICA

COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DEPARTAMENTO DE SISTEMAS INTEGRADOS

DISSERTAÇÃO DE MESTRADO ACADÊMICO

A Unified Approach to Homography-Based Image Registration

Uma Abordagem Unificada para Registro de Imagens baseado em Homografia

Autor: Lucas Afonso Casanova de Oliveira Nogueira
Orientador: Prof. Dr. Ely Carneiro de Paiva
Coorientador: Dr. Geraldo Figueiredo da Silveira Filho

A banca examinadora composta pelos membros abaixo aprovou esta dissertação :

Prof. Dr. Ely Carneiro de Paiva
FEM - UNICAMP

Prof. Dr. Niederauer Mastelari
FEM - UNICAMP

Prof. Dr. Eric Rohmer
FEEC - UNICAMP

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

Campinas, 28 de Agosto de 2019

Dedication

To my mother, Ozeneide.
To my father, Thomaz.
To my soon-to-be wife, Gabi.

"Tudo vale a pena, se a alma não é pequena"
Fernando Pessoa

Acknowledgements

A worthy endeavour is rarely ever the result of one man's work alone. This dissertation is not an exception. It is the product of several people contributions, directly and indirectly. I thank them here for their support.

Thanks to my advisor, Professor Ely Carneiro de Paiva, head of the LEVE lab, for assuring perfect conditions to conduct my research. and for his confidence in my work. With his support, I was able to work on many different projects throughout this Masters, not only those in display in this dissertation.

Thanks to my co-advisor, Dr. Geraldo Silveira, from CTI Renato Archer, who proposed a very interesting research topic, guided me throughout the process and always tried to make me the best researcher I could be.

Thanks to Professor André Fioravanti, for the logistical support during Prof. Ely's post-doc period and important contributions in the qualification process.

Thanks to the members of the dissertation committee: Professors Niederauer Mastelari (FEM-UNICAMP) and Eric Rohmer (FEEC-UNICAMP) and Luiz Mirisola (ITA), for their time, attention and valuable inputs.

Thanks to Dr. Samuel Bueno, my undergraduate research advisor, who started me towards the path to a career in robotics. He opened many doors for me.

Thanks to my friends and fellow colleagues from the College of Mechanical Engineering: Randerson, Henrique, Rafael, Miguel, Apolo, Alexandre, Vinícius, Ricardo, César and Alan. Their contributions range from their technical experience to making graduate life much more enjoyable.

Huge thanks to my family: my parents Ozeneide and Thomaz; my sisters Mariana and Rebeca, my brothers-in-law Tomás and João. They are my safe haven and my biggest supporters. Also thanks to my extended Leite, Casanova and Nogueira families. They are a true blessing.

Thanks to Gabriela, the person who has been by my side during the best and worst of times. You made every day better.

Finally, I would like to thank Thematic Project InSAC, under grants (CAPES/CNPq 465755/2014-3, FAPESP 2014/50851-0). They provided the much-needed financial support for this project.

Resumo

A matriz de homografia define uma relação que surge quando um plano é observado a partir de dois pontos de vista diferentes. Estimar essa matriz é uma tarefa fundamental em muitas aplicações da robótica e pode ser formulada como um problema de registro de imagens. Ele é definido como uma busca pelos parâmetros que melhor definem uma transformação entre duas imagens. Métodos de estimação visual dividem-se em duas classe: baseados em primitivas ou intensidade. Por um lado, métodos baseados em primitivas tem um domínio de convergência grande, mas baixa precisão. Por outro lado, aqueles baseados em intensidade possuem propriedades complementares: um domínio de convergência menor, com alta precisão. Esse trabalho apresenta um método que possui o domínio de convergência dos métodos baseados em primitivas e a precisão daqueles baseados em intensidade. Ele unifica a informação geométrica e fotométrica advinda de cada classe em um único método de otimização, combinando suas funcões custo por meio de pesos cuidadosamente selecionados e executando rejeição de *outliers*. A otimização utiliza o método de Minimização de Segunda-ordem Eficiente porque ele permite uma aproximação de segunda ordem da série de Taylor sem a realização de cálculos custosos da matriz Hessiana. Adicionalmente, dois métodos são apresentados como passos intermediários: um baseado em intensidade capaz de tratar robustamente oclusões desconhecidas e outro baseado em primitivas que implementa o a Minimização de Segunda-ordem Eficiente. Todos os métodos desenvolvidos nessa dissertação são disponibilizados para uso pela comunidade acadêmica.

Palavras-chave: Robótica, Visão Computacional, Matriz de Homografia

Abstract

The homography matrix defines a relation that arises when a plane is observed from two different viewpoints. Estimating it is a fundamental task in many robotic applications and can be formulated as an image registration problem. It is defined as a search for the parameters that best define the transformation between a pair of images. Vision-based algorithms are generally divided in two classes: feature-based and intensity-based. Feature-based methods have a large convergence domain with relatively low precision. Intensity-based methods have complementary properties: a smaller convergence domain with better precision. This work presents a method that has the convergence domain of feature-based methods with the precision of the intensity-based ones. It unifies the photometric and geometric information from each classes into a single optimization method, combining their cost functions with carefully selected weights and performing outlier rejection. The optimization uses the Efficient Second-order Minimization (ESM) because it allows for a second-order approximation of the Taylor series without performing any computationally expensive Hessian calculations. Additionally, two methods are also presented as intermediate steps: a intensity-based one that is able to robustly handle unknown occlusions and a feature-based method that implements the same ESM optimization framework. All the methods developed in this work are made available for use by the research community.

Keywords: Homography estimation, Computer vision, Robotics

List of Figures

List of Tables

Glossary

LIDAR Light Detection and Ranging

IMU Inertial Measurement Unit

UAV Unmanned Aerial Vehicles

AR Augmented Reality

ESM Efficient Second-Order Minimization

IR Image Registration

SLAM Simultaneous Localization and Mapping

DLT Direct Linear Transformation

SSD Sum of Squared Differences

ZNCC Zero-mean Normalized Cross Correlation

SW Sliding Window

SIFT Scale-Invariant Feature Transform

SURF Speeded Up Robust Features

ORB Oriented Brief feature detector

FAST Features from Accelerated Segment Test

MAD Median Absolute Deviation

RMSD Root-Mean Squared Error

IB Intensity-Based visual estimation algorithms

FB Feature-Based visual estimation algorithms

OpenCV Image processing library

ROS Robot Operating System

RANSAC Random Sample and Consensus

Contents

1 INTRODUCTION

1.1 Motivation

In the last decade, the number of real-world applications where robots can be useful has exploded. One of the many factors that help explain this phenomenon is the continuous miniaturization of sensors and computing power, coupled with a decrease in their price. For instance, it is common knowledge that a single modern smartphone surpasses the processing power that was necessary to take mankind to the moon, for a fraction of the cost. Additionally, smartphones now come with a multitude of sensors such as inertial measurement units, infrared lasers, and cameras. In particular, the miniaturization of camera sensors has made it almost ubiquitous in our society.

Among all the sensors that are customarily used in robotics, cameras stand out because of their high information density with relatively low price points. They can provide important information to a robotic system, and are used in a variety of applications. Some examples are: object recognition, 3D reconstruction and motion estimation. In some scenarios, a camera can replace other common and more expensive sensors, such as light detection and ranging (LIDAR) and inertial measurement units (IMU). For instance, some self-driving car companies have dispensed with the use of LIDARs in favor of a camera-oriented approach (HAWKINS, 2018).

As it moves around the environment, a robot's camera produces a sequence of images. This sequence often portraits the same scene structure for an extended period of time, from different viewpoints. The need arises for these images to be related to each other, in order to understand the relative motion between scene and robot. The homography matrix is an important way of describing this motion when a plane is observed from two different viewpoints.

The plane-induced homography encodes the scene structure and the camera motion, and has been used in a variety of vision-based applications, such as image mosaicking (FAUGERAS; LUONG; PAPADOPOULO, 2001), visual servoing (BENHIMANE; MALIS, 2007), and visual tracking (SILVEIRA; MALIS, 2010). Homographies arise in urban environments, where the Manhattan world assumption (COUGHLAN; YUILLE, 1999) holds. In this case, billboards and traffic signs may be observed by a self-driving car. Another instance is in the industrial robotics area, where packages and objects that need to be manipulated by an robotic arm have one or more planar surfaces (NEUBERGER et al., 2019). Finally, they are also present in the field of Unmanned Aerial Vehicles (UAV), specially when the ground is observed from a distance in such a way that it is perceived as a plane (PLINVAL et al., 2011).

The field of augmented reality (AR) also makes extensive use of the homography matrix. It consists of overlaying virtual data onto a video sequence in a manner that is con-

sistent with the real world data. To accomplish this task, planar object tracking(LIANG et al., 2018; WU et al., 2019) is commonly used and usually involves an homography estimation step (VALOGNES; DASTJERDI; AMER, 2019). In particular, the methods in this work could enable a markerless AR system.

1.2 Image Registration

Figure 1.1: Inputs to the Image Registration Problem. Left: Reference Image and Template; Right: Current Image

The homography estimation task can be formulated as an image registration problem. This problem is defined as a search for the parameters that best define the transformation between corresponding pixels in a pair of images. The first and second images are typically referred to as the reference and current image, respectively, as shown in Fig. 1.1. Initially, a region of interest in the reference image is selected, which creates a reference template. In these terms, the image registration problem tries to find where in the current image is the reference template. Solutions to this problem involve the definition of four important characteristics (BROWN, 1992): the transformations models; the information space; the similarity measures; the search strategy; and optionally of a robust method. In particular, a homography estimation task can be seen as an image registration problem where the transformation model uses a homography to explain the geometric part of the transformation. In addition, the model can contain other components, such as a photometric one.

Regarding the information space, the vast majority of vision-based algorithms use a feature-based approach. In this approach, first an extraction algorithm searches each image for some form of geometric primitive (e.g., point, lines, ellipses, etc) and selects the best candidates, which are referred to as features. These features are then encoded as a feature vector by a feature description algorithm. Then, a matching algorithm is responsible for finding correspondences between features in different images. After these correspondences are found, the actual estimation takes place. The convergence domain of these algorithms is usually very high, because the entire image is searched for features. However, both the extraction and matching steps are error-prone and can create correspondence outliers that affect the quality of the estimation, leading to a decrease in precision. Additionally, it can be noted that the intermediate steps throw away useful information.

In contrast, intensity-based methods have no extraction, description and matching

steps. These methods are also referred to as direct methods, because they exploit the pixel intensity values directly. This allows the estimation algorithm to work with more information than feature-based methods. In turn, it leads to more precise estimation results. It also eliminates the errors from bad correspondences. In a way, the extraction and matching steps are done implicitly by the estimation process. However, one drawback from direct methods is that they require a small interframe displacement, i.e. sufficient overlap between two images. Additionally, unknown occlusions in the image have the potential to lead the estimation to completely erroneous result, if not treated correctly. Table 2.1 summarises the differences between these two classes.

Table 1.1: Comparison of Feature-Based and Intensity-based methods

Property	Feature-Based	Intensity-Based
Intermediate Steps	Feature extraction and matching	None
Outliers	Bad Correspondences	Occlusions
Information Space	Feature Coordinates	Pixel Intensities
Precision	Lower	Higher
Convergence Basin	Large	Small

The algorithms presented in this work use multidimensional optimization methods as the main search strategy for the image registration problem. When formulated as such, an initial solution is iteratively refined using a nonlinear optimization method. Specifically, the algorithms presented here are derived from the Efficient Second Order Minimization (ESM) algorithm. As will be explained, this algorithm is particularly suited to this application domain, and its advantages include both a higher convergence rate and a larger convergence domain than standard iterative methods. It allows for a second-order approximation of the Taylor series while dispensing with the need to perform any computationally expensive Hessian calculations.

The use of the ESM framework has shown great results for intensity-based methods. However, its has not been applied in a feature-based setting. As discussed, the two classes of estimation methods have complementary strengths. This observation naturally leads for the search of a hybrid method that has the precision of intensity-based algorithms with the larger convergence domain of feature-based ones. This work presents such a method. As intermediate steps, two methods are developed: a intensity-based method that is able to robustly handle unknown occlusions and a feature-based method that implements the same ESM optimization framework as the intensity-based one. Finally, we present the hybrid method that unifies the approaches.

The proposed method estimates a homography that relates two images using photometric (IB) and geometric (FB) information. It unifies these categories under a single non-linear optimization. This is accomplishing by considering a unified cost function with carefully selected weights for each component. A coarse-to-fine scheme is used to achieve better convergence properties. Additionally, this unified result is used iteratively to reject outliers in the feature correspondence set.

All the methods developed in this thesis are made available as ready-to-use ROS packages and a C++ library. These implementations make it possible to easily deploy the estimation algorithms in a variety of real-time applications. Two type of applications will also be presented in this dissertation. First is the visual tracking application, which is

possibly the most direct use of the image registration algorithms. The second application is visual servoing, which consists of controlling a robot using feedback provided by a camera sensor.

1.3 Related Works

This work focuses on the estimation of the homography matrix, and solves this problem by formulating it as an image registration (IR) problem. It consists of finding the transformation parameters that best align different images. Most modern IR algorithms can be traced back to the Lucas-Kanade (LK) algorithm (LUCAS; KANADE, 1981). According to (BROWN, 1992), an IR algorithm can be characterized by analysing four components: information space; transformation model; similarity measure; and search strategy.

The information space considers the inputs of the algorithm. In this category, there are two main classes of approaches. The first one is feature-based (FB). It requires the extraction and association of geometric primitives in different images before the actual estimation can occur (HARTLEY; Andrew ZISSERMAN, 2003; HUA et al., 2018). These primitives can be points (HARRIS; STEPHENS, et al., 1988), lines (SMITH; REID; DAVISON, 2006), or even more complex structures. The second approach is intensity-based (IB). It uses the photometric information, i.e. the intensity of the pixels. It simultaneously solves for the estimation problem and pixel correspondences, without any intermediate steps (IRANI; ANANDAN, 1999; BENHIMANE; MALIS, 2007; ENGEL; SCHÖPS; CREMERS, 2014).

The transformation model dictates which parameters are estimated. For example, the original LK algorithm only estimated translations in the image space. This was later extended to more sophisticated warp functions (BERGEN et al., 1992). Simultaneous Localization and Mapping (SLAM) algorithms commonly use IR to estimate a 3D pose (J. ZHANG; S. SINGH, 2015). The homography matrix is often used as a transformation model when dealing with predominantly planar regions of interest (SILVEIRA; MALIS; RIVES, 2008; MUR-ARTAL; MONTIEL; TARDOS, 2015; LIU; G. ZHANG; BAO, 2016; DETONE; MALISIEWICZ; RABINOVICH, 2016), and is used in all the algorithms proposed in this dissertation. Additionally, illumination parameters can be considered as a component of the transformation model (SILVEIRA; MALIS, 2007; SILVEIRA, 2014; BARTOLI, 2008).

The quality of the IR estimation can be defined by a similarity measure. When an optimization method is used, this measure is often used as a cost function. Some algorithms use the Sum of Squared Differences (SSD) (LUCAS; KANADE, 1981; SILVEIRA; MALIS, 2010). Other possibilities include correlation-based metrics (EVANGELIDIS; PSARAKIS, 2008; FONSECA; MANJUNATH, 1996; YAN et al., 2014) and mutual information (VIOLA; WELLS III, 1997).

The last component of IR algorithms is the search strategy. Most real-time applications use a multidimensional optimization approach, based on Gradient Descent search. They use the first and second derivatives of the similarity measures with respect to the transformation parameters. The ESM algorithm (BENHIMANE; MALIS, 2004) is one

example. Alternative optimization approaches include Gauss-Newton and Levenberg-Marquardt (BAKER; MATTHEWS, 2004). These techniques are most suited to applications with small interframe displacements. Some algorithms overcome this issue using sampling-based methods such as RANSAC (FISCHLER; BOLLES, 1981) to cover a larger portion of the search space. It is also possible to use additional information about the robot to improve the estimation. For instance, (HUA et al., 2018) uses partial velocity information in a temporal filter to narrow the search space and make the estimation more robust. A thorough review and comparison of image registration algorithms can be found in (ZITOVA; FLUSSER, 2003) and (A. K. SINGH, 2017).

This work develops an approach that merges two different information sources into a single optimization framework, by creating a unified cost function. Namely, it merges feature-based and intensity-based informations. Some algorithms in the literature have proposed similar hybrid methods. For instance, (GEORGEL; BENHIMANE; NAVAB, 2008) merges point correspondences with photometric information, but estimates a 3D pose, instead of the homography matrix. In (MEILLAND; COMPORT; RIVES, 2011), two techniques are also unified: a model-based tracking that suffers when dealing with large illumination changes; and a visual odometry that compares consecutive image frames. This technique is better at handling illumination changes, but accumulates drifts. The techniques are merged to produce an algorithm that handles large illuminations changes without accumulating drift. In this method, both techniques use intensity-based information and a 3D pose is estimated, which contrasts to the method proposed in this work. The method proposed in (MORENCY; DARRELL, 2002) unifies Normal Flow Constraint (intensity-based) and Iterative Closest Point (feature-based) methods into a single optimization framework, similar to the methods in this work. However, it estimates a 3D pose and consider stereo cameras whilst we consider monocular ones. Finally, (YAN et al., 2014) uses a Maximum Likelihood approach to create a unified IB/FB cost function. The enhanced cross-correlation is used as the IB similarity measure and it uses RANSAC for robust optimization.

Most recently, deep learning algorithms have been proposed to tackle the homography estimation problem. (DETONE; MALISIEWICZ; RABINOVICH, 2016) propose an end-to-end supervised learning approach with convolutional neural networks. The algorithm requires an offline learning step that takes as input image pairs labeled with ground truth. The process of obtaining such datasets is either costly because it requires manual labeling of images, or it is restricted to synthetic datasets. In turn, (NGUYEN et al., 2017) propose the use of a intensity-based metric as a loss function in order to enable a unsupervised learning approach. In (RANFTL; KOLTUN, 2018), an iteratively reweighted least squares problem is used to robustly estimate a fundamental matrix, with the robusts weights estimated using deep networks. These methods, as deep learning methods in general, require an extensive offline training step and the use of GPUs, while those presented in this work require neither.

1.4 Objectives

In summary, the objective of this work is the development of a vision-based estimation algorithm that:

- Robustly estimates the homography matrix between two images.
- Unifies the intensity-based and feature-based approaches under a single optimization framework with a unified cost function and a coarse-to-fine strategy.
- Is computationally efficient enough to be applied in real-time applications, such as visual tracking and visual servoing.

Additionally, the algorithms are made available to the research community as a C++ library and ROS (QUIGLEY et al., 2009) package.

2 THEORETICAL BACKGROUND

This chapter introduces the basic theory concepts neessary to understand the proposed methods. In Section 2.1, the special properties that arise from observing the same scene in multiple images are presented. Then, Section 2.2 presents an overview of the Efficient Second-order Minimization algorithm. Finally, Section 2.3 details each of the individual components that are part of an image registration algorithm and how our algorithms approach each of them.

2.1 Two-view Geometry

This section aims to present the mathematical framework that explains the spatial relations between corresponding scene points in two different images and their 3D world coordinates. For this purpose, it presents an overview of projective geometry theory and the pin-hole camera model. Additionally, the homography matrix properties and some of its practical applications are also presented.

2.1.1 Projective geometry

Projective Geometry is an extension of Euclidian Geometry that makes it possible to model not only rotation and translations, but also the peculiar phenomena that occur when the 3D world is projected onto a 2D image, such as:

- Any set of parallel lines in the 3D world converge to a single point when projected to a 2D image. This point is referred to as the *vanishing point*;
- Points that are very far away in the 3D world, also known as points *at infinity*, are projected to the same point even as the observer moves around locally. This explains to how the mooon is perceived as in the same place as a person moves around in a city.

This dissertation applies the convention that a geometric object is in normal type and its coordinate vector is in bold type, such as point p and its coordinate vector $\mathbf{p}$.

Consider a point $p \in \mathbb{R}^2$ that belongs to a plane. Its Euclidian coordinates are $[u, v]^\top$, and its homogeneous projective coordinates are obtained by adding a third element equal to 1 at the end, such that

$$\mathbf{p} = [u, v, 1]^\top \in \mathbb{P}^2, \tag{2.1}$$

with the property that all triples that differ only by a scale factor represent the same image point:

$$\mathbf{p} \equiv \lambda\mathbf{p} = [\lambda u, \lambda v, \lambda]^\top, \forall\lambda \neq 0. \tag{2.2}$$

In order to convert from projective coordinates to Euclidian coordinates, it suffices to divide every element in the representation by the last one and drop the last element.

$$\mathbf{p} = [\lambda u, \lambda v, \lambda]^\top \in \mathbb{P}^2 \rightarrow [u, v]^\top \in \mathbb{R}^2 \tag{2.3}$$

Besides the usual points from the Euclidian space, the Projective space also contains the so-called points *at infinity*. These are points in which the last projective coordinate is equal to zero. They have no Euclidian equivalent, as can be observed from (2.3). However, they are regular points in Projective space.

In the Projective space, a line has the same representation as a point. Given a point p with projective coordinates $\mathbf{p} = [x, y, z]$ that belongs to a line l, the following relation holds:

$$\mathbf{l}^\top\mathbf{p} = \mathbf{p}^\top\mathbf{l} = ax + by + cz = 0, \tag{2.4}$$

where $\mathbf{l} = [a, b, c]^\top$ is the projective representation of line l. This property induces a *duality* between lines and points in the projective space that can be observed by the following properties.

- If a line l contains points p and p^*, then their projective representations satisfy:

$$\mathbf{l} \equiv \mathbf{p} \times \mathbf{p}^* \tag{2.5}$$

- If a point p is at the intersection of two lines l and l', then their projective representations satisfy:

$$\mathbf{p} \equiv \mathbf{l} \times \mathbf{l}' \tag{2.6}$$

From the equations above, it is possible to observe that the cross product is a specially important operator when dealing with the projective space. Its advantage is that it can be written as a linear operator, replacing it by the following matrix multiplication.

$$\mathbf{v} \times \mathbf{x} = [\mathbf{v}]_\times\mathbf{x}, \tag{2.7}$$

with $\mathbf{v} = [v_1, v_2, v_3]$ and $\mathbf{x} = [x_1, x_2, x_3]$ and $[\mathbf{v}]_\times$ being the skew-simmetric matrix such as:

$$[\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}. \tag{2.8}$$

The concept of line *at infinity* is also present in the projective space. It is composed of all points at infinity and has the representation:

$$\mathbf{l}_\infty = [0, 0, 1]^\top. \tag{2.9}$$

2.1.2 Pinhole camera model

The relation between the coordinates of a point in the 3D world and its projection in a 2D image can be explained with the pinhole camera model, as shown in Fig. 2.1. The basic components of this model are:

- A camera center C, which also acts as the origin for the *camera reference frame*;
- A retinal plane R that is parallel to the xy plane of the camera frame and located at a distance f from C. Therefore, the z axis crosses the retinal plane at the *principal point* c, which has Euclidian coordinates $\mathbf{c} = [0, 0, f]$. f is called the focal length of the camera. Without loss of generality, it is chosen as the unit.

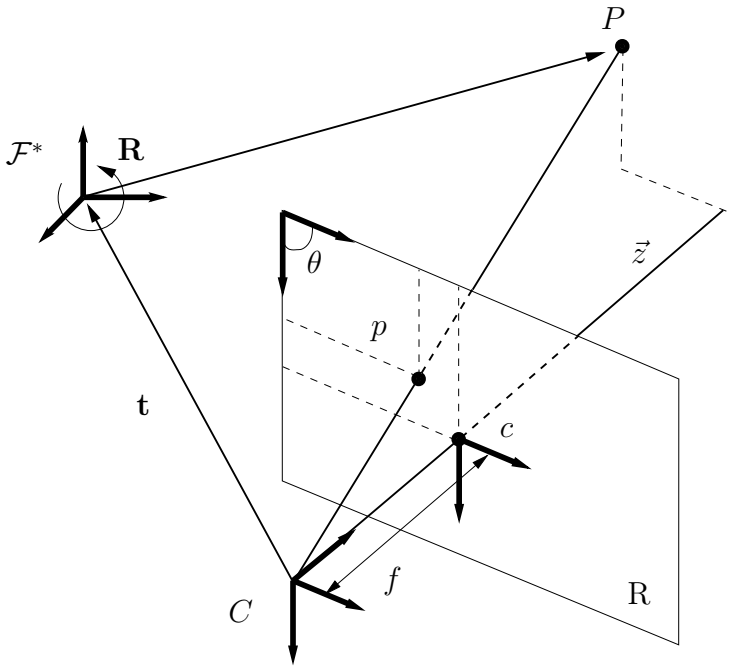


Figure 2.1: The pinhole camera model

Given a 3D point P with coordinates $[X, Y, Z]^\top \in \mathbb{R}^3$, its projection to the retinal plane p is $[u, v] \in \mathbb{R}^2$ and their relation is obtained with the following equations:

$$u = f\frac{X}{Z}, v = f\frac{Y}{Z}; \tag{2.10}$$

It can be observed that (2.10) and (2.3) are quite similar. From this similarity stems the applicability of projective geometry to the study of computer vision. Also, it is clear that the projection loses information, since there are multiple points in a line passing through the camera center that map to the same point in the image. By consequence, it is impossible to recover depth information using only the *geometry* of a single image. Humans are able to induce depth information from a single image by using semantic reasoning *alongside* geometrical relations.

The nonlinear equations (2.10) can be converted to linear using tools from projective geometry. Consider the projective coordinates of 3D point P:

$$\mathbf{P} = [X, Y, Z, 1]^\top \in \mathbb{P}^3, \tag{2.11}$$

and the following relation holds:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathcal{P}_0 \mathbf{P}, \tag{2.12}$$

where the matrix $\mathcal{P}_0$ is called the projection matrix. When the pinhole camera model is perfect, it has this simple structure. However, for real-world cameras, its components depend on physical properties of the device and has the general form:

$$\mathcal{P} \equiv \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathcal{P}_0 \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{A}[\mathbf{R}\,\mathbf{t}], \tag{2.13}$$

where its components can be split in two groups:

1. *Intrinsic parameters* are related to the construction process of the camera. Its components are:
   - α_u and α_v represent the focal length expressed in pixel units. They are related to the focal distance f and the width and height of the image in pixels. Their ratio is called aspect ratio;
   - γ represents the skew between the axes of the image such that $\gamma \propto \tan(\theta) \approx 0$, where θ is the angle between the image axes. In modern cameras its value is very close to zero, which means the $\vec{u}$ and $\vec{v}$ are nearly perfectly perpendicular;
   - u_0 and v_0 represent the coordinates of the principal point. Because pixels are usually counted from the corner of the image, they are usually not equal to zero. Instead, they are roughly half the image resolution.
2. *Extrinsic parameters* are related to the camera's relative pose w.r.t. to the world reference frame $\mathcal{F}^*$. Its components are:
   - $\mathbf{R} \in \mathbb{SO}(3)$ represents the rotation of the camera frame w.r.t to the world frame;
   - $\mathbf{t} \in \mathbb{R}^3$ represents the translation of camera frame w.r.t to the world frame.

The process of obtaining the intrinsic parameters of a given camera sensors is commonly called *intrinsic calibration*. It is a common challenge in computer vision and required whenever Euclidean information has to be recovered from the images. The visual estimation methods that are the core of this master thesis require no calibration whatsoever, as will be shown. The visual servoing algorithms that use these estimation results only need coarse calibration.

Finally, it is possible to define the image function as the mapping between discrete pixel coordinates and its intensity.

$$\mathcal{I}\colon \begin{array}{rl} \Omega \subset \mathbb{Z}^2 & \to \mathbb{R}_+ \\ \mathbf{p} & \mapsto \mathcal{I}(\mathbf{p}), \end{array} \tag{2.14}$$

where $\Omega = [0, L-1] \times [0, C-1]$ defines the size of the image grid, commonly known as the image resolution. The intensity can be thought of as the color of the 3D point P that is projected onto the point p in the retinal plane. In this work, only black-and-white images will be considered, so a single intensity value is produced by the image function. Colored images, by contrast, typically produce three intensity values (red, green and blue).

2.1.3 Homographies

Consider a point $P \in \mathbb{P}^3$ that lies in a plane Π and consider that the world reference frame is attached to this plane such that its xy plane coincides with Π. Therefore, in this reference frame, P has the following coordinates:

$$\mathbf{P} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \tag{2.15}$$

The projection of P on the retinal plane is obtained with the aid of (2.12):

$$\mathbf{p} = \begin{bmatrix} u\lambda \\ v\lambda \\ \lambda \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{14} \\ P_{21} & P_{22} & P_{24} \\ P_{31} & P_{32} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \tag{2.16}$$

which can be further simplified to:

$$\mathbf{p} = \begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{2.17}$$

The (3×3)-matrix $\mathbf{H}$ encodes a transformation between two $\mathbb{P}^2$ projective spaces, that are the planes themselves. $\mathbf{H}$ has eight degrees of freedom and requires a minimum of 4 correspondences between points in each space to be calculated.

Now, consider the case where the same plane Π is observed from two different viewpoints with the same camera, as shown in Fig. 2.2. In this case, a point P in this plane will have a projection p in the first image $\mathcal{I}$ and a projection p^* in the second image $\mathcal{I}^*$. $\mathcal{I}$ and $\mathcal{I}^*$ are commonly referred throughout this work to as current and reference image, respectively. The following relation holds:

$$\mathcal{P}\mathbf{P} \propto \mathbf{p} \tag{2.18}$$

$$\mathcal{P}^*\mathbf{P} \propto \mathbf{p}^* \tag{2.19}$$

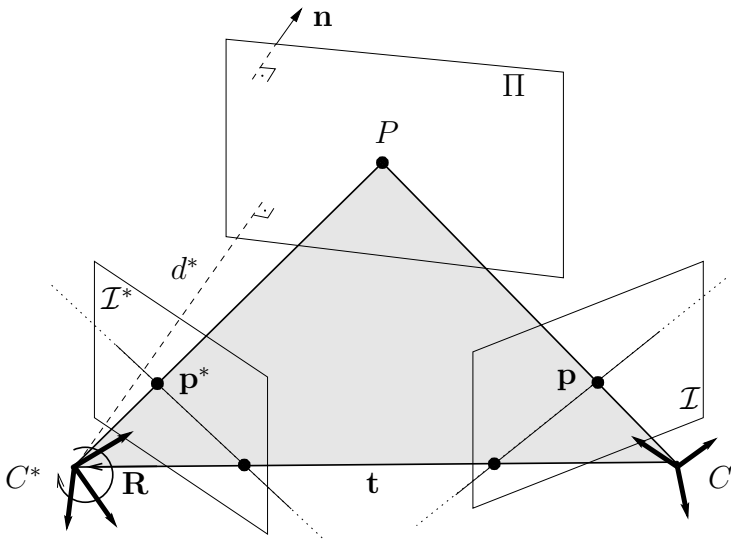


Figure 2.2: Epipolar geometry.

where $\mathcal{P}$ and $\mathcal{P}^*$ are the projection matrices for the first and second image, respectively. Without loss of generality, it can be considered that both images were captured with the same camera sensor. In this case, the projection matrices will differ only by their extrinsic parameters, i.e. rotation and translation of the camera w.r.t to the world frame.

It has been established that there is a homography between plane Π and the retinal plane of the first camera. Likewise, there is *another* homography between Π and the retinal plane of the second camera. By composition, it is clear that there should also be a homography between the two retinal planes. This is called a *planar homography* and is said to be *induced by* the plane Π. This relation can be written as:

$$\mathbf{p} \propto \mathbf{H}\,\mathbf{p}^*. \tag{2.20}$$

As before, four point correspondences are necessary to fully define a homography between the two retinal planes. A traditional method for calculating it from these correspondence set is the Direct Linear Transformation (DLT) as explained in Chapter 4 of (HARTLEY; Andrew ZISSERMAN, 2003). Additionally, it can be shown that the same relation also holds if the camera undergoes a pure rotational movement. This phenomenon is exploited in smartphone applications where a panoramic image is constructed by the user carefully moving the device.

2.1.4 Examples of operations with homographies

This section presents some of the operations that can be achieved using homographies. The intent here is to develop intuition about the structure of the homography and how images are transformed by it. First, consider (2.20). It gives rise to an operation called warping.

Formally, the warping operator is defined as follows:

$$\mathbf{w}: \begin{array}{ll} \mathbb{SL}(3) \times \mathbb{P}^2 & \to \mathbb{P}^2 \\ (\mathbf{H}, \mathbf{p}^*) & \mapsto \mathbf{p} = \mathbf{w}(\mathbf{H}, \mathbf{p}^*) = \left[\frac{h_{11}u^*+h_{12}v^*+h_{13}}{h_{31}u^*+h_{32}v^*+h_{33}}, \frac{h_{21}u^*+h_{22}v^*+h_{23}}{h_{31}u^*+h_{32}v^*+h_{33}}, 1 \right]^\top \end{array} \tag{2.21}$$

where $\mathbf{H} \in \mathbb{SL}(3)$ is the projective homography matrix, $\{h_{ij}\}$ comprise its elements, $\mathbf{p}^* = [u^*, v^*, 1]^\top \in \mathbb{P}^2$ is the projective coordinates of the source pixel.

It is said that point p is the result of the warping of point p^* with homography $\mathbf{H}$. For example, let point $\mathbf{p}^* = [100, 100, 1]^\top$ and the homography

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 20 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

Then, by applying the warping operation to p^*, the warped point p is obtained, and its converse is also possible:

$$\mathbf{p} = [110, 120, 1]^\top = \mathbf{w}(\mathbf{H}, \mathbf{p}^*), \tag{2.23}$$

$$\mathbf{p}^* = [100, 100, 1]^\top = \mathbf{w}(\mathbf{H}^{-1}, \mathbf{p}). \tag{2.24}$$

The warping operation first multiplies the projective coordinates $\mathbf{p}^*$ by the homography $\mathbf{H}$ and then divides every element of this new vector by its third element, always maintaining the last element equal to 1. For instance, applying the warping operator with the homography in (2.22) to the following set of four points that form a square, we obtain:

$$\begin{aligned} \mathbf{p}_1^* &= [0, 0, 1]^\top && \to [10, 20, 1]^\top; \\ \mathbf{p}_2^* &= [0, 100, 1]^\top && \to [10, 120, 1]^\top; \\ \mathbf{p}_3^* &= [100, 0, 1]^\top && \to [110, 20, 1]^\top; \\ \mathbf{p}_4^* &= [100, 100, 1]^\top && \to [110, 20, 1]^\top. \end{aligned}$$

It is easy to verify that this simple homography parametrizes a translation operation. Now, we will consider how a homography can be applied to a entire image $\mathcal{I}(\mathbf{p})$, instead of just a single point. Warping every pixel p_i in the domain of $\mathcal{I}$ maps the pixels to its corresponding location in the warped image $\mathcal{I}^*$. For illustration purposes, consider the following homography that parametrizes a translation of 1 pixel in the $\vec{u}$ direction and its inverse:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \mathbf{H}^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.25}$$

Figure 2.3 shows the warping of a very simple image of resolution 3×3. Starting from the first pixel p in image $\mathcal{I}$, the corresponding location p^* on the warped image is obtained by applying the warping operation to the coordinates of that pixel with the inverted homography. The first pixel correspondences are marked with a green circle.

After obtaining this correspondence, the destination pixel receives the intensity value of the origin pixel. The following equation summarizes the attribution.

$$\mathcal{I}^*(\mathbf{w}(\mathbf{H}^{-1}, \mathbf{p})) \leftarrow \mathcal{I}(\mathbf{p}). \tag{2.26}$$

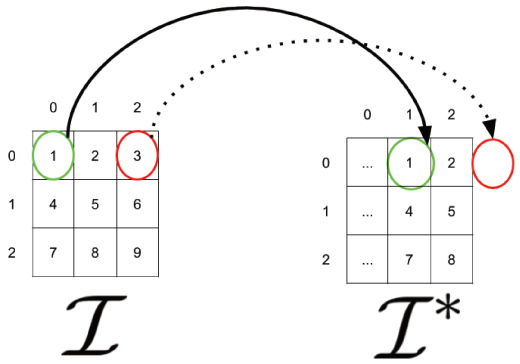


Figure 2.3: Image Warping Example

Note that, in Fig. 2.3, the pixel highlighted in red is warped to an out-of-bounds location in the destination image. In this case, the values are ignored. Conversely, the points in the first column of $\mathcal{I}^*$ do not "receive" any values in this process. By convention, they should receive a zero value.

These cases indicate two problems with this warping algorithm, called *forward warping* because it starts with the original image coordinates. The first problem is that some pixels get mapped to out-of-bounds pixels in the warped image, and therefore unnecessary calculations are made. The second one is worse. By using forward warping, there is no guarantee that all pixels in the destination image are filled with values. If the homography also encodes scaling operations, gaps between pixels will appear.

Because of these problems, the actual implementation of the warping function works backwards. Pixels of the destination image are iterated to obtain the corresponding pixels in the source image. If the source location does not lies perfectly over a single pixel, but instead *between* pixels, then an interpolation is performed using the neighboring pixels to obtain the pixel value at the sampled location. The following attribution summarizes this algorithm, called "inverse warping" because it starts with the warped image coordinates.

$$\mathcal{I}^*(\mathbf{p}^*) \leftarrow \mathcal{I}(\mathbf{w}(\mathbf{H}, \mathbf{p}^*)). \tag{2.27}$$

Using this equation, it is possible to state that $\mathcal{I}^*$ is the result of warping $\mathcal{I}$ with $\mathbf{H}$.

In the previous example, the warped image had the same size as the original image. In some cases, however, it is useful to use the warping operation to generate a smaller image. This smaller image is usually referred to as a *template*. In Fig. 2.4, the original image has a resolution of 512×512 pixels. The warped image has a resolution of 200×200.

The white square in the original image indicates the area that was selected for the template. The square corner coordinates are obtained by warping the coordinates of the extreme points of the destination image. It is important to note that the size of the selected area is not encoded in the homography matrix itself, but depends on the warped image size. Also notice that while we consider that the original image is warped by $\mathbf{H}$ onto the template, the pixels of the template are the ones that are effectively warped by $\mathbf{H}$ onto their corresponding coordinates in the original image.

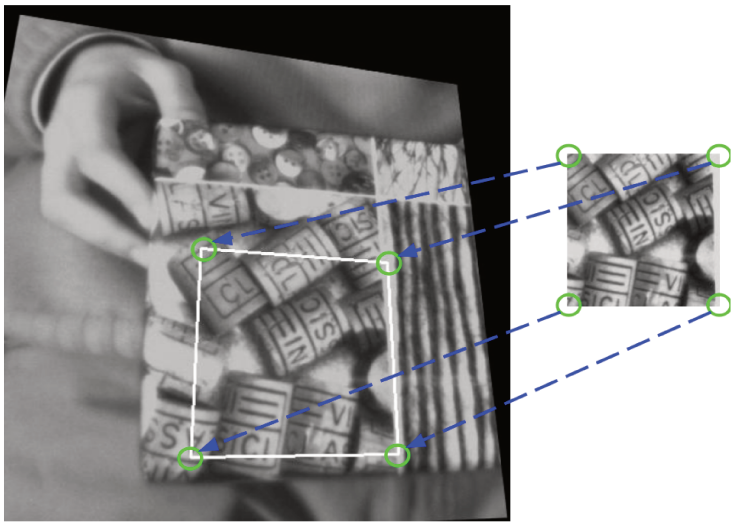


Figure 2.4: Extracting a template from an image.

Another use for warping operations is that of scaling an image, i.e. changing its resolution. Consider the following homography:

$$\mathbf{H} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.28}$$

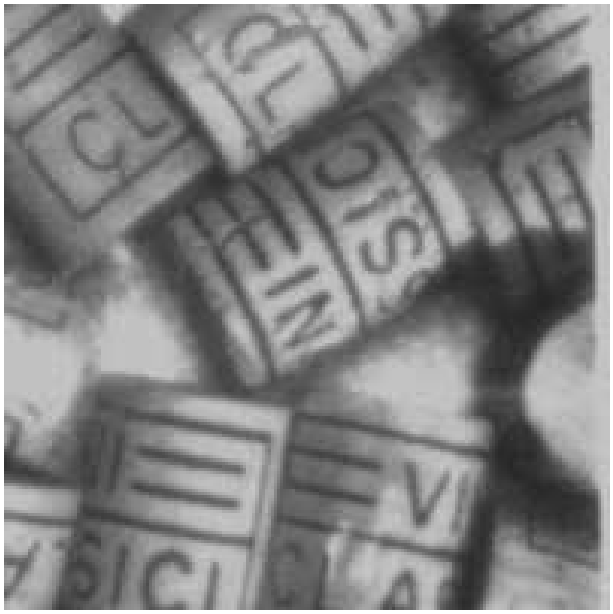
If the warped image has a resolution of $[100 \times 100]$, for instance, the following set of point correspondences are obtained for its corners.

$$[(0,0),(0,99),(99,0),(99,99)] \rightarrow [(0,0),(0,198),(198,0),(198,198)] \tag{2.29}$$

From this, it is clear that the area in the original image is four times bigger than in the warped image. This is an scale down operation, or downsizing. An example of this warping can be seen in Fig. 2.5. The downsizing has been applied to the template obtained in the previous section. Scale operations are specially important in this work because they enable the use of multiresolution pyramids.

2.2 Efficient Second-order Minimization

An important part of this work is the parametric estimation that is involved in the image registration problem. This is solved by formulating it as a nonlinear least squares optimization. In turn, this type of optimization is solved via numerical iterative methods. A traditional example is the Newton-Raphson method. Here, however, another iterative method is used, the Efficient Second-order Minimization (ESM) algorithm (MALIS, 2008). Its advantages when registering images include both a higher convergence rate and a larger convergence domain than standard iterative methods, with the absence of costly Hessian calculations.



(a) The original image.

(b) A scaled down image.

Figure 2.5: Scaling down an image with warping.

The goal of this section is to present the ESM method. In order to understand the method, it is first presented as a root-finding algorithm. Later, it is shown how it can be reformulated as a method to solve nonlinear least squares problems.

2.2.1 Root-finding algorithms

This section considers the problem of finding the root of an equation. The goal is to find the solution x^* that satisfies:

$$f(x^*) = 0. \tag{2.30}$$

Iterative methods solve this problem starting from an initial approximation x_0. Then, it seeks to find a sequence of approximations $\widehat{x}_n$ such that:

$$\lim_{n\to\infty} \widehat{x}_n = x^*. \tag{2.31}$$

The next approximation $\widehat{x}_{n+1}$ is obtained by calculating an increment $\widetilde{x}_n$ that is added to the current one $\widehat{x}_n$:

$$\widehat{x}_{n+1} = \widehat{x}_n + \widetilde{x}_n \tag{2.32}$$

It is important to note that at the true solution x^*, the following equations hold:

$$f(x^*) = 0 \tag{2.33}$$

$$f'(x^*) \neq 0 \tag{2.34}$$

The first root-finding method presented is Newton-Raphson's algorithm. It is used here as a reference to which the ESM can be compared. Each algorithm mainly dictates

how the increment $\widetilde{x}_n$ is calculated. The idea is to find one that brings the current approximation $\widehat{x}_n$ closer to the ideal solution x^*:

$$\widehat{x}_n + \widetilde{x}_n = x^*. \tag{2.35}$$

Applying function f to both sides of (2.35), using (2.30) and the Taylor series expansion up to first order terms, it becomes:

$$f(\widehat{x}_n + \widetilde{x}_n) = f(x^*) = 0 \tag{2.36}$$

$$f(\widehat{x}_n) + \widetilde{x}_n f'(\widehat{x}_n) \approx 0 \tag{2.37}$$

and an expression for the increment is obtained:

$$\widetilde{x}_n \approx -\frac{f(\widehat{x}_n)}{f'(\widehat{x}_n)} \tag{2.38}$$

From this expression, it is clear that, in the Newton-Raphson method, each increment is calculated using only information about the function f at the current evaluation point $\widehat{x}_n$. This is a key difference to the ESM method, as will be shown. The ESM considers the second-order terms of the Taylor expansion. Therefore, (2.37) is replaced by:

$$f(\widehat{x}_n) + \widetilde{x}_n f'(\widehat{x}_n) + \frac{\widetilde{x}_n^2}{2} f''(\widehat{x}_n) \approx 0 \tag{2.39}$$

Now, consider the first order Taylor expansion of the first derivative of f.

$$f'(\widehat{x}_n + \widetilde{x}_n) \approx f'(\widehat{x}_n) + \widetilde{x}_n f''(\widehat{x}_n) \tag{2.40}$$

Rearranging the terms:

$$\widetilde{x}_n f''(\widehat{x}_n) \approx f'(\widehat{x}_n + \widetilde{x}_n) - f'(\widehat{x}_n) \tag{2.41}$$

Replacing in (2.39)):

$$f(\widehat{x}_n) + \widetilde{x}_n f'(\widehat{x}_n) + \frac{\widetilde{x}_n}{2}[f'(\widehat{x}_n + \widetilde{x}_n) - f'(\widehat{x}_n)] \approx 0 \tag{2.42}$$

$$f(\widehat{x}_n) + \frac{\widetilde{x}_n}{2}[f'(\widehat{x}_n + \widetilde{x}_n) + f'(\widehat{x}_n)] \approx 0 \tag{2.43}$$

And now, the ESM increment is obtained:

$$\widetilde{x}_n \approx -\frac{2f(\widehat{x}_n)}{f'(\widehat{x}_n) + f'(\widehat{x}_n + \widetilde{x}_n)} \tag{2.44}$$

$$\approx -\frac{2f(\widehat{x}_n)}{f'(\widehat{x}_n) + f'(x^*)} \tag{2.45}$$

Note that the presence of the $f'(x^*)$ term implies that the ESM increment depends on the ability of calculating the first-order derivative of function f at the root solution point x^* where $f(x^*) = 0$. Also, since the formulation stems from a Taylor expansion that considers the second-order terms, this formulation is a second-order approximation that does not require the calculation of any second derivatives of f (MALIS, 2008).

For higher-dimensional systems of nonlinear equations, the ESM increment is defined as:

$$\widetilde{\mathbf{x}}_n = -2(\mathbf{J}(\widehat{\mathbf{x}}_n) + \mathbf{J}(\mathbf{x}^*))^+ \mathbf{f}(\widehat{\mathbf{x}}_n), \tag{2.46}$$

where $\mathbf{J}(\widehat{\mathbf{x}}) = \mathbf{f}'(\mathbf{x}) = \nabla \mathbf{f}(\mathbf{x})$ is the Jacobian matrix of function $\mathbf{f}$. The ESM Jacobian can thus be defined as:

$$\mathbf{J}_{ESM} = \frac{1}{2}\big[\mathbf{J}(\widehat{\mathbf{x}}_n) + \mathbf{J}(\mathbf{x}^*)\big]. \tag{2.47}$$

2.2.2 Nonlinear optimization formulation

The root-finding problems presented in the previous section can be reformulated as a minimization problem. Likewise, the methods for solving one can be used to solve the other. For instance, consider the problem of finding x such that:

$$f(x) = b. \tag{2.48}$$

This is equivalent to the root-finding problem

$$f(x) - b = 0. \tag{2.49}$$

Which, in turn, can be formulated as an minimization problem by choosing:

$$\min_x \|f(x) - b\|, \tag{2.50}$$

since the minimum value of $f(x) - b$ is zero.

If a system with m nonlinear equations and n unknowns is considered, finding $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ may be an overdetermined problem that is impossible to solve exactly, due to errors in the measurement points. In this case, it is useful to convert the problem to a least-squares problem in the form of:

$$\min_x c(\mathbf{x}) = \frac{1}{2}\mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}), \tag{2.51}$$

where $c(\mathbf{x})$ is the cost function to be minimized. A local minimum of it is found when

$$\nabla c(\mathbf{x}) = 0. \tag{2.52}$$

Again, the method starts from an initial solution $\mathbf{x}_0$ and applies an increment such that the sequence of solutions converges to the local minimum. A similar process as the one in the previous section can be applied to obtain each algorithm's increment. Table 2.1 summarizes the increments for Newton-Raphson and ESM methods. In the least squares case, the increments are calculated using pseudo-inverses $(\cdot)^+$, since the problem is overdetermined.

Table 2.1: Nonlinear Least Squares Increments

Method	Increment
Newton-Raphson	$-\mathbf{J}(\widehat{\mathbf{x}}_n)^+\mathbf{f}(\widehat{\mathbf{x}}_n)$
ESM	$-2(\mathbf{J}(\widehat{\mathbf{x}}_n) + \mathbf{J}(\mathbf{x}^*))^+\mathbf{f}(\widehat{\mathbf{x}}_n)$

2.3 Image Registration

This work solves the homography estimation problem by formulating it as an image registration task. The goal of the image registration problem is to estimate the transformation

that optimally aligns two images of the same scene. Pixels in different images that correspond to the same scene point are then mapped to each other. The first image is commonly referred to as the reference image, and the second one as the current image. According to (BROWN, 1992), existing methods to solve this problem can be viewed as a combination of four components:

1. information space;
2. transformation model;
3. similarity measure;
4. and search strategy.

The following sections describe each one of these components.

2.3.1 Information space

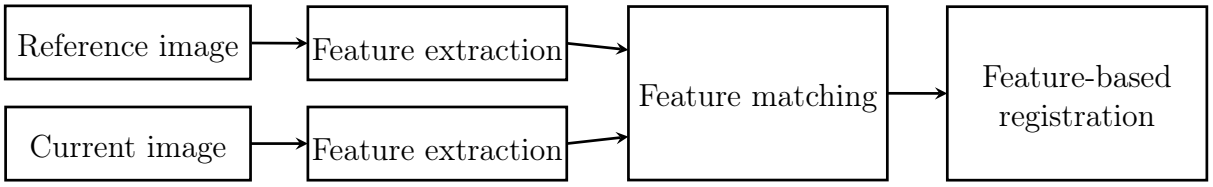
The information space dictates what information from the images is exploited, i.e. what information is the input to the registration process. In this sense, most existing methods can be classified into two categories, and their processing pipelines are shown in Fig. 2.6:

The first category is composed by feature-based methods. In it, geometric primitives (e.g., points, lines, ellipses, etc.), also known as image features, are extracted from each image. The information space regards the parametrization of these features in the images. Primitives from different images are compared and matched to obtain correspondences between them. The set of image coordinates of corresponding features is the input to the image registration algorithm. In general, each feature contributes two equations to the estimation procedure. These methods thus depend heavily on the accuracy of the extraction and matching steps. They can even fail due to the outliers in these steps.

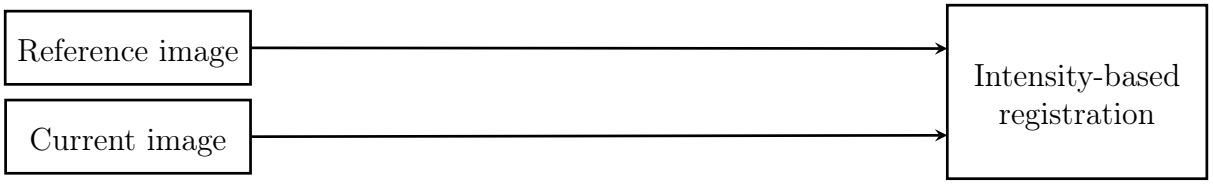
The second category contains the intensity-based methods. In this category, the image registration algorithm directly exploits the pixel intensities, with no intermediate steps. The information space is composed of the intensity values of the pixels in the image. They are also known as direct methods, appearance-based, and texture-based. Intensity-based methods can achieve high levels of accuracy due to the fact that they can exploit all image information. In general, each pixel in the region of interest contributes one equation to the estimation procedure. These methods often assume a sufficient overlapping between the two images. This is a reasonable condition when dealing with robotics applications.

2.3.2 Transformation models

This component defines the admissible transformations to the images that will be considered by the algorithm. The more complex the model, more parameters are needed to completely define the transformation. Parameters that are defined by the transformation



(a) Feature-based



(b) Intensity-based

Figure 2.6: Feature- vs intensity-based registration pipeline.

model are the output of the image registration algorithm. The guiding principle when choosing a transformation model is to have one that is flexible enough to account for the likely changes that occur in the image, while also balancing the complexity involved. This is a key tradeoff in the solution design. This thesis uses mainly two models: one that parametrizes the changes that are due to geometric motion and another for the photometric, i.e., lighting changes in the image.

Geometric transformation

The geometric transformations model image changes due to variations in the scene structure and/or the camera motion. For this purpose, this work considers the homography matrix. For a given pixel in the reference frame, its change of position in the current one is modeled as

$$\mathbf{p} \propto \mathbf{H}\,\mathbf{p}^* \tag{2.53}$$

$$= \left[\frac{h_{11}u^* + h_{12}v^* + h_{13}}{h_{31}u^* + h_{32}v^* + h_{33}}, \frac{h_{21}u^* + h_{22}v^* + h_{23}}{h_{31}u^* + h_{32}v^* + h_{33}}, 1 \right]^\top, \tag{2.54}$$

where $\mathbf{H} \in \mathbb{SL}(3)$ is the projective homography matrix, $\{h_{ij}\}$ comprise its elements, $\mathbf{p}^* = [u^*, v^*, 1]^\top \in \mathbb{P}^2$ is the homogeneous pixel coordinates in the reference image, and $\mathbf{p}$ is its corresponding coordinates in the current image. This is, clearly, the same warping operation warping presented in (2.21). Equation (2.54) is appropriate to model geometric transformations when at least one out of the three conditions below is true:

- Planar surfaces: The observed object is planar, with no constraints on the camera motion;
- Objects at infinity: The scene is far from the camera, with no constraints on the scene geometry or on the camera motion;

- Purely rotational motion: The camera undergoes a pure rotation between images, with no constraints on the scene geometry.

The homography matrix $\mathbf{H}$ is related to the scene structure and camera motion via

$$\mathbf{H} \propto \mathbf{K}\left(\mathbf{R} + \frac{1}{d^*}\mathbf{t}\mathbf{n}^{*\top}\right)\mathbf{K}^{-1}, \tag{2.55}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ contains the camera intrinsic parameters; $\mathbf{R} \in \mathbb{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the rotation matrix and translation vector associated with the camera motion, respectively; and $d^* > 0$ and $\mathbf{n}^* \in \mathbb{R}^3 : \|\mathbf{n}^*\| = 1$ are, respectively, the distance to the surface plane and its normal vector with respect to the reference frame, as described in Fig. 2.2. The decomposition of $\mathbf{H}$ into those Euclidean components is out of the scope of this work. A standard method for this decomposition has been proposed by (MALIS; VARGAS, 2007) and an open-source implementation is available inside the OpenCV library (BRADSKI, 2000). Additionally, a step-by-step guide to the homography decomposition can be found in (MA et al., 2012).

It is possible to vary the number of degrees of freedom in a homography, creating different categories. Three of them are most relevant:

1. Full: This is the most general homographic transformation. Thus, it contains eight d.o.f.;
2. Affine: This homography contains six d.o.f., which are related to the two image translations (vertical and horizontal), the two image scalings, a 2D image rotation and a shear transformation;
3. Stretch: This homography contains four d.o.f., which are related to the two image translations and the two scaling transformations.

Examples of each homography category are shown in Fig. 2.7. Depending on the application at hand, the methods proposed here may use a different category of homography that suits its problem domain better. The correct choice eliminates unnecessary calculations and increases the algorithms performance. This is enabled in the computational package that is publicly available. The rest of this work will consider only the "Full" case, which is the most generic.

Photometric transformation

This transformation model aims to explain the changes in the image due to variations in the lighting conditions of the scene. This work models only global illumination changes, i.e., changes that apply equally to all pixels in the images. This model can be defined as

$$\mathcal{I}'(\mathbf{p}) = \alpha\,\mathcal{I}(\mathbf{p}) + \beta, \tag{2.56}$$

where $\mathcal{I}(\mathbf{p})$ is the intensity value of the pixel $\mathbf{p}$ in image $\mathcal{I}$, $\mathcal{I}'(\mathbf{p})$ denotes its transformed intensity, and the gain $\alpha \in \mathbb{R}$ and the bias $\beta \in \mathbb{R}$ are the parameters that fully define the transformation. These parameters represent the adjustments in the image contrast and brightness, respectively.

(a) The original image (b) A full warping example

(c) An affine warping example (d) A stretch warping example

Figure 2.7: Examples of different homographic transformations applied to Van Gogh's *Starry Night*.

2.3.3 Similarity measures

Similarity measures give an indication of the registration quality. They are important because they allow different solutions to the optimization problem to be compared to each other. This work uses two of these measures, as described below.

The first similarity measure is the Sum of Squared Differences (SSD) over the pixel intensities:

$$\mathrm{SSD}(\mathcal{I}', \mathcal{I}^*) = \sum_i \left[\mathcal{I}'(\mathbf{p}_i, \mathbf{x}) - \mathcal{I}^*(\mathbf{p}_i^*) \right]^2, \tag{2.57}$$

where $\mathcal{I}^*(\mathbf{p}_i^*)$ is the intensity value of $\mathbf{p}_i^*$ in the reference image, $\mathcal{I}'(\mathbf{p}_i, \mathbf{x})$ is the intensity value of the current image photogeometrically transformed using the parameters $\mathbf{x} = \{\mathbf{H}, \alpha, \beta\}$. The SSD is used in the registration as the cost function to be minimized. Because the optimal transformation results in a minimal value, the SSD is technically a dissimilarity measure.

The other similarity measure used to assess the registration quality is the Zero-mean Normalized Cross Correlation:

$$\mathrm{ZNCC}(\mathcal{I}^*, \mathcal{I}') = \left\langle \frac{\mathcal{I}_v^* - \bar{\mathcal{I}}_v^*}{\left\| \mathcal{I}_v^* - \bar{\mathcal{I}}_v^* \right\|}, \frac{\mathcal{I}_v' - \bar{\mathcal{I}}_v'}{\left\| \mathcal{I}_v' - \bar{\mathcal{I}}_v' \right\|} \right\rangle, \tag{2.58}$$

where $(\cdot)_v$ denotes the vectorized form of an image, $\bar{(\cdot)}$ represents its mean value, and $\langle \cdot, \cdot \rangle$ is the inner product. This similarity measure varies between -1 (very bad) and $+1$ (perfect). Especially because of such normalizations, the ZNCC is useful both for evaluating the final quality of the estimation and as a sliding window predictor.

2.3.4 Search strategy

The algorithms developed in this work formulate the search strategy as a nonlinear optimization problem. To solve the nonlinear Least Squares optimization problem (2.57), the Efficient Second-order Minimization method is applied, as presented in Section 2.2. Its advantages when registering images include both a higher convergence rate and a larger convergence domain than standard iterative methods, without any costly Hessian calculations (SILVEIRA; MALIS, 2010). The use the ESM method is possible here because the image at the solution is known beforehand: it is the reference image. Its first order derivatives are obtained by calculating the gradient of the image in the $\vec{u}$ and $\vec{v}$ direction.

All proposed methods in this work also make use of a multiresolution pyramid. This strategy consists of creating a sequence of smaller templates until a predefined minimum size is reached. Two pyramids are built, one for the current template and another one for the reference template. The algorithms solve the optimization from the lowest resolution templates to the highest ones. The benefits of this coarse-to-fine strategy include an increase in the computational efficiency, avoidance of spurious local minima, and a larger domain of convergence (IRANI; ANANDAN, 1999).

2.3.5 Working conditions

The presented theory leads to a set of conditions that have to be respected for the proposed algorithms to work correctly and efficiently. These conditions are summarized below:

- the inter-frame displacement is relatively small;
- for unknown camera motion, the observed object is planar or is at infinity;
- for pure rotational camera motion, the observed object can be of any shape and distance;
- the observed object is sufficiently textured;
- the observed object is subject only to global illumination changes;
- the observed object is mostly unoccluded.

3 ROBUST INTENSITY-BASED HOMOGRAPHY ESTIMATION

Intensity-based methods can exploit all image pixels, and can thus attain high levels of accuracy and versatility. They are also referred to as direct methods, because they skip intermediate steps that are required in the feature-based approach. A possible challenge of direct methods is handling large illumination changes. (L. CHEN et al., 2017) tackle this problem by considering gradient orientations as dense image features and solving the ESM optimization with multidimensional images. (ALISMAIL; BROWNING; LUCEY, 2016) use the Lucas-Kanade method coupled with illumination-invariant binary descriptors.

In this chapter, the idea proposed in (SILVEIRA; MALIS, 2010) is used, but only global illumination changes are modeled. Therefore, in order to robustly handle illumination changes, the proposed method considers a photogeometric transformation model. The geometric part is composed of the homography matrix and can be considered the main goal of the estimation process. The photometric part is composed of a gain and a bias values that parameterize a global affine transformation. By considering both photometric parameters jointly with the geometric ones, the estimation is made robust to global illumination changes. Figure 3.1 shows examples of such changes.

Figure 3.1: Example of an image with varying illumination and occlusions

The parametric direct estimation process is here formulated as a multidimensional optimization problem. Given two images, the objective consists of finding the optimal set of photogeometric parameters that minimize the Sum of Squared Differences (SSD) over the pixel intensities of an region of interest. Typically, an initial solution is iteratively refined using a nonlinear optimization method. However, if a partial occlusion, as shown in Fig. 3.1, is present in the images, then the SSD may lead to a completely erroneous result. In this chapter, a suitable M-estimator is proposed to treat unknown occlusions in intensity-based efficient methods. This technique involves defining a robust function to reduce (or eliminate) the effects of large residuals. The Talwar function (HINICH; TALWAR, 1975) is used because the homography is estimated jointly with photometric values and the algorithm needs to reach real-time performance whilst being able to completely eliminate the perceived outliers (AHMED et al., 2016; IKAMI; YAMASAKI; AIZAWA, 2018). (MEILLAND; COMPORT; RIVES, 2011) use a similar robust technique to an

intensity-based visual estimation method. However, they estimate the 3D camera pose, not a homography, and apply the Huber robust function. (TORR; A. ZISSERMAN, 1998) do use the Talwar robust function, but in a feature-based setting.

3.1 Problem Modelisation

The homography estimation is formulated as an image registration problem. Two images are considered: the reference image $\mathcal{I}^*$ and the current image $\mathcal{I}$. Both images observe the same planar region. This region undergoes a photogeometric transformation that can be completely parametrized via a homography matrix $\mathbf{H}$, a gain α and a bias β. If the region of interest, i.e. the reference template, has m pixels, the goal is to find such parameters that transform each pixel $\mathbf{p}_i^*, i \in \{1, \ldots, m\}$ of the reference image into its correspondent $\mathbf{p}_i$ in the current image, such that:

$$\alpha\mathcal{I}(\mathbf{w}(\mathbf{H}, \mathbf{p}_i^*)) + \beta = \mathcal{I}^*(\mathbf{p}_i^*), \quad \forall i = 1, \ldots, m \tag{3.1}$$

Since there are 10 parameters to be estimated, and m is typically higher than 10, the problem is overdetermined. Therefore, it can be reformulated as a nonlinear least squares problem:

$$\min_{\mathbf{x}=\{\alpha,\beta,\mathbf{H}\}} \quad \frac{1}{2}\sum_{i=1}^{m} \left[\ \alpha\mathcal{I}(\mathbf{w}(\mathbf{H}, \mathbf{p}_i^*) + \beta - \mathcal{I}^*(\mathbf{p}_i^*)\ \right]^2. \tag{3.2}$$

3.2 Variable Parametrization

For simplicity, consider just the geometric part of the transformation, i.e. the homography matrix. Also, that an approximation $\widehat{\mathbf{H}}$ of the solution homography $\mathbf{H}^*$ is given. The problem then becomes of finding the incremental homography $\widetilde{\mathbf{H}}$ such that the difference between the current image warped by the homography $\widehat{\mathbf{H}}\widetilde{\mathbf{H}}$ and the reference image is zero for every pixel i in the region of interest. The residual y_i can be defined as:

$$y_i(\mathbf{x}) = \mathcal{I}(\mathbf{w}(\widehat{\mathbf{H}}\widetilde{\mathbf{H}}, \mathbf{p}_i^*)) - \mathcal{I}^*(\mathbf{p_i}^*) = 0 \tag{3.3}$$

As seen previously, the homography $\widetilde{\mathbf{H}} \in \mathbb{SL}(3)$ is a 3×3 matrix with only eight degrees-of-freedom. In general, this situation leads to the need of adding a reprojection step after each iteration of the minimization algorithm to bring the estimated matrix back into the Special Linear Group. To avoid this problem, the proposed algorithm uses the Lie Algebra formulation that parametrizes the incremental homography using its tangential space (BENHIMANE; MALIS, 2007). This is accomplished via the matrix exponential function, which maps a region around the identity matrix $\mathbf{I} \in \mathbb{SL}(3)$ to a region around the nul matrix $\mathbf{0} \in \mathfrak{sl}(3)$ in its Lie Algebra. With it, a matrix $\mathbf{A}(\mathbf{v}) \in \mathfrak{sl}(3)$ can be written as:

$$\mathbf{A}(\mathbf{v}) = \sum_{i=1}^{8} v_i \mathbf{A}_i, \tag{3.4}$$

where the matrices $\mathbf{A}_i, i = \{1, 2, \ldots, 8\}$ form a base of the Lie Algebra and $v_i, i = \{1, 2, \ldots, 8\}$ are the components of vector $\mathbf{v}$. And an homography is thus parameterized using the exponential map:

$$\mathbf{H}(\mathbf{v}) = \exp(\mathbf{A}(\mathbf{v})) = \sum_{i=0}^{\infty} \frac{1}{i!}(\mathbf{A}(\mathbf{v}))^i \tag{3.5}$$

The matrices that compose the base of the Lie Algebra used in this work are the same as the ones presented in (BENHIMANE, 2006), which are:

$$\begin{aligned}
\mathbf{A}_1 &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\
\mathbf{A}_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{A}_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{A}_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
\end{aligned} \tag{3.6}$$

Using this tool, the photogeometric image transformation model defined by $\mathbf{x} = \{\mathbf{H}, \alpha, \beta\}$ can be replaced by an incremental parameter vector $\mathbf{z} = \{\mathbf{v}, \widetilde{\alpha}, \widetilde{\beta}\}$, with $\mathbf{v}$ comprising the eight Lie Algebra elements of the homography reparametrization. If $\widehat{\mathbf{x}} = \{\widehat{\mathbf{H}}, \widehat{\alpha}, \widehat{\beta}\}$ is an approximation of the true solution $\mathbf{x}^*$, then:

$$\mathbf{x} = \widetilde{\mathbf{x}}(\mathbf{z}) \circ \widehat{\mathbf{x}} \tag{3.7}$$

where $\circ$ denotes the composition operation. For α and β the composition operation is addition; for the homography, it is matrix multiplication. With this parametrization, the incremental transformation to an image is:

$$\mathcal{I}'(\mathbf{p}_i^*, \widetilde{\mathbf{x}}(\mathbf{z}) \circ \widehat{\mathbf{x}}) = (\widehat{\alpha} + \widetilde{\alpha})\mathcal{I}(\mathbf{w}(\widehat{\mathbf{H}}\widetilde{\mathbf{H}}(\mathbf{v}), \mathbf{p}_i^*)) + (\widehat{\beta} + \widetilde{\beta}) \tag{3.8}$$

Now, it is possible to rewrite the minimization problem from (3.2) as:

$$\min_{\mathbf{z}=\{\mathbf{v},\widetilde{\alpha},\widetilde{\beta}\}\in\mathbb{R}^{10}} \quad \frac{1}{2}\left\|\mathcal{I}'(\mathbf{p}_i^*, \widetilde{\mathbf{x}}(\mathbf{z}) \circ \widehat{\mathbf{x}}) - \mathcal{I}^*(\mathbf{p}_i^*)\right\|^2 \tag{3.9}$$

3.3 The Jacobian Matrices

Usage of the Efficient Second-order Minimization algorithm requires the calculation of two Jacobian matrices in order to obtain the increment vector, as shown in Table 2.1. The first Jacobian matrix is called the *current* Jacobian, and is evaluated at the current evaluation point $\mathbf{x} = \mathbf{x}_n$. The second Jacobian is called the *reference* Jacobian and is evaluated at the solution point $\mathbf{x} = \mathbf{x}^*$. Using the Lie Algebra parametrization presented in the previous section it is also possible to say that the current Jacobian is evaluated at point $\mathbf{z} = \mathbf{0}$ and the reference Jacobian matrix at the point $\mathbf{z} = \mathbf{z}^*$.

Each pixel i in the region of interest contributes to one equation in the least-squares formulation. Therefore, it will also contribute to one line to the Jacobian. The residual $y_i(\mathbf{z})$ associated with the i-th pixel can be written as:

$$y_i(\mathbf{z}) = (\alpha + \widehat{\alpha})\mathcal{I}(\mathbf{w}(\widehat{\mathbf{H}}\widetilde{\mathbf{H}}(\mathbf{v}), \mathbf{p}_i^*)) + (\beta + \widehat{\beta}) - \mathcal{I}^*(\mathbf{p}_i^*) \tag{3.10}$$

(BENHIMANE, 2006) shows that the *geometric* components of these Jacobian matrices are the composition of three separate Jacobians:

$$\nabla_{\mathbf{v}} y_i(\mathbf{v}) = \mathbf{J}_{\mathcal{I}}\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{H}}, \tag{3.11}$$

where

1. $\mathbf{J}_{\mathcal{I}}$: This Jacobian is a (1×3) matrix that corresponds to the partial derivative of the image function w.r.t. its projective coordinates. This is also known as the image gradients. Figure 3.2 shows an example of these gradients.
2. $\mathbf{J}_{\mathbf{w}}$: This Jacobian is a (3×9) matrix that corresponds to the partial derivative of the warping function w.r.t. the 9 elements of the homography matrix. Therefore, it defines how the projective coordinates of a point change due to small changes in the elements of the homography that is used in the warp function.
3. $\mathbf{J}_{\mathbf{H}}$: This Jacobian is a (9×8) matrix that corresponds to the partial derivative of the Homography matrix w.r.t to its Lie Algebra parametrization vector $\mathbf{v}$.

An important result from (BENHIMANE, 2006) is the multiplication invariance property, defined as:

$$\mathbf{J}_{\mathbf{H}}\big|_{\mathbf{z}=\mathbf{z}^*}\mathbf{z}^* = \mathbf{J}_{\mathbf{H}}\big|_{\mathbf{z}=\mathbf{0}}\mathbf{z}^*. \tag{3.12}$$

where $\mathbf{J}_{\mathbf{H}}\big|_{\mathbf{z}=\mathbf{z}^*}$ is the Jacobian $\mathbf{J}_{\mathbf{H}}$ calculated at the reference; $\mathbf{J}_{\mathbf{H}}\big|_{\mathbf{z}=\mathbf{0}}$ is the same Jacobian, but calculated at the current image.

This result implies that it suffices to calculate the $\mathbf{J}_{\mathbf{H}}$ for one case, so it can be considered invariant for both the reference and current Jacobians. The $\mathbf{J}_{\mathbf{w}}$ is also constant in both cases. Therefore, the difference is only in the $\mathbf{J}_{\mathcal{I}}$ component. On one hand, for the reference, it only needs to be calculated once during the optimization process, since the reference image does not change, and neither do its gradients. On the other hand, this component needs to be recalculated at every iteration for the current Jacobian.

Now, considering also the **photometric** part of the Jacobian matrix, it is possible to write:

$$\nabla_{\mathbf{z}} y_i(\mathbf{z}) = \begin{bmatrix} \frac{\partial y_i}{\partial \mathcal{I}}\mathbf{J}_{\mathcal{I}}\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{H}} & \frac{\partial y_i}{\partial \alpha} & \frac{\partial y_i}{\partial \beta} \end{bmatrix} \tag{3.13}$$

For the complete current Jacobian, Eq. 3.13 is extended to vector form and thus becomes:

$$\mathbf{J}(\mathbf{0}) = \nabla_{\mathbf{z}}\mathbf{y}(\mathbf{z} = \mathbf{0}) = \begin{bmatrix} \alpha\mathbf{J}_{\mathcal{I}}\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{H}} & \mathcal{I} & \mathbf{1} \end{bmatrix}, \tag{3.14}$$

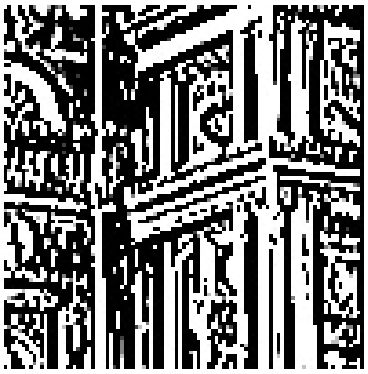
where $\mathcal{I}$ is the current image and $\mathbf{J}_{\mathcal{I}}$ its gradient. Likewise, the complete reference Jacobian is:

$$\mathbf{J}(\mathbf{z}^*) = \nabla_{\mathbf{z}}\mathbf{y}(\mathbf{z} = \mathbf{z}^*) = \begin{bmatrix} \alpha\mathbf{J}_{\mathcal{I}^*}\mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{H}} & \mathcal{I}^* & \mathbf{1} \end{bmatrix}, \tag{3.15}$$

where $\mathcal{I}^*$ is the reference image and $\mathbf{J}_{\mathcal{I}^*}$ its gradient. The ESM Jacobian for the intensity-based case can then be defined as:

$$\mathbf{J}_{IB} = \frac{1}{2}\big[\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{z}^*)\big] \tag{3.16}$$

(a) The original image.



(b) $\vec{u}$ gradient.

(c) $\vec{v}$ gradient.

Figure 3.2: Image gradients.

3.4 Sliding Window Prediction

Intensity-based methods require relatively small interframe displacements. This means that the approximation to the solution that initializes the optimization needs to be close enough to the true solution for the algorithm to converge. Different solutions exist that perform an initialization step to refine this initial estimate before passing it on to the optimization. This work implements a simple sliding window (SW) predictor for this purpose.

The SW algorithm consists of generating multiple candidates solutions, evaluating them according to a given metric, and choosing the best one. A common method to generate these candidates in Computer Vision is done by translating a window, i.e. a region of interest, across a bigger image. Each position of the window is therefore a candidate solution.

In this case, the algorithm is based upon the Homography matrix parametrization. Consider that the estimation has already been initialized and therefore a reference template of resolution $(p \times q)$ has been stablished. Then, it receives a current image $\mathcal{I}$ and an initial approximation of the solution $\widehat{\mathbf{x}} = \{\widehat{\mathbf{H}}, \widehat{\alpha}, \widehat{\beta}\}$. It is possible to use $\mathcal{I}$ and $\widehat{\mathbf{x}}$ to build a current template of same size. However, the SW predictor actually uses a modified version of $\widehat{\mathbf{x}}$ to build a *sliding window template*, which is slightly bigger than the reference template, i.e. a resolution of $(\lfloor Np \rfloor \times \lfloor Nq \rfloor)$, with $N > 1$.

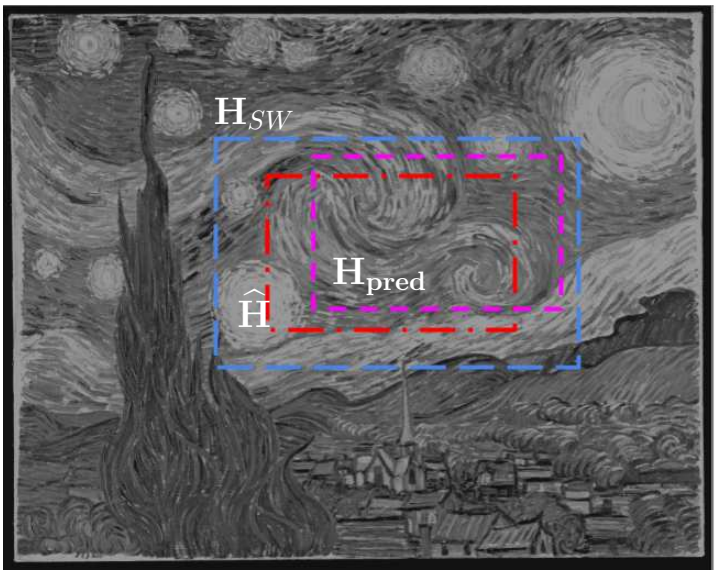


Figure 3.3: The sliding window algorithm

The modified version of the initial approximation is obtained by applying a translation matrix operation to $\widehat{\mathbf{H}}$. This is done by building a homography $\mathbf{H_t}$ that encodes a translation operation, such as:

$$\mathbf{H_t} = \begin{bmatrix} 1 & 0 & \delta u \\ 0 & 1 & \delta v \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.17}$$

where the translational elements are calculated as such:

$$\delta u = \frac{(Np - 1)}{2} \tag{3.18}$$

$$\delta v = \frac{(Nq - 1)}{2} \tag{3.19}$$

where N is the multiplication factor, which is 1.2 in this case. A bigger value for N allows for bigger interframe displacement, but also entails a higher computational cost. This is specially important because correlation-based methods are very costly. In the presented method, the sliding window prediction occurs only in the highest level of the multiresolution pyramid, i.e. the lowest resolution. This allows for faster computing, at the cost of precision.

The sliding window homography is obtained by matrix multiplication:

$$\mathbf{H}_{SW} = \widehat{\mathbf{H}}\mathbf{H_t} \tag{3.20}$$

Once $\mathbf{H}_{SW}$ is obtained, a warping operation is performed to obtain the sliding window template from the current image. Then, the smaller reference template is superimposed over a region of the SW template and sequentially translated in $\vec{u}$ and $\vec{v}$ directions. For each position it assumes, the reference template is compared to the corresponding region in the sliding window template. For this purpose, the Zero-Mean Normalized Cross-Correlation is used to assign a score. This process is similar to performing a convolution

of the sliding window template section with the reference template. It generates a matrix that contains the ZNCC scores associated with each position. The position (u, v) that generates the best score is then used to produce another translation homography that is composed with $\mathbf{H}_{SW}$. This resulting homography $\mathbf{H}_{pred}$ replaces $\widehat{\mathbf{H}}$ as the initial approximation to the optimization procedure.

Figure 3.3 showcases the three "windows" that are created by the algorithm. Each one can be generated by warping the current image with the appropriate homography. The red dash-and-dotted window is associated with the initial approximation $\mathbf{H}$. The blue with longer dashes line is created from $\mathbf{H}_{SW}$ and can be seen as a stretching of the previous window in the translational directions. Finally, $\mathbf{H}_{pred}$ is the predicted homography, chosen because it will have the biggest ZNCC value among all candidates.

3.5 Robust Method

Real-world applications often require the use of a robust estimation method to deal with model uncertainties and measurement errors (outliers), such as unknown occlusions. Least Squares algorithms are not robust since in theory a single outlier can lead to an estimate arbitrarily far from the true solution. In an intensity-based algorithm, outliers are often the result of an occlusion. An strategy to make the estimation robust to occlusions is presented in the sequel.

The robust equivalent of the Least Squares family are the M-estimators. In this case the cost function to be minimized is modified to

$$\sum_i \rho\big(r_i(\mathbf{x})\big), \tag{3.21}$$

where r_i is the i-th normalized residual from (2.57), and $\rho(\cdot)$ is a robust function (at least C^0) that penalizes the largest residuals (HUBER, 1981). Specifically, the Talwar robust function is chosen:

$$\rho(z) = \begin{cases} z^2/2 & \text{if } |z| \leq c, \\ c^2/2 & \text{if } |z| > c, \end{cases} \tag{3.22}$$

with its constant $c = 2.795$, because of its hard redescending property and its low computational cost (C. CHEN, 2009). That constant is obtained for 95% asymptotic efficiency on the standard normal distribution. Its discontinuous first derivative is not an issue as will be shown next. As a remark, note that for $\rho(\mathtt{r}_i(\mathbf{x})) = \mathtt{r}_i^2(\mathbf{x})$ with an unnormalized residual $\mathtt{r}_i$, it becomes the original (nonrobust) Least Squares cost function (2.57).

An important advantage of M-estimators is that they can be implemented using a simple Iteratively Reweighted Least Squares procedure. The weights reflect the confidence of each datum and are computed as

$$w_i = \frac{1}{r_i}\frac{\partial \rho(r_i)}{\partial r_i}. \tag{3.23}$$

The procedure hence estimates $\mathbf{x}$ by solving a weighted Least Squares system, and reiterates until convergence. In any case there still exists a trade-off between efficiency and

robustness. Relatively small interframe displacements or a suitable predictor should thus always be applied.

3.6 Operation Modes

Considering the methods presented in this chapter, it is possible to define four modes of operation for the algorithm:

1. *Regular (IBG)*: This mode performs the optimization *without* the robust method presented in this chapter. Therefore, it always considers all pixels in the template to obtain its increment. Additionally, it does not use any predictor method to aid the initialization of the algorithm.

2. *Robust (RIBG)*: This mode performs the otimization *with* the robust method presented in this section. Thus, it will constantly try to estimate the occlusions that might be present in the template. Additionaly, it does not use any predictor method.

3. *Regular + Predictor (IBG_P)*: This mode uses the Regular optimization method in conjunction with the ZNCC-based sliding window predictor method presented in Section 3.4.

4. *Robust + Predictor (RIBG_P)*: This modes uses the Robust optimization method in conjunction with the ZNCC-based sliding window predictor method presented in Section 3.4.

4 FEATURE-BASED HOMOGRAPHY ESTIMATION

The vast majority of computer vision estimation algorithms are based on the extraction and matching of features in different images. These features are geometric primitives that can be easily detected in an image, such as points belonging to the corner of a physical structure; or lines that correspond to walls or windows.

In these methods, each image is processed in order to extract its most prominent features. Then, the set of features from different images are compared to each other, with the goal of matching features that correspond to the same physical entity. Finally, this set of matches is then used as an input to the parametrical estimation procedure.

4.1 Feature Detectors and Descriptors

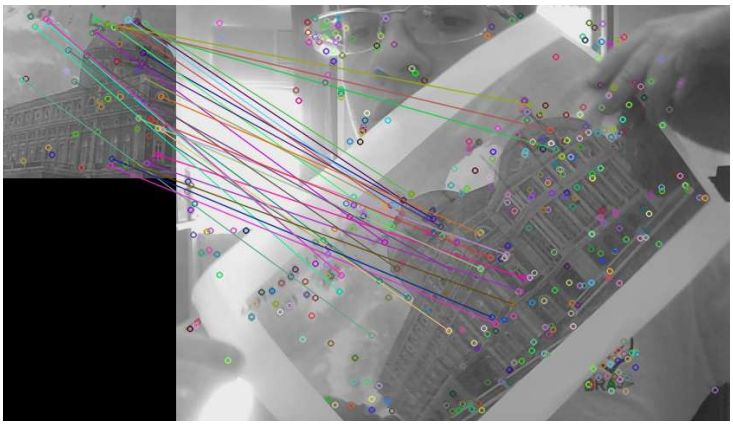


Figure 4.1: Example of the feature matching process.

Ideally, the feature extraction step has two important properties: repeatability and precision. The first means that the same feature can be extracted from different images independently of the geometric and/or photometric transformations that might have been applied to them. The second is related to the error between the detected feature position in the image and its actual position. This is specially important because the error introduced in this step is never corrected in subsequent steps.

Once a feature is detected, a descriptor is generated for it. In general, it consists of a vector that encodes information about the feature. For instance, some descriptors analyze the surrounding pixels of the feature and compute important metrics that help to characterize it, such as gradients and histograms. After a descriptor has been calculated

for each feature detected, the feature matching step begins. In general, a similarity measure is calculated between the descriptors in order to find those that correspond to the same observed object. A threshold value is traditionally used in this step. If the similarity measure is bigger than the threshold, the features are matched. Table 4.1 lists a few of the most important feature extraction and matching algorithms in modern computer vision. Note that some algorithms provide both a detector and a descriptor, while others focus only on one of these steps. A thorough review and evaluation of feature extraction methods is available in (CANCLINI et al., 2013).

Method	Detector	Descriptor	Reference
SIFT	✓	✓	(LOWE, 2004)
HARRIS	✓		(HARRIS; STEPHENS, et al., 1988)
SURF	✓	✓	(BAY; TUYTELAARS; VAN GOOL, 2006)
ORB	✓	✓	(RUBLEE et al., 2011)
FAST	✓		(ROSTEN; DRUMMOND, 2006)

Table 4.1: Feature Detectors and Type

The feature extraction and matching steps produce a set of feature correspondences. They relate the coordinates of a feature in an image with its coordinates in another image. These correspondences are then used to estimate the transformation parameters that best describe the changes in the coordinates. Therefore the feature *coordinates* are used as the information explored by the estimation algorithms, which differs from the intensity-based methods.

This work is not focused on feature extraction and matching algorithms. Indeed, here they are used in a black-box approach. Instead, the focus is on the estimation procedure that happens after those steps have been performed. In particular, the SURF method has been used for feature detection and description. Investigating which methods provide the best results in relevant scenarios is a research problem that remains open, as is highlighted in the Conclusion chapter.

4.2 Problem Modelisation

As in the previous chapter, the homography estimation is formulated as an image registration problem. Again, a nonlinear least squares approach will be proposed. An important difference is that the feature-based method considers only a geometric transformation. It ignores illumination changes. This occurs because the information space, i.e. the feature coordinates, does not contain any type of photometric property. Therefore, in this case, the transformation space is composed only of the homography matrix $\mathbf{H}$.

Consider that the estimation algorithm has been initialized with a *reference template*. This is usually a region of interest with predefined resolution inside a larger reference image. Then, a second image, referred to as *current image* is given to the estimation algorithm. The goal is to find the transformation parameters that, when applied to the current image, results in a *current template* that is, ideally, equal to the the reference

template. The detection algorithm scans the reference template for features, and does the same with the current image. Then, these two sets of features are matched in order to create a correspondence set. Consider that these steps have provided n correspondences. Then, the goal is to find an homography $\mathbf{H}^*$ such that

$$\mathbf{w}(\mathbf{H}^*, \mathbf{p}_i^*) = \mathbf{p}_i, \quad \forall i = 1, \ldots, n. \tag{4.1}$$

If the feature extraction produced four perfect correspondences between the reference and the current image, it would be possible to calculate $\mathbf{H}^*$ perfectly. However, this is not the case. Traditionally, an over-complete set of correspondences is provided. Therefore, the problem is reformulated as a nonlinear least squares problem.

$$\min_{\mathbf{x}=\{\mathbf{H}\}} \quad \frac{1}{2}\sum_{i=1}^{n} \left[\mathbf{w}(\mathbf{H}, \mathbf{p}_i^*) - \mathbf{p}_i \right]^2 \tag{4.2}$$

Using the same Lie Algebra parametrization from Equ 3.5, it is possible to replace the transformation parameter vector $\mathbf{x} = \{\mathbf{H}\}$ with $\mathbf{z} = \{\mathbf{v}\}$ such that $\mathbf{x} = \widetilde{\mathbf{x}} \circ \widehat{\mathbf{x}}$, where $\widehat{\mathbf{x}}$ is an approximation to the true solution. Now, it is possible to rewrite the minimization problem as:

$$\min_{\mathbf{z}=\{\mathbf{v}\}\in\mathbb{R}^8} \quad \frac{1}{2}\sum_{i=1}^{n} \left[\mathbf{w}(\widehat{\mathbf{H}}\widetilde{\mathbf{H}}(\mathbf{v}), \mathbf{p}_i^*) - \mathbf{p}_i \right]^2 \tag{4.3}$$

where $\widehat{\mathbf{H}}$ is an initial approximation of the homography matrix. An equivalent formulation is then:

$$\min_{\mathbf{z}=\{\mathbf{v}\}\in\mathbb{R}^8} \quad \frac{1}{2} \left\| \mathbf{w}(\widetilde{\mathbf{x}}(\mathbf{z}) \circ \widehat{\mathbf{x}}, \mathbf{p}_i^*) - \mathbf{p}_i \right\|^2 \tag{4.4}$$

4.3 The Jacobian Matrices

In the Efficient Second-order Minimization framework, there are two important Jacobian matrices that need to be calculated. They are referred to as the current and reference Jacobians. However, due to the Lie Algebra formulation presented in (BENHIMANE, 2006), a special result is achieved that makes it unnecessary to calculate both.

Each feature i contributes with two equations in the least-squares formulation. Therefore, it will also contribute with two lines to the Jacobian. The residual $\mathbf{y}_i(\mathbf{z})$ associated with the i-th feature correspondence can be written as:

$$\mathbf{y}_i(\mathbf{z}(\mathbf{v})) = \mathbf{w}(\widehat{\mathbf{H}}\widetilde{\mathbf{H}}(\mathbf{v}), \mathbf{p}_i^*) - \mathbf{p}_i \tag{4.5}$$

Note that $\mathbf{y}_i$ is a 3×1 vector, but the last element is always zero and so it is discarded.

The feature-based Jacobian matrix is the composition of two separate Jacobians:

$$\nabla_{\mathbf{v}}\mathbf{y}_i(\mathbf{v}) = \mathbf{J}_{\mathbf{w}}\mathbf{J}_{\mathbf{H}}, \tag{4.6}$$

where

1. $\mathbf{J_w}$: This Jacobian is a (3×9) matrix that corresponds to the partial derivative of the warping function w.r.t. the 9 elements of the homography matrix. Therefore, it defines how the projective coordinates of a point change due to small changes in the elements of the homography that is used in the warp function. This Jacobian is the same for both the current and reference images.

2. $\mathbf{J_H}$: This Jacobian is a (9×8) matrix that corresponds to the partial derivative of the Homography matrix w.r.t to its Lie Algebra parametrization vector $\mathbf{v}$.

For the current and reference Jacobians, (4.6) becomes:

$$\mathbf{J(0)} = \nabla_{\mathbf{z}}\mathbf{y}_i(\mathbf{z} = \mathbf{0}) = \mathbf{J_w}\mathbf{J_H}\big|_{\mathbf{z}=\mathbf{0}} = \mathbf{J_w}\mathbf{J_{H_0}} \tag{4.7}$$

$$\mathbf{J}(\mathbf{z}^*) = \nabla_{\mathbf{z}}\mathbf{y}_i(\mathbf{z} = \mathbf{z}^*) = \mathbf{J_w}\mathbf{J_H}\big|_{\mathbf{z}=\mathbf{z}^*} = \mathbf{J_w}\mathbf{J_{H^*}} \tag{4.8}$$

,

The Taylor expansion of the cost function $\mathbf{y}(\mathbf{z})$ using the ESM formulation can be written as:

$$\mathbf{y}(\mathbf{z}^*) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2}(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{z}^*))\mathbf{z}^* \tag{4.9}$$

Using (4.7) and (4.8), it becomes:

$$\mathbf{y}(\mathbf{z}^*) \approx \mathbf{y}(\mathbf{0}) + \frac{1}{2}(\mathbf{J_w}\mathbf{J_{H_0}} + \mathbf{J_w}\mathbf{J_{H^*}})\mathbf{z}^* \tag{4.10}$$

This equation can then be simplified using (3.12).

$$\mathbf{y}(\mathbf{z}^*) \approx \mathbf{y}(\mathbf{0}) + \mathbf{J_w}\mathbf{J_{H^*}}\mathbf{z}^* \tag{4.11}$$

This result shows that, in the feature-based case, the ESM Jacobian can be greatly simplified by using only the reference Jacobian. At the same time, it maintains its second-order properties. This is specially useful because throughout the minimization process, the coordinates of the features in the reference image do not change, while for the current image they change at every iteration. Therefore, it is possible to compute the Jacobian once for the entire estimation process. In summary, the ESM Jacobian for the feature-based case is defined as:

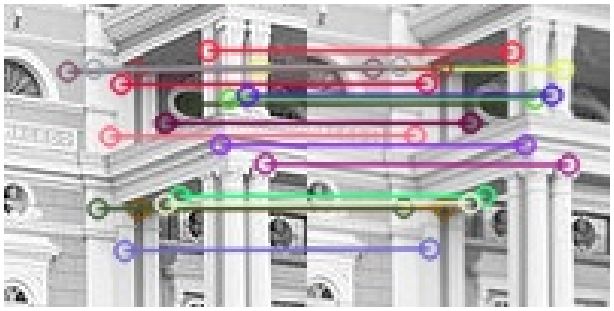
$$\mathbf{J}_{FB} = \mathbf{J_w}\mathbf{J_{H^*}} \tag{4.12}$$

4.4 Local versus Global Feature Search

Figure 4.2: Global and Local Initialization

In this section, a method for improving the estimation speed is presented. This method tackles an issue that arises in the feature detection step. Recall that this work uses SURF, but the issue is common to all feature detectors. The problem is that searching the complete current image for features can be computationally expensive. The proposed solution is to use the initial approximation $\widehat{\mathbf{H}}$ to narrow the feature detection search space.

The first step is the ZNCC test. The idea is to use $\widehat{\mathbf{H}}$ to warp the current image $\mathcal{I}$ into a tentative current template. This template is then compared to the reference one using the ZNCC similarity measure. If the ZNCC between them is higher than a threshold ϵ, the approximation is deemed to be close enough. In this case, the feature detection method searches only this current template. Otherwise, if the ZNCC is lower than ϵ, the entire image is searched, as it would normally be. In the first case, the feature search is referred to as *local* while in the second, *global*. Figure 4.2 illustrates this algorithm.



(a) Local estimation case: Features are detected in the current *template*.

(b) Global estimation case: Features are detected in the current *image*.

Figure 4.3: Local versus global feature detection and matching.

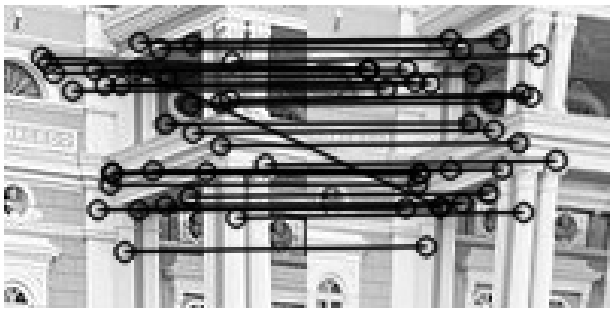
If the local feature search is used, then the current template is searched for features, and matched with those found in the reference template, as shown in Fig. 4.3a. The correspondence set obtained is used to calculate a homography using the DLT algorithm, which becomes the initial approximation passed onto the actual optimization method.

If the global feature search is used, then the complete current image is searched for features and matched with the reference template, as shown in Fig. 4.3b. The correspondence set, after an outlier rejection process (described in Section 4.5), is used to obtain a new approximation $\widehat{\mathbf{H}}$, also using the DLT method. With this approximation, a new current template is generated, and the estimation resumes with the local feature search.

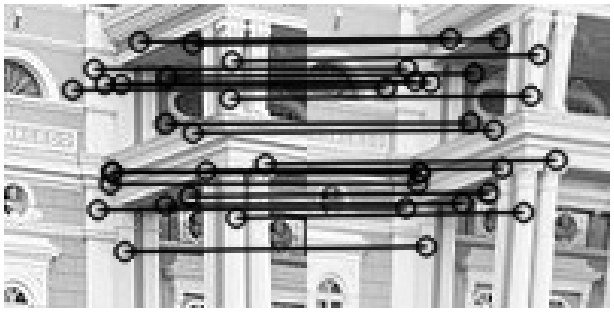
The goal of this preprocessing step is to speed up the estimation process. This is

important for real-time applications. If the estimation is used in a visual tracking setting, for instance, then it is a reasonable assumption that a previous result can be used to better set up the feature detection. However, one of the advantages of the feature based methods is that they do not require such small interframe displacements between the images as direct methods do. Therefore, it is important to maintain the ability to search the entire image when the initial approximation is far from the actual solution.

4.5 Outlier Rejection



(a) The original set of matches.



(b) A set of matches after outlier rejection.

Figure 4.4: Example of outlier rejection in feature matches.

The treatment of outliers in the feature matching process requires careful consideration. These outliers occur when the matching algorithm pairs two features in different images that do not correspond to the same scene point. In theory, a single outlier can move the estimated solution arbitrarily far from the true solution. Therefore, treating outliers is an important step to ensure that the optimization method does not use bad inputs. This section presents an outlier rejection method that is used in both the global and local feature searches before the DLT method is applied. Additionally, it is also applied at each iteration of the optimization procedure.

The outlier rejection method here is very similar to the method presented in section 3.5. It modifies the feature-based cost function $\mathbf{y}_{FB}$ using a robust function and recalculates weights at every iteration. Since the Talwar robust function is used, the weights are either 0 or 1. This allows for an implementation where lines with weights 0 are removed from the Jacobian and cost vector, instead of multiplying the weights by the corresponding lines in those matrices.

The method for computing the outliers is as follows. Given an initial approximation of the solution homography $\widehat{\mathbf{H}}$, the Euclidian distance between each feature match $j \in 1, \ldots, n$ is computed:

$$d_j(\widehat{\mathbf{H}}) = \left\| \mathbf{w}(\widehat{\mathbf{H}}, \mathbf{q}_j^*) - \mathbf{q}_j \right\|_2 \tag{4.13}$$

Then, two important metrics are calculated using the series d_j. The first one is the median:

$$\widetilde{d} = \text{median}(\mathbf{d}) \tag{4.14}$$

where $\mathbf{d}$ is the vector composed of the elements d_j. The second metric is the median absolute deviation (MAD):

$$MAD(\mathbf{d}) = \text{median}(|\mathbf{d} - \widetilde{d}|) \tag{4.15}$$

With these metrics, it is possible to filter out the outliers by choosing a range of acceptable distance values. For a given feature match j, it is considered an *inlier* if its distance to the median is less than $k \cdot MAD(\mathbf{d})$, where $k = 1.4826$ is a constant scale factor chosen for the case where the data is normally distributed. Thus, the following condition should be true:

$$|d_j - \widetilde{d}| < k \cdot MAD(\mathbf{d}) \tag{4.16}$$

The features matches that satisfy the condition above are kept, while those that do not are considered outliers and are removed from the feature-based cost function. Figure 4.4 demonstrate the matches before and after outlier rejection.

5 UNIFIED HOMOGRAPHY ESTIMATION

This chapter presents an approach that unifies the intensity-based and feature-based methods using the Efficient Second-order Minimization framework. Its goal is to obtain a method that can provide the advantages of both methods while trying to mitigate the effects of their disadvantages. This is the main contribution of this dissertation.

5.1 Problem Modelisation

Consider that the estimation algorithm has been initialized with a *reference template*. This is tipically a region of interest with predefined resolution inside a larger reference image. Then, a second image, referred to as the *current image*, is given to the estimation algorithm. The goal is to find the transformation parameters that, when applied to the current image, results in a *current template* that is equal to the reference template. In turn, these parameters allows us to align the images according to a common reference frame.

The transformation space considered in this case will be similar to the intensity-based case, composed of a geometric and a photometric part. This is because using the photometric transformation is crucial to the success of a intensity-based method where lighting conditions change.

Consider that the reference template is composed of m pixels. Also, consider that the feature matching algorithm provides n feature correspondences between the reference template and the current image. Ideally, it would be possible to find a vector $\mathbf{x}^* \in \mathbb{SL}(3) \times \mathbb{R}^2 = \{\mathbf{H}^*, \alpha^*, \beta^*\}$ such that:

$$\alpha^* \mathcal{I}(\mathbf{w}(\mathbf{H}^*, \mathbf{p}_i^*)) + \beta^* = \mathcal{I}^*(\mathbf{p}_i^*), \qquad \forall i = 1, \ldots, m \tag{5.1}$$

$$\mathbf{w}(\mathbf{H}^*, \mathbf{q}_j^*) = \mathbf{q}_j, \qquad \forall j = 1, \ldots, n \tag{5.2}$$

where $\mathcal{I}$ and $\mathcal{I}^*$ are the current and reference image functions, respectively; $\mathbf{w}$ is the warping operator; $\mathbf{p}_i^* \in \mathbb{P}^2$ contains the projective coordinates of the i-th pixel of the reference template; and $\mathbf{q}_j, \mathbf{q}_j^* \in \mathbb{P}^2$ are the projective coordinates of the j-th feature correspondence set in the current image and reference template, respectively.

The perfect calculation of $\mathbf{x}^*$ is, in practice, impossible due to a variety of reasons, including noise in the camera sensor and outliers in the feature matching. This leads to the reformulation of this problem as a nonlinear least-squares problem.

First, two separate cost-functions are defined: the Intensity-Based (IB) and the Feature-Based (FB) ones. These cost functions are the same as the ones that have been presented in the previous chapters. Each pixel i of the reference template contributes to the following

residual to the IB cost function:

$$y_i(\mathbf{x}) = \alpha\mathcal{I}(\mathbf{w}(\mathbf{H}, \mathbf{p}_i^*)) + \beta - \mathcal{I}^*(\mathbf{p}_i^*), \tag{5.3}$$

or in compact vectorial form:

$$\mathbf{y}_{IB}(\mathbf{x}) = \alpha\mathcal{I}(\mathbf{w}(\mathbf{H}, \mathbf{p}^*)) + \beta - \mathcal{I}^*(\mathbf{p}^*), \tag{5.4}$$

where $\mathbf{p}^*$ contains the projective coordinates of all points of the reference template, with abuse of notation.

The FB cost function receives the contribution of the distance between the features coordinates in each image:

$$y_j(\mathbf{x}) = \mathbf{w}(\mathbf{H}, \mathbf{q}_j^*) - \mathbf{q}_j = \begin{bmatrix} y_j^u \\ y_j^v \\ 0 \end{bmatrix} \tag{5.5}$$

where y_j^u, y_j^v are distance between the features in the u and v directions, respectively. The third element is disregarded since it is always zero. Concatenating all y_j, it is possible to write:

$$\mathbf{y}_{FB}(\mathbf{x}) = \mathbf{w}(\mathbf{H}, \mathbf{q}^*) - \mathbf{q}. \tag{5.6}$$

Using (5.4) and (5.6), the nonrobust unified nonlinear least squares problem can be compactly written as

$$\min_{\mathbf{x}=\{\mathbf{H},\alpha,\beta\}} \quad \frac{1}{2}\Big(w_{IB}\left\|\mathbf{y}_{IB}(\mathbf{x})\right\|^2 + w_{FB}\left\|\mathbf{y}_{FB}(\mathbf{x})\right\|^2\Big) \tag{5.7}$$

where w_{IB}, w_{FB} are carefully selected weights given to the intensity-based and feature-based components of the cost function, respectively. In this case, the weights respect the constraint $w_{IB} + w_{FB} = 1$.

Using the same Lie Algebra parametrization from (3.5), it is possible to reparametrize the transformation model into $\mathbf{x} = \mathbf{x}(\mathbf{z})$ with $\mathbf{z} = \{\mathbf{v}, \widetilde{\alpha}, \widetilde{\beta}\}$ as described in Section 3.2. Consider that an initial approximation $\widehat{\mathbf{x}} = \{\widehat{\mathbf{H}}, \widehat{\alpha}, \widehat{\beta}\}$ of the actual solution is given. Now, it is possible to rewrite the nonrobust unified minimization problem as:

$$\min_{\mathbf{z}=\{\mathbf{v},\widetilde{\alpha},\widetilde{\beta}\}} \quad \frac{1}{2}\Big(w_{IB}\left\|\mathbf{y}_{IB}(\mathbf{x}(\mathbf{z}) \circ \widehat{\mathbf{x}})\right\|^2 + w_{FB}\left\|\mathbf{y}_{FB}(\mathbf{x}(\mathbf{z}) \circ \widehat{\mathbf{x}})\right\|^2\Big) \tag{5.8}$$

Given that a normalization factor is included, a unified residual vector can thus be defined as:

$$\mathbf{y}_{UN} = \begin{bmatrix} \frac{w_{IB}}{m}\mathbf{y}_{IB} \\ \frac{w_{FB}}{2n}\mathbf{y}_{FB} \end{bmatrix} \tag{5.9}$$

and a more concise formulation is achieved:

$$\min_{\mathbf{z}=\{\mathbf{v},\widetilde{\alpha},\widetilde{\beta}\}} \quad \frac{1}{2}\left\|\mathbf{y}_{UN}(\mathbf{x}(\mathbf{z}) \circ \widehat{\mathbf{x}})\right\|^2. \tag{5.10}$$

5.2 Jacobian Matrices

The derivation of the Jacobian matrices used in the unified case are greatly simplified because most of the work has already been presented in the previous chapters. These matrices are naturally the combination of those used in the intensity- and feature-based cases, which can be found in (3.16) and (4.12).

$$\nabla_{\mathbf{z}}\mathbf{y}_{UN} = \mathbf{J}_{UN} = \begin{bmatrix} \frac{w_{IB}}{m}\mathbf{J}_{IB} \\ \frac{w_{FB}}{2n}\mathbf{J}_{FB} \end{bmatrix} \tag{5.11}$$

5.3 Weight Choices

The weights w_{IB} and w_{FB} should be carefully selected to ensure the best convergence properties for the algorithm. The proposed method used in this work is one of many possibilities, and the best choice is still an open research problem. Recall that the following constraint applies to the weights:

$$w_{IB} + w_{FB} = 1; w_{FB} > 0; w_{IB} > 0. \tag{5.12}$$

It is clear that only one of the weights needs to be defined, as the other one can be obtained from this constraint. The idea behind the proposed method for determining the weights is to let the feature-based error be more important to the optimization when the current solution is far from the solution point. As the FB error decreases, then the IB component becomes more important. This is consistent with the idea that the FB method is better suited to handle large displacements, while IB methods have better precision, but only work when the solution is close enough. Consider the root mean squared deviation (RMSD) associated with the feature-based error:

$$RMSD(\mathbf{y}_{FB}) = \sqrt{\frac{\sum_{j=1}^{n} d_j^2}{n}} = d_{FB} \tag{5.13}$$

where d_j is as defined in 4.13, and n is the number of feature matches. The proposed weight function is:

$$w_{FB} = 1 - e^{-d_{FB}} \tag{5.14}$$

This function allows for a continuous transition where the feature-based weight decreases as it gets lower, and the intensity-based component becomes gradually more important in the optimization.

5.4 Local versus Global Feature Search

The unified case also requires a method for improving the estimation speed, similar to the one presented for the feature-based case. It is necessary because a feature detection step is also present in the unified case. First, a current template is generated by warping the current image with the initial approximation $\widehat{\mathbf{H}}$. Then, this current template is assigned a score by comparing it with the reference template using the Zero-mean Normalized Cross-Correlated. If this score is higher than a predefined threshold, then the feature detection algorithm searches only this current template. Otherwise, the current template and $\widehat{\mathbf{H}}$ are discarded. In this case, the feature detection algorithm searches the entire current image for features. The first scenario is referred to as a local search and the second as a global search.

When the global search is used, it is necessary to recalculate an initial approximation $\widehat{\mathbf{H}}$. This is done by calculating the homography solely from the features matches between the current image and the reference template. The outlier rejection algorithm presented in Section 4.5 is also used here.

6 EXPERIMENTAL RESULTS

In this chapter, the homography estimation algorithms presented in Chapters 3, 4 and 5, and their variants, are tested and evaluated. They are also compared to existing algorithms. Section 6.2 explains the testing procedure used to generate the results presented in this chapter. Then, Section 6.3 demonstrates how the intensity-based algorithm presented is robust to large illumination changes. In Section 6.4, it is shown how the robust version of the intensity-based algorithm handles occlusions. Next, a general comparison the convergence domain of all algorithms is presented in Section 6.5. It is followed by an analysis of the rate of convergence of each algorithm presented here. Section 6.7 compares the processing speed that each algorithm is able to achieve, in order to verify their applicability in real-time applications. Finally, in Section 6.8 a discussion of the best algorithms is presented.

6.1 List of Algorithms Evaluated

The following algorithms are evaluated in this chapter. They are divided among intensity-based; feature-based and unified categories.

1. Intensity-based (IB) Methods:
   (a) **IB_0**: The classic version of the intensity-based ESM algorithm. It does not estimate the photometric transformation. It is provided as a precompiled library in (MALIS, 2011).
   (b) **IBG**: The *IBG* version of the algorithm presented in Chapter 3.
   (c) **IBG_P**: This is the "Regular + Predictor" version of the algorithm presented in Chapter 3.
   (d) **RIBG**: The "Robust" version of the algorithm presented in Chapter 3.
   (e) **RIBG_P**: This is the "Robust + Predictor" version of the algorithm presented in Chapter 3.
2. Feature-based (FB) Methods:
   (a) **OPENCV**: An algorithm is based on the open-source implementation of the `cv::findHomography` function in the OpenCV library (BRADSKI, 2000). It uses the same feature detection and extraction steps as the algorithm implemented in Chapter 4, but replaces the actual estimation algorithm with the aforementioned OpenCV function. Additionally, it uses the RANSAC robust method (FISCHLER; BOLLES, 1981) to handle outliers.

(b) **FB_0**: The estimation method presented in Chapter 4.

3. Unified Methods:

   (a) **UNIF**: The estimation method presented in Chapter 5.

   (b) **UNIF_P**: The estimation method presented in Chapter 5 with a ZNCC Sliding Window predictor.

6.2 Validation Setup

Figure 6.1: The image used for the validation procedure, with the reference template annotated.

To validate the algorithm, the same testing procedure used by (BAKER; MATTHEWS, 2001) is implemented. It consists of generating progressively larger known perturbations to an image, and feeding the perturbed images to the estimation algorithms. This perturbation can be in any of the transformation parameters, but isn't restricted to them. Since the actual transformation is always known, it can be compared to the result given by the estimation algorithm.

To generate perturbations in the geometric transformation space, the procedure is as follows. First, a reference image is chosen and a region of size 100×100 pixels is selected as the reference template, as shown in Fig. 6.1. This template size is used in all the experiments shown in this chapter, unless otherwise stated. The set of coordinates of the four reference template corners $\mathbf{p}^*$ is separately perturbed in the $\overrightarrow{u}$ and $\overrightarrow{v}$ direction with a zero mean Gaussian noise and standard deviation of σ pixels, obtaining a perturbed set of corners $\mathbf{p}$. The relation between $\mathbf{p}^*$ and $\mathbf{p}$ defines a test homography $\overline{\mathbf{H}}$.

The reference image is then transformed using $\overline{\mathbf{H}}$. Figure 6.2 shows some sample of perturbed images with varying σ. The algorithm receives the reference template and the transformed image with the zero element $\mathbf{z} = \{\mathbf{v}, \alpha, \beta\} = \mathbf{0}$ as the initial guess for

(a) Examples of images perturbed with $\sigma = 1$.

(b) Examples of images perturbed with $\sigma = 5$.

(c) Examples of images perturbed with $\sigma = 10$.

Figure 6.2: Different geometric perturbation levels showcase.

the photogeometric transformation. Recall that the zero element from the Lie Algebra $\mathbf{v} = \mathbf{0}$ maps to the identity element $\mathbf{H} = \mathbf{I}$. From this input, the estimation algorithm produces an estimated homography $\widehat{\mathbf{H}}$. This is used to transform each reference template corner point to evaluate the quality of this result. If the average residual error in pixels between the actual perturbed corner points and the estimated perturbed ones is less than a predefined value, the result is considered to have converged. 1,000 test cases are randomly generated for each value of the perturbation $\sigma \in [0, 20]$ and used as input for each version of the algorithm that is evaluated.

A similar procedure is used to generate perturbations in the photometric parameters α and β. In this case, however, the perturbations are generated directly on their values. The perturbations levels for these parameters are defined by the standard deviations σ_α and σ_β. Whenever only σ is used, it refers to the geometric component.

6.3 Robustness to Illumination Changes

In Chapter 3, an intensity-based algorithm was proposed that is able to handle large illumination changes. In order to test this claim, an experiment was designed that generated different perturbations levels in three dimensions: geometric ($\mathbf{H}$), gain (α) and bias (β).

(a) Examples of images perturbed with $\sigma_\alpha = 0.05$.

(b) Examples of images perturbed with $\sigma_\alpha = 0.25$.

(c) Examples of images perturbed with $\sigma_\alpha = 0.45$.

Figure 6.3: Different contrast perturbation levels showcase.

The perturbations were zero mean Gaussian noise with standard deviation σ_α or σ_β added to the default values of $\alpha = 1$ and $\beta = 0$. The standard deviation value σ_α was increased in steps of 0.05, therefore it varied from 0 to 0.45. For σ_β, a step of 0.5 was used, and its value varied from 0.0 to 4.5. Figures 6.3 and 6.4 display some examples of perturbations the contrast (gain) and brightness (bias) levels, respectively.

The test mixed perturbations in α, β and $\mathbf{H}$ equally. 10 perturbations levels were used for all variables totaling 1,000 possible combinations for each $\sigma_\alpha, \sigma_\beta, \sigma_H$. For each combination, 10 test cases were generated. This implies that for each perturbation level in any of the variables, there are 1,000 test cases. Algorithm 1 shows how the experiment was structured. In these tests, 3 levels of the multiresolution pyramid are used. In each level, a maximum of 3 iterations of the algorithm are allowed to execute. The threshold for convergence was 1 pixel.

Two algorithms were tested in this experiment. The first one is *IB_ 0*, that does not estimate the photometric parameters jointly with the homography matrix. The second one is *IBG*. This algorithm does estimate the gain and bias parameters for each image.

Figure 6.5 displays the results of this experiment. In 6.5a, the frequency of convergence of both algorithms is shown for each magnitude of the perturbation σ that was applied to the corners of the template. There are 1000 test cases for each magnitude of perturbation. Note that the *IB_ 0* algorithm has a consistently lower frequency of convergence than the *IBG* version of the algorithm proposed in this work. This value decreases for both cases as the geometric perturbation increases, which is expected.

(a) Examples of images perturbed with $\sigma_\beta = 0.5$.

(b) Examples of images perturbed with $\sigma_\beta = 2.5$.

(c) Examples of images perturbed with $\sigma_\beta = 4.5$.

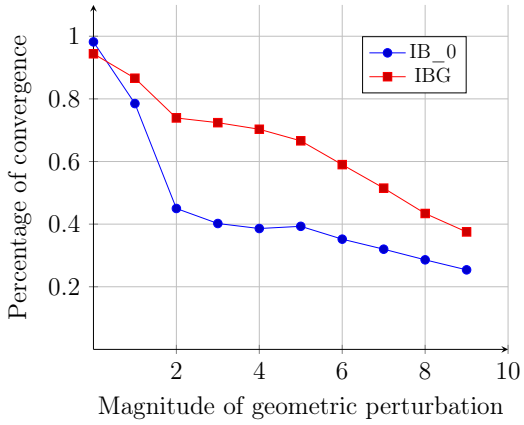
Figure 6.4: Different brightness perturbation levels showcase.

Figure 6.5b shows the frequency of convergence for varying levels of the perturbation in the gain α. Note that the default value of this parameters is 1.0 and it multiplies the entire image. Again, the *IBG* algorithm consistently outperforms the *IB_0* version. In this case, for both algorithms the frequency decreases as the magnitude of perturbation in the gain increases.

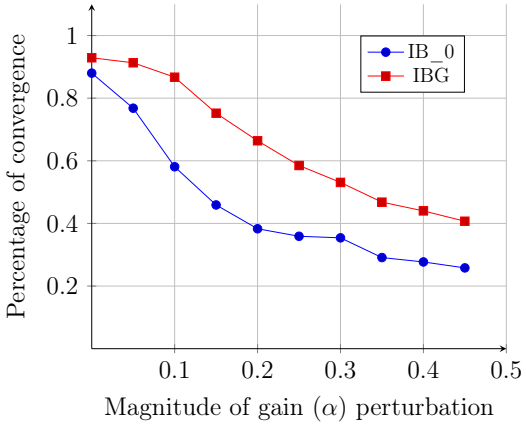
Finally, Fig. 6.5c presents an interesting result. While it confirms once again that the *IBG* algorithm is better than the *IB_0* version, it also shows that both algorithms are insensitive to increases in the magnitude of perturbation in the bias parameter. This indicates that the contrast in images is more decisive than brightness for the success of the intensity-based algorithm.

Algorithm 1 Arbitrary Illumination Test

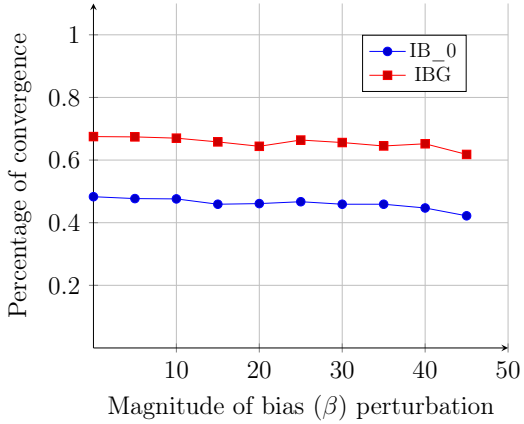
```
1: procedure TestIllumination
2:     for σ ← 0, ..., 9 do
3:         for σ_α ← 0, 0.05, ..., 0.45 do
4:             for σ_β ← 0, 5, ..., 45 do
5:                 for i ← 1, ..., 10 do
6:                     T ← GENERATE_TEST(σ, σ_α, σ_β)
7:                     for every method do
8:                         RUN_TEST(T)
```



(a) % of convergence vs geometric perturbation levels.



(b) % of convergence vs α perturbation levels.

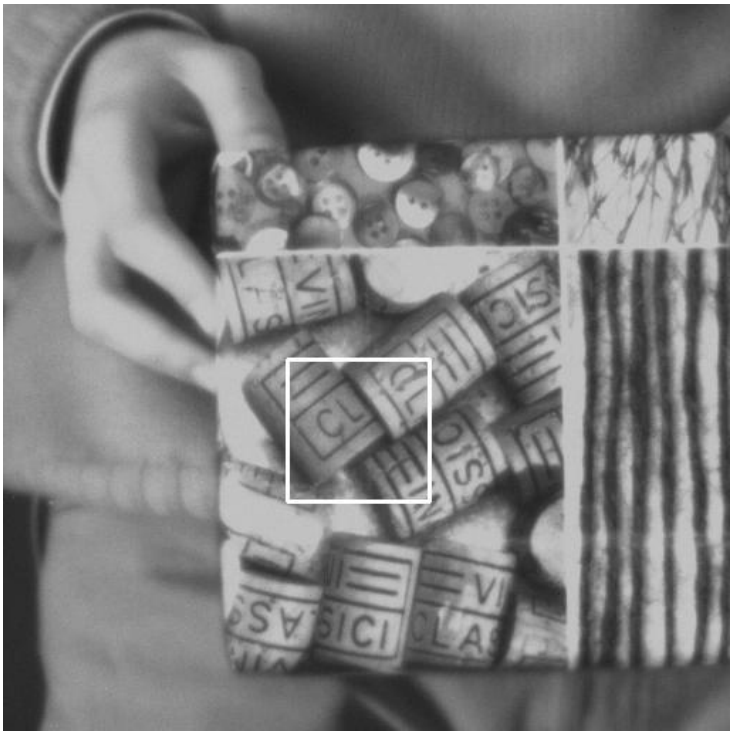


(c) % of convergence vs β perturbation levels.

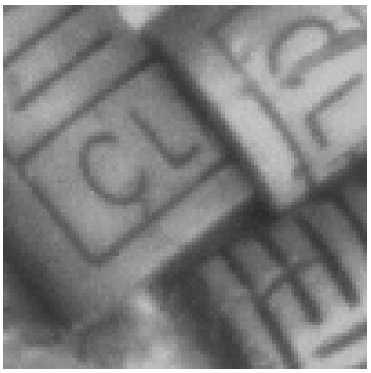
Figure 6.5: Frequency of convergence for different photogeometric perturbation levels.

6.4 Robustness to Unknown Occlusions

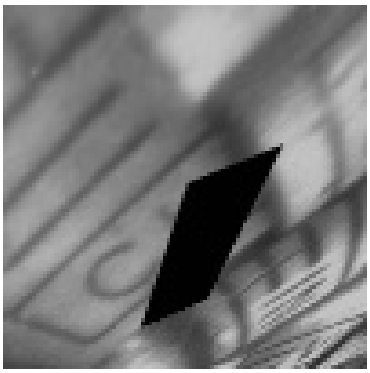
In Chapter 3, a version of the intensity-based algorithm was presented that implemented a robust optimization using M-estimators in order to handle occlusions. This was considered the robust mode of the algorithm. This section shows how this algorithm performs better



(a) Reference image annotated with reference template.



(b) Unoccluded reference template

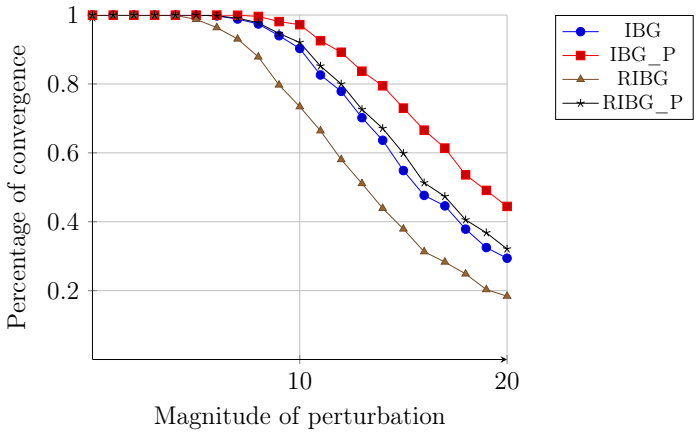


(c) Transformed 10%-occluded template

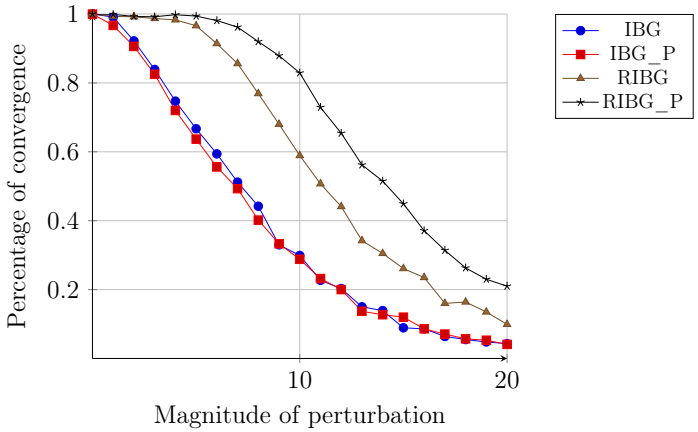
Figure 6.6: Images and Templates used for Occlusion Testing.

than the regular mode when the current image contains occlusions.

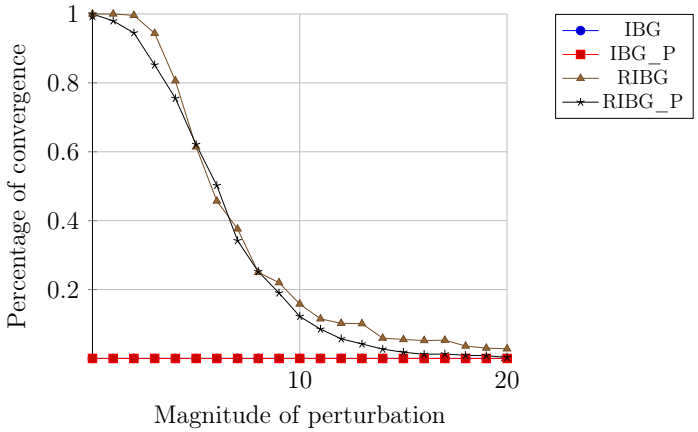
In order to test the robustness to occlusion of the algorithm, the testing procedure is enhanced. In additional to the perturbation described in section 6.2, synthetic occlusions are also added to the current image. Two different occlusion sizes are used. Both cases use an all-black occluder and a rectangular shape. The first one has 20×50 pixels, whilst the second, 40×50 pixels. They correspond to 10% and 20% occlusions of the reference template, respectively. These occlusions are applied before transforming the reference image with the test homography, and are randomly positioned around the center of the reference template. The estimation procedure then receives the unnocluded reference template and the transformed occluded image as an input. Figure 6.6 shows an example of this template pair for a perturbation of $\sigma = 20$ pixels with 10% occlusion. In these tests, 3 levels of the multiresolution pyramid are used. In each level, a maximum of 5



(a) No occlusion



(b) 10% occlusion



(c) 20% occlusion

Figure 6.7: Frequency of convergence for different setups and occlusion levels.

iterations of the algorithm are allowed to execute. The threshold for convergence was 1 pixel.

Figure 6.7 presents the frequency of convergence of the algorithm (over 1,000 cases for each magnitude of perturbation) for different setups. The *IBG* setup uses the default processing. The *IBG_P* setup makes use of a 2D sliding window approach to initialize the position of the template on the current image before starting the optimization procedure. The *RIBG* setup uses the strategies outlined in Section 3.5 to give weights to pixels in the image. Lastly, the *RIBG_P* setup uses both setups together.

The same figure shows the frequency of convergence data for each setup with varying levels of occlusion. In the scenario with no occlusion (Fig. 6.7a), the best performance comes from *IBG_P*. In this case, the *RIBG* algorithm suffers from a smaller basin of convergence. This occurs because it is always throwing away some piece of data, even if there is no occlusion. The importance of the *RIBG* can already be seen in the 10% occlusion scenario (Fig. 6.7b). Indeed, nonrobust methods suffer because they cannot handle occlusions appropriately. The best algorithm is *RIBG_P* in this case. For the last scenario of 20% occlusion (Fig. 6.7c), nonrobust methods fail completely, while the robust methods still work for relatively small perturbations.

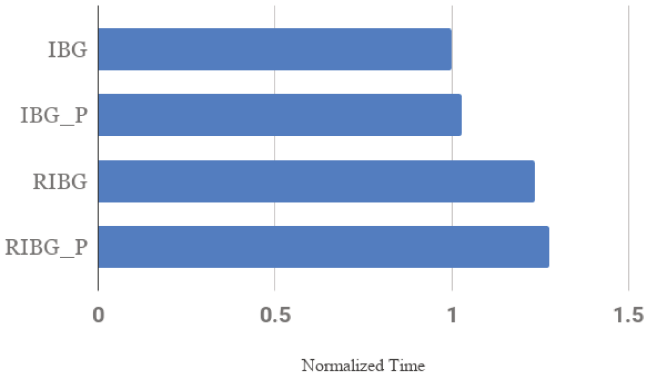


Figure 6.8: Average processing time for each setup using "Regular" as reference.

Figure 6.8 shows the comparison of the average time needed by each setup to process an image when the occlusion is at the 10% level. The data is normalized using the *IBG* setup timing value as the reference. It is clear that robust methods are slightly more computationally costly ($\approx$ 25%). This represents a trade-off to be decided by the end user.

6.5 Convergence Domain

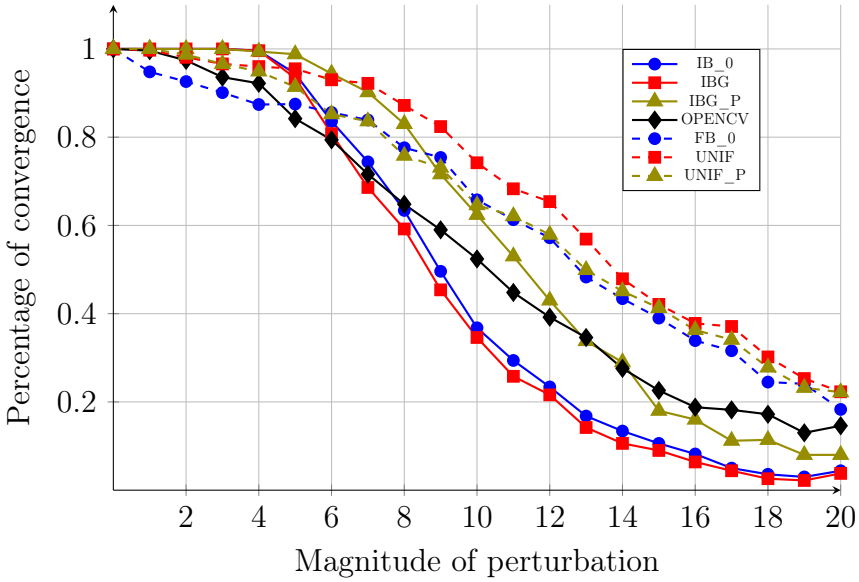


Figure 6.9: Frequency of convergence versus magnitude of perturbation for different homography estimation algorithms.

This section compares the algorithms developed in this work with regards to their convergence domain. The convergence domain dictates how the algorithm handles increasingly large perturbations. The same testing method presented in Section 6.2 is used. In this test, no perturbation was generated in the photometric parameters α and β. 3 pyramid levels with 2 iterations in each level are used. In this case, the convergence threshold was 1.5 pixels. This slight increase in value was necessary because the FB algorithms are not precise enough. All algorithms presented in Section 6.1 are compared, except for the robust to occlusion versions ($RIBG$ and $RIBG\_P$). They were not considered because this experiment does not involve occlusions.

The results of the experiment can be seen in Figure 6.9. It shows that the unified algorithms have a larger convergence domain than the FB or IB versions. It can also be observed that the IB_0 and IBG algorithm have very similar performance. This is expected because in this experiment there are no lighting changes, which would make IB_0 suffer since it does not estimate the photometric parameters.

The use of the ZNCC predictor in the unified case effectively decreases its frequency of convergence. This observation seems counter-intuitive, but is explained by looking at the convergence rate analysis. However, the $UNIF\_P$ is still the second-best algorithm when considering only the frequency of convergence.

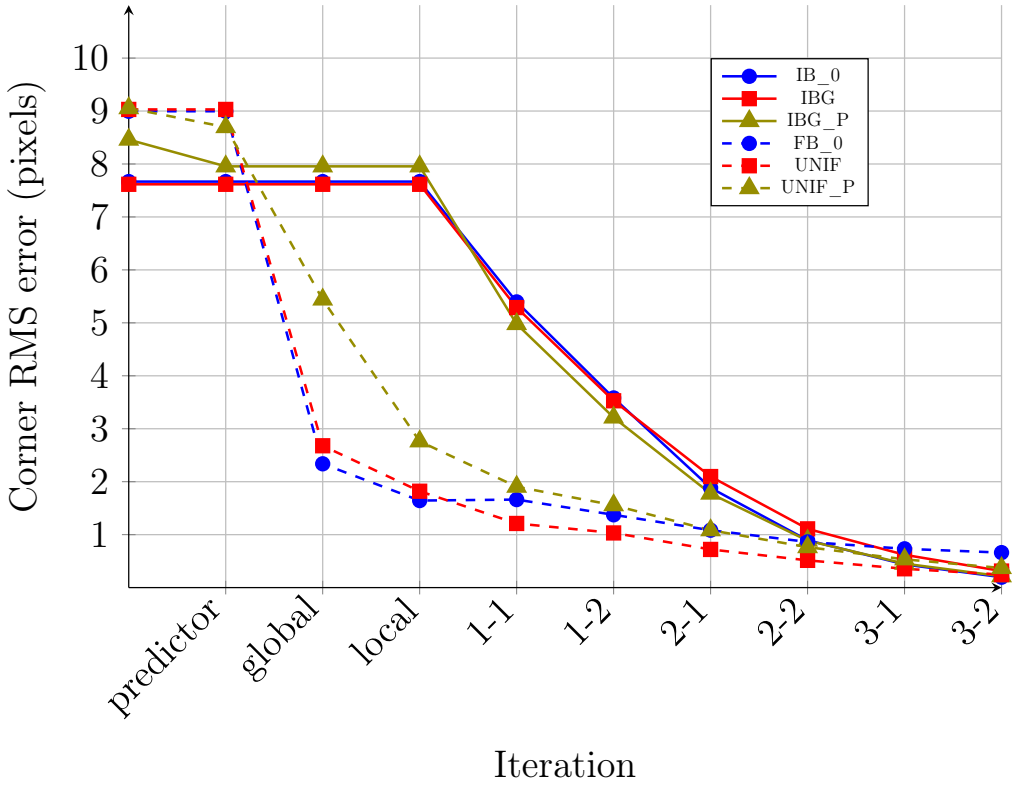


Figure 6.10: Rate of convergence after each optimization iteration for different homography estimation algorithms with a magnitude of perturbation $\sigma = 10$.

6.6 Convergence Rate

Figure 6.10 compares the rate of convergence for the homography estimation algorithms under a perturbation of magnitude $\sigma = 10$. This rate is displayed as the progression of the root mean squared (RMS) error between the coordinates of the 4 corners of the reference template and the estimated transformed current template. Out of the 1000 test case, only those where the estimation converged are considered here. Note that the **OPENCV** algorithm is omitted because it was used as a black-box and therefore the sequence of homographies in each iteration cannot be accessed.

The x-axis of Figure 6.10 contains each important step in the optimization. The first step, labeled "predictor" is the result of the ZNCC prediction step. The second step, labeled "global" is the step where the algorithm decides to search the entire current image for feature and obtains a new approximation. The next step, "local", refers to the local feature search. These last two steps are described in Section 4.4. They are not always present in all the algorithms. The following steps are regular steps in the iterative optimization method. They are separated by pyramids level. In these cases, the designation "X-Y" means: pyramid level X, iteration Y.

The convergence rate graph allows several observations regarding the algorithms presented in this work. First, the FB_0 performance is very dependent on the "global" step. After this step, it is the algorithm with the best RMS value. However, it is incapable

of decreasing this value too much in the subsequent optimization steps. By the time the other algorithms reach the third level of the pyramid, they surpass the RMS of the FB_0 algorithm.

The behaviour of IB_0, IBG and IBG_P are very similar and showcases the second-order nature of the optimization methods. A small difference between them is that the IBG_P does use the "predictor" step. Thus, it is able to converge even for cases with a slightly higher initial RMS error. After the prediction step, however, the three algorithms are very similar.

From the graph, it can also be observed how the Unified methods have a behaviour that mixes the FB and IB methods, as expected. The use of the ZNCC predictor stage in the unified (*UNIF_P*) method leads decreases the number of occasions where the "'global" step is actually used, when compared to the *UNIF* method. This is observed by the higher error value in global step of the the *UNIF_P* method. It explains why the use of the predictor actually decreases the frequency of convergence, as seen in the previous section. However, this decrease in the usage of the global step also lead to a improvement in the processing time, as shown in the next section.

6.7 Timing Analysis

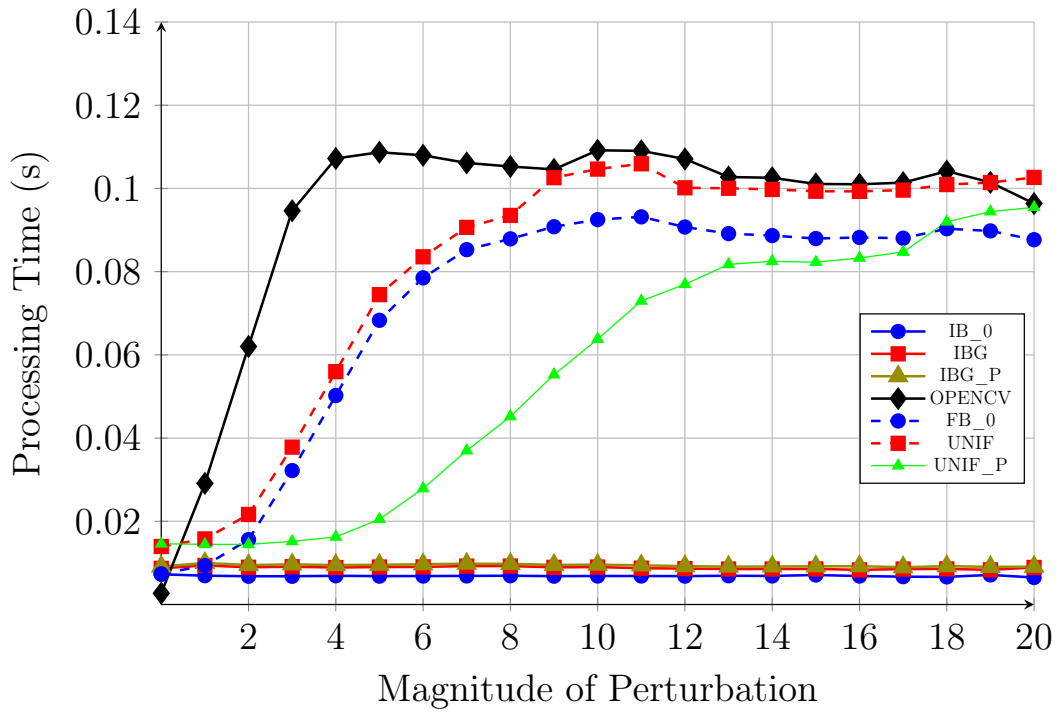


Figure 6.11: Processing time variation with respect to increasing perturbation levels.

Figure 6.11 shows how the average time needed to run the estimations algorithms varies depending on the magnitude of perturbation. This time was measured in a Intel i7-6700HQ processor. This time is averaged over the subset of the 1000 cases where

the estimation converged. The most noticeable aspect of this graph is that pure IB algorithms have nearly constant time, independent of the perturbation level. In constrast, the algorithms that have a feature-based component need more time to process images with higher perturbation levels.

This phenomenon can be explained by considering the effect of the global and local feature searches. As the perturbation level increases, the number of occasions where the algorithm decides to use the global search also increases. The global step, however, is very computationally expensive. The UNIF_P manages to have a lower processing time because the prediction step increases the probability that the local search will be used instead of the global one. Therefore, the UNIF_P can be seem as a compromise between having the advantage of being capable of performing global search, without taking a big penalty in processing time.

However, the results in this graph ultimately show that more work is needed to create a method that is able to reliably perform in real-time settings even for large perturbations. The IB methods are already capable of that, requiring less than 0.02 seconds per image. However, the FB and Unified methods may need up to 0.1 seconds, which in some applications may be unnacceptable.

6.8 Comparison Discussion

Considering the three criteria presented in previous sections, some of the algorithms emerge as better options than others. In particular, the *UNIF_P* and *IBG_P* stand out. First, the *UNIF_P* has the precision level of pure intensity-based methods, with a larger convergence basin. Considering the processing time, it is a better option than the *UNIF* method because the predictor decreases the probability that the global feature search will be necessary, with a small loss in the convergence basin size.

If the application domain guarantees a smooth camera-scene relative motion and a good initialization, then the *IBG_P* is an appropriate option because its processing time is much lower than any of the methods involving a feature-based component. The processing time is also independent of the perturbation level. Considering also the frequency of convergence, it is possible to conclude that *IBG_P* is the best method in its class of intensity-based methods and suitable to several robotic applications, as will be shown in the next chapter.

7 USE CASES

The algorithms developed in this project are available as a ready to use C++ library and as a ROS package. The ROS package is available at the link below:

`https://github.com/lukscasanova/vtec_ros`

An accompanying technical report can be found in (NOGUEIRA; PAIVA; SILVEIRA, 2019). This distribution and documentation allows the algorithms to be used in different applications and by different researchers. This section presents some of these applications.

7.1 Intensity-Based Visual Tracking Robust to Global Illumination Changes

This application uses the intensity-based homography estimation algorithm presented in Chapter 3 (IBG_P) to track a planar surface in a video sequence. The result is available at the link:

`https://www.youtube.com/watch?v=FFRPL2mo0f0`

This experiment shows how the visual tracking is able to handle large and abrupt illumination changes that arise when the lights in the room are switched on and off. It also undergoes relatively large translational and rotational movements in order to show the robustness of the algorithm.

Figure 7.1 shows some sample screenshots of the tracking application. The first one is before the initialization of the algorithm. A white square indicates to the user the area that will be tracked if tracking starts at that moment. The user then presses the “S" key and tracking starts. The larger image shows the current image annotated with a black rectangle that indicates the location of the tracked region. The top-left image shows the current template and the mid-left image shows the reference template. The remaining screenshots show different poses where the algorithm successfully tracked the planar region. Note that the last screenshot shows a situation where the lights are off, and the target is almost indiscernible, but the visual tracking succeeds nonetheless.

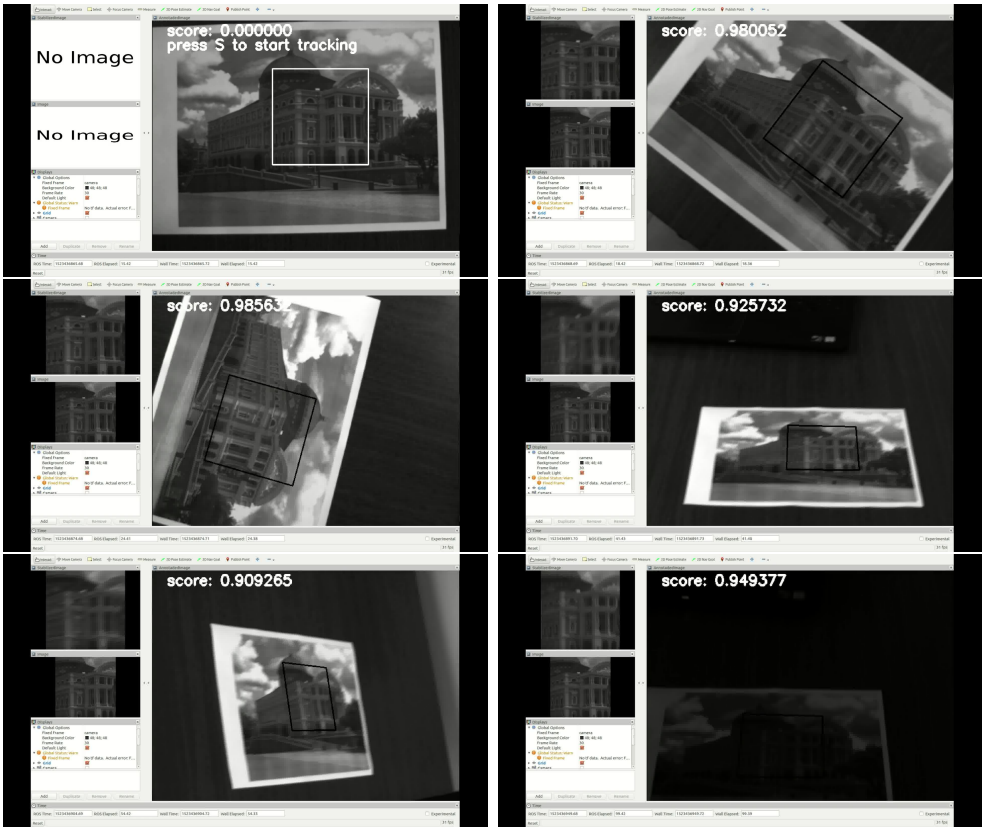


Figure 7.1: Visual tracking robust to illumination changes.

7.2 Direct Visual Tracking Robust to Unknown Occlusions

The results of visual tracking using a nonrobust and a robust version of the intensity-based algorithm are compared using the same video stream. It is available at the link:

`https://youtu.be/qhAFe8IbIHc`.

On one hand, Fig. 7.2 shows that the nonrobust algorithm (IBG_P) fails quickly even if the occlusion is small. On the other hand, Fig. 7.3 shows that the robust algorithm ($RIBG\_P$) performs that task even with increasing occlusion levels. Two types of occlusion are present in this video: A synthetic occlusion composed of an all-black square that is added to each image frame always at the same coordinates; and physical occlusions in the form of real coins that are progressively added to the tracked image.

7.3 Unified Visual Tracking

The $UNIF\_P$ algorithm was used to create a visual tracking application. A prediction step was used, as is recommended for all real-time tracking application. Results are

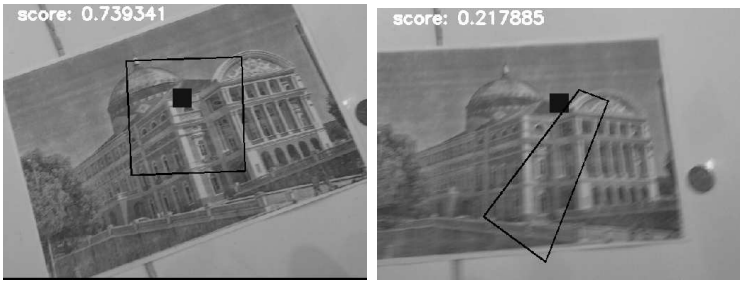


Figure 7.2: Visual tracking not robust to partial occlusions.

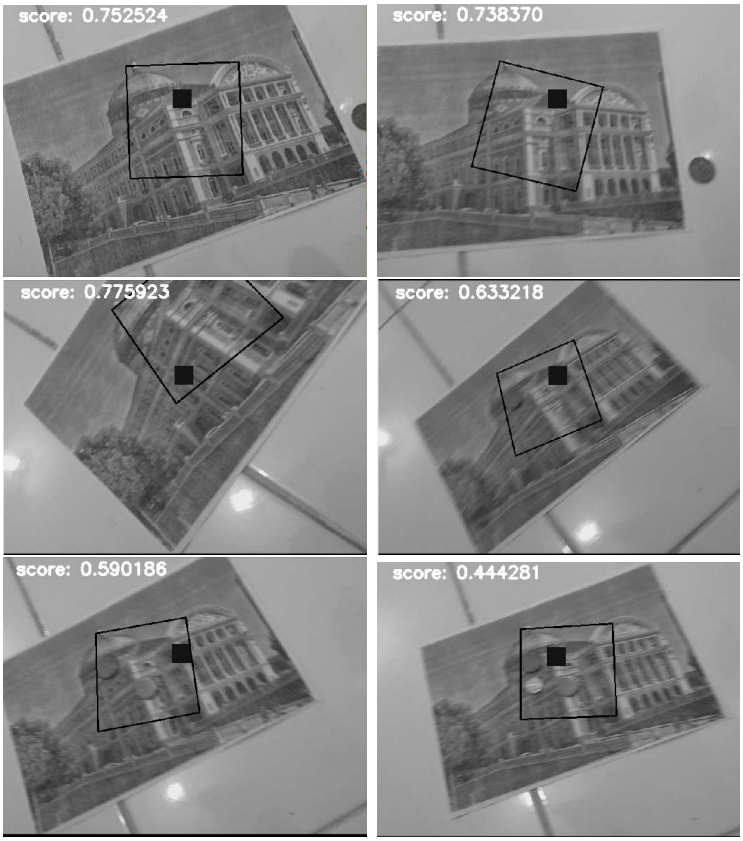


Figure 7.3: Visual tracking robust to partial occlusions.

available at the link:

`https://youtu.be/oArw449qp1E`

.

Figure 7.4 shows some samples of the tracking results. An interesting result is that this visual tracker can recover from complete failure. Even after completely removing the tracked region from the current image, the tracker later recovers, due to the feature-based component ability to perform the "global" feature search. Additionally, it can be seem that the algorithm is robust to large illumination changes, and that in some cases it can recover from complete failure even under severe illumination changes.

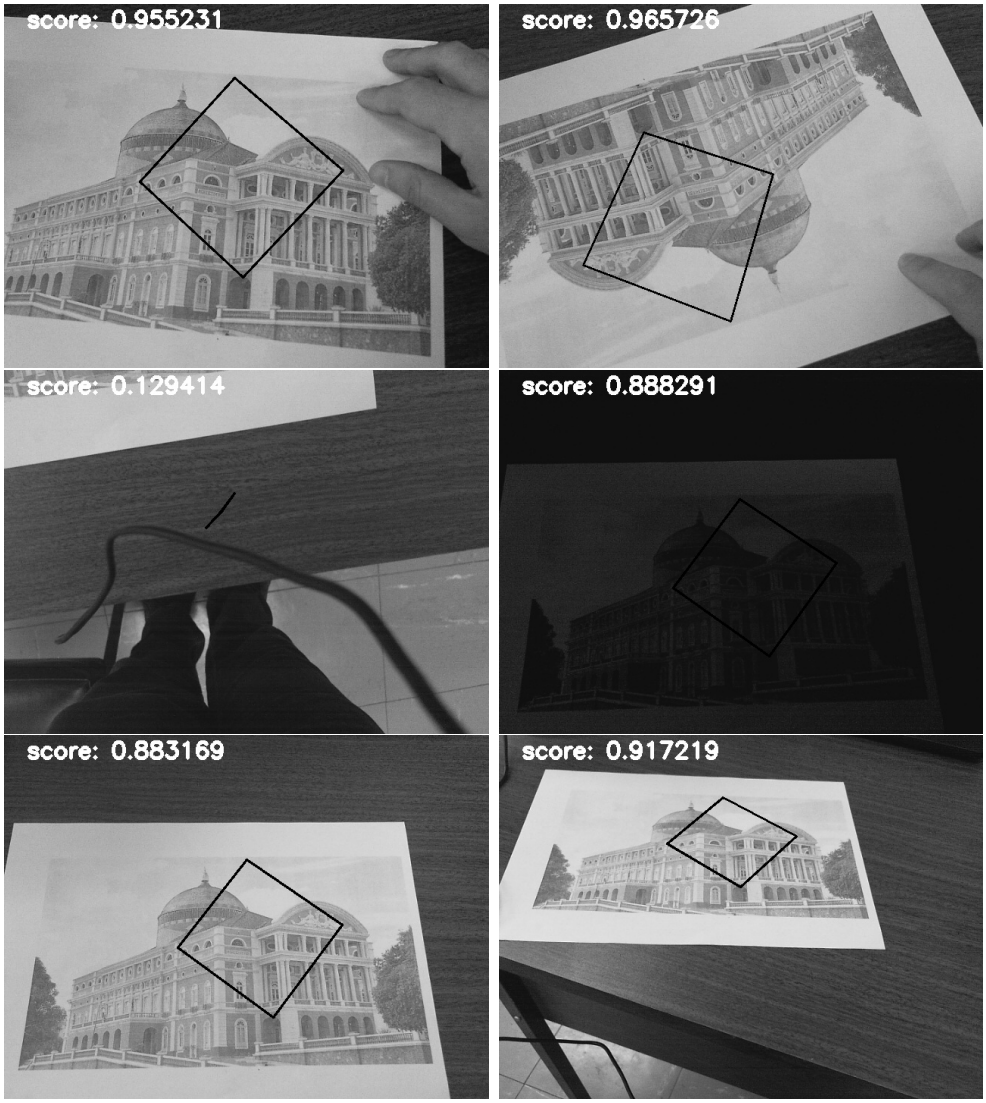


Figure 7.4: Samples of unified visual tracking.

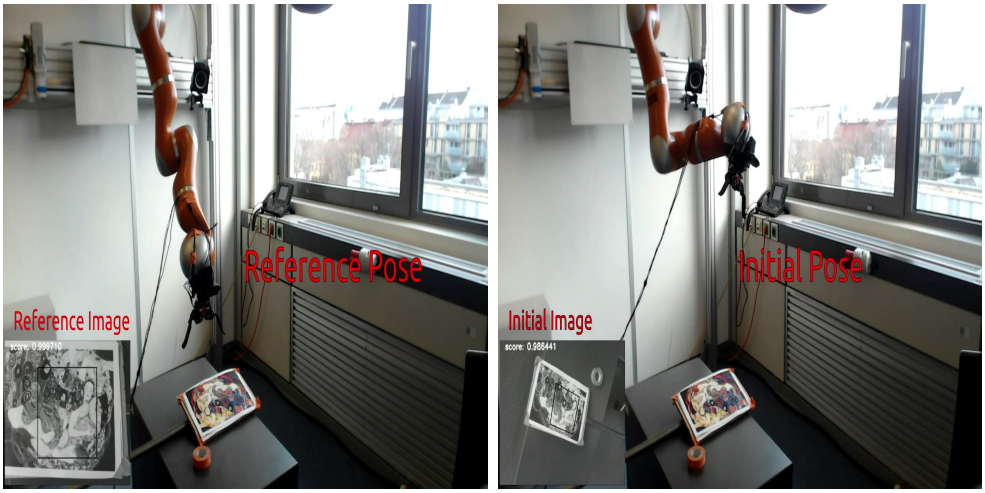
7.4 Direct Visual Servoing

This application is developed at the V4R group at the Vienna University of Technology (TUW) in Austria. A target is tracked using the intensity-based *IBG* setup. The estimated homography is used for controlling a 7-DoF industrial arm with a variant of the visual servoing technique proposed by (SILVEIRA; MIRISOLA; MORIN, 2018). The video is available at

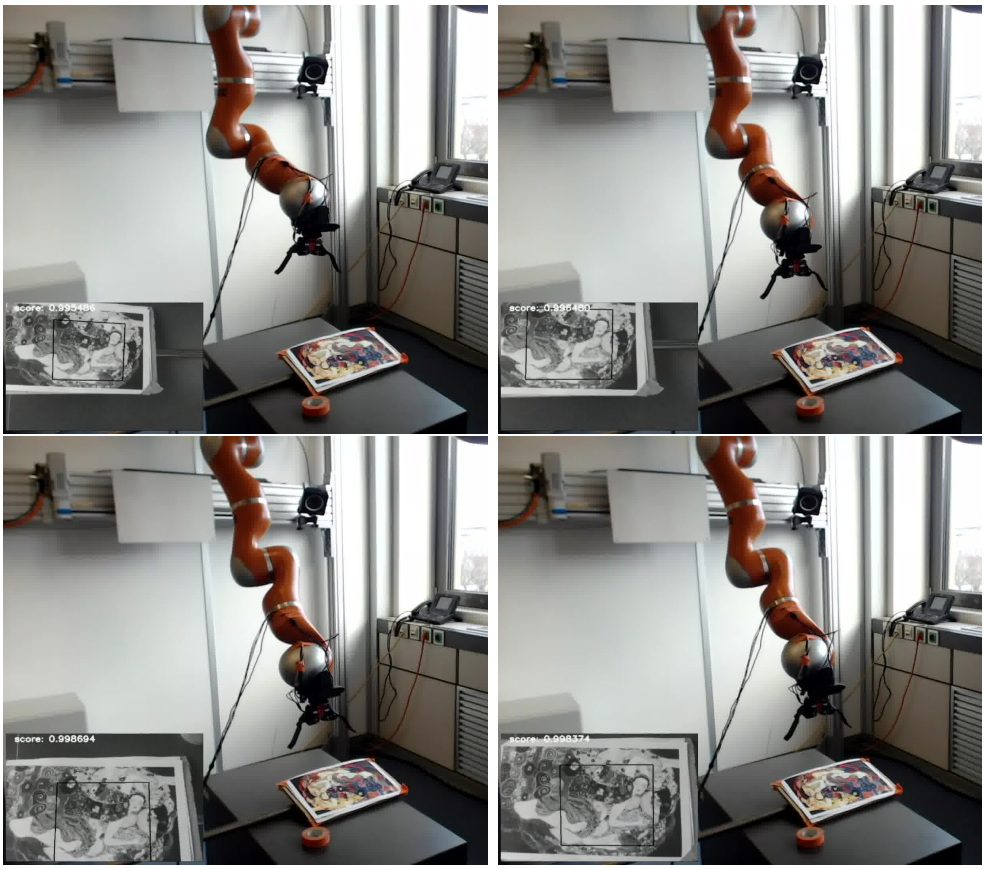
`http://tinyurl.com/vs-tuw`

This experiment shows that the servoing is robust to global illumination changes, and to errors of 50% in the camera intrinsic parameters (NEUBERGER et al., 2019). In this application, the camera is mounted on the robot end-effector, and the reference pose is completely defined by a reference image. Afterwards, the arm is repositioned at a different initial pose and then visual servoing is performed to drive the camera pose back

to reference one.



(a) Reference and initial camera poses with respect to the observed object/scene, respectively. Corresponding images are shown in their bottom left corner.



(b) Excerpts of the robot evolution towards its convergence.

Figure 7.5: Example of visual servoing application.

8 CONCLUSIONS AND FUTURE WORKS

This dissertation investigated the problem of homography estimation using a image registration paradigm. The goal was to build a hybrid method that was capable of unifying the intensity-based and feature-based approaches. In the path to accomplishing this goal, each of these approaches were considered separately in order to better understand them and their most glaring challenges. The contributions to each class of algorithms are summarised here.

In Chapter 3, a refinement for existing intensity-based algorithms based on the Efficient Second-order Minimization framework was proposed. Namely, two improvements were verified: robustness to global illumination changes and to occlusions. The former was accomplished by introducing photometric parameters into the optimization cost function, in addition to the standard geometric parameters. The latter required the use of M-estimator in order to calculate which pixels were statistical outliers, and therefore considered as occluded.

In Chapter 4, a reformulation of the feature-based approach was proposed in order to accomodate it in the ESM framework. This was a necessary step in the process of creating a unified algorithm. An important result of this formulation is that the Jacobian matrix in the feature-based case is greatly simplified using the Lie algebra formulation. It accomplishes a second-order approximation using only information from the reference image for the Jacobian, which is constant throughout the optimization procedure.

Finally, Chapter 5 presented a unified intensity- and feature-based approach for homography estimation, based on the approaches devised in the previous chapters. It was shown that this method was able to achieve a higher convergence domain among the algorithms proposed in this work. Its precision level was comparable to the IB approaches. When used in visual tracking, it displayed the property of being capable of recovering from complete failure, which was not possible natively using only the IB approach. However, the processing time for this unified approach is highly dependent on the interframe displacement, which may be an impediment for some robotic applications.

The body of work presented in this dissertation is intended as a first step towards building a unified approach for homography estimation, and visual estimation more generally. Some of the results point towards future research directions. One possibility is further investigating the feature-based approach in order to improve its performance. A better outlier rejection method, for instance, could greatly increase the stability of the unified algorithm. A traditional method that could be used for this purpose is RANSAC, which is used in the OpenCV implementation of the homography estimation function.

Still regarding the feature-based approach, a more systematic approach can be used in order to evaluate different feature detectors and descriptors. The validation setup proposed in Section 6.2 will facilitate this comparison analysis. The currrent implementation could also be enhanced to provide a modular approach that would allow

the user to set different descriptor/detection algorithms at runtime. This philosophy could also be applied in exposing more of the hyperparameters to the application developers. This would allow them to better tailor the algorithm towards their domain.

Other future works include extending the proposed methods to handle different image types, such as color images and those created by omnidirectional camera. In addition, the extension of the transformation model from global illumination changes to arbitrary illumination changes is yet another implementation possibility.

Regarding the robotics research community, an interest application of the proposed method is to use them in a sensor-fusion context. For instance, using a inertial measurement unit could help provide a better initialization for the optimization algorithms. This would in effect work as a predictor for the algorithm, and therefore further increase its convergence domain.

In conclusion, the work done towards this Masters dissertation is a first step in the development of a efficient and robust unified approach for visual estimation. There are promising results and immediate applications, but also several improvements that can be achieved in the short and mid term. The algorithms proposed and its validation framework will hopefully be used as a platform for future research, specially as they are made available to the global community.

Bibliography

AHMED, Faraz; ERMAN, Jeffrey; GE, Zihui; LIU, Alex X; WANG, Jia; YAN, He. Detecting and localizing end-to-end performance degradation for cellular data services. **IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications**, pp. 1–9, 2016.

ALISMAIL, Hatem; BROWNING, Brett; LUCEY, Simon. Robust Tracking in Low Light and Sudden Illumination Changes. **2016 Fourth International Conference on 3D Vision (3DV)**, pp. 389–398, 2016.

BAKER, Simon; MATTHEWS, Iain. Equivalence and efficiency of image alignment algorithms. vol. 1, pp. i–1090, 2001.

_____. Lucas-kanade 20 years on: A unifying framework. **International journal of computer vision**, vol. 56, 2004.

BARTOLI, Adrien. Groupwise geometric and photometric direct image registration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, vol. 30, no. 12, pp. 2098–2108, 2008.

BAY, Herbert; TUYTELAARS, Tinne; VAN GOOL, Luc. Surf: Speeded up robust features. **European conference on computer vision**, pp. 404–417, 2006.

BENHIMANE, Selim. **Vers une approche unifiée pour le suivi temps réel et l'asservissement visuel**. 2006. PhD thesis – Paris, ENMP.

BENHIMANE, Selim; MALIS, Ezio. Homography-based 2D Visual Tracking and Servoing. **The International Journal of Robotics Research**, vol. 26, no. 7, pp. 661–676, 2007.

_____. Real-time image-based tracking of planes using efficient second-order minimization. **2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)**, vol. 1, pp. 943–948, 2004.

BERGEN, James R; ANANDAN, Patrick; HANNA, Keith J; HINGORANI, Rajesh. Hierarchical model-based motion estimation. **European conference on computer vision**, pp. 237–252, 1992.

BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000.

BROWN, L. A survey of image registration techniques. **ACM computing surveys**, vol. 24, no. 4, pp. 325–376, 1992.

CANCLINI, Antonio; CESANA, Matteo; REDONDI, Alessandro; TAGLIASACCHI, Marco; ASCENSO, João; CILLA, Rodrigo. Evaluation of low-complexity visual feature detectors and descriptors. **2013 18th International Conference on Digital Signal Processing, DSP 2013**, pp. 1–7, July 2013. DOI: 10.1109/ICDSP.2013.6622757.

CHEN, Colin. Bayesian adaptive nonparametric M-regression. **Statistics and Its Interface**, vol. 2, pp. 71–81, 2009. ISSN 1938-7989.

CHEN, Lin; ZHOU, Fan; SHEN, Yu; TIAN, Xiang; LING, Haibin; CHEN, Yaowu. Illumination insensitive efficient second-order minimization for planar object tracking. **2017 IEEE International Conference on Robotics and Automation (ICRA)**, pp. 4429–4436, 2017.

COUGHLAN, James M; YUILLE, Alan L. Manhattan world: Compass direction from a single image by bayesian inference. **Proceedings of the Seventh IEEE International Conference on Computer Vision**, vol. 2, pp. 941–947, 1999.

DETONE, Daniel; MALISIEWICZ, Tomasz; RABINOVICH, Andrew. Deep Image Homography Estimation, 2016.

ENGEL, Jakob; SCHÖPS, Thomas; CREMERS, Daniel. LSD-SLAM: Large-scale direct monocular SLAM. **European conference on computer vision**, pp. 834–849, 2014.

EVANGELIDIS, Georgios D; PSARAKIS, Emmanouil Z. Parametric image alignment using enhanced correlation coefficient maximization. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, vol. 30, no. 10, pp. 1858–1865, 2008.

FAUGERAS, O.; LUONG, Quang-Tuan; PAPADOPOULO, Theo. **The geometry of multiple images**. [S.l.]: The MIT Press, 2001.

FISCHLER, Martin A; BOLLES, Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, ACM, vol. 24, no. 6, pp. 381–395, 1981.

FONSECA, Leila MG; MANJUNATH, BS. Registration techniques for multisensor remotely sensed imagery, 1996.

GEORGEL, Pierre; BENHIMANE, Selim; NAVAB, Nassir. A Unified Approach Combining Photometric and Geometric Information for Pose Estimation. **BMVC**, pp. 1–10, 2008.

HARRIS, Christopher G; STEPHENS, Mike, et al. A combined corner and edge detector. **Alvey vision conference**, vol. 15, no. 50, pp. 10–5244, 1988.

HARTLEY, Richard; ZISSERMAN, Andrew. **Multiple view geometry in computer vision**. [S.l.]: Cambridge university press, 2003.

HAWKINS, Andrew J. Elon Musk Still Doesn't Think LIDAR is Necessary for Fully Driverless Cars. **The Verge**, 2018.

HINICH, Melvin J; TALWAR, Prem P. A simple method for robust regression. **Journal of the American Statistical Association**, Taylor & Francis Group, vol. 70, no. 349, pp. 113–119, 1975.

HUA, Minh-Duc; TRUMPF, Jochen; HAMEL, Tarek; MAHONY, Robert; MORIN, Pascal. Feature-based recursive observer design for homography estimation and its application to image stabilization. **Asian Journal of Control**, Wiley Online Library, 2018.

HUBER, P. J. **Robust Statistics**. [S.l.]: John Wiley & Sons, 1981.

IKAMI, Daiki; YAMASAKI, Toshihiko; AIZAWA, Kiyoharu. Fast and robust estimation for unit-norm constrained linear fitting problems. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, pp. 8147–8155, 2018.

IRANI, M.; ANANDAN, P. All about direct methods. **Proc. Workshop on Vision Algorithms: Theory and practice**, 1999.

LIANG, Pengpeng; WU, Yifan; LU, Hu; WANG, Liming; LIAO, Chunyuan; LING, Haibin. Planar object tracking in the wild: A benchmark. **2018 IEEE International Conference on Robotics and Automation (ICRA)**, pp. 651–658, 2018.

LIU, Haomin; ZHANG, Guofeng; BAO, Hujun. Robust Keyframe-Based Monocular SLAM for Augmented Reality. **2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)**, pp. 1–10, 2016.

LOWE, David G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, Springer, vol. 60, no. 2, pp. 91–110, 2004.

LUCAS, Bruce D; KANADE, Takeo. An iterative image registration technique with an application to stereo vision, 1981.

MA, Yi; SOATTO, Stefano; KOSECKA, Jana; SASTRY, S Shankar. **An invitation to 3-d vision: from images to geometric models**. [S.l.]: Springer Science & Business Media, 2012. vol. 26.

MALIS, Ezio. **ESM Software Development Kits**. [S.l. S.n.], 2011. `http://esm.gforge.inria.fr/ESM.html`. [Online; accessed 12-July-2019].

_____. **Vision-based estimation and robot control**. 2008. PhD thesis – Université Nice Sophia Antipolis.

MALIS, Ezio; VARGAS, Manuel. **Deeper understanding of the homography decomposition for vision-based control**. 2007. PhD thesis – INRIA.

MEILLAND, Maxime; COMPORT, Andrew I; RIVES, Patrick. Real-time dense visual tracking under large lighting variations. **British Machine Vision Conference**, pp. 45–1, 2011.

MORENCY, L-P; DARRELL, Trevor. Stereo tracking using ICP and normal flow constraint. **Object recognition supported by user interaction for service robots**, vol. 4, pp. 367–372, 2002.

MUR-ARTAL, Raul; MONTIEL, J. M. M.; TARDOS, Juan D. ORB-SLAM: a Versatile and Accurate Monocular SLAM System. **IEEE Transactions on Robotics**, vol. 31, no. 5, pp. 1147–1163, 2015.

NEUBERGER, Bernhard; SILVEIRA, Geraldo; POSTOLOV, Marko; VINCZE, Markus. Object Grasping in Non-metric Space Using Decoupled Direct Visual Servoing. **Proceedings of the ARW & OAGM Workshop**, 2019.

NGUYEN, Ty; CHEN, Steven W; SHIVAKUMAR, Shreyas S; TAYLOR, Camillo J; KUMAR, Vijay. Unsupervised Deep Homography: A Fast and Robust Homography Estimation Model, 2017.

NOGUEIRA, Lucas; PAIVA, Ely de; SILVEIRA, Geraldo. **VTEC robust intensity-based homography optimization software**. Brazil, 2019.

PLINVAL, Henry de; MORIN, Pascal; MOUYON, Philippe; HAMEL, Tarek. Visual servoing for underactuated VTOL UAVs: A linear, homography-based approach. **2011 IEEE International Conference on Robotics and Automation**, pp. 3004–3010, 2011.

QUIGLEY, Morgan; CONLEY, Ken; GERKEY, Brian; FAUST, Josh; FOOTE, Tully; LEIBS, Jeremy; WHEELER, Rob; NG, Andrew Y. ROS: An open-source Robot Operating System. **ICRA workshop on open source software**, 2009.

RANFTL, René; KOLTUN, Vladlen. Deep fundamental matrix estimation. **Proceedings of the European Conference on Computer Vision (ECCV)**, pp. 284–299, 2018.

ROSTEN, Edward; DRUMMOND, Tom. Machine learning for high-speed corner detection. **European conference on computer vision**, pp. 430–443, 2006.

RUBLEE, Ethan; RABAUD, Vincent; KONOLIGE, Kurt; BRADSKI, Gary R. ORB: An efficient alternative to SIFT or SURF. **International Conference on Computer Vision**, vol. 11, no. 1, p. 2, 2011.

SILVEIRA, Geraldo. Photogeometric Direct Visual Tracking for Central Omnidirectional Cameras. **Journal of Mathematical Imaging and Vision**, vol. 48, no. 1, pp. 72–82, 2014.

SILVEIRA, Geraldo; MALIS, Ezio. Real-time Visual Tracking under Arbitrary Illumination Changes. **2007 IEEE Conference on Computer Vision and Pattern Recognition**, pp. 1–6, 2007.

_____. Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images. **International Journal of Computer Vision**, vol. 89, pp. 84–105, 2010.

SILVEIRA, Geraldo; MALIS, Ezio; RIVES, Patrick. An efficient direct approach to visual SLAM. **IEEE transactions on robotics**, IEEE, vol. 24, no. 5, pp. 969–979, 2008.

SILVEIRA, Geraldo; MIRISOLA, L.; MORIN, P. Decoupled Intensity-Based Nonmetric Visual Servo Control. **IEEE Transactions on Control Systems Technology**, 2018.

SINGH, Abhineet K. **Modular tracking framework: a unified approach to registration based tracking**. 2017. MA thesis – University of Alberta.

SMITH, Paul; REID, Ian D; DAVISON, Andrew J. Real-time monocular SLAM with straight lines. **British Machine Vision Conference**, BMVA, 2006.

TORR, P.; ZISSERMAN, A. Robust computation and parametrization of multiple view relations. **Sixth International Conference on Computer Vision**, pp. 727–732, 1998.

VALOGNES, Julien; DASTJERDI, Niloufar Salehi; AMER, Maria. Augmenting Reality of Tracked Video Objects Using Homography and Keypoints. **International Conference on Image Analysis and Recognition**, pp. 237–245, 2019.

VIOLA, Paul; WELLS III, William M. Alignment by maximization of mutual information. **International journal of computer vision**, Springer, vol. 24, no. 2, pp. 137–154, 1997.

WU, Ziming; GUO, Jiabin; ZHANG, Shuangli; ZHAO, Chen; MA, Xiaojuan. An AR Benchmark System for Indoor Planar Object Tracking. **2019 IEEE International Conference on Multimedia and Expo (ICME)**, pp. 302–307, 2019.

YAN, Qing; XU, Yi; YANG, Xiaokang; NGUYEN, Truong. HEASK: Robust homography estimation based on appearance similarity and keypoint correspondences. **Pattern Recognition**, Elsevier, vol. 47, no. 1, pp. 368–387, 2014.

ZHANG, Ji; SINGH, Sanjiv. Visual-lidar odometry and mapping: Low-drift, robust, and fast. **2015 IEEE International Conference on Robotics and Automation (ICRA)**, pp. 2174–2181, 2015.

ZITOVA, Barbara; FLUSSER, Jan. Image registration methods: a survey. **IMAGE AND VISION COMPUTING**, vol. 21, pp. 977–1000, 2003.