Robust Self-calibration and Fundamental Matrix Estimation in 3D Computer Vision

by

Houman Rastgar

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the Ph.D. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

The recent advances in the field of computer vision have brought many of the laboratory algorithms into the realm of industry. However, one problem that still remains open in the field of 3D vision is the problem of noise. The challenging problem of 3D structure recovery from images is highly sensitive to the presence of input data that are contaminated by errors that do not conform to ideal assumptions. Tackling the problem of extreme data, or outliers has led to many robust methods in the field that are able to handle moderate levels of outliers and still provide accurate outputs. However, this problem remains open, especially for higher noise levels and so it has been the goal of this thesis to address the issue of robustness with respect to two central problems in 3D computer vision. The two problems are highly related and they have been presented together within a Structure from Motion (SfM) context. The first, is the problem of robustly estimating the fundamental matrix from images whose correspondences contain high outlier levels. Even though this area has been extensively studied, two algorithms have been proposed that significantly speed up the computation of the fundamental matrix and achieve accurate results in scenarios containing more than 50% outliers. The presented algorithms rely on ideas from the field of robust statistics in order to develop guided sampling techniques that rely on information inferred from residual analysis. The second, problem addressed in this thesis is the robust estimation of camera intrinsic parameters from fundamental matrices, or self-calibration. Self-calibration algorithms are notoriously unreliable for general cases and it is shown that the existing methods are highly sensitive to noise. In spite of this, robustness in self-calibration has received little attention in the literature. Through experimental results, it is shown that it is essential for a real-world self-calibration algorithm to be robust. In order to introduce robustness to the existing methods, three robust algorithms have been proposed that utilize existing constraints for self-calibration from the fundamental matrix. However, the resulting algorithms are less affected by noise than existing algorithms based on these constraints. This is an important milestone since self-calibration offers many possibilities by providing estimates of camera parameters without requiring access to the image acquisition device. The proposed algorithms rely on perturbation theory, guided sampling methods and a robust root finding method for systems of higher order polynomials. By adding robustness to self-calibration it is hoped that this idea is one step closer to being a practical method of camera calibration rather than merely a theoretical possibility.

Acknowledgements

First and foremost I would like to thank my supervisor Dr Eric Dubois. It is thanks to his patience and guidance and support that this thesis has been possible. I thank him for his support during a prolonged illness, several topic changes and many research excursions into areas that did not end up being directly used in this thesis. Also, his insights and encyclopedic knowledge in mathematics have helped improve this thesis significantly. I also am grateful for his very thorough proof reading of my thesis and all my publications and all the invaluable comments he has given me to improve them. It has been a great privilege to work with him and I am forever indebted to him for his support, which went well above his call of duty. I would also like to thank my co-supervisor, Liang Zhang. It was a pleasure working with him in CRC and I cherish the many hours we spent discussing research ideas on the whiteboard in his office. I also thank him for helping me improve the manuscripts of several papers and for helping guide my research directions.

I also thank the members of my committee for their time and effort. The valuable comments I received from them during my proposal have helped me improve this thesis. I especially thank Dr Anthony Whitehead for pointing me to some hard-to-find image sets for self-calibration. I also am very grateful for Dr Lang's extensive feedback after my proposal which helped me fix many errors.

This work was primarily supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). Also, some of the work in Chapter 4 has been supported by the Communications Research Centre Canada (CRC).

I would also like to thank my colleagues at the VIVA lab and the university of Ottawa, especially Alan, Saied, Aboud, Roy, Oana, Mohamed and Ehsan. My friends Patrick, Greg and Steph also deserve credit for always being there for me.

I also am grateful to my mother, sister and my aunt Katy for their continuing support. Last but not least I thank Dawn for believing in me and supporting me during the good and the bad times.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Scope of the Work

The issue of modeling the 3D world through photographs is an issue that has been intensely researched in the scientific community for years. Since Marr's seminal work on stereo vision [65], many computational methods have been proposed to solve the problem of retrieving the three dimensional structure of the world through images. One of the most important methods of determining 3D geometric information from images has been the process of Structure from Motion (SfM) where the pinhole model of the camera is utilized to represent the image projection process and the structure of the scene is inferred through the motion of the camera(s) in the scene. Various ideas from the field of photogrammetry [100] are then utilized to "reverse" the image projection process and to retrieve 3D models from the images acquired. Even though the basic ideas in SfM were known as early as the 19th century, the field has been in a constant flux and new methods and ideas are constantly being proposed in order to improve the existing algorithms. This is due to the fact that SfM is an inherently ill posed problem and to this day no general-purpose algorithm exists that can handle arbitrary scenes and provide accurate 3D geometry in all cases. In spite of this, several milestones have been achieved in the SfM research field due to recent innovations in the field and outside of it. These innovations include:

1. Faster CPUs.
2. Parallel implementations of 3D geometry recovery [34].
3. Availability of vast image databases on the Internet that provide redundancy.

4. Improved feature matching techniques [60].

5. Improved dense surface reconstruction techniques [35, 9, 85].

These innovations have in fact provided us with several technologies that bring the innovations in SfM to the forefront of commercial use. These include for example Photosynth™ which combines large Internet photo collections and creates a powerful navigation experience to the user that draws upon both geometrical approaches of recreating an environment and also image-based techniques [103]. Others include methods that are able to reconstruct the geometry of large scale city models such as those presented in [9, 34, 83]. These methods have enabled researchers to build accurate models of large scale environments based purely on images via the SfM framework.

In spite of the multitude of the advances in the field of SfM, one of the most challenging issues is the problem of noise. In fact, a large proportion of the algorithms presented in the field aim to reduce the impact of non-ideal input data, or noise, and to enable more robust algorithms. Robustness is essentially enabling algorithms to perform accurately regardless of deviations in the assumptions made. As stated in [67]:

> The ultimate goal of computer vision is to mimic human visual perception. Therefore, in the broadest sense, robustness of a computer vision algorithm is judged against the performance of a human observer performing an equivalent task. In this context, robustness is the ability to extract the visual information of relevance for a specific task, even when this information is carried only by a small subset of the data, and/or is significantly different from an already stored representation.

Therefore, a robust SfM algorithm must be able to discern 3D geometry regardless of noise in the input data. In fact, the goal of this thesis is to improve the robustness of the SfM process in order to make a general and seamless SfM framework closer to practice. However, since SfM is a "pipeline" rather than a single algorithm, robustness is a multifaceted issue in this regard. One of the most important of these facets is the process of camera tracking or pose estimation. This intermediate stage in the SfM pipeline is defined in this thesis as spatially localizing the position of the cameras in 3D space *and* determining the optical properties of the cameras. Since the SfM framework is able to retrieve the 3D geometry of the scene using the motion of the cameras, quantifying the camera positions and optical characteristics must be carried out accurately. The details of the SfM pipeline and background information on camera tracking have been presented in Chapter 2. At this point it suffices to mention that given a set of accurate point matches across images in a sequence, and once cameras have

been specified (their position and properties) the task of creating a sparse 3D cloud from the images is simplified. Subsequently, a dense 3D reconstruction based on this initial estimate of cameras and a limited set of points can be carried out via a dense reconstruction method [35].

Before specifying the adopted approach and the specific problems addressed in the SfM pipeline, an overview of some of the existing techniques to the problem of scene modeling will be provided. In order to put the issues addressed in this thesis in a larger context it is important to mention various existing frameworks to the problem of 3D reconstruction.

1.2 Summary of Existing Methods

It is important to note one of the fundamental assumptions in this thesis. The goal of the methods proposed herein is to enhance the robustness of the SfM framework that aims to recover the geometry of rigid objects. In other words, we assume that objects in the scene do not undergo non-rigid deformations. Even though some of the methods proposed in this thesis aim to account for local motion, which could include deformations, these are considered outliers with respect to the environment and no geometry is inferred from such objects. There are however established methods for estimating non-rigid structure, for instance in [121]. In all subsequent discussions presented in this thesis, the assumption is that the scene that is to be visualized is rigid and that the SfM framework is based on the rigidity assumption. As a result, all discussions that follow focus on the reconstruction of rigid scenes.

As mentioned, the SfM framework is a method of visualization that proceeds by finding the underlying geometry of a scene using photogrammetry. However, a completely separate paradigm exists which aims to bypass geometric modeling of the 3D environment. One of the routes that the research community adopted in the 90s was the attempt to visualize the world through photographs without having to find explicit geometry of the scene thus avoiding the difficulties of explicit 3D geometric modeling. These methods are referred to as "image-based rendering" [66, 99] and they found great use in modeling smaller objects [53, 38] or in limited navigable slideshow of an environment. One of the earliest methods belonging to this category is Quicktime VR™ [17]. However, this method only enabled users to jump from various points where photographs were perviously taken. Similarly, the popular Street View™ tool of Google enables users to merely "fade" from one pre-taken panorama to another. The reason these methods are not able to provide a seamless navigation to the user is due to their lack of any underlying geometrical information.

Given the two separate paradigms presented, there is also an array of intermediate methods that leverage varying levels of geometry and purely image based information to create a visu-

alization of the world. The interested reader is referred to [99] for a detailed survey of some of these methods. In addition to the spectrum of algorithms that aim to visualize the world based on varying levels of geometry and image data, there are also those that incorporate non-visual depth cues and those that use active sensors. One such method is the use of structured light [96] where a light pattern is projected into the environment and the recorded intensities are used to build a depth map. One popular technology leveraging this method is the Microsoft Kinect sensor [5] which is able to build accurate 3D maps in real-time. The Kinect device contains an infrared laser emitter, an infrared camera and an RGB camera. The infrared emitter projects a specific pattern into the scene and the infrared sensors record this patterns and depth is inferred by the displacement of known points on the patterns with respect to some reference plane [52]. This can also be considered a structured light technique based on infrared patterns. Many other methods also exist based on time-of-flight sensors such as laser and radar sensors [49]. The approach adopted in this thesis however is the geometric approach based on purely visual information. However, within this subfield of 3D modeling framework, there are also several subcategories.

Visual odometry is the field of research concerned with localizing the camera with respect to an environment by establishing a set of stable landmarks in that environment. Rooted in the field of robotics, a prominent example of this would be the Monocular Simultaneous Localization and Mapping (monoSLAM) as presented in [26]. Even though SLAM is not necessarily a subfield of SfM and can be considered more of a separate entity, it shares many similar aspects with SfM. The goal of SLAM is real-time, high frame rate tracking of the location of a sensor, geared towards robotic navigation. However, SfM is generally considered an off-line process (but not always [76]) and the goal is more focused on geometry modeling rather than sensor localization. Even though real-time performance is not the goal of many SfM algorithms, achieving results within reasonable time constraints is still a desirable goal. Considering some of the new directions of SfM which focus on building scenes from very large Internet databases, it is important not to neglect the computational cost aspect of the underlying algorithms. To this end, computational efficiency has been one of the goals of this thesis and several of the algorithms improve this aspect of SfM as will be shown by the experimental results.

As explained, SfM is a pipeline which proceeds in several steps by repeatedly estimating the pose of the cameras and performing 3D reconstruction. One other approach to structure recovery which can find camera location and 3D locations of points in space in one step is the factorization method [114]. This algorithm uses the singular value decomposition to find the pose of the cameras and the 3D location of a set of point matches across a set of views. However, this method is rather restrictive since it can only work with affine cameras (a subclass

of general image acquisition scenarios) and also it requires that the points whose 3D locations are found be seen by all cameras. This renders the algorithm unsuitable for many practical scenarios where objects disappear and reappear in the images of a scene.

Within the SfM research, there are also several approaches for achieving 3D geometry. One way to categorize some of these methods is based on the amount of knowledge one has with respect to the scene, how much information is available with respect to the cameras and what type of geometric reconstruction is desired. Here, knowledge with respect to the cameras is information regarding the optical properties of cameras (i.e., intrinsic parameters) rather than the 3D position of the cameras. Using this approach we can divide SfM algorithms to:

1. Projective 3D reconstruction. (no knowledge of cameras, and no knowledge of the scene)
2. Affine 3D reconstruction. (some knowledge of cameras, and no knowledge of the scene)
3. Calibrated Euclidean 3D reconstruction. (full knowledge of cameras, and no knowledge of the scene)
4. Semi-calibrated Euclidean 3D reconstruction. (some knowledge of cameras, and no knowledge of the scene)
5. Euclidean 3D reconstruction with known points in the scene or known scene geometric entities (e.g., planes, parallel lines, etc.). (no knowledge of cameras, some knowledge of the scene)
6. Uncalibrated Euclidean 3D reconstruction. (no knowledge of cameras, and no knowledge of the scene)

The first two methods reconstruct the scene but up to some geometric ambiguity as will be explained in Chapter 2. This obviously renders them inadequate for visualization purposes. However, such reconstructions are important as intermediate steps for 3D Euclidean reconstruction or other computer vision applications (e.g., camera localization [92]). As a result, whenever 3D reconstruction is mentioned throughout this thesis without being qualified with the "projective" or "affine" prefix, it is assumed to be Euclidean. Under a Euclidean reconstruction, the aim is to create a model of the world that is most realistic in terms of the geometric properties of the scene (this will be expounded upon in Chapter 2). Calibrated 3D reconstruction is essentially when the properties of cameras are fully known in advance. This is of interest to scenarios where the image acquisition process can be carefully controlled. Semi-calibrated 3D reconstruction is where most properties of the camera are known, except the focal length.

Moreover, methods that use scene constraints depend on assumptions on the existence of various geometric entities in the scene, such as vanishing lines or planes. The goal of this thesis has been the last item presented, or the uncalibrated Euclidean 3D reconstruction which uses the least number of assumptions. Under this umbrella of methods, no assumptions about the scene or the camera properties are made except that the camera parameters do not vary across the views. Although many of the principles in this thesis applies to the case of unknown but varying camera parameters, the experiments and methodology focus on the case of fixed camera parameters. In fact, extensive experiments have been carried out showing the robustness of the proposed algorithms with respect to varying camera parameters.

The particular focus of this thesis within the field of uncalibrated SfM has been twofold. This has been the robust and efficient estimation of the fundamental matrix, and self-calibration. The following section will outline the SfM pipeline and how these two processes fall within the SfM context.

1.3 Problem Statement

The issue tackled in this thesis is improving robustness and efficiency in the estimation of the fundamental matrix and camera parameters from purely image data (i.e., without using extra constraints). The fundamental matrix is a geometric entity which relates the corresponding points in an image pair via the geometric relationship between the two views. This entity which is represented by a 3×3 matrix is of utmost importance in many computer vision application. The second issue tackled in this thesis is the robust and efficient estimation of the camera intrinsic parameters using only visual information in an image sequence (i.e., only using the images themselves). This is also referred to as "self-calibration" since no external means are utilized to find these parameters. Although self-calibration is a difficult process and obtaining accurate results are not always possible, the idea of a method that is able to obtain camera parameters without using complicated calibration methods requiring specially made calibration targets is of great interest to the field of computer vision. As a result, it has been the goal of this thesis to improve the self-calibration accuracy in order to make it possible for an uncalibrated SfM algorithm to leverage this process. Interestingly, the input to the self-calibration approach utilized in this thesis is the set of fundamental matrices obtained between the frames in a sequence. Therefore, the estimation of the fundamental matrix and self-calibration are highly related. In fact, one could consider our approach a progressive estimation technique where initially fundamental matrices are estimated and then camera parameters are found based on these fundamental matrices. Following this, the fundamental matrix can be upgraded to its calibrated

counterpart (i.e., the essential matrix) and the uncalibrated SfM can be converted to a calibrated SfM problem. Figure 1.1 shows the adopted SfM framework in this thesis. The areas that are addressed herein are highlighted. This diagram shows the adopted approach to SfM where a series of images are utilized to create a 3D model of the imaged environment. The areas that have been addressed in this thesis are prior to the structure recovery. In other words, one could consider the estimation of the fundamental matrix and self-calibration as intermediate steps in SfM. Whereas, feature detection (finding prominent image location) and feature matching (finding correspondences between image features) are the initial stages and structure recovery (triangulation) and bundle adjustment (refining structure and camera pose) are the final stages. The illustration is presented at this point in order to illuminate the focus of this thesis with respect to an uncalibrated SfM framework. Note that the left box contains the steps related to matching features and finding camera parameters and the right box contains the steps involved in structure recovery. There are several different methods of handling uncalibrated SfM and the methodology presented in Figure 1.1 is one particular approach. Alternative methods could include for example, Euclidean reconstruction obtained from an uncalibrated sequence via stratified reconstruction where initially a projective reconstruction is found and then upgraded to an affine and then a Euclidean one. Chapter 5 contains a more detailed explanation on these methods. Also the presented framework is similar to the approach adopted in [102] with the exception that the camera parameters are estimated via self-calibration, rather than being retrieved from the tagged information in the jpg files (that are not always available and when they are, not always reliable).

Even though the problem of the estimation of the fundamental matrix and self-calibration are presented within an SfM framework, these could be considered independent general computer vision techniques that have wide applicability in several fields. For instance, the fundamental matrix is used in motion segmentation, tracking, image rectification just to name a few. Similarly, self-calibration is used whenever camera parameters are required, such as: robotics, navigation, surveillance and etc. Since the problem of SfM is perhaps one of the most challenging and well-studied problem in computer vision, this framework has been adopted in order to present the proposed algorithms. However, since the estimation of the fundamental matrix and self-calibration have broad applicability in computer vision outside of SfM both ideas have been discussed independent of one another in a general sense and their performance is evaluated outside of an SfM algorithm. The next section outlines the motivation behind the emphasis on self-calibration and the fundamental matrix and also the importance of robustness in these algorithms.

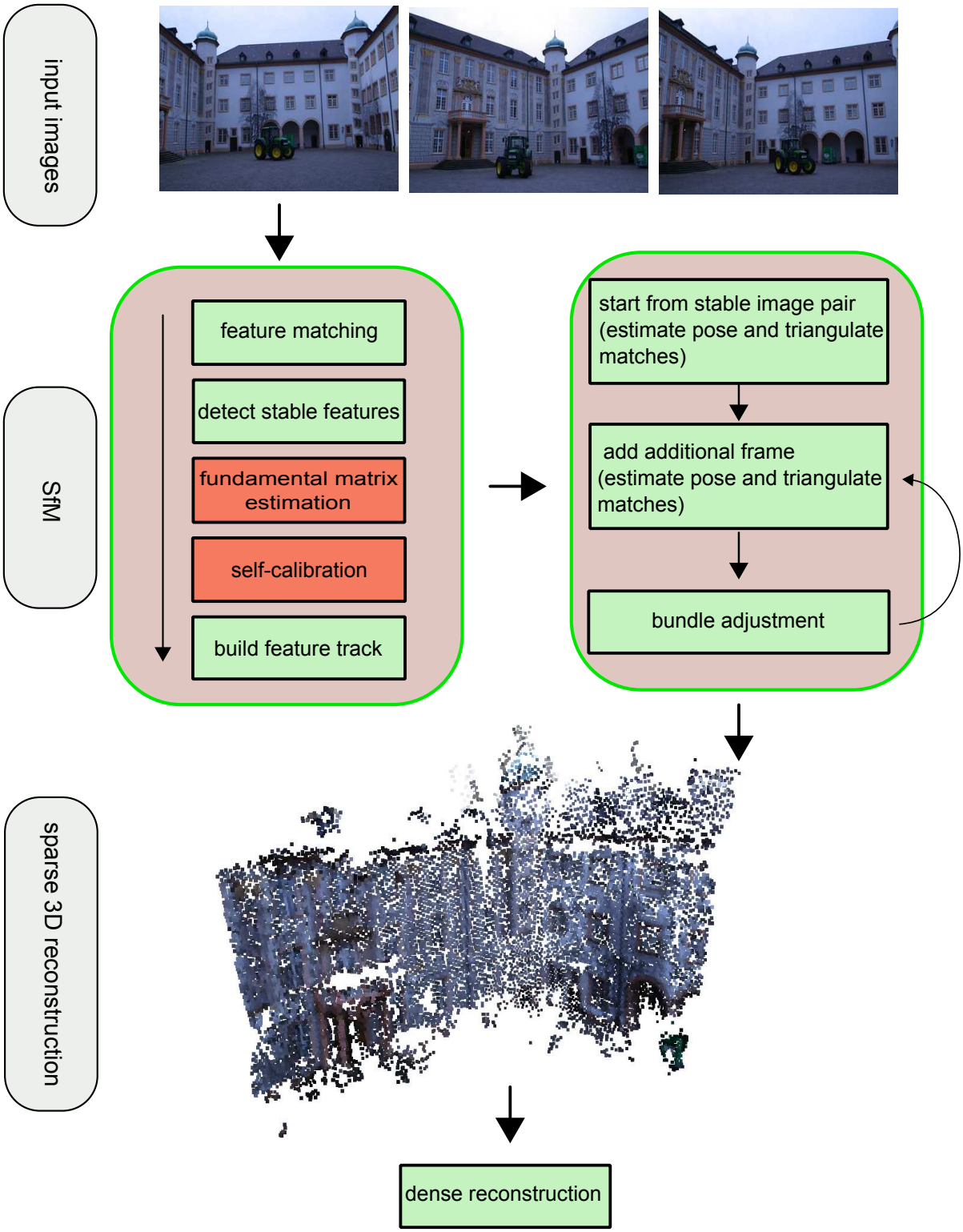


Figure 1.1: Uncalibrated Euclidean Structure from Motion (SfM) pipeline.

1.4 Motivation

So far it has been argued that estimation of the fundamental matrix and reliable self-calibration offer the ability to perform uncalibrated geometric modeling using the SfM framework. This effectively enables an automatic structure recovery pipeline where a user is able to estimate 3D geometry of a scene without resorting to a calibration process. Therefore, as a result of their importance to 3D computer vision, there is a large interest in the areas of fundamental matrix estimation and self-calibration. However, one of the areas that still remains challenging is the issue of robustness as mentioned earlier. More specifically, the issue of noise can adversely affect the SfM framework which means that for many sequences SfM will either fail or produce less than adequate results. Therefore, it has been the goal of this thesis to address the issue of robustness in both areas of fundamental matrix estimation and self-calibration. As shown in Figure 1.1, the left column of processes are responsible for ensuring that no outliers make their way to the structure recovery in the right hand column processes. Improving the robustness in the fundamental matrix estimation and self-calibration will significantly improve the overall robustness of the SfM pipeline relying on the two methods.

Even though robustness in the field of fundamental matrix estimation has been addressed in the literature, starting with the seminal work presented in [119, 116], the area still remains a challenging one. Since the goal of this thesis is to enable seamless uncalibrated reconstruction via self-calibration, it is essential that the fundamental matrices used as inputs to self-calibration be as accurate as possible. In fact, it will be shown in Chapter 5 that self-calibration is highly sensitive to the quality of the estimated fundamental matrix and therefore a higher expectation is placed on the accuracy of the fundamental matrix than is normally required. Also, one of the problems that has not received sufficient attention in the field of robust estimation of the fundamental matrix is the problem of local motion. This basically is the presence of independently moving objects in a scene that is being captured. As a result, it is one of the goals of thesis to address this issue in order to improve the estimation of the fundamental matrix. Such scenes contain very large outlier ratios and also require the algorithms processing their correspondences to avoid the typical method of relying on matching scores in order to infer validity of the data. This has been thoroughly examined in Chapter 4.

Unlike the fundamental matrix, robust self-calibration has not received much attention in the research community. Chapter 5 begins with a section demonstrating the sensitivity of the self-calibration procedure to various types of outliers. In fact, it will be argued in this thesis that the reason for the lack of reliability of most self-calibration algorithms is their lack of robustness with respect to the various types of noise. As a result, three algorithms that rely on

different ideas in robustness are proposed to tackle the issue of robustness in self-calibration. It is the goal of this thesis to move self-calibration closer to practice by improving the robustness of the process. A reliable self-calibration procedure has numerous applications such as SfM, augmented reality, visual navigation and novel view synthesis.

1.5 Overview of the Proposed Approach

The common theme in this thesis has been robustness and its advancement in the estimation of the fundamental matrix and self-calibration. In Chapter 4 two methods have been proposed in order to improve the efficiency and accuracy of the estimation of the fundamental matrix. The methods in this chapter leverage ideas from robust statistics to improve the sampling process in the estimation of the fundamental matrix. Using the analysis of the residuals in an iterative framework two methods have been proposed to infer validity data in order to steer the sampling process in a way that outliers are pruned. The ideas have been tested rigorously using synthetic correspondences. The test setups contain virtual image correspondences where all the parameters of the cameras in the scene are known and so comparisons with ground truth parameters can be made. The results show the improved robustness and speed of the robust estimator for the fundamental matrix.

The second part of the contributed methods rely on improving the robustness of self-calibration. Three different algorithms have been proposed each with a different set of advantages and application scenarios. The proposed methods are based on ideas from robust sampling techniques, parameter estimation, perturbation theory and algebraic geometry. The proposed methods offer a degree of robustness and reliability that is not achieved by existing methods. The proposed methods have been tested under a variety of scenarios including synthetic and real image sequences. The results demonstrate the effectiveness of the proposed self-calibration strategies.

Overall the aim has been to improve the robustness in two highly related and very fundamental areas in 3D vision. The results presented show the efficacy of the proposed methods and their limitations.

1.6 Thesis Organization

Chapter 2 presents the background and the terminology in 3D reconstruction and multiple view geometry. The subsequent derivations and methodology are based on the foundation

built in this chapter. Following this, Chapter 3 presents the background the terminology in robust statistics. The formulations in this chapter are used to present the proposed ideas with respect to the robust estimation of the fundamental matrix. Chapter 4 contains a survey and analysis of some of the advances in the field of robust estimation of the fundamental matrix. The limitations of these methods are then outlined and two methods are proposed to improve the robustness of the existing estimators. The chapter concludes with the experimental results comparing the two proposed algorithms and some of the existing methods.

Chapter 5 presents the portion of the thesis relating to robust self-calibration. This chapter starts with a survey of some of the existing methods. These methods are presented based on the terminology presented earlier in Chapter 2. The issue of robustness is then discussed and the inadequacy of existing methods in dealing with even the smallest amount of outliers is then discussed. The chapter then presents the three proposed algorithms. Each individual algorithm is presented in its own section with a stand-alone set of experimental results. The chapter finally concludes with a more extensive set of experimental results which includes comparison and analysis of the respective merits of the proposed algorithms.

Finally, Chapter 6 presents the conclusion of this thesis. This chapter contains a summary of the whole thesis in addition to an outline of the contributions made. Furthermore, the open questions and challenges remaining in the areas that have been addressed will be presented.

Chapter 2

Multiple View Geometry

2.1 Overview

The goal of computer vision is to reconstruct, interpret and understand a three dimensional scene from its two dimensional images. Similar to the human visual system, this entails such diverse tasks as: recognition (face, gait, emotion, etc.), measurement, navigation, etc. There are different methods of categorizing subdisciplines within the computer vision research; one way to outline the different problems is by dividing computer vision into three levels:

- Low level vision: feature extraction, texture extraction, etc.
- Mid level vision: 3D reconstruction, pose estimation,calibration, etc.
- High level vision: scene recognition and analysis

The context of this thesis as mentioned, is the robust estimation of camera parameters from correspondences and the robust estimation of the fundamental matrix, which are mid-level vision tasks. These algorithms pertain to the goal of reconstruction of a 3D model from two dimensional images and resolving the location of the camera(s). Also it has been assumed that efficient low-level vision algorithms exist and can be used as input data for the mid-level tasks. Specifically I have made the assumption that reasonably effective feature extraction and matching techniques exist and can be applied to a wide range of images.

As a research area that started in photogrammetry in the 19th century [44] and could even be traced back to the 14th century, scene reconstruction from multiple images is a somewhat aged science. In fact, many existing real-world applications rely on techniques developed by the researchers in this field. Some existing applications domains are:

- Machine inspection [3]
- 3D model building such as Bing map [4]
- Image stitching [1]

However, as explained earlier, there are still various areas of computer vision that are unsolved or inadequately solved. Scene reconstruction and camera tracking is still a challenging task and research is still ongoing towards improving the robustness, scalability and reliability of scene reconstruction.

Before outlining various relevant areas of scene reconstruction, a brief review of projective vision will be provided. This lays the foundation for the subsequent discussions in 3D computer vision. Following this, the problem of camera calibration, including the intrinsic and extrinsic parameters will be provided. Next, the basics of two-view geometry will be outlined, including the fundamental matrix, homography and the essential matrix. Also pose estimation will be discussed in this section. Subsequently, 3D reconstruction will be defined where stereo matching and triangulation will be explained. Then the problem of degeneracy will be discussed within the context of 3D reconstruction.

2.2 Projective Geometry

The familiar Euclidean geometry is inadequate for studying the relationship between a scene and its two dimensional images. This is due to the fact that we wish to denote the process of capturing a 2D slice of our 3D world with a transformation. However, images often do not preserve the shape of images, in other words, the picture of a circular object is not necessarily a circle. Therefore, using a Euclidean transformation is not appropriate for studying the imaging process.

Projective geometry provides us with the toolkit required to analyze and study the process of imaging. The following sections will outline some of the important concepts of projective geometry needed for later chapters. For a more detailed explanation see [29, 44].

2.2.1 Homogenous Coordinates

When thinking of pixels on an image, every point represents a possible line of sight of an incoming light ray. In other words, any 3D point along the ray projects to the same image point, so only the direction of the ray is relevant, not the distance of the point along it [71].

In computer vision we need to represent this "visual sphere" of incoming rays. The solution offered by homogenous coordinates is by arbitrarily choosing some 3D point along each ray to represent the ray's direction. This will effectively match the point to its visual ray. In addition it has the significant advantage of making the image projection process much easier to mathematically deal with.

For example, suppose we have a point (x, y) in the Euclidean plane. To represent this same point in the projective plane, we simply add a third coordinate of 1 at the end: $(x, y, 1)$. Since Overall scaling along such a ray is unimportant, the point $(x, y, 1)$ is the same as the point $(\alpha x, \alpha y, \alpha)$, for any nonzero α. In other words:

$$(x, y, w) = (\psi x, \psi y, \psi w) \tag{2.1}$$

for any $\psi \neq 0$.

2.2.2 Lines in Projective Space

The two dimensional line with equation $ax + by + c = 0$ is represented in homogeneous coordinates by the homogeneous equation $(a, b, c) \cdot (x, y, 1) = ax + by + c = 0$. Note that lines are represented homogeneously as 3 component vectors, just as points are. This represents the duality that exists in projective geometry between points and lines and is a recurrent concept in various computer vision discussions.

Using this notation, we can claim that a line $\mathbf{l}$ goes through a point $\mathbf{x}$ if $\mathbf{l}^T\mathbf{x} = 0$. Similarly, in 3D space points are represented by a 4-element vector $(X, Y, Z, 1) = (\psi X, \psi Y, \psi Z, \psi)$. In 3D projective space the duals of points are planes and we can claim that a point $\mathbf{X}$ is on plane $\mathbf{\Pi}$ if $\mathbf{\Pi}^{\boldsymbol{T}}\mathbf{X} = 0$.

2.2.3 Conics

This section will briefly review the concept of conics and quadrics in projective space. These geometric entities are of importance in the discussion of self-calibration in Chapter 5.

A conic is basically a shape described by a second degree equation in the plane. Euclidean geometry contains three different conic types: ellipses, parabolas and hyperbolas. The equation of a conic in non-homogenous coordinates is simply:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{2.2}$$

which is a polynomial of second degree. In homogenous coordinates the equation of a conic becomes:

(a) Conic. (b) Dual conic.

Figure 2.1: Point and line conics.

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0 \tag{2.3}$$

after replacing (x, y) with (x_1, x_2, x_3) in our homogenous coordinates. In matrix format this can be written more compactly as:

$$\mathbf{x}^\mathbf{T} C \mathbf{x} = 0 \tag{2.4}$$

where the conic coefficients are embedded in C as:

$$C = \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix} \tag{2.5}$$

Note that multiplying the above matrix with a non-zero constant will not change the equation and that the matrix is symmetric. Therefore, there are only five degrees of freedom in the equation of the conic.

Similar to points and lines, the intersection of a line $\mathbf{l}$ tangent to a conic C at point $\mathbf{x}$ can be written as: $\mathbf{l} = C\mathbf{x}$, the proof for this is omitted but the interested reader can refer to [44].

In addition to regular conics, another class of conics are important in the study of multiple view geometry. These are dual conics, or conics that are defined on lines rather than points. Similar to lines and points that are duals of one another in projective geometry, dual conics and regular conics share the same mathematical definition. However, rather than a regular conic which is defined by a set of points, the dual conic is defined by the set of tangent lines to the conic. The equation for a dual conic, C^* is defined in terms of the envelope of its tangent lines as: $\mathbf{l}^\mathbf{T} C^* \mathbf{l} = 0$. Figure 2.1b shows a regular conic and the dual conic.

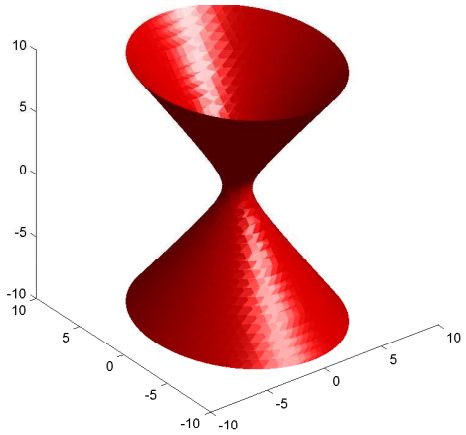


Figure 2.2: Example of a Quadric.

2.2.4 Quadrics and Dual Quadrics

A quadric is surface defined in 3D space, $\mathbb{P}^3$ according to a quadratic polynomial. Similar to a conic, the form for a quadric in homogenous coordinates can be written as:

$$\mathbf{X}^{\mathbf{T}} Q \mathbf{X} = 0 \tag{2.6}$$

where $\mathbf{X}$ is the homogenous location of a point in $\mathbb{P}^3$ space and $\mathbf{Q}$ is the 4×4 symmetric matrix that represents the coefficients of the quadric. Similar to conics in $\mathbb{P}^2$, the intersection of a quadric Q with the plane $\boldsymbol{\pi}$ at point $\mathbf{X}$ is defined by: $\boldsymbol{\pi} = Q\mathbf{X}$.

Similar to the case for conics, the dual quadric is defined on planes rather than points. In other words, the equation for the dual quadric Q^* can be defined in terms of its tangent planes $\boldsymbol{\pi}$ and can be written as:

$$\boldsymbol{\pi}^T Q^* \boldsymbol{\pi} = 0. \tag{2.7}$$

Figure 2.2 shows an example of a quadric surface.

2.2.5 Transformations in Projective Space

A projective transformation is often called a collineation or a homography. For example, in $\mathbb{P}^2$, a collineation h maps points $\mathbf{x}$ to $\mathbf{x}'$ according to the following:

$$\mathbf{x}' = H\mathbf{x} = h(\mathbf{x}) \tag{2.8}$$

Table 2.1: Transformations in projective plane and space under collineation H

Entity	before transformation	after transformation
point	$\mathbf{x}$	$H\mathbf{x}$
line	$\mathbf{l}$	$H^{-T}\mathbf{l}$
plane	$\boldsymbol{\pi}$	$H^{-T}\boldsymbol{\pi}$
conic	C	$H^{-T}CH^{-1}$
dual conic	C^*	HC^*H^T
quadric	Q	$H^{-T}QH^{-1}$
dual quadric	Q^*	HQ^*H^T

Here, the collineation H is a linear and invertible mapping of the homogenous coordinates, in other words:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{2.9}$$

Similar to points, other geometric entities can be transformed in the projective space using a collineation. For instance, under a projective collineation H, a line $\mathbf{l}$ is transformed into $\mathbf{l}'$ according to : $\mathbf{l}' = H^{-T}\mathbf{l}$. Table 2.1 describes how the projective collineation transforms various entities in the projective space.

2.2.6 Stratification of 3D Geometry

Often when the inverse problem of reconstructing the world from images is carried out, the geometry is defined up to an arbitrary transformation in a particular space. Under ideal circumstances, one obtains a geometry of the world up to an arbitrary metric transformation, T_{metric}. A metric transformation is a rigid transformation defined by a rotation and translation as defined by:

$$T_{metric} = \begin{bmatrix} r_{21} & r_{22} & r_{23} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.10}$$

In other words, the reconstruction of geometry up to an arbitrary metric transformation would resemble the real geometry, except for a rigid transformation from the reference coordinate

system. This means that all important geometric entities are preserved, angles and parallelism and volume are all preserved under a metric transformation. Also absolute measurements can be made under this set of transformations since lengths are preserved and there is no scaling involved in the transformation. This transformation contains 6 degrees of freedom, three for the rotation and three for the translation.

Under less ideal circumstances, one obtains a Euclidean geometry, $T_{Euclidean}$ of the scene. This is similar to the metric case except that there is now a scaling σ involved. This transformation is defined as:

$$T_{Euclidean} = \begin{bmatrix} \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & \sigma t_X \\ \sigma r_{21} & \sigma r_{22} & \sigma r_{23} & \sigma t_Y \\ \sigma r_{31} & \sigma r_{32} & \sigma r_{33} & \sigma t_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

Under these circumstances the geometry of the scene resembles that of the world, however metric measurements cannot be made under such a geometry due to the missing scale factor. This set of transformations contain an additional scale factor to the metric case, therefore there are 7 degrees of freedom.

Under yet less ideal circumstances, one obtains an "affine" geometry, T_{affine} of the scene. This set of transformations contain 12 degrees of freedom and are thus more general than the previous cases and contain them as subclasses. An affine transformation T_{affine} can be written as:

$$T_{affine} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

As a result of the shearing effects of this set of transformations, often an affine reconstruction seems a like scaled and sheared version of the world. The only entities that are preserved under this set of transformations are parallelism and ratios of volumes.

In cases where there is no information regarding the parameters of the cameras, the geometry of the world can be reconstructed up to an arbitrary "projective" transformation, $T_{projective}$. This is the most general set of transformations and has 15 degrees of freedom. Under this set of transformations the only things that are persevered are tangency and cross ratio. This

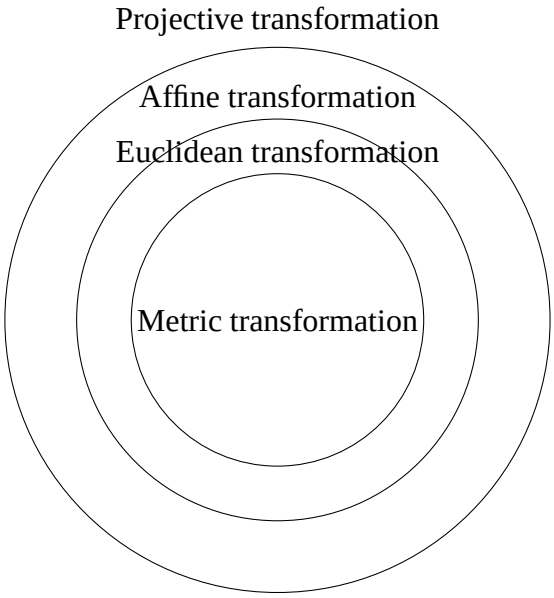


Figure 2.3: Family of transformations.

transformation can be denoted as:

$$T_{projective} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix} \tag{2.13}$$

Figure 2.3 outlines the set of transformations discussed above. As it is shown, the most general set of transformations are the projective ones. On the other hand, the most restrictive, and thus desirable, in case of scene reconstruction are the metric and Euclidean reconstructions. In these cases, the reconstructed scene differs from the actual scene by only a translation and a rotation (and possibly scale), making it ideal for visualization.

Figure 2.4 shows the visual outcome of applying the various families of transformations to a simple cubic shape. It is clear that the projective reconstruction often renders a scene unrecognizable since none of its geometric properties are preserved. On the other hand, the metric reconstruction only applies a rigid transformation to the cube.

2.2.7 Lines and Plane at Infinity

The projective space contains a set of additional points to the Euclidean space. Note the intersection of two parallel lines (a, b, c) and (a, b, c'). In Euclidean geometry these two lines

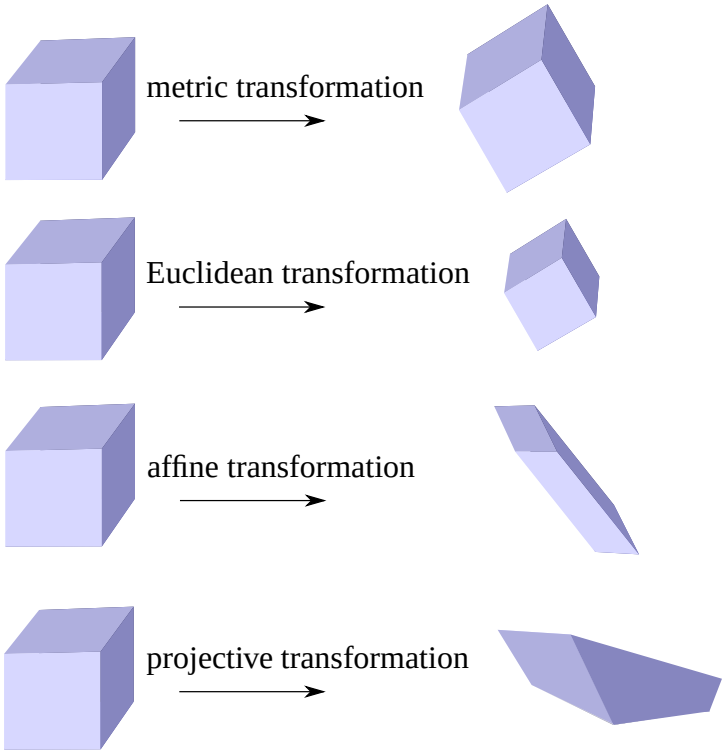


Figure 2.4: Various transformations applied to a cube.

do not intersect. However, in projective space the intersection of these lines can be found as: $(a, b, c) \times (a, b, c') = (b, -a, 0)$. This point has no non-homogenous equivalent since converting it back to non-homogenous coordinates one obtains: $(\frac{b}{0}, \frac{-a}{0})$. In projective geometry, such points are referred to as ideal points or points at infinity [44]. In fact, all parallel lines in this space meet at points at infinity.

Similarly in $\mathbb{P}^3$ all parallel planes meet at the plane at infinity denoted by $\boldsymbol{\pi}_\infty$. An interesting property of the plane at infinity is that it can be used to upgrade a reconstruction from a projective space to an affine space. In other words, given an existing projective reconstruction, one can "upgrade" this to an affine reconstruction by localizing the plane at infinity (i.e. finding its projective coordinates) and moving it to its canonical location at $(0, 0, 0, 1)$ by an appropriate transformation. This would account for the three additional degrees of freedom that a projective reconstruction has over an affine reconstruction. So, by localizing the plane at infinity these three degrees of freedom are fixed and one can make affine measurements from the reconstruction. For instance, once a scene is upgraded from projective to affine reconstruction, one can detect actual parallelism in the scene. This is because now the scene is defined up to an arbitrary affine transformation and as explained, parallelism is invariant under affine transformations.

2.2.8 Absolute Conic and Dual Quadric

Another important geometric identity defined in the $\mathbb{P}^3$ projective space is the absolute conic Ω_∞. This is a conic that is defined on the plane at infinity $\boldsymbol{\pi}_\infty$. In a metric frame, the plane at infinity would be located in its canonical position : $\boldsymbol{\pi}_\infty = (0, 0, 0, 1)$ and the absolute conic would be defined by the equation:

$$\begin{aligned} X_1^2 + Y_2^2 + Z_2^2 = 0 \\ \psi = 0. \end{aligned} \tag{2.14}$$

Considering only the plane at infinity where all the points have coordinates such that $\psi = 0$, the equation of this conic can be defined by:

$$(X, Y, Z) I (X, Y, Z)^T = 0 \tag{2.15}$$

Therefore, using the compact definition of conics explained in Eq. 2.5, we can say $\Omega_\infty = I$, defining a conic of purely imaginary points on the plane at infinity $\boldsymbol{\pi}_\infty$. One of the interesting properties of the absolute conic is that it is unchanged under any Euclidean transformations. Therefore, given an arbitrary reconstruction of the geometry of a scene, if we locate the absolute conic and moved it to its canonical position such that it has the coefficients represented by

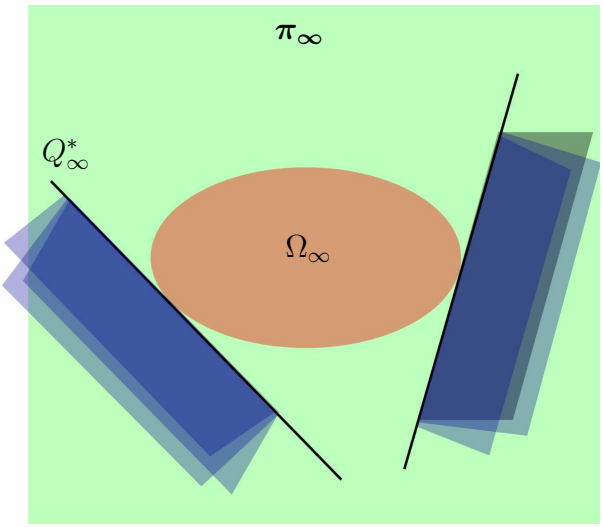


Figure 2.5: The absolute dual quadric.

the identity matrix, I, we can claim the scene is defined up to only a Euclidean transformation. This means we can now measure all Euclidean properties of the scene such as angles and parallelism. In fact, the five degrees of freedom that the affine transformation has over a Euclidean one, can be eliminated by locating the absolute conic and transforming the scene such that it is located in its canonical position. In fact, as shown in Section 2.5.4, this is one of the ways in which a projective geometry is upgraded to a Euclidean geometry in self-calibration.

One of the disadvantages of using the absolute conic for the task of changing a geometry from projective to Euclidean, is that it is represented by two equations Eq. 2.14, one to represent the plane at infinity and one to represent the absolute conic. A more convenient form of representing both geometric entities, is the dual of the absolute conic. This is referred to as the absolute dual quadric, represented by Q^*_∞ which consists of all planes tangent to the absolute conic as shown in Figure 2.5.

As is seen in Figure 2.5, the absolute dual quadric is the set of planes tangent to the absolute conic on the plane at infinity. The canonical representation of the absolute quadric in a Euclidean reconstruction is:

$$Q^*_\infty = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.16}$$

Since the absolute quadric contains implicitly the locations of the plane at infinity and the

absolute conic, one can upgrade a projective reconstruction to a Euclidean one after localizing the absolute quadric and transforming a scene such that it is placed in its canonic position. In fact, the absolute quadric contains 8 degrees of freedom which are fixed after locating it, making it possible to go from the projective geometry which has 15 degrees of freedom to the 7 degrees of freedom for a Euclidean reconstruction. In fact, one of the most prominent self-calibration methods uses the localization of the absolute quadric in order to upgrade a projective geometry to a Euclidean one [122].

2.3 Camera Model

The camera model used throughout this thesis is the standard pinhole camera model [44, 125] as shown in Figure 2.6. An ideal pinhole camera maps scene points into the image by a process that is completely determined by choosing a projection center (or camera center), $\mathbf{C}$ and a retinal or image plane. The projection of a scene point is then obtained as the intersection of a ray coming from the scene point and passing through the camera center with the image plane.

Figure 2.6 shows the process of projecting two scene points into the image plane. The example shows two corners of a cube being projected to the image plane. Most cameras are described adequately by this model for the purposes of SfM. However, in certain cases additional effects such as radial distortion have to be taken into account. In this thesis we have assumed that all images follow this pinhole camera model or that they have been undistorted to adhere to this model.

Note that the center of projection, $\mathbf{C}$ is a point where all rays must go through in order to form a pixel on the image sensor. This is also referred to as the camera center. In addition, the line from the camera center perpendicular to the image plane is called the principal axis of the camera and the point where the principal axis meets the image plane is called the principal point, or the optical center as shown in Figure 2.6. Note that the optical center is a point on the image plane and so its location is denoted with pixel coordinates.

The process of projecting a scene point into an image point involves three transformations:

1. A $\mathbb{P}^3 \rightarrow \mathbb{P}^3$ transformation from the world coordinate system to the camera coordinate system.

2. A $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ transformation from the camera coordinate system to the image plane coordinate system.

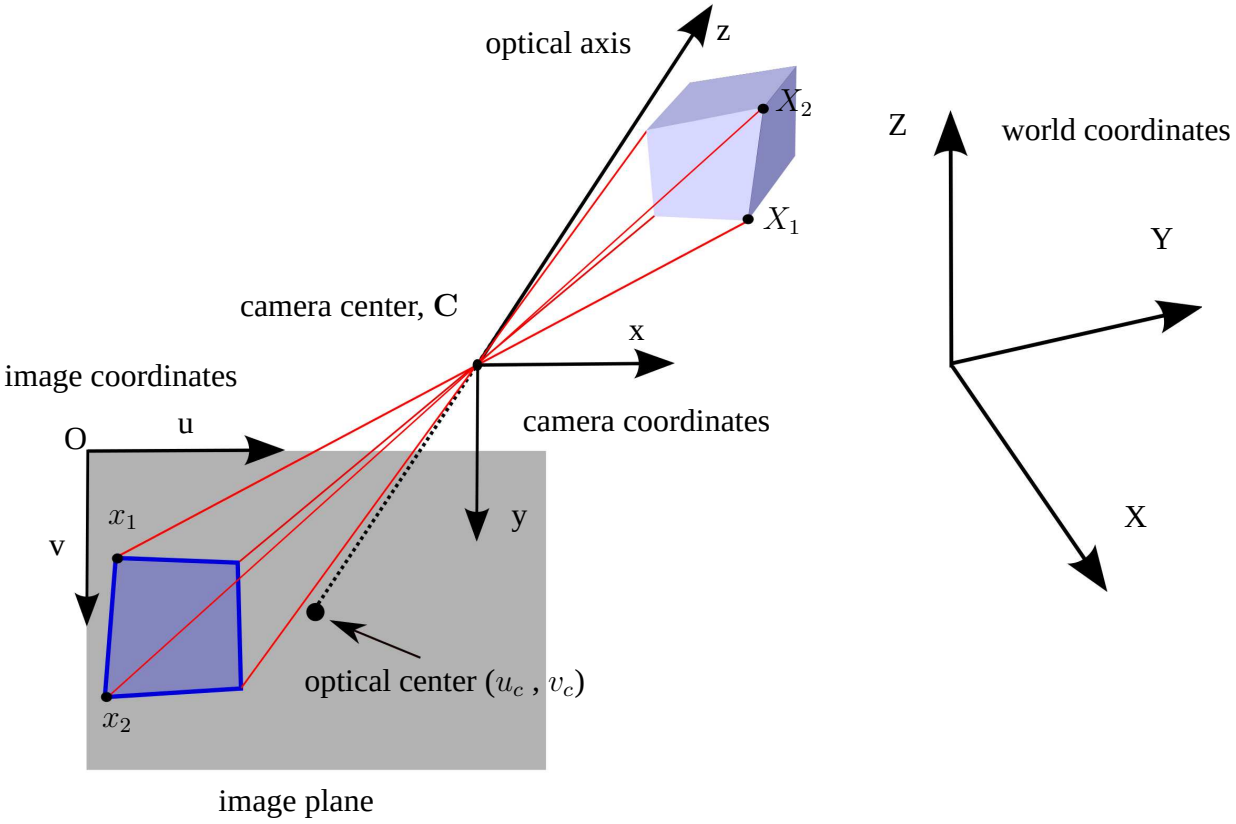


Figure 2.6: Pinhole camera model.

3. A $\mathbb{P}^2 \rightarrow \mathbb{P}^2$ transformation from the image plane coordinate system to the image pixel coordinates.

Using homogenous coordinate systems the point $\mathbf{X}$ is used to represent the coordinates of a point in the world coordinate systems, $\mathbf{X}_c$ as its coordinates in a camera coordinate systems and $\mathbf{x}$ as its image coordinates.

It is often with respect to a coordinate system that is independent of the camera that the reconstruction is carried out. Therefore, a similarity transformation is often used to transfer the coordinates of scene points to the camera's coordinate system as shown in Figure 2.6 . This similarity matrix maps coordinates according to:

$$\mathbf{X}_c = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \mathbf{X} \tag{2.17}$$

where R and $\mathbf{t}$ denote the rotation and translation between the camera and the world coordinate system. The parameters of this similarity transformation are often referred to as the extrinsic parameters of a camera. This is due to the fact that these parameters are independent of the actual camera and are related only to its position in space with respect to the world coordinate system. In addition, the mapping from the camera coordinate system to the pixel coordinates are carried out by:

$$\mathbf{x} = K[I|0]\mathbf{X}_c \tag{2.18}$$

where K denotes the intrinsic parameters of the camera as explained in the next section. Also, note that the overall transformation is a $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ transformation which embeds the transfer of the scene point from the camera coordinates to the image coordinates and that of transferring the image coordinates to pixel coordinates.

Combining all the transformations together, the process of transferring a scene point, $\mathbf{X}$, in the world coordinate system to an image point $\mathbf{x}$ in the image coordinates can be written as:

$$\mathbf{x} = K[R|\mathbf{t}]\mathbf{X} \tag{2.19}$$

For convenience, we can denote this set of transformation as matrix P such that: $P = K[R|\mathbf{t}]$. This matrix is of rank three and its null-space is the camera center $\mathbf{C}$ so that $P\mathbf{C} = 0$.

2.3.1 Intrinsic Parameters

Eq. 2.18 shows the matrix representation of the intrinsic parameters of the camera. These parameters are due to the mechanics and optics of the camera and are thus referred to as the

intrinsic parameters or the intrinsics matrix. The actual components of this matrix can be more clearly shown in the following:

$$K = \begin{bmatrix} f_x & s & u_c \\ 0 & f_y & v_c \\ 0 & 0 & 1 \end{bmatrix} \tag{2.20}$$

where f_x and f_y represent the focal length in the x and y dimensions in pixel units. More specifically, $f_x = fm_x$ where m_x is the number of pixels in unit distance along the x dimension and f represents the distance between the focal plane and the camera center. Similarly $f_y = fm_y$. Also the point (u_c, v_c) represents the optical center in terms of pixel coordinates. Here we can again write $u_c = m_x p_x$ and $u_v = m_y p_y$ where (p_x, p_y) is the physical location of the optical center. Throughout this thesis focal length has been represented by its x and y components f_x and f_y except in cases where for illustrative purposes we have assumed aspect ratio $\frac{f_x}{f_y}$ is one. In such cases the subscript is dropped and focal length is simply represented as f. Also at times, for brevity the x and y coordinates are combined into a vector representation as $\mathbf{f}$.

In addition, the skew parameter s can be decomposed as: $s = f\tan(\alpha m_y)$ where α is the skew angle between the sensors in the x and y directions. The skew factor is often assumed to be zero since most modern cameras have almost perfectly perpendicular sensors [44]. This assumption has been made throughout this thesis and the estimation of the skew has been ignored.

2.3.2 Camera Calibration

Calibrating a camera is an important step in most computer vision applications. This involves resolving the coefficients in the projection matrix P. This includes estimating the five intrinsic parameters as outlined in the previous section, and six extrinsic parameters including the translation and rotation with respect to the world coordinate system.

Some of the original methods of calibrating the camera parameters involve using calibration objects as shown in Figure 2.7. There are various methods that have proven to be effective in resolving these parameters. One of the earliest such methods was the one presented in [126] where the optical center is assumed to be given and various components of the projection matrix are calculated explicitly using a linear method.

The process involves detecting the markers in the calibration object and using them to find the coefficients of the projection matrix. Since the location of the markers are known in the world coordinate system, every single point detected gives two linear constraints on

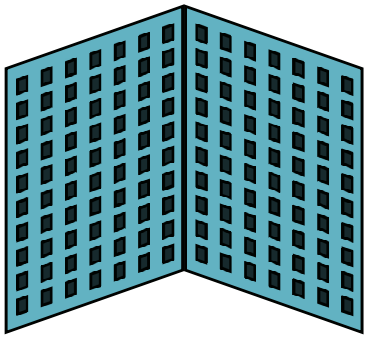


Figure 2.7: Calibration grid.

the coefficients of the projection matrix, at least six points are required to fully estimate the coefficients.

In other words, a single point $\mathbf{x}_i = (x_i, y_i, w_i)$, once detected in the image must obey the projection equation: $\mathbf{x}_i = P\mathbf{X}_i$ where $\mathbf{X}_i = (X_i, Y_i, Z_i, W_i)$ is the space coordinates of the point in our world coordinate system. Note that the location of both points are given in homogenous coordinates. This gives two linear constraints on the unknown coefficients as:

$$\begin{bmatrix} \mathbf{0^T} & -w_i\mathbf{X^T}_i & y_i\mathbf{X^T}_i \\ \mathbf{X^T}_i & \mathbf{0^T} & -x_i\mathbf{X^T}_i \end{bmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = 0 \tag{2.21}$$

where m_{ab} is the ab-th element of the projection matrix. Stacking these constraints, given a sufficient number of points detected on the calibration target, we obtain the typical least squares form of $A\mathbf{V} = 0$ where $\mathbf{V}$ is the column format of the projection matrix. This equation can be simply solved using a linear method. Once the projection matrix is found, the actual coefficients of the extrinsic and intrinsics matrix can be found using QR decomposition [44].

Another prominent method of finding the calibration parameters of a camera is the method

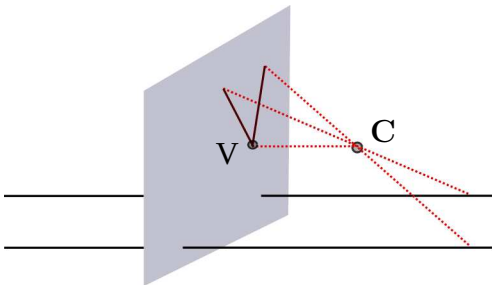


Figure 2.8: Vanishing point.

utilizing planar objects, presented in [132]. Using this calibration method, various snapshots of a planar calibration grid are taken. The locations of the markers on the calibration grid are then detected. However, in this case, it is assumed that the world coordinate system is aligned with the planar target and that the Z coordinate of all the marker points is zero since we have assumed that the plane resides on the plane where $Z = 0$. Combined with Eq. 2.23 our projection equation for such planar scene points becomes:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ \psi \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & t_1 \\ r_4 & r_5 & t_2 \\ r_7 & r_8 & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ \psi \end{bmatrix} = KH \begin{bmatrix} X \\ Y \\ \psi \end{bmatrix} \tag{2.22}$$

where H is a 3×3 collineation in $\mathbb{P}^3 \rightarrow \mathbb{P}^3$ and r_i and t_i elements are the coefficients of the rotation and translation of the extrinsic parameters. This is a linear one-to-one mapping from the planar object to the image locations as will be discussed in Section 2.4.2. Every such homography provides two independent constraints on the projection matrix as proven in [132]. With a sufficient number of views of the planar object enough such collineations can be accumulated to solve for the calibration coefficients.

Another class of calibration techniques takes advantage of the concept of vanishing points. Two such methods can be found in [16, 22]. Vanishing points are the result of the fact that perspective geometry can make infinite scenes look finite. For example, parallel railway lines are imaged as covering lines and their image intersection is the vanishing point for the direction of the railway. Figure 2.8 shows the vanishing point of two parallel scene lines. Point C denotes the projection center and V denotes the vanishing point for the two shown parallel lines.

The way vanishing lines are used in calibrating a camera can be summarized by the follow-

ing equation. Considering the points at infinity corresponding to the three orthogonal directions we can derive simple constraints on the elements of the projection [22]:

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ w_1 & w_2 & w_3 \end{bmatrix} = P \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.23}$$

where the right matrix contains the location of three orthogonal vanishing directions and the left matrix is the projection of those scene points in the image. Once the image location of the orthogonal vanishing points are detected, one can solve the above equation using an optimization routine. However, there are a few drawbacks with this approach. The main drawback being the lack of orthogonal parallel lines in scenes. Although suited for man-made environment, this approach fails when there are not enough vanishing points in the scene.

2.4 Two View Relations

Two-view relations are mathematical entities that relate image pairs. The fundamental matrix, homography and the essential matrix will be discussed. The most general of these entities is the fundamental matrix. Accurate estimation of the fundamental matrix is an important stage in the 3D reconstruction toolchain. This issue will be further discussed in Chapter 4. Also the problem of pose estimation from the two view geometry will be addressed in this section.

2.4.1 Fundamental Matrix

The fundamental matrix is the algebraic representation of the epipolar geometry. In essence, the epipolar geometry is the geometric relationship between the intersections of two image planes with the set of virtual planes that contain the camera centers of both cameras and scene points. This is shown in more detail in Figure 2.9.

Figure 2.9 shows the projection of a scene point, $\mathbf{X}$ into two image planes. The image points are denoted by $\mathbf{x}$ and $\mathbf{x}'$ and the camera centers are denoted by $\mathbf{C}$ and $\mathbf{C}'$. At first glance, it might seem that the projections of the scene point $\mathbf{X}$ in the two images do not have any geometric relation. However, in closer inspection, the scene point $\mathbf{X}$, the image points $\mathbf{x}$ and $\mathbf{x}'$ and the two camera centers are all coplanar. In fact, for every single point in the scene, its image points in the two cameras and the two camera centers are coplanar. This means that for every scene point, given the two camera centers one obtains a plane that intersects the two

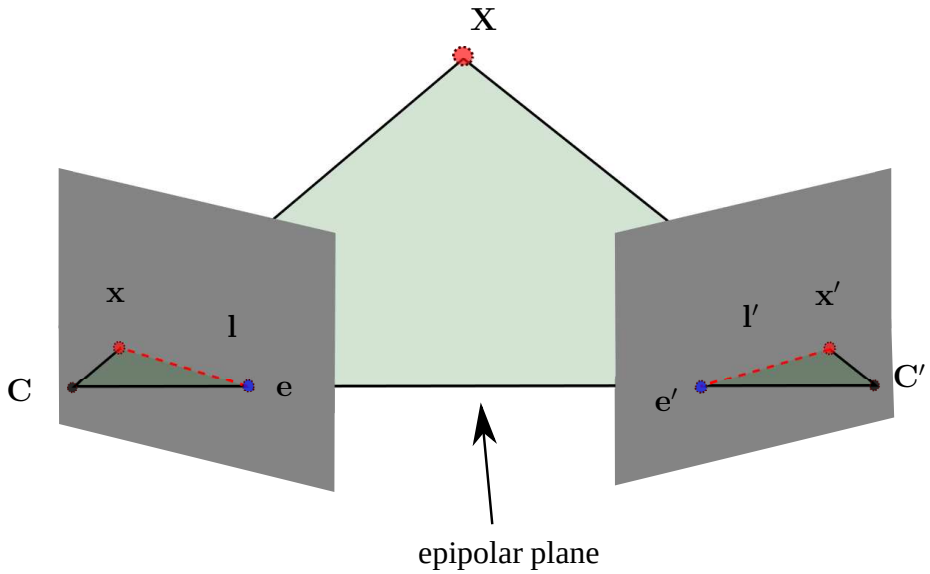


Figure 2.9: Epipolar geometry of two views.

images in two lines. Therefore, having located a pixel in one image, one can limit the location of the image of the same scene point in the other image to be on a line given by the intersection of the epipolar plane with the other image. In fact, this relationship is algebraically captured by the matrix representation of the fundamental matrix. This matrix represents a mapping from points in one image, into lines in the other image, $\mathbf{x} \rightarrow \mathbf{l}'$. This can be written as:

$$\mathbf{l}' = F\mathbf{x} \tag{2.24}$$

where F is the matrix form of the fundamental matrix and $\mathbf{l}'$ is the epipolar line in the right image corresponding to image point $\mathbf{x}$ as shown in Figure 2.9. In fact, since the point $\mathbf{x}'$ lies on the line $\mathbf{l}'$, the equation can also be written as:

$$\mathbf{x}'^{\mathbf{T}} F\mathbf{x} = 0 \tag{2.25}$$

As it can be seen from Figure 2.9, all epipolar planes intersect with each image at the so called "epipoles" as shown in the diagram by $\mathbf{e}$ and $\mathbf{e}'$. This is in fact the intersection of the baseline, or the line connecting the two camera centers, with the two image planes.

The problem of scene reconstruction depends to a high degree to the accuracy of finding matching image points. These are correspondences between points in pairs or more views of a scene. Using the fundamental matrix this can be done much more effectively since the search region is narrowed significantly to a 1D search rather than the whole image. Therefore, finding the fundamental matrix is an essential step in this process. Also, as will be discussed many

self-calibration methods rely on the fundamental matrix to produce accurate camera calibration parameters.

One method of finding the fundamental matrix between two views is using the projection matrices of the given views. For instance, if image one has projection matrix $P = K[I|0]$ and image two has $P' = K'[R|\mathbf{t}]$, the fundamental matrix can be found as [44]:

$$F = K'^{-T} R K^T [K R^T \mathbf{t}]_\times = K'^{-T} ([\mathbf{t}]_\times R) K^{-1} \tag{2.26}$$

where $[]_\times$ denotes the skew-symmetric matrix of a vector. Also under the assumption that is used in this thesis all cameras in a sequence or a image pair have the same set of intrinsic parameters and so $K' = K$. This will be assumed throughout the thesis and the notation will reflect this by using a single set of K parameters for all images.

However, the fundamental matrix is often found before the elements of the projection matrix become available. Thus a numerical method is required to find the fundamental matrix. Similar to the projection matrix in Eq. 2.21, this matrix can be solved using a linear method given a set of correspondences between the two images. In other words, having found a number of $\mathbf{x} \leftrightarrow \mathbf{x}'$ one can estimate the fundamental matrix between two views.

In this case, every image point correspondence provides a single linear constraint over the 9 coefficients of the 3×3 fundamental matrix. Seven point correspondences are at least required to estimate these parameters. The reason for this is that the fundamental matrix is estimated up to a scale and also there is a requirement that $\det F = 0$. Typically, there are many more than seven points present and so a least squares solution can be found using the formulation below:

$$\begin{bmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{bmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = 0 \tag{2.27}$$

where $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ and $\mathbf{x'^T} F \mathbf{x} = 0$. In this equation the fundamental matrix is represented in column format and the system of equations can be written more compactly as: $A\mathbf{b} = 0$ where $\mathbf{b}$ is the elements of the 3×3 fundamental matrix F arranged in a column. Stacking a sufficient number of points gives us enough constraints to solve this

equation using least squares. Note that the constraint on the determinant of the fundamental matrix can be imposed after this estimation by using singular value decomposition and setting the smallest singular value to zero. Also it is important to perform a data normalization on the matrix A in order to prevent numerical instability [43, 118].

In most real scenarios, the above set of equations fail to produce a reasonable solution due to erroneous data points in the A matrix due to matching errors. In fact, it is the goal of this thesis to improve the robustness of the process of estimating the fundamental matrix. Chapter 4 is devoted to the discussion on the robust estimation of the fundamental matrix where the pitfalls of using simple linear methods will be discussed.

2.4.2 Homography

Another important relation in multiple view geometry is the homography. This is a collineation in $\mathbb{P}^2$ that maps points to other points, unlike the fundamental matrix that maps points to lines. In other words, if the homography is denoted by H, this transformation maps point $\mathbf{x}$ to point $\mathbf{x}'$ by : $\mathbf{x}' = H\mathbf{x}$. Since the points are denoted in their homogenous notation this means that the matrix H is 3×3. The scenario that is of relevance to this thesis is where two images are related by a homography. This happens only when:

- Two camera are related only by a rotation around the center of projection (null translation).
- Two cameras both contain only images of the same scene plane.

The first case can simply be shown by the following example. Assuming the first camera is at the origin of the world coordinate system so that $P = K[I|0]$ and the second camera undergoes a rotation so that: $P = K[R|0]$ we can see that the projection of a scene point $\mathbf{X}$ in the first camera will be $K[I|0]\mathbf{X} = K\mathbf{X}$ and the projection in the second camera will be: $K[R|0]\mathbf{X} = KR\mathbf{X}$. Therefore, the relation between the projection of an image point with its corresponding point in the other one will simply be: $\mathbf{x}' = KRK^{-1}\mathbf{x} = H\mathbf{x}$ where the homography H is simply KRK^{-1}.

In the absence of information regarding the intrinsic parameters of the two cameras, K and the rotation R, a numerical method can be used to find the coefficients of the homography. A single point correspondence provides two independent linear constraints on the unknown coefficients of the homography such that:

$$\begin{bmatrix} 0 & 0 & 0 & -xw' & -yw' & -ww' & xy' & yy' & wy' \\ w'x & w'y & w'w & 0 & 0 & 0 & -x'x & -x'y & -x'w \end{bmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = 0 \qquad (2.28)$$

where $\mathbf{x} \leftrightarrow \mathbf{x}'$ are corresponding points and $\mathbf{x} = (x, y, w)^T$ and $\mathbf{x}' = (x', y', w')^T$ are their homogenous coordinates. Here we have again formed a homogenous set of equations in the form of $A\mathbf{h} = 0$ where $\mathbf{h}$ is the coefficients of the homography arranged in a column vector. This set of equations can be solved using a least squares method.

2.4.3 Essential Matrix

One can think of the essential matrix as the calibrated case of the fundamental matrix. Whereas the fundamental matrix maps points to lines in uncalibrated image pairs (i.e. where the intrinsic parameters, K are unknown) the essential matrix does the same in the camera coordinate system. This can be clarified by noting a pair of cameras where one is located in the canonical position at the origin of the world coordinate system so $P = K[I|0]$ and $P' = K[R|\mathbf{t}]$. If the intrinsic parameters of the cameras are known, one can obtain the "normalized" camera coordinates by pre-multiplying all image coordinates with K^{-1} so that we obtain the new normalized coordinates $\hat{\mathbf{x}}$ from the image coordinates $\mathbf{x}$ as : $\hat{\mathbf{x}} = K^{-1}\mathbf{x}$. This means that are projection matrices are now reduces to their extrinsic parameters and the intrinsic parameters can now be assumed to be identity. In other words: $P_{normalized} = [I|0]$ and $P'_{normalized} = [R|\mathbf{t}]$. This means that according to Eq. 2.26 the fundamental matrix is now:

$$E = F_{normalized} = K^{-T}RK^T[KR^T\mathbf{t}]_\times = I^{-T}RI^T[IR^T\mathbf{t}]_\times = R[R^T\mathbf{t}]_\times = [\mathbf{t}]_\times R \qquad (2.29)$$

and so the essential matrix can be found from the fundamental matrix if the intrinsic parameters are known by:

$$E = K^{-T}FK^{-1} \qquad (2.30)$$

Similar to the fundamental matrix, the essential matrix maps points in one image to lines in the other, so that:

$$\hat{\mathbf{x}}^T E \hat{\mathbf{x}}' = 0 \tag{2.31}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ are corresponding points expressed in the camera coordinate system.

Since the essential matrix consists of a rotation and a translation, there are five degrees of freedom in an essential matrix. This is because the translation has a scale ambiguity and so it can only be determined up to a scale.

One of the important properties of the essential matrix that will be utilized in the discussion of self-calibration in Section 2.5.4 is the one imposed by its singular values [46]. This states that the first two singular values of the essential matrix must be identical and the last must be zero for the matrix to be a valid essential matrix. In other words, any matrix with such properties will be decomposable to the constituent rotation and translation, otherwise this decomposition will not be valid.

2.4.4 Pose Estimation

Before we can reconstruct a scene from images, the extrinsic parameters of the images have to be found. This is to say that we must estimate the rotation and translation between every image with the world coordinate system. We can further simplify this problem by assuming that one of the images is located at the origin of the world coordinate system. This means that all we have to estimate is the "relative" orientation and translation between the given image pairs. Also, the translations are estimated up to a scale in our SfM framework since there is a scale ambiguity inherent in the problem [125]. As a result we can only reconstruct our scene up to an arbitrary Euclidean reconstruction.

The problem set up is illustrated in Figure 2.10 where the rigid transformations between three frames need to be estimated.

Clearly, the simplest method of estimating these parameters is by calibrating the cameras as in Eq. 2.21. However, it is often not practical to perform a full camera calibration using a target-based method. In the adopted SfM framework, the camera intrinsic and extrinsic parameters are inferred separately. Once the camera intrinsic parameters are found using self-calibration, the extrinsic parameters can easily be found.

Given a set of point correspondences between a set of images, it is easy to find the fundamental matrices as shown in Section 2.4.1. Given the fundamental matrices one can estimate the intrinsic parameters, K using self-calibration as will be shown in Chapter 5. Then according to Eq. 2.30 one can estimate the essential matrix E for each image pair. This matrix can

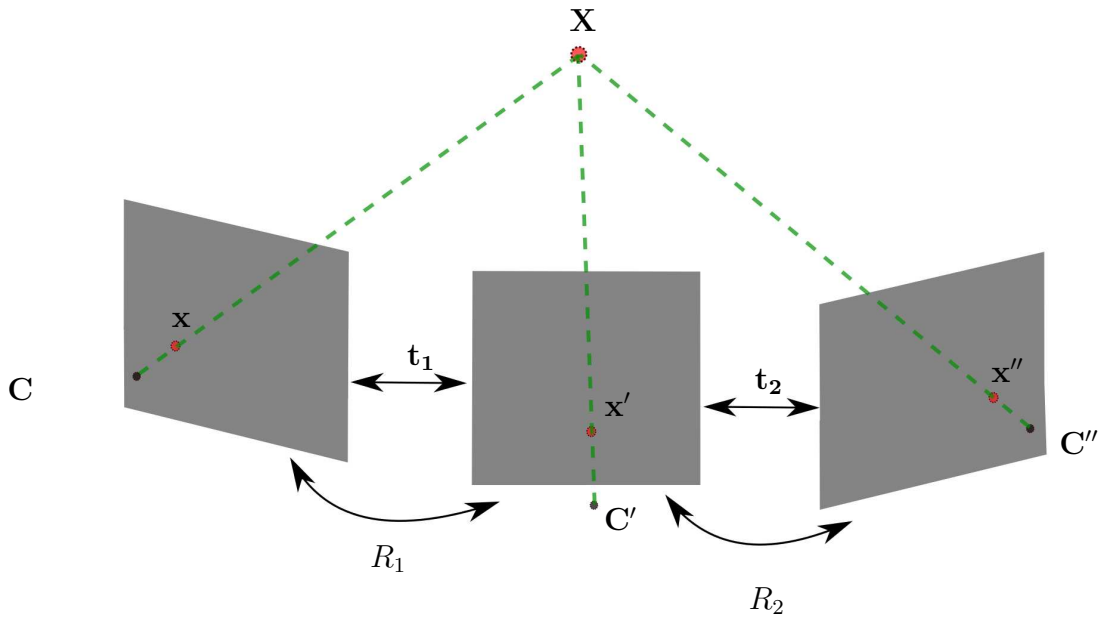


Figure 2.10: Pose estimation.

then be decomposed to give the relative orientation R and $\mathbf{t}$. The decomposition of E using singular value decomposition provides four different solutions for rotation and translation [44], so if $E = UWV^T$ then:

$$\begin{cases} R = UWV^T & \mathbf{t} = +\mathbf{u}_3 \\ R = UW^TV^T & \mathbf{t} = +\mathbf{u}_3 \\ R = UWV^T & \mathbf{t} = -\mathbf{u}_3 \\ R = UW^TV^T & \mathbf{t} = -\mathbf{u}_3 \end{cases} \tag{2.32}$$

This four-fold ambiguity can be resolved by performing a 3D reconstruction on a single point in the scene using all four different parameters and then choosing the setting that gives a scene where the scene point falls in front of both scene planes.

Note that once the parameters relating a pair of frames are estimated and the scene points reconstructed, the remaining frames can be added to the reconstruction using the process of "resectioning". This is basically using Eq. 2.21 where the camera parameters are estimated based on a set of known world points, but now the world points are estimated using a reconstruction method rather than being predetermined using a fabricated calibration target. This process can be repeated for any number of cameras once an initial reconstruction using two pairs of images is carried out.

2.5 Structure from Motion

Structure from motion (SfM) , or more accurately "structure AND motion" is the process of estimating world geometry from a set of images of a scene in addition to the position and orientation of the cameras used in taking those images. As explained, in our uncalibrated SfM framework it is assumed that no prior knowledge exists in terms of the parameters of the given cameras or the scene. This section gives an overview of the necessary components for a SfM framework using the previously defined concepts.

2.5.1 Image Matching and Triangulation

The first step in almost all SfM algorithms is the process of image matching. This is the process of establishing correspondence between images of scene entities across two or more views. These entities can be lines [11] points [60, 101] or other geometric entities. The main features used in this work are the SIFT and SIFT-PCA [60, 51] due to their reliability and invariance to large perspective transformations between the views. There have been many advances in feature matching such as SURF [12] that improve the efficiency of SIFT. However, this thesis focuses on later stages of SfM, and throughout the work it is assumed an initial set of feature matches exist across the views. The accuracy or quality required of these matches are however more relaxed than some competing techniques. In other words, the goal of the techniques presented in this work is to be able to cope with "low-quality" matches and still present a valid reconstruction.

Figure 2.11 shows an example of the process of finding point correspondences across two views of the Merton sequence [6]. This is done through the use of SIFT matching technique. The green lines show correct matches and red ones show incorrect matches.

Once point matches and camera pose has been estimated, the 3D geometry of the point matches can be estimated. This is done by back-projecting point matches into space and finding their intersection which effectively finds their originating scene point. This is shown in Figure 2.12 where point in one image and its correspondence in the other image has been found. Once this correspondence is established, the 3D location of the scene point whose image points were found can be estimated. However, for this to take place the projection matrices of the two cameras are required. As mentioned, once sparse correspondence is established, one can use various techniques to find the extrinsic and intrinsic parameters of every camera. So given image points $\mathbf{x}$ and $\mathbf{x}'$ and projection matrices P and P' such that: $\mathbf{x} = P\mathbf{X}$ and $\mathbf{x}' = P'\mathbf{X}$ the

Figure 2.11: SIFT-based point correspondences using the implementation found in [7], images from the Oxford sequence in [6].

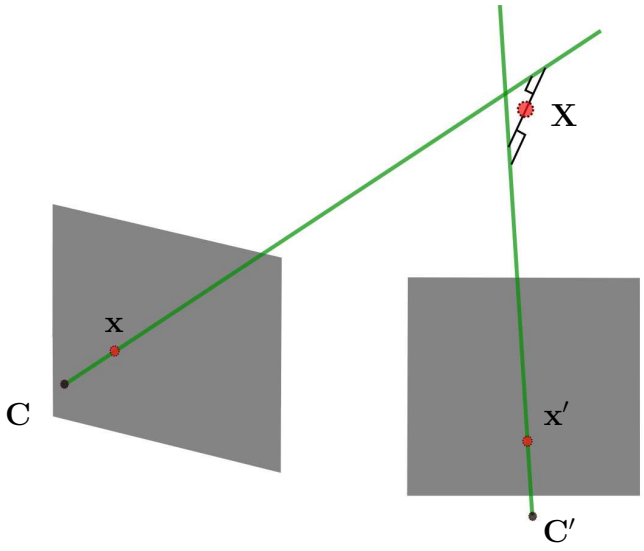


Figure 2.12: Triangulation to find 3D scene geometry.

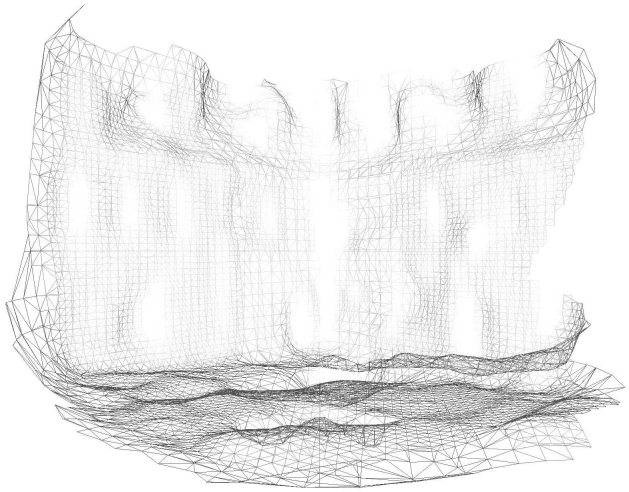


Figure 2.13: 3D reconstruction of Merton College sequence [6].

goal is to estimate $\mathbf{X}$. This can be done through the following:

$$\begin{cases} \mathbf{x} = P\mathbf{X} \Rightarrow & \mathbf{x} \times P\mathbf{X} = 0 \\ \mathbf{x}' = P'\mathbf{X} \Rightarrow & \mathbf{x}' \times P'\mathbf{X} = 0 \end{cases} \Rightarrow \begin{pmatrix} [\mathbf{x}]_\times P \\ [\mathbf{x}']_\times P' \end{pmatrix} \mathbf{X} = 0 \tag{2.33}$$

which is an over-determined set of equations. This is equivalent to finding the point that most closely fits between the rays starting from the camera centers and going through each image pixel. Since these rays rarely intersect, this over-determined set of equations effectively finds the most likely scene point to have originated the image points [33].

After pose estimation is carried out and the geometry of an initial set of sparse points is found, one can obtain a 3D reconstructions of the scene. An example of performing a dense reconstruction is shown in Figure 2.13, where the Merton scene shown in 2.11 is fully reconstructed. Here the point cloud that is triangulated using Eq. 2.33 is used to create a mesh via Delaunay triangulation. Further processing of the point clouds can take place by using a dense reconstruction algorithm [95] and texture mapping the polygons as done in [35].

2.5.2 Projective Reconstruction

As shown in Figure 2.4 a projective reconstruction often leaves a scene appearing significantly distorted. However, one advantage of a projective reconstruction of a scene is that there are no calibration requirements [28, 41]. In other words, using a set of image correspondences one can estimate an initial scene geometry without any knowledge of the camera poses or intrinsic parameters. Once this initial reconstruction is carried out, various methods can be used to upgrade this to an affine or Euclidean reconstruction.

One way to form a projective reconstruction from a pair of frames without any knowledge of cameras is to find the fundamental matrix between a pair of cameras. Given the fundamental matrix, two projection matrices of the pair of images can be found by:

$$\begin{cases} P_p = [I|0] \\ P'_p = [[\mathbf{e}']_\times F + \mathbf{e}'\mathbf{v}^\mathbf{T}|\lambda\mathbf{e}'] \end{cases} \tag{2.34}$$

where λ can be set to unity and $\mathbf{v}$ is an arbitrary 3-vector [85] and the subscript "p" denotes a projective reconstruction and $\mathbf{e}'$ is the epipole in the second image. This is also referred to as a canonic camera corresponding to a fundamental matrix.

Such a reconstruction suffers from the so-called projective ambiguity problem. In other words, if triangulation is carried out on all the corresponding points in the image pair via the two projection matrices as found above, the results will satisfy a whole family of world points and projection matrices. Considering that $\mathbf{x} = P_p\mathbf{X}_p$ one can also obtain $\mathbf{x} = (P_pH^{-1})(H\mathbf{X}_p)$ where H is an arbitrary collineation in 3D space and (P_pH^{-1}) is an entirely different projection matrix and $(H\mathbf{X}_p)$ is a different world point location. In fact, the search for the Euclidean geometry of the scene is the search for such a collineation that would map the projective space to the Euclidean space, $H_{p\to e}$.

2.5.3 Bundle Adjustment

One of the fundamental tools in photogrammetry is the process of bundle adjustment [124]. Bundle adjustment is a global optimization that aims to find the maximum likelihood estimate of a set of projection matrices and triangulated world points. Often, this is the last stage of a complete 3D reconstruction. Due to the complexity of this objective function, it is necessary to find a reasonable estimate of the structure and projection matrices to start the optimization with. Letting a set of camera projection matrices and image matches and their triangulated 3D world points be: $\mathbf{x}_i^k = P_k\mathbf{X}_i^k$, where k denotes the frame number, the bundle adjustment optimization can refine the structure by minimizing the reprojection error $|\mathbf{x}_i^k - P_k\mathbf{X}_i^k|$. This

is found by solving the following optimization:

$$\underset{P_k, \mathbf{X}_i^k}{min} \sum_{ik} d(P_k \mathbf{X}_i^k, \mathbf{x}_i^k) \tag{2.35}$$

where $d(\mathbf{x}, \mathbf{y})$ denotes geometric distance. A popular implementation which takes advantage of the sparse structure of this optimization problem is the implementation of the Sparse Bundle Adjustment in [58].

Even though the above method of optimizing projection matrices can provide a good estimate of the camera intrinsic parameters, it is important to start the optimization process with as accurate of an estimation of these parameters as possible. Since there are many local minima, the better an initial estimate used in the above formulation, the more likely it is to get a more accurate set of final values.

2.5.4 Self-calibration

The goal of self-calibration is to find the camera intrinsic parameters. However, unlike the methods presented in Section 2.3.2 the aim of self-calibration algorithms is to find the camera intrinsic parameters without resorting to the use of calibration targets or vanishing points or other scene constraints. In fact, the only information that is used in a typical self-calibration algorithm is a set of point correspondences across an image pair or an image set. Robust self-calibration is in fact one of the goals of this thesis and a more detailed background will be provided in Chapter 5. However, using the already introduced notations, two general approaches to self-calibration will be briefly reviewed.

As explained in Section 2.2.8, the absolute quadric is a dual quadric which encodes two important geometric entities that are of great importance in 3D reconstruction. The first entity is the plane at infinity whose localization in a projective reconstruction allows for an upgrade to an affine reconstruction. The second is the absolute conic, Ω_∞ which allows a further upgrade to a Euclidean reconstruction. In fact, the absolute quadric projects to the dual image of the absolute conic (the dual of the image of the absolute conic $\omega^* = \omega^{-1}$) [82, 122]. This is shown by the equation of the projection of the absolute quadric as:

$$\omega^* = P Q_\infty^* P^T \tag{2.36}$$

where P is the projection matrix of a camera. Also it is known that the image of the absolute conic ω^* is equivalent to KK^T [44]. Therefore, in a sequence of images with identical parameters, the absolute quadric projects to exactly the same position in all the images. Using this equation it is possible to transfer constraints on the camera parameters to the coefficients of

the absolute quadric. It is possible to solve for the coefficients of the absolute quadric using a linear method if certain restrictive assumptions are made on the camera parameters. For instance, if the camera is assumed to be calibrated except for a focal length, the above leads to a linear equation in the parameters of the absolute quadric. However, in the most general sense, this requires the use of a nonlinear minimization method.

One important aspect of the above method for self-calibration is the use of an initial projective reconstruction. Even though such a reconstruction has no value in terms of visualization, it enabled the estimation of the absolute quadric. In fact, the collineation T in 3D space that would map the estimated location of the absolute quadric to its canonical location would also transform the projective reconstruction to a Euclidean one. So if the absolute quadric is found in our projective reconstruction, the collineation T such that:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = TQ_{\infty}^{*}T^{T} \tag{2.37}$$

can be used to upgrade the projective reconstruction from Eq. 2.38 as:

$$\begin{cases} P_e = P_p T^{-1} \\ \mathbf{X}_e = T\mathbf{X}_p \end{cases} . \tag{2.38}$$

Therefore, the above method provides a set of camera parameters using the information obtained from a projective reconstruction. Once the absolute quadric is found, the projective reconstruction can be upgraded to a Euclidean one. Also the locations of the plane at infinity and the absolute conic can be found. This in turns gives us the camera parameters.

A second group of self-calibration methods rely solely on the fundamental matrix rather than a projective reconstruction of a sequence of images. An example of this category of methods is the use of the properties of the essential matrix for self-calibration. As explained in Section 2.4.4, the formula for the essential matrix is $E = [\mathbf{t}]_{\times}R$. It was shown in [46] that because the essential matrix is the product of a rotation matrix and a skew symmetric matrix, it has the special property that its first two singular values are identical. As a result, using a fundamental matrix whose equation is given by: $F = K^{-T}(R[\mathbf{t}]_{\times})K^{-1}$ one can search in the space of all intrinsic parameters such that the resulting essential matrix from $E = K^{-T}FK^{-1}$ has identical first and second singular values. This is performed by minimizing the difference between the singular values of the resulting essential matrix. In fact, the adopted self-calibration strategy in this thesis uses this constraint as will be detailed in Chapter 5.

2.5.5 Issue of Degeneracy in SfM

Oftentimes an SfM routine fails due to degeneracies inherent in the input images. The problem of degeneracy is one that affects a large number of vision algorithms. Three types of degeneracies are often encountered in the SfM context. These are:

1. Structure degeneracy: this is when structure cannot be instantiated from a pair of images or an image set due to camera geometry or the scene structure or noise in the correspondences.

2. Fundamental matrix degeneracy: this is when a unique fundamental matrix cannot be estimated due to camera geometry or the scene structure or noise in the correspondences.

3. Self-calibration degeneracies: when it is difficult or impossible to infer camera parameters from correspondences alone due to camera geometry or the scene structure or noise in the correspondences or noise in the estimation of the initial projective reconstruction or the estimation of the fundamental matrix.

In order for an SfM algorithms to be robust it needs to handle such scenarios. Even though a degeneracy often means a solution cannot be determined, the detection of such cases is of importance in order to avoid providing an erroneous set of results. Degeneracies in self-calibration are addressed in this thesis in Chapter 5. The other types of degeneracies are outside the scope of this work but the interested reader is referred to [117, 86].

In order to lay the ground work for the presentation of the robust algorithms presented in this thesis, the next chapter presents basic ideas in the area of robust statistics that will be utilized in the proposed algorithms.

Chapter 3

Robust Statistics

3.1 Introduction

Robust statistics is a field of study concerned with estimating parameters when certain ideal assumptions are violated. Robust techniques have a long history in the field of computer vision [105, 68, 69, 119]. In fact, the popular RANSAC algorithm that will be discussed in Section 3.3.6 is one in a plethora of robust techniques used in computer vision. Several of the most widely used robust techniques in computer vision will be briefly reviewed in this chapter. Initially the concept of linear regression is discussed and subsequent discussions of robustness are presented based on the notation presented herein.

3.2 Linear Regression

The idea of linear regression is to fit a linear model to a set of data points. So for instance, if there are N data points that we need to fit a model to, and for each data point i we know the p "predictor" values: $x_{i1}, ..., x_{ip}$ and "response" value y_i and we wish to fit a model to describe a linear relationship between the predictor and response values, we can write:

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + u_i, \qquad i = 1, ..., N \tag{3.1}$$

Or, rearranging the predictor values in matrix form $\mathbf{y} = X\boldsymbol{\beta} + \mathbf{u}$. The vector $\boldsymbol{\beta}$ is the vector of coefficients that need to be estimated. In addition, u_is are the "errors" affecting the measurements of the data points.

One might observe that many of the problems mentioned in Chapter 2 are similar to this formulation. For instance the fundamental matrix 2.4.1 is a linear model where the noise component $\mathbf{u}$ is the localization and matching errors as will be discussed in Chapter 4.

In order to estimate the unknown coefficients in the above formulation of linear regression, an explicit formula exists. This is done by finding:

$$\sum_{i=1}^{N} r_i^2(\boldsymbol{\beta}) = min. \tag{3.2}$$

where $r_i^2(\boldsymbol{\beta})$ is the squared residual of the i-th data point as a function of the unknown coefficients $\boldsymbol{\beta}$. In fact, the maximum likelihood estimation of the parameters $\boldsymbol{\beta}$ are found by minimizing the squared of the residuals [72]. Here the residuals are defined to be the error terms of the estimation or the difference between the estimated value of the response and the measured value of the response such that: $r_i(\boldsymbol{\beta}) = y_i - \hat{y_i}(\boldsymbol{\beta})$. In order to accomplish this minimization a simple explicit formula can be derived to find this "least squares" fit as:

$$\boldsymbol{\beta} = (X^T X)^{-1} X^T \mathbf{y}. \tag{3.3}$$

Minimizing the squared error, or the least squared solution, is in fact the optimal method of estimating the unknown parameters if the noise values are normally distributed [72]. However, most vision applications deal with problems that are far from this ideal case. In such cases, using a least squares method leads to highly erroneous results. Figures 3.1a and 3.1b illustrate examples of least squares fitting in the presence of noise. As it is shown in the figure, the left column's data points are affected by Gaussian noise and thus a least squares fit to the data points provides a reasonable model for the data. On the other hand, the right column shows the case where the noise affecting the data contains two points that have such large errors that they can be referred to as "outliers". While the first row shows the histogram of the errors affecting the data, the second row shows the QQ plot of this noise. A QQ plot is a graphical tool for checking whether or not a sample data set belongs to a particular distributions. This is done by plotting the quantiles of the sample data versus those of a theoretical distribution to see if the data belongs to that class of distributions [64]. A quantile is essentially the point under which a given proportion of data lies, so the 40% quantile is the point where 40% percent of the data fall under (and the 50% quantile is the median). Deviations from a straight line in a QQ plot as shown in the outlier contaminated data set in Figure 3.1b are indicative that the data does not belong to the assumed distribution (the normal distribution in this case). Even though there are only two outliers present in Figure 3.1b, the least squares solution has failed to provide a reasonable solution. This examples illustrates the inadequacy of least squares solutions in

dealing with outliers. This clearly shows the need for using alternative methods to cope with the non-ideal nature of the noise affecting measured data.

3.3 Robust Linear Regression

Oftentimes a number of assumptions must be made in order for an estimation technique to provide an optimal solution. For instance, in the presented example of least squares line fitting, the error in the data points is assumed to be Gaussian. In many situations such assumptions are invalid as is the case in computer vision. In addition, the cost of making such assumptions in cases where they are not applicable is significant as shown in Figure 3.1b. This motivates the need for estimators that do not depend on such assumptions.

However, before introducing the robust estimators, it is essential to review the types of noise that are most likely to effect data in vision applications. There are generally three types of noise that will be of concern in the discussions that follow.

3.3.1 Gaussian Noise

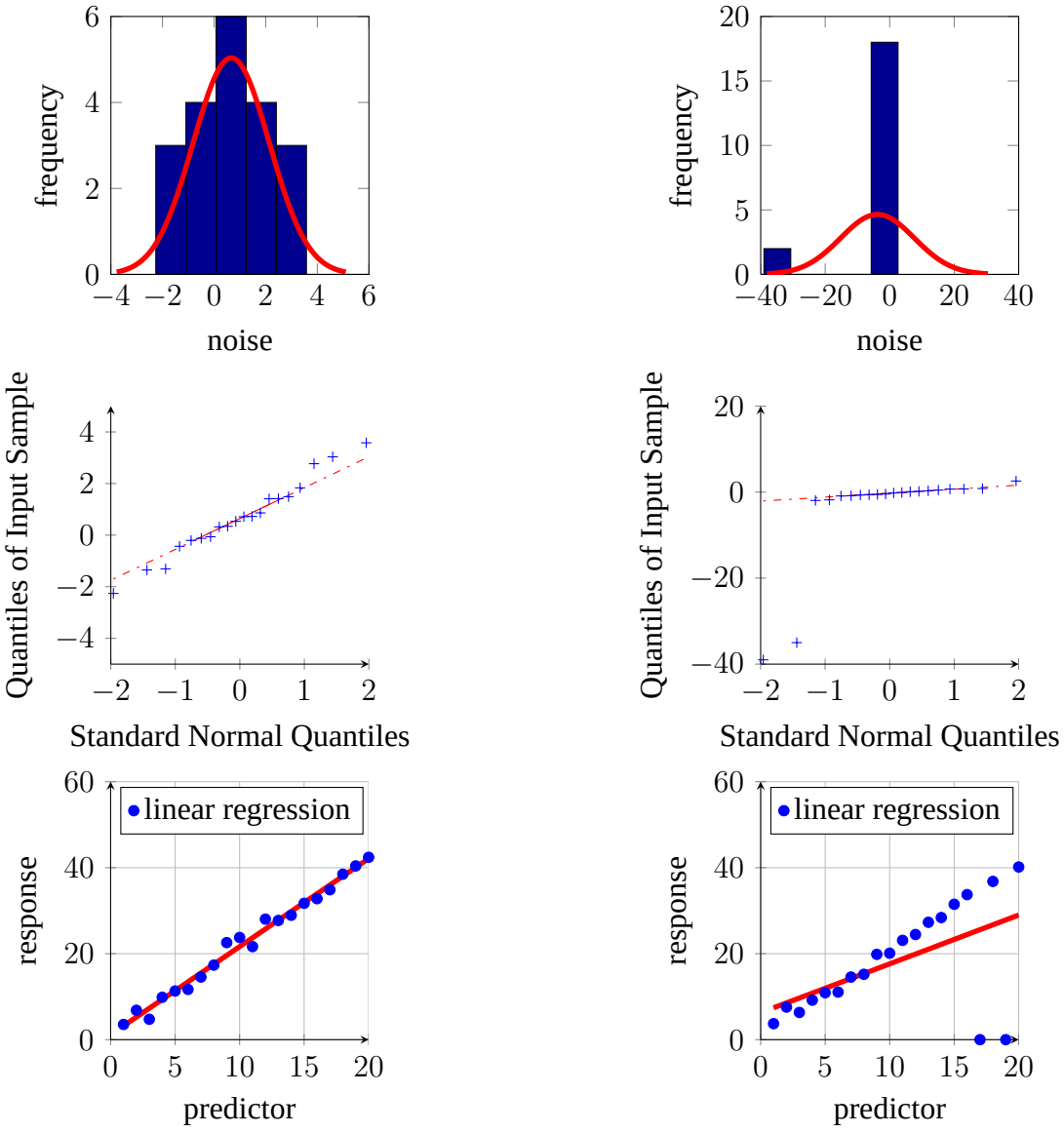
In this case the error term in Eq. 3.2 is drawn from a normal or Gaussian distribution: $u_i = N(0, \sigma^2)$. If the mean is not zero we consider the data to have a bias. An example of this is the feature localization error inherent in most feature detectors [115]. Having assumed Gaussian noise, the probability of a point having error r can be denoted as:

$$p(r|v) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{r-\mu}{\sigma}\right)^2}. \tag{3.4}$$

Here the label v is an indicator variable that denotes the event that a data point is not an outlier (i.e., an inlier) and $\hat{v}$ denotes the even of the point being a outlier.

3.3.2 Non-Gaussian Noise

In this case the noise terms, u_i are generated by more than one process. There is often a Gaussian component as is described in the previous section, and a non-Gaussian component affecting the underlying data. This second component is often produced by gross errors in measurement. In the case of SfM, one source of this error can be mismatches between pixel locations which can be modeled as a uniform distribution, so $p(r|\hat{v}) = \frac{1}{b-a}$ [115] where a and b define the space of possible error values. In future chapters this the outlier distribution is simply abbreviated to $p(r|\hat{v}) = \frac{1}{w}$ where w is the range of these outlier values defined in



(a) Least squares fit, Gaussian noise.

(b) Least squares fit, non-Gaussian noise.

Figure 3.1: Least squares fitting of lines for a data set containing Gaussian noise and one with non-Gaussian noise. The fit to the data containing non-Gaussian noise is quite inaccurate. The non-Gaussian data was created by merely adding two outlying points to the set.

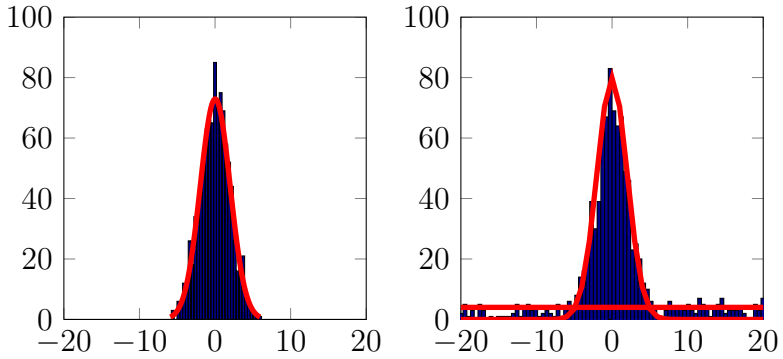


Figure 3.2: Distributions for a Gaussian noise and for a Gaussian noise with outliers.

$b - a$. Having assumed Gaussian noise plus a non-Gaussian component, the probability of a data point having error r can be denoted as:

$$p(r) = p(v)\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-1}{2}\left(\frac{r-\mu}{\sigma}\right)^2} + (1 - p(v))\frac{1}{b-a}. \tag{3.5}$$

where $p(v)$ denotes the probability of a point being an inlier, and $(1 - p(v))$ denotes the probability of a point being an outlier, also indicated by $\bar{v}$. Figure 3.2 shows the difference between the distributions of Gaussian noise and one with outliers. Here the outliers are modeled with a uniform distribution as in Eq. 3.5, but it is also possible that the outliers can take on other distributions such as a Gaussian with very long tails. Often the mixture model is a Gaussian with "fat" tails which means points with very large errors are now possible.

3.3.3 Multiple Structures

An even more challenging scenario is when the data not only contains Gaussian and non-Gaussian noise (i.e., outliers), but it contains multiple structures. In other words, the underlying models generating the data are not merely a single model. For example, if there are two models M_1 and M_2 that generate the data we can write:

$$y_i = \begin{cases} \sum_{j=1}^{p} x_{ij}\beta_j + u_i & i \in M_1 \\ \sum_{j=1}^{p} z_{ij}\theta_j + \varepsilon_i & i \in M_2 \end{cases} \tag{3.6}$$

where the noise can be a combination of Gaussian distributions and outlier distributions so for example:

$$u_i = p(v|M_1)N(0, \sigma_2^2) + (1 - p(v|M_1))U(0, a) \tag{3.7}$$

$$\epsilon_i = p(v|M_2)N(0, \sigma_1^2) + (1 - p(v|M_2))N(0, c\sigma^2) \tag{3.8}$$

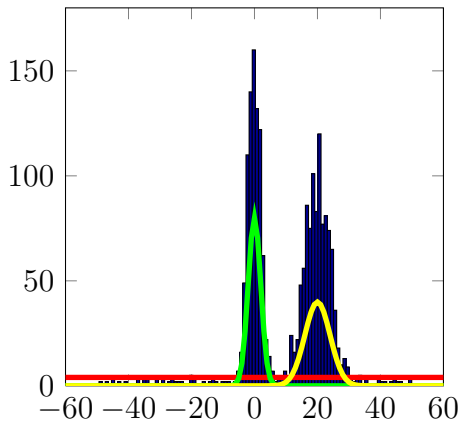


Figure 3.3: Error distribution for data with multiple structures. The errors also contain Gaussian noise and outliers as indicated by the fat tails of the distributions. The errors have been calculated with respect to one of the models.

where one of the models contains outliers from a uniform distribution and the other from a Gaussian with a high variance. As it is shown, there are two models present, M_1 and M_2. So there are now two sets of coefficients to be estimated. Figure 3.3 shows an example of the error distributions for a case with two underlying structures in addition to outlier noise. There have been few methods offered in solving such problems, such as sequentially estimating the most dominant model until only outliers are left. However, such models are difficult to estimate even with a robust method [104]. In such cases the outlier ratio can be extremely high since the inliers of one model act as pseudo-inliers with respect to another model as shown in Figure 3.3. Eq. 3.7 shows an example with two structures; however the general case can contain an arbitrary number of structures.

In addition to changing the way a regression problem is solved, the nature of the noise affecting the data changes the measure of the "quality" of a fit. In other words, one can opt for minimizing the least squared value of the residuals in the case of purely Gaussian noise. But in case of an arbitrary noise distribution this will not be the optimal quantity to minimize. The way this problem is approached is using a maximum likelihood estimation technique. This set of approaches attempt to estimate the coefficients $\boldsymbol{\beta}$ such that the probability of the errors of all data points, denoted by D is maximized. In other words we wish to maximize:

$$p(D|\boldsymbol{\beta}) = \prod_{i=1}^{n} p(r_i|\boldsymbol{\beta}) \tag{3.9}$$

where the logarithm is often used in order to turn the product into a sum, so that in the case of

Gaussian noise for instance, the following is minimized:

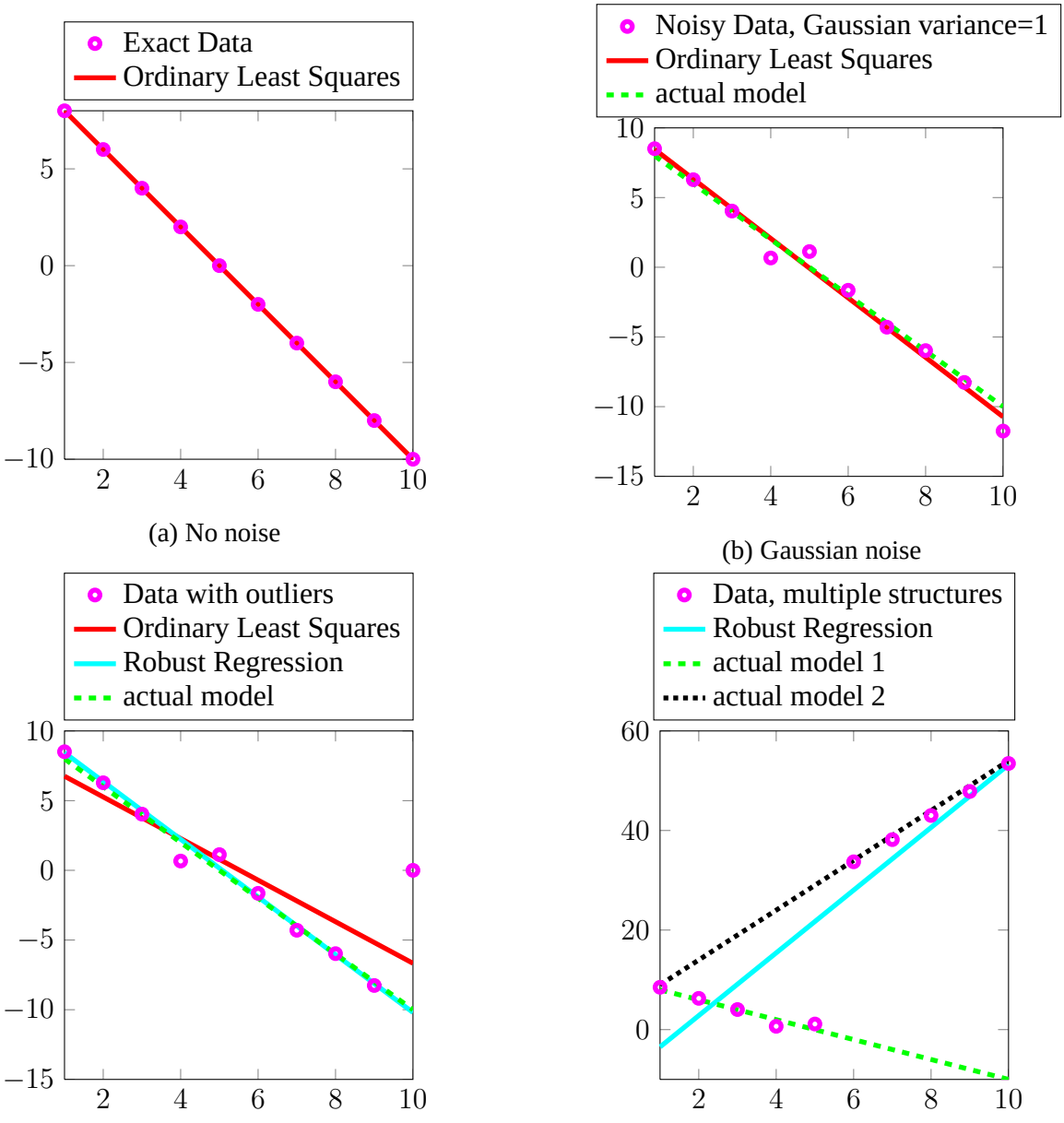
$$\log\left(\prod_{i=1}^{n} p(r_i|\boldsymbol{\beta})\right) = \sum_{i=1}^{n} \log\left(\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{r_i-\mu}{\sigma}\right)^2}\right). \tag{3.10}$$

This amounts to minimizing the square root of the errors [93]. However, in the presence of a non-Gaussian noise, this optimality condition will not hold. Also note the presence of μ and σ in Eq. 3.10. Although the mean is often assumed zero, the scale is an unknown that must be estimated before this expression can be minimized. This is another reason why it is important to carry out a density estimation on the errors in the problem that is to be solved before devising an optimal estimator. As a result, in Chapter 5 a large portion of the discussion on the robust estimation of self-calibration is dedicated to analyzing the types of noise that are present in this scenario.

As argued, the assumption of Gaussian noise is often too restrictive in real case scenarios including those arising in the field of computer vision. Therefore, it is essential to devise robust estimation techniques that are able to deal with minimal assumptions on the measured data. Figure 3.4 summarizes the four cases and how least squares fitting handles each case. Figure 3.4a shows the simple case of fitting a line to a noise-free set. Figure 3.4b shows least squares fitting to a case with Gaussian noise. As predicted, the fit is adequate and close to the actual model. Figure 3.4c shows the inadequacy of using a least squares fit in the presence of outliers. However, this graph introduces a simple robust estimator that is able to handle outliers. As it is shown, this estimator is able to fit a line to the data even in spite of the outliers. This is referred to as an M-estimator and will be discussed further in Section 3.3.5. And finally, Figure 3.4d shows the inadequacy of the least squared and the M-estimator in dealing with multiple structured data.

3.3.4 Leverage and Influence

Robustness often entails assessing the influence of single data points and removing the ones that exert an unusually large amount of influence on the final fit. Such points are often referred to as "leverage" points and they are often outliers. One of the simplest ways of assessing whether or not a point is an outlier is by analyzing the residuals of an initial least squares fit. As mentioned earlier, residuals are the difference between observations, y_i and their corresponding fitted values $\hat{y}_i$. Therefore, a residual can be thought of as the deviation between the data and the fit. Also, under certain circumstances a residual, r_i can also be thought of as the model error, u_i. Therefore analyzing residuals can be an effective way of assessing points that



(a) No noise

(b) Gaussian noise

(c) non-Gaussian noise

(d) multiple structures in data.

Figure 3.4: Summary of different line fitting methods in the presence of noise.

are outliers. This is in fact one of the adopted strategies in solving the fundamental matrix estimation problem as will be discussed in Chapter 4.

However, one of the disadvantages of using pure residuals in assessing outliers is the fact that oftentimes the variance of the errors of the different data points are different. In order to scale the residuals properly, first the relationship between the model errors and the residuals must be explained. Given Eq. 3.3 one can write the transformation between the fitted values of $\mathbf{y}$ denoted as $\hat{\mathbf{y}}$ and the observed values of $\mathbf{y}$ as:

$$\hat{\mathbf{y}} = X\hat{\boldsymbol{\beta}} = X(X^T X)^{-1} X^T \mathbf{y} = H\mathbf{y} \tag{3.11}$$

where the matrix H is referred to as the hat matrix and is simply: $H = X(X^T X)^{-1} X^T$. According to this, the residuals can be rewritten as:

$$\mathbf{r} = (I - H)\mathbf{y}. \tag{3.12}$$

In fact, the covariance of the residuals can be written as: $Cov(\mathbf{r}) = \sigma^2(I - H)$. Where σ is the standard deviation of the model errors $\mathbf{u}$. As a result, one can estimate the variance of a single data point as:

$$var(r_i) = \sigma^2(1 - h_{ii}) \tag{3.13}$$

where h_{ii} is the i-th element of the diagonal of the hat matrix. Therefore, the scaling of the residuals can be done by dividing them by their standard deviation. This is referred to as studentized residuals and can be written as [64]:

$$d_i = \frac{r_i}{\sqrt{MS_{res}(1 - h_{ii})}} \tag{3.14}$$

where MS_{res} is the estimate of the variance of the model errors. One estimate of this value can be found by using the variance of the residuals as:

$$MS_{res} = \frac{\sum_{i=1}^{N}(r_i - \bar{r})^2}{N - p} \tag{3.15}$$

where N is the number of data points and p is the number of parameters.

In addition to using residuals and studentized residuals, the measure of leverage for a data point i, denoted by h_{ii} is a telling measure of the amount of its influence in the regression estimate (also denoted simply by h_i). The leverage can be interpreted as the standardized measure of the distance of the i-th observation from the center of the space of all measured points. As a result, a large value of leverage, h_{ii} often reveals observations that are potentially

influential due to their distance from the rest of the data. In fact, points having leverage more than $\frac{2p}{N}$ are often analyzed further due to their high likelihood of being outliers [64].

In order to more accurately detect outliers, it is desirable to combine both residuals and leverage. There are many measures offered to combine these two, but one of the most prominent measures is the Cook's distance D_i [64] which is defined as:

$$D_i = \frac{d_i}{p} \frac{h_{ii}}{1 - h_{ii}}. \tag{3.16}$$

This formula is in fact the product of the square of the studentized residuals with $\frac{h_{ii}}{1-h_{ii}}$ which can be interpreted as the distance of i-th data point to the rest of the data. Therefore, D_i is consists of a component that indicates how well the model fits the i-th observation, y_i in addition to a term that measures how far the point is from the rest of the data. However, there is no known threshold for checking certain values of the Cook distance, and an approximate measure is to look for points having Cook distance larger than unity. There are also many other metrics for assessing influence, such as DFBETAS and DFFITS [93]. Most of these quantities also use various combinations of leverage and the data residuals.

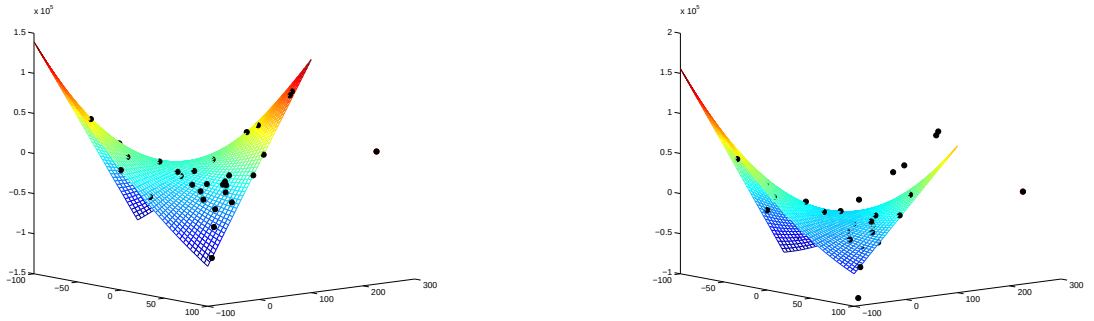
To see the comparison of the mentioned values and their role in discovering influential points, a simple regression is carried out for a quadratic surface in Figure 3.5. The example consists of creating a quadratic surface and choosing 30 random points from the model in addition to artificially creating an outlier that will be used for the fitting example. Figure 3.5a shows that all points are ground truth data (i.e., noise free or inliers) except one outlier shown in red which is deliberately inserted into the mix. Figure 3.5b shows the least squares fit to all the data points including the outlier. Table 3.1 shows the comparison of the various metrics on the used data points including the outlier which is in the last row and is highlighted. It can be seen that most of the quantities except for the pure residuals are highly indicative of the outlier point. Therefore, one method of robustly fitting a model to noisy data is to use an initial least squares fit and then by inspecting some of the outlined quantities in measuring influence removing points that are likely to be outliers. However, this method is only effective when the number of outliers are fairly small. When a moderate number of outliers are present in the data they tend to mask each other's influence [64] and render such an approach ineffective.

3.3.5 M-Estimators

Even though the previous section outlines several measures of assessing whether or not a point is an outlier, none provide an explicit objective function to be minimized in order to robustly fit a model to noisy data. Also as mentioned, such methods are highly ineffective at moderate

Point number	residual	studentized residuals	leverage	Cook's distance
1	-4913.7015	-0.7419	0.16896	0.019007
2	-339.8279	-0.048332	0.085214	3.784e-005
3	9011.7265	1.4518	0.22174	0.095669
4	-5285.7539	-0.76835	0.10188	0.011356
5	-4720.3725	-0.66528	0.0506	0.004025
6	-6024.5182	-0.88519	0.11376	0.016915
7	4141.4113	0.64825	0.23104	0.021565
8	-3660.9325	-0.51634	0.059101	0.0028791
9	-2779.4403	-0.40023	0.10145	0.0031235
10	3390.1373	0.50433	0.15469	0.0080061
11	10023.0814	1.704	0.27903	0.17353
12	17030.701	3.5313	0.33632	0.71261
13	-3915.3506	-0.55451	0.065213	0.0036814
14	-5130.5323	-0.725	0.052251	0.0049271
15	-4485.8732	-0.65382	0.11282	0.0092817
16	-2823.0048	-0.40328	0.08696	0.002675
17	8566.6017	1.4196	0.26721	0.11751
18	-2248.6565	-0.31818	0.071815	0.0013563
19	-4441.2866	-0.62574	0.052027	0.0036747
20	5106.3114	0.75505	0.13278	0.014813
21	10934.6197	2.0379	0.37113	0.36106
22	-7154.535	-1.1184	0.20158	0.052086
23	-4406.5411	-0.62036	0.050824	0.0035248
24	3699.5727	0.76425	0.55542	0.12376
25	-8427.4113	-1.3611	0.23356	0.090865
26	-876.3568	-0.12382	0.072527	0.00020836
27	-5369.085	-0.79956	0.1425	0.017976
28	2688.5126	0.51835	0.49645	0.045538
29	8354.328	1.2832	0.15957	0.050736
30	-5943.8235	-40810982.5325	0.97158	136.7431

Table 3.1: Measuring influence in the regression example of the quadratic surface.



(a) Correct model plus one outlier shown in red.

(b) Fitting to all points including outlier.

Figure 3.5: Measuring influence in regression. The example shows a set of points generated on a quadratic surface plus an outlier. The fitted surface significantly deviates from the inliers due to the influence of the single outlier.

noise ratios and often depend on heuristics to find a cutoff for deciding whether or not an influential point is an outlier. A more effective way of finding an estimate of the unknown coefficients without having to check for outliers is to use the concept of M-estimators (or maximum likelihood estimators).

As mentioned earlier, in order to find the regression coefficients under Gaussian noise, the maximum likelihood estimator is the one minimizing the squared errors $\sum_{i=1}^{N} r_i^2$. However, without the normality assumption on the distribution of the noise, one can define the maximum likelihood estimator to be:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \sum_{i=1}^{N} \rho(r_i) = \underset{\boldsymbol{\beta}}{\text{minimize}} \sum_{i=1}^{N} \rho(y_i - \mathbf{x^T}_i \boldsymbol{\beta}) \tag{3.17}$$

where $\mathbf{x^T}_i$ denotes the transpose of the i-th row of the observation matrix $\mathbf{X}$. This is referred to as an M-estimator, or a maximum likelihood estimator. As we saw previously, the $\rho(r_i)$ function amounts to a squared function of the errors for the Gaussian case. However, the M-estimator definition is more general and can cope with a much larger array of noise distributions.

In order to solve Eq. 3.18 the derivative with respect to $\boldsymbol{\beta}$ must be found:

$$\sum_{i=1}^{N} \psi \left(\frac{y_i - \mathbf{x^T}_i \boldsymbol{\beta}}{s} \right) \mathbf{x}_i \tag{3.18}$$

where ψ is the derivative of ρ and s is a preliminary estimate of the scale which is introduced to guarantee the scale invariance of the solution. This initial parameter of scale can be found robustly using a measure such as the interquartile range or the median absolute deviation [64].

Since Eq. 3.18 leads to a nonlinear set of equations, an iterative method can be used to solve for the coefficients. The most common way of doing this is through iteratively reweighted least squares [72].

The important design issue when using an M-estimator is deciding what the $\rho(r_i)$ should be. The goal is that "instead of minimizing a sum of squares, we minimize a sum of less rapidly increasing functions of the residuals" [47]. This removes the large impact that outliers have on the fit of the coefficients. There are in fact a large number of M-estimators, each with unique properties. The Table 3.2 outlines three popular M-estimators [131]. Note that each has a tuning parameter k which depends on the particular M-estimator.

M-estimator	ρ	ψ	w
Huber	$\begin{cases} \frac{r^2}{2} & \text{if } \lvert r\rvert \le k \\ k(\lvert r\rvert - \frac{k}{2}) & \text{if } \lvert r\rvert > k \end{cases}$	$\begin{cases} k \\ k\text{sgn}(x) \end{cases}$	$\begin{cases} 1 \\ \frac{k}{\lvert r\rvert} \end{cases}$
Cauchy	$\frac{k^2}{2}\log\left(1+(\frac{r}{k})^2\right)$	$\frac{x}{1+(\frac{r}{k})^2}$	$\frac{1}{1+(\frac{r}{k})^2}$
Bisquare	$\begin{cases} \frac{k^2}{6}(1-[1-(\frac{r}{k})^2]^3) & \text{if } \lvert r\rvert \le k \\ \frac{k^2}{6} & \text{if } \lvert r\rvert > k \end{cases}$	$\begin{cases} r[1-(\frac{r}{k})^2]^2 \\ 0 \end{cases}$	$\begin{cases} [1-(\frac{r}{k})^2]^2 \\ 0 \end{cases}$

Table 3.2: Three different M-estimators.

These M-estimators are graphically captured in Figure 3.6. As it is shown, the idea of an M-estimator is to weigh the data points according to their residuals so that the penalty associated with regression does not increase proportionately with the error. For example, ordinary least squares has an unbounded penalty function that increases proportionately with the error squared. Therefore, points having large errors will impose a large amount of influence on the regression coefficients. However, the robust M-estimators shown in Figure 3.6 all taper off or increase less rapidly than least squares when the error exceeds a threshold.

3.3.6 RANSAC

RANdom SAmple Consensus or RANSAC is an iterative sampling strategy for robust estimation [31]. RANSAC is a highly effective strategy for estimating a model in a high-noise situation. This hypothesis-and-verify algorithm proceeds by repeatedly creating a hypothesized model by drawing minimal subsets of data and fitting a model to these points. Following this, a score is assigned to the parameters estimated using this minimal set with respect to the

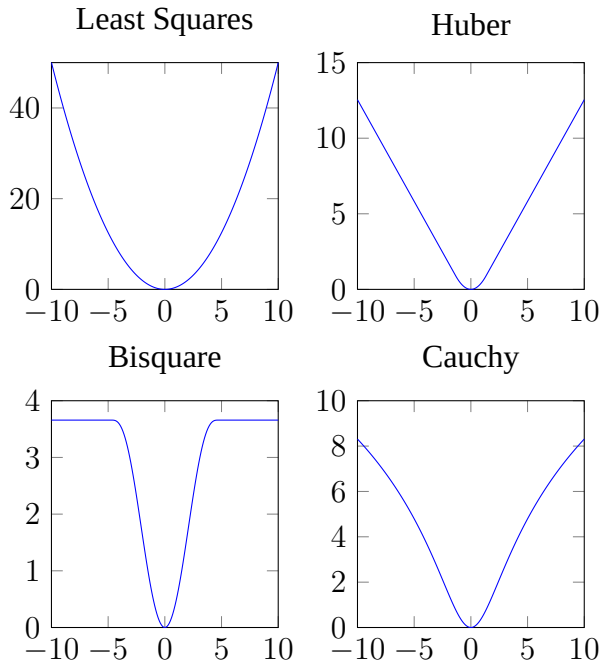


Figure 3.6: Four different M-estimator functions.

data. At the end, the best set of parameters that has achieved the highest score is selected. The flowchart presented in Figure 3.7 summarizes the iterative process of RANSAC.

The number of iterations that the RANSAC algorithm has to run through is calculated statistically. This number is based on the number of iterations that it takes to draw a minimal sample consisting of only inliers with a given user defined confidence level. This can be expressed mathematically as:

$$I_{max} = \frac{\log(1-\eta)}{\log(1-p(v)^m)} \tag{3.19}$$

where η is the user desired confidence level, often set to 0.99 and $p(v)$ is the inlier ratio and m is the number of points in a minimal subset, i.e., 2 in the case of a line, and I_{max} is the number of iterations. Note that here $p(v)$ is not the probability of a single inlier, but the probability of any given point being an inlier, which amounts to the inlier ratio.

Interestingly, casting RANSAC as an M-estimator, it is clear that the penalty function of RANSAC amount to:

$$\rho(r) = \begin{cases} Z & \text{if } r > T \\ 0 & \text{if } r \leq T \end{cases} \tag{3.20}$$

where T is the threshold and Z is any constant positive value. This threshold is directly proportional to the variance of the errors affecting the data. One method of setting this threshold

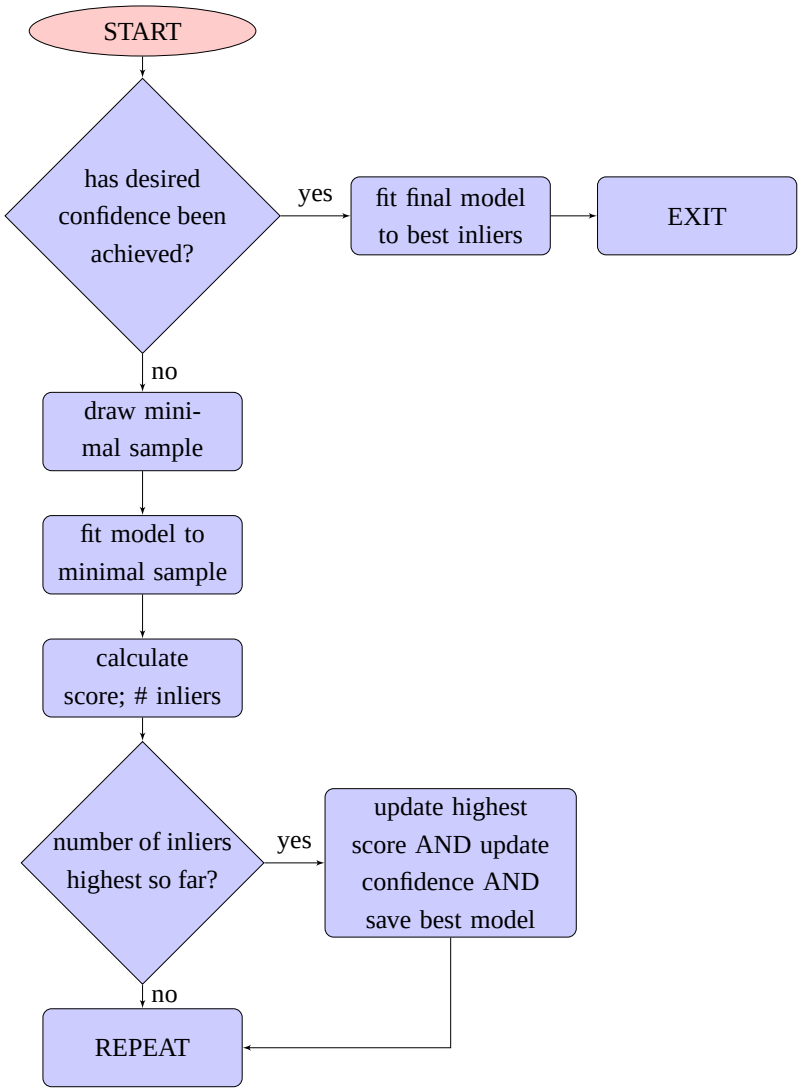


Figure 3.7: RANSAC flowchart.

is by: $T = 1.96\sigma$ where σ is the standard deviation of the input data [120] where the noise distribution is assumed to be Gaussian. The threshold is chosen so that Gaussian inliers are only incorrectly rejected five percent of the time.

Figure 3.8 shows an example of fitting a line to a set of data that is generated by two different models. In addition, the data has been corrupted by Gaussian noise. The first graph illustrates the underlying models used to generate the data points. The second through the last figure shows the successive progression of the RANSAC algorithm. Here the black points denote the minimal samples. In each graph a line has been fit to two randomly chosen points, also referred to as the "minimal sample". The score for each model is calculated based on the number of points that fall within a certain threshold to the model. In this case the dashes lines on either side of the estimated model show the limit of how close a point ought to be to be considered an inlier. The model with the highest score, defined as the number of inliers, is the final estimate of the model under consideration. It is clear that RANSAC is able to discover the underlying model in spite of multiple structures within the data.

3.4 Nonlinear Regression

Unlike the case for linear regression, where the equation is linear in terms of the unknown parameters $\boldsymbol{\beta}$, the nonlinear models can take on any form. The general form of a nonlinear model can be written as:

$$\mathbf{y} = f(X, \boldsymbol{\beta}) + \mathbf{u} \tag{3.21}$$

As opposed to the linear regression case where: $\mathbf{y} = X\boldsymbol{\beta} + \mathbf{u}$. In this case no explicit solution for $\boldsymbol{\beta}$ exists and the use of an iterative method is required. Using this form, the general least squares solution for a nonlinear model can be found by minimizing:

$$\sum_{i=1}^{N} [y_i - f(\mathbf{x}_i, \boldsymbol{\beta})]^2. \tag{3.22}$$

There are various existing methods for finding the solution to the above set of equations. One of the most notable examples is the Levenberg-Marquardt nonlinear least squares [97] whose implementation at [57] is utilized in the coming chapters.

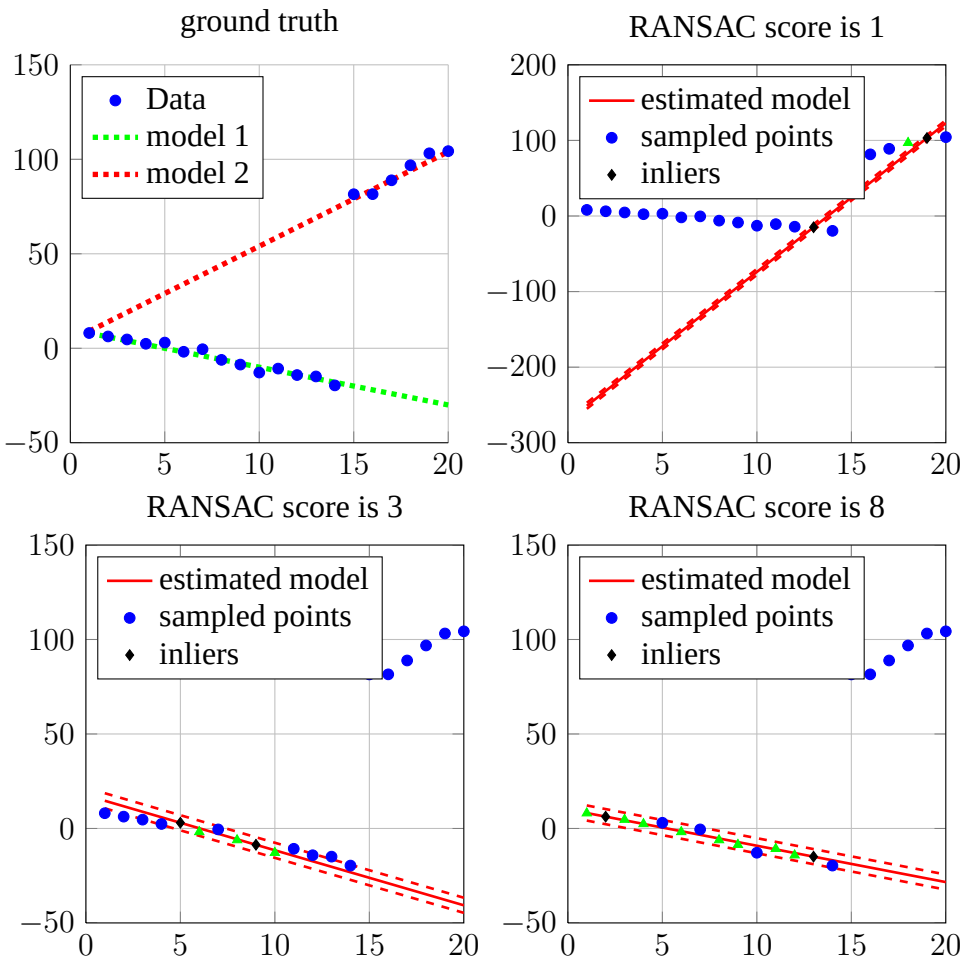


Figure 3.8: Example of using RANSAC for fitting a line to data set which consists of multiple structures (two underlying models) and Gaussian noise. The ground truth models and three iterations of RANSAC are displayed. Each hypothesized model has been shown with a solid line and the dashed lines show the support region where a point would be considered an inlier.

Chapter 4

Robust Motion Estimation

4.1 Introduction

As outlined in Section 2.4.1, the fundamental matrix is of great importance for many computer vision tasks. This geometric entity is used in many domains, such as:

- Dense point matching: by reducing the search space for image correspondences.
- Outlier removal for correspondence data.
- Stratified Reconstruction: by providing an initial projective reconstruction, refer to Eq. 2.34.
- Image rectification: by providing the image transformation that leads to a rectified image. [63]
- Self-calibration as discussed thoroughly in Chapter 5.

However, the accurate estimation of the fundamental matrix is a challenging task due to the very high noise ratio in the data. This data consists of the point matches between frames and often contains large number of outliers. Figure 4.1 demonstrates an example of the process of point matching between two image pairs using the SIFT point matcher [60]. The right column of Figure 4.1 shows the QQ plot for the errors (or residuals) of the correspondences with respect to the ground truth fundamental matrix which was found from a set of correct matches that were provided with the image sets [6]. Note that QQ plots were briefly described in Section 3.2 along with their usage for visually inspecting the normality assumption of a data set. The fact that both QQ plots show a strong deviation from a straight line demonstrates the

fact that the errors are far from being simply Gaussian noise. In fact, the underlying errors are indicative of a large number of outliers which are very typical of an image matching scenario. These outliers or image mismatches are marked in red in the left column.

As a result of the large number of gross outliers affecting the calculation of the fundamental matrix, combined with the essential role that the fundamental matrix plays is many computer vision tasks, there has been a great interest in devising new and robust estimators to solve this problem more effectively.

Before introducing some existing techniques to robustly estimate the fundamental matrix, some error metrics for assessing the quality of the fundamental matrix will be introduced.

4.2 Error Metrics for the Fundamental Matrix

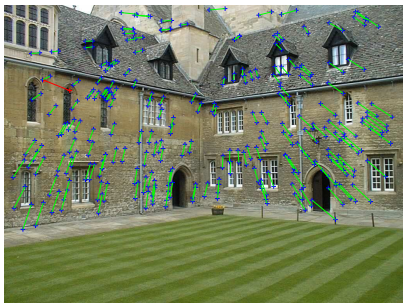
The simplest error metric to minimize in order to estimate the fundamental matrix is the algebraic error. Referring to Eq. 2.28, this can be done by finding the minimum of $||A\mathbf{b}||$ where $\mathbf{b}$ contains the elements of the 3×3 fundamental matrix F arranged in a column and A is the matrix formed as shown in Eq. 2.28. In addition, the rank 2 condition can be imposed after this minimization by finding the closest rank 2 matrix to the estimated F. This can be done by singular value decomposition by setting $F' = U\text{diag}(\sigma_1, \sigma_2, 0)V^T$ where $F = UWV^T$ and σ_1 and σ_2 are the first and second singular values of F and F' is the fundamental matrix with the rank 2 condition properly imposed.

A more meaningful metric to minimize would be the "symmetric epipolar distance". Here the metric that is minimized is simply the distance of a point to its corresponding epipolar line in each image. Since ideally one expects the points to be on their corresponding epipolar line, this would be an intuitive metric to minimize. This distance can be written as:

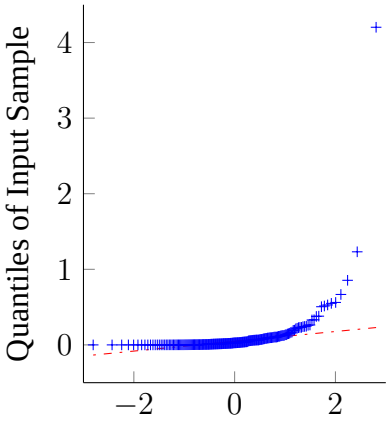
$$\sum_{i=1}^{n} d(\mathbf{x'}_i, F\mathbf{x}_i)^2 + d(\mathbf{x}_i, F^T\mathbf{x'}_i)^2 \tag{4.1}$$

where the i-th point in the first image is $\mathbf{x}_i$ and its match in the second is $\mathbf{x'}_i$ and so we can write $\mathbf{x}_i \leftrightarrow \mathbf{x'}_i$. Also, $d(\mathbf{x'}_i, F\mathbf{x}_i)$ is simply the perpendicular distance between point $\mathbf{x'}_i$ and its corresponding epipolar line $F\mathbf{x}_i$. Also, note F^T is the reverse mapping of a point in the second image into a line in the first image. Therefore Eq. 4.1 is a minimum when the epipolar lines are perfectly matched with their corresponding points.

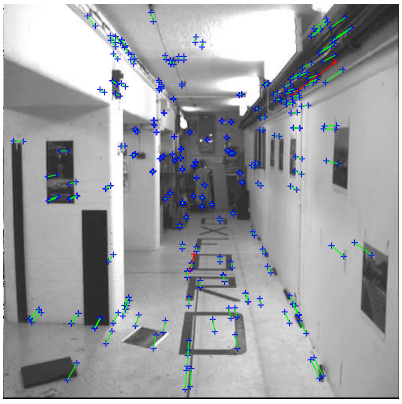
An even more useful error metric is the Sampson error that is known to give better results than the symmetric epipolar error [44]. This is the geometric distance to the first-order



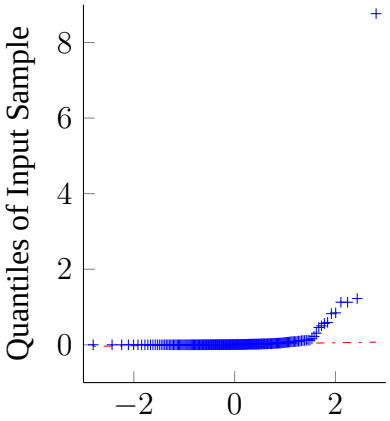
(a) Merton SIFT matches.



(b) QQ plot for the residuals of Merton matches.



(c) Corridor SIFT matches.



(d) QQ plot for the residuals of Corridor matches.

Figure 4.1: Two images from two separate image sequences have been shown with their raw SIFT matches. Inliers have been shown in green and outliers in red. The right diagrams show the errors of all the matches with respect to the fundamental matrix estimated from the inliers. The error magnitudes plotted in the QQ plots are calculated using the Sampson error metric defined in Eq. 4.2.

approximation of the fundamental matrix:

$$\sum_{i=1}^{n} \frac{(\mathbf{x'}^{\mathbf{T}}{}_i F \mathbf{x}_i)^2}{(F\mathbf{x}_i)_1^2 + (F\mathbf{x}_i)_2^2 + (F^T\mathbf{x'}_i)_1^2 + (F^T\mathbf{x'}_i)_2^2} \tag{4.2}$$

where $(F\mathbf{x}_i)_m^2$ denotes the square of the m-th entry of the vector $(F\mathbf{x}_i)$. Even though this error cannot be minimized using an explicit formula, it can be minimized using a nonlinear optimization method. In fact, the Sampson error has been used throughout this thesis as the measure of the accuracy of the fundamental matrix. All score functions, as outlined in Section 4.4.3 use this as a measure of error in fundamental matrices. Note that the Sampson error is the geometric distance to the first order approximation of the fundamental matrix. For the derivation of a more accurate measure of the geometric error refer to [40].

Finally, the minimization of the reprojection error of the fundamental matrix is a computationally expensive method of obtaining a good solution. This is an ideal metric to minimize when computation time is not an issue and a set of inliers have been identified. The reprojection error of the fundamental matrix can be written as:

$$\sum_{i=1}^{n} d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x'}_i, \hat{\mathbf{x}}'_i)^2 \tag{4.3}$$

where $\hat{\mathbf{x}}_i$ is the reprojection of point $\mathbf{x}_i$ using the projection matrix which was obtained by the fundamental matrix through Eq. 2.34. In other words, $\hat{\mathbf{x}}_i = P_p\mathbf{X}_i$ and $\hat{\mathbf{x}}'_i = P'_p\mathbf{X}_i$ where P_p and P'_p are the canonical projective cameras obtained from F and $\mathbf{X}_i$ is the triangulation of the points $\mathbf{x}_i$ and $\mathbf{x'}_i$ using these cameras. Therefore, the reprojection error is the minimization of the reprojection of matching points through a projective reconstruction due to the fundamental matrix. In other words, a projective reconstruction is carried out using the obtained fundamental matrix and its reprojection error minimized to find the most suitable fundamental matrix. The minimization is carried out over the projective reconstruction and the triangulated points.

Even though the minimized metrics do have an impact on the accuracy of the fundamental matrix, the essential idea is that the fitting is done to a set of inliers. Any of the aforementioned metrics will lead to an appropriate objective function if the data used for the estimation is free of outliers. However, if too many outliers exist in the data, none of the outlined objective functions when minimized will produce a suitable fundamental matrix. As a result, a robust method is required that is able to discern inliers from outliers and to effectively fit an accurate fundamental matrix to a set of image correspondences even if it contains a large number of outliers.

4.3 Basic Estimation Methods

A RANSAC based algorithm, as discussed in Section 3.3.6 requires two types of data fitting strategies, one for the minimal solver and one for the final stage where a set of inliers has been identified. This is the case regardless of the model that is to be fitted. These two techniques are generally different since in the case of the minimal solver one can estimate an exact fit but in the case of the final fit there are generally more constraints than unknowns and some measure of noise has to be minimized. In the case of the fundamental matrix, the minimal solver is referred to as the "7-point" algorithm and the general solver is referred to as the "8-point" algorithm. Whereas the minimal solver is used iteratively on randomly selected minimal samples, the 8-point algorithm is used once to fit the model to the final set of inliers. Note that it is also desirable to perform a nonlinear optimization in the final stage of fitting the fundamental matrix in order to further improve the accuracy. In this case the output of the 8-point algorithm can be used as a starting point for the nonlinear optimization algorithm and either the Sampson error or the reprojection error can be minimized.

4.3.1 Seven Point Algorithm

It was previously explained that the fundamental matrix has seven degrees of freedom. As a result, given seven points it is possible to estimate a fundamental matrix that exactly fits the data. This is done by forming the usual matrix form of $A\mathbf{b} = 0$. This would lead to a rank 7 matrix which would produce a family of solutions in the form of $\alpha F_1 + (1 - \alpha)F_2$. The parameter α can be estimated by enforcing the $\det F = 0$ constraint so that: $\det(\alpha F_1 + (1 - \alpha)F_2) = 0$. This equation can be solved numerically and leads to an exact fit to the seven input correspondences.

4.3.2 Eight Point Algorithm

This is a method that solves for the fundamental matrix given more than seven correspondences. In a RANSAC framework it is generally used for the final fit of the data. The idea is to use the general least squares formulation of $A\mathbf{b} = 0$ matrix and find the right nullspace using singular value decomposition.

There are two problems when applying the 8-point algorithm. If a proper data normalization is not applied, the results can be erroneous [118]. Also, the rank 2 condition must be imposed after the solution has been found which is not desirable since ideally this constraint should be imposed during the estimation. One way to enforce this constraint in the estimation

process is by using the epipolar parametrization [44]. This parametrization leads to a set of nonlinear equations and must be solved using a nonlinear optimization method.

4.4 Existing Algorithms

In order to estimate the fundamental matrix accurately, one needs a robust estimator to discern inliers from outliers, in addition to having an accurate optimizer that would refine the estimated fundamental matrix once inliers have been found. However, the contributions and the outline of the existing methods presented herein focus on the first step; the fast and accurate classification of data points as inliers and outliers. This is a far more important step in the process of estimating the fundamental matrix since once inliers are found most simple nonlinear minimization routines can be used to refine an initial estimate of the fundamental matrix found by a linear method.

In order to improve the robust estimation of the fundamental matrix we have adopted the RANSAC framework since the other algorithms reviewed in Chapter 3 do not perform as effectively as RANSAC in the robust estimation of the fundamental matrix. A thorough analysis of the use of various robust estimators for the estimation of the fundamental matrix has been carried out in [119] and it was concluded that the best algorithm would be one where RANSAC is used to find the set of inliers and then the results refined using another method. Other methods such as M-estimators are usually unsuitable since they are not as effective in handling higher error ratios. Therefore, the focus of this chapter will be on the RANSAC framework for robust estimation.

The following is a review of some existing methods for robust estimation of the fundamental matrix. They have been grouped according to the stage that they address in the RANSAC pipeline as shown in Figure 3.7. Following this, the proposed algorithms are presented and the chapter concluded with the experimental results.

4.4.1 Methods for Improving Score Metric

The following is a discussion on the different score measures in the literature for aggregating the minimal hypotheses results in the RANSAC framework. The variable Q is used to indicate the score of the whole dataset whereas q_i is used to denote the penalty incurred by a single correspondence point. This is similar to the idea of the robust M-estimator functions presented in Section 3.3.5. In fact, using the terminology used in Section 3.3.5, $q_i = \rho(r_i)$. Note that, the terms “score” and “penalty” have been used interchangeably in this thesis. These two terms

are used to denote either a value that is to be maximized or one to be minimized respectively, therefore a score function is simply the reverse of a penalty function.

As described in Section 3.3.6, RANSAC is an iterative algorithm that draws minimal samples and scores the generated hypotheses on these minimal samples according to the number of inliers. In other words, once a random sample is drawn, a model is fit to this random sample and then the number of points having less than a threshold error with respect to this model is counted. This would amount to drawing seven data points, or correspondences, in the case of the fundamental matrix and using the 7-point algorithm to fit a model to each sample. A penalty is then calculated based on:

$$q_i = \begin{cases} Z & \text{if } r_i > T \\ 0 & \text{if } r_i \leq T \end{cases} \tag{4.4}$$

where r_i is the square root of Sampson error of the i-th point with respect to a given minimal hypothesis and Z is a fixed score. There are a few problems with this score however, one being the unknown parameter T. The second being the fact that this is a binary score function and it assigns a fixed penalty to the error values. Therefore a point having a zero error and one having error T get assigned the same score. A more plausible score function that scores inliers not only based on the fact that they are inliers, but also on how well these inliers fit the model, is the MSAC (M-estimator SAmple Consensus) [120] penalty:

$$q_i = \begin{cases} T & \text{if } r_i > T \\ r_i & \text{if } r_i \leq T \end{cases} \tag{4.5}$$

where the penalty is merely the error value when the data point is an inlier, and a constant value otherwise. This is an improvement over the RANSAC score since the quality of inliers is also taken into account. However, an even more efficient score function was introduced in MLESAC [120] where a maximum likelihood framework is used to penalize error values. This score function is a continuous function and is based on a mixture distribution that accounts for the distribution of the outliers and inliers as discussed in Eq. 3.3.2. This score function is given by:

$$Q = -\sum_{i=1}^{n} \log p(r_i|F) = -\log \sum_{i=1}^{n} \left(p(v) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{r-\mu}{\sigma}\right)^2} + (1 - p(v)) \frac{1}{w}. \right). \tag{4.6}$$

In this score formulation, there are still three parameters that need to be estimated. One is the scale value σ, and one is the inlier ratio for the entire data set $p(v)$. The third variable is the

"window size" w or the range of possible error values that an outlier can take as explained in Section 3.3.2. This depends on the search window size and can be set according to the size of the image. The inlier ratio $p(v)$ must be estimated using expectation maximization as shown in [120]. This has to be done every time a new best hypothesis is found.

One issue with the MLESAC score is the fact that the parameters of the distribution change during iterations. In other words, one can have an entirely different function of the residuals in one iteration than the next. Therefore, comparing two score functions in two different iterations does not always yield sensible results.

An improvement over the MLESAC score function was presented in [115] where each point is assigned an individual probability of being an inlier $p(v_i)$ and this is used in the calculation of the score. This probability is also used in the sampling process, as will be described in the next section, and is estimated from the matching scores between the correspondences. Therefore, this matching score is a modification of Eq. 4.6 where we have:

$$Q = -\sum_{i=1}^{n} \log p(r_i|F) = -\log \sum_{i=1}^{n} \left(p(v_i) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{r-\mu}{\sigma}\right)^2} + (1 - p(v_i)) \frac{1}{w}. \right). \qquad (4.7)$$

Even though, unlike the MLESAC score, the parameters of this score function do not change during the iterations, the score is still dependent on an *a priori* probability for the correspondences that is unknown and has to be estimated separately based on matching scores. As will be argued in Section 4.5, inlier probabilities that are driven from matching scores are often less than adequate in many cases.

Another measure of the score of each data point as presented in [130] is the value of the kurtosis (or peakedness) of the distribution of the residuals of that point throughout all the iterations of RANSAC. This is used by generating hypotheses, saving the error for each data point at every iterations and then calculating the kurtosis of the resulting histogram. The authors argue that the kurtosis of inliers and outliers will form separable clusters in the kurtosis space. The authors argue that the distributions of the errors of an inlier point is more likely to have strong peak near zero and thus have a high kurtosis, whereas an outlier will form a uniform distribution without any strong peaks. Based on this they use a clustering method to classify each data point depending on the kurtosis of its error distribution. To this end, the value of the kurtosis is used as a feature in a classification framework. Figure 4.2 shows the error distributions of two sample inliers versus two sample outliers for the Corridor sequence during 500 RANSAC trials. The top row shows the histogram of the residuals of the two inliers and the lower row shows the histogram of the residuals of the two outliers. Clearly the distributions are different and this information can often be used to discern between inliers and outliers.

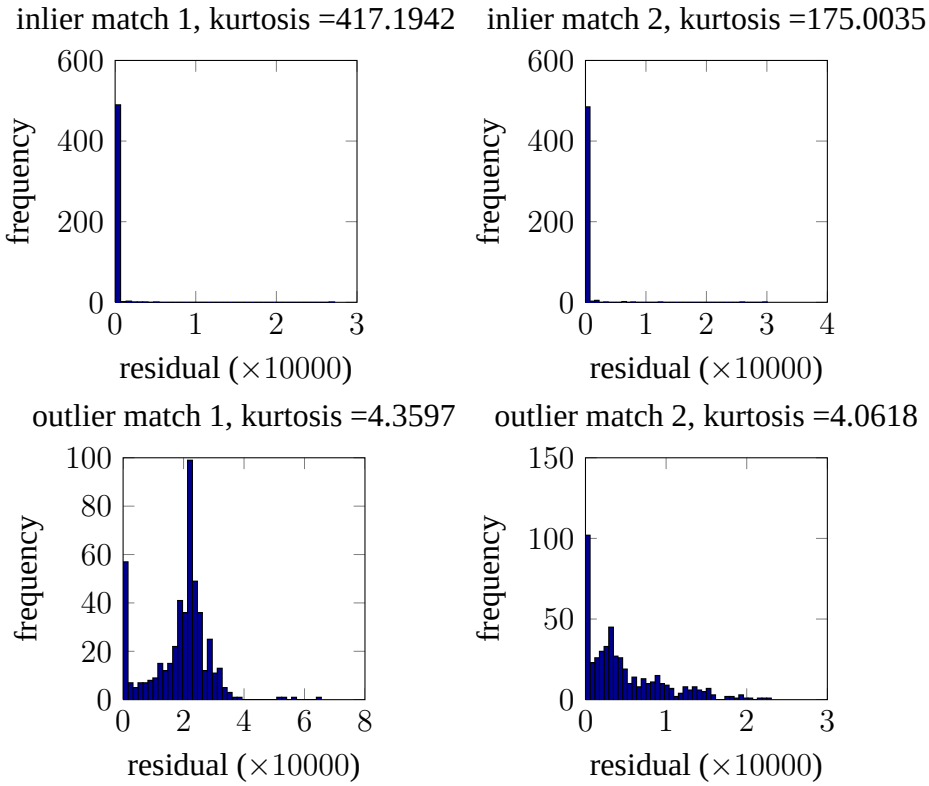


Figure 4.2: Distributions of the residuals of two sample inliers and two sample outliers over 500 RANSAC iterations and their kurtosis.

The authors admit that the method begins to deteriorate beyond a 0.65 outlier ratio. In addition, the method requires a fixed set of iterations, independent of the number of outliers. Also, a question that arises here is whether or not "phantom" outliers will be formed by the classifier if no actual outliers exist in the underlying data.

Figure 4.3 summarizes the three main penalty functions described so far. The MLESAC function is shown with three different values for $p(v)$. As discussed, each penalty function is a function of the matching errors with respect to a fundamental matrix. The goal is to minimize such a penalty function in order to obtain the more accurate fundamental matrix. Clearly the MLESAC function shown in Figure 4.3 is superior to the other two; since unlike MSAC and RANSAC functions MLESAC is a smooth function of the error.

There are also additional penalty measures, such as Least of Median squares (LMedS) [131] where the median of the residuals is used as the score, or Least trimmed squares (LTS) where the value of a certain subset of the residuals is used as the value to be minimized [93]. These methods are robust but they have a breakdown point of 0.5 outlier ratio.

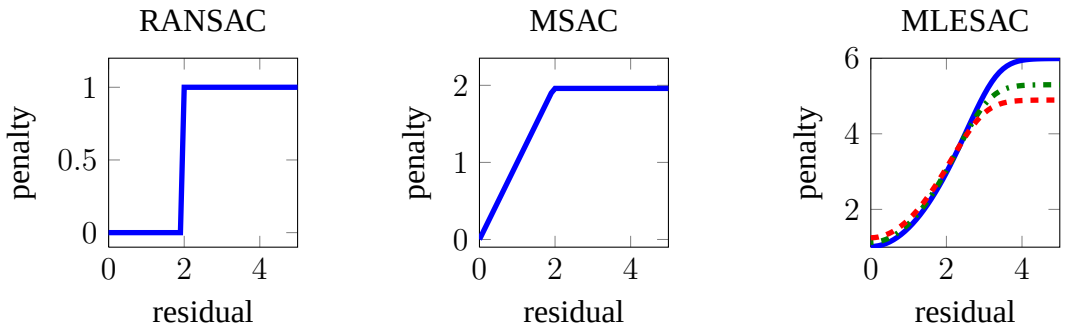


Figure 4.3: Summary of three popular penalty functions in the RANSAC framework which have been discussed so far. RANSAC in Eq. 4.4, MSAC in Eq. 4.5 and MLESAC in Eq. 4.6 for three different values of $p(v)$.

4.4.2 Methods for Improving Hypothesis Generation

Once a minimal sample is chosen, the traditional RANSAC algorithm uses the 7-point algorithm to fit a model to this minimal sample. However, there are alternative methods of fitting minimal samples. One would be to add additional points by drawing more than the required minimal sample, such as 8 points in the case of the fundamental matrix. Also, it is possible to fit a new model to the inliers found from the minimal samples using the 8-point algorithm as done in Lo-RANSAC [21]. The estimated fundamental matrix at each iteration can also be further refined by using a nonlinear optimization method to re-fit the inliers. Figure 4.4 shows the difference between the residuals of the 7-point algorithm and those of a further refinement by refitting to the inliers of the minimal sample. It is shown that the residuals decrease when the inliers are further processed. The refitting to inliers via the 8-point algorithm has a slight advantage since it uses a larger number of constraints to fit the hypothesized fundamental matrix. The authors in [21] also propose an additional method whereby sampling is done on the inliers of the minimal fit and new models are fit to this new inlier set. This improves the results but comes at a computational cost that might not be generally acceptable.

4.4.3 Methods for Improving the Sampling Process

An important effort in improving RANSAC has been to improve the sampling process. Ordinarily, points are picked randomly from the entire set of correspondences. The selection of which points are chosen is highly relevant to the final outcome of RANSAC. If no *a priori* information is provided on the correspondences, as is the case in the original RANSAC algorithm, points are chosen with equal probability in the sampling process. This presents a worst

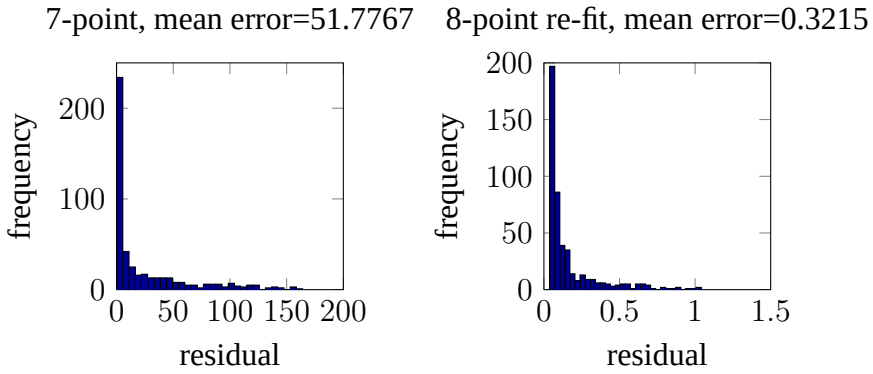


Figure 4.4: Examples of only fitting using the 7-point algorithm to the minimal samples versus refining the hypothesized fundamental matrix by refitting it to the inliers of the 7-point fit. The above shows the residuals of the fundamental matrices fit to the inlier correspondences of two frames in the Merton College sequence using these two methods. The experiment is repeated 100 times and the results of the mean errors over all correspondences are plotted.

case scenario since all points have the same likelihood of being chosen regardless of whether or not they are outliers or inliers. The solution to alleviating this problem is to use non-uniform sampling to give higher priority to points that are more likely to be inliers. The proposed algorithms RES-RANSAC and LEV-RANSAC are detailed in Sections 4.6 and 4.7 also belong to this set of methods that attempt to improve the sampling process. Basically a sampling method that is more likely to draw inliers, and "good" inliers at that, is the faster and more accurate robust estimator. In this context, a good inlier is one that provides a better span of the parameter space so that the model can extrapolate the rest of the data. It is not sufficient to merely find inliers, but "good" inliers that can provide more useful constraints in the parameter space. In other words, inliers are not created equal!

The following sections outline some of the existing methods that attempt to improve the sampling process and therefore the speed of convergence and the quality of the estimated model.

Sampling Based on Residuals

Methods based on residuals or the error terms use the error of each data point with respect to all generated hypotheses as a measure for guiding the sampling process. The error terms can be indicative of an outlier if they are consistently high and inlier if otherwise.

RES-RANSAC [90] is one of the contributed algorithms of this thesis and will be discussed in Section 4.6. This method uses the value of residuals and their distribution to guide the

sampling process.

Multi-GS RANSAC [18] presented, 2 years after RES-RANSAC, offers a sampling method based on the proximity of the residuals in a preprocessing step where residual neighborhoods are formed. This algorithm is capable of discerning multiple structures in the underlying data.

Sampling Based on Geometric Relationship of Data

These set of methods attempt to infer *a priori* data information based on the geometric properties of input data.

NAPSAC [73] is one of these methods. The argument behind NAPSAC is that points that belong to the same manifold cluster together. Therefore sampling is performed by taking an initial point and selecting the remaining points from a hypersphere centered on this initial point. The distances are estimated in the 4D space of correspondences. The algorithm increases the accuracy over traditional RANSAC. However, points that are nearby are often on the same object and form nearly planar structures, often leading to degenerate configurations. Therefore, unless planar structures are scarce in the scene, application of this algorithm leads to a poor estimate of the fundamental matrix.

SCRAMSAC [94] is another algorithm that attempts to improve the samples by using a spatial consistency filtering process. The proposed sampling process results in favoring matches that belong to the same structure. Again, the problem with this approach is the high likelihood in using samples that belong to fronto-parallel degenerate planar configurations.

GroupSAC [75] is another algorithm that attempts to find clusters of data points for sampling, arguing that inliers are more likely to belong to the same color segment. The segments are made using a color segmentation or an optical flow clustering algorithm. Since this method relies on color segmentation and a clustering algorithm for optical flow, the outcome is highly dependent on the performance of these two auxiliary algorithms.

In conclusion, methods that rely on grouping data together and sampling based on various spatial relationships that lead to the formation of these groups are often prone to inadvertently picking degenerate samples on planar scenes. These effects might not be as prominent on scenes devoid of large planar structures, but this disadvantage will reveal itself on sequences containing dominant planes. Also, the assumption that inliers cluster together is often incorrect in cases with large amounts of textures where inliers and outliers can exist side by side. Also, the idea of clustering data into neighborhoods means that a scale or threshold for determining membership to such clusters must exist, which can be difficult to set and is usually set heuristically in the above algorithms.

Sampling Based on Matching Scores

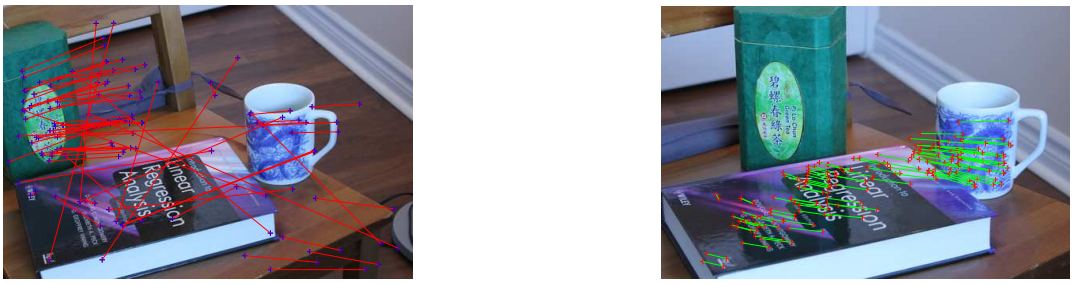
Another set of algorithms that attempt to find *a priori* information on input data are those relying on matching scores. These methods argue that a correspondence with a higher matching score is more likely to be an inlier.

GUIDED-MLESAC [115] is the first method in this category to use a Bayesian approach to infer *a priori* on the sample matches. The authors use the correlation scores to form a distribution over each point on how likely it is to be an inliers. GUIDED-MLESAC uses the zero-normalized cross-correlation value between grey-level patches around the match points in the source and destination images to find a measure of the validity of a correspondence. Letting S_i be the set of matching scores of point i in the left image against all its possible candidates in the right image: $S_j = \{s_{i1}, s_{i2}, ..., s_{ik}\}$ where s_{ik} denotes the correlation score between points i and k. Also, letting v_i be the event that the i-th correspondence is valid and $\bar{v}_i$ otherwise and similarly letting v_{ik} the event that point i correctly matches against point k, the algorithm presents a measure of validity as:

$$p(v_i) \leftarrow p(v_{ik}|s_{i1}, ..., s_{in}) \tag{4.8}$$

where an experimental probability density function is devised that maps the set of correlation scores of a given point to a set of probabilities. Even though this scheme leads to higher accuracy in the results presented in the paper, the method only works when there is only one structure present in the data. In other words, the correlation scores can be used as a measure of the likelihood that a point when there is only one motion present in the scene. For instance, Figure 4.5 shows an example where the scene contains local motion in addition to the egomotion of the camera. It is shown that even though many of the features are correctly match on the object that undergoes a movement between the pairs, these correspondences are still gross outliers with respect to the fundamental matrix. Such matches might have high correlation scores but are in fact outliers.

PROSAC [19] is another method that relies on the matching scores to calculate *a priori* values for the data points. PROSAC orders the correspondences by their similarity score and operates on progressively increasing correspondence sets to generate hypothesis. Although this method leads to higher quality sample sets, it is also only applicable to cases with a single motion is present in the data, and it is prone to degeneracies since points with high correlation scores are often on the same fronto-parallel structures.



(a) Outlying correspondences.

(b) Inlying correspondences.

Figure 4.5: Inliers and outliers for an image pair containing two motions; the egomotion of the camera and the movement of the tea container. The motion vectors of the inliers are shown with green lines in the right image and the outliers are shown with red lines in the left. Many of the features on the tea container have been matched correctly but are considered outliers with respect to the fundamental matrix.

4.4.4 Methods Based on Improved Hypothesis Verification

The runtime of a RANSAC based robust estimator can be expressed as [20]: $t = k(t_m + \bar{m_s}N)$, where k is the number of samples drawn, t_m is the time to instantiate a model from a minimal sample, $\bar{m_s}$ is the average number of models per sample (2.3 for the fundamental matrix) and N is the number of samples. Here the time it takes to verify a single correspondence with respect to a hypothesis is set to the unit amount of time. Figure 4.6 shows the proportion of the time that is spent in RANSAC on verification of the generated hypotheses. Clearly this occupies the largest portion of the RANSAC computation time.

As a result of this several, methods have been proposed to reduce the computation time of RANSAC by improving the hypotheses verification step. Also, since the contributed algorithms, RES-RANSAC and LEV-RANSAC have been devised to address the issues of sampling, they are easily combined with the hypothesis verification acceleration algorithms. In fact, the experimental results show that the fastest relative algorithm amongst the implemented methods is the combination of LEV-RANSAC with the probability ratio test that will be discussed shortly.

Bail-out test RANSAC [15] proposes an exit strategy from the verification process. In other words, when a certain condition is met, the data verification loop prematurely exits, thus saving the time that would have been needed to go through all correspondences. More specifically, whenever the probability of finding more inliers than the best found hypotheses drops below a threshold, the evaluation is terminated, and the hypothesis is discarded as incorrect.

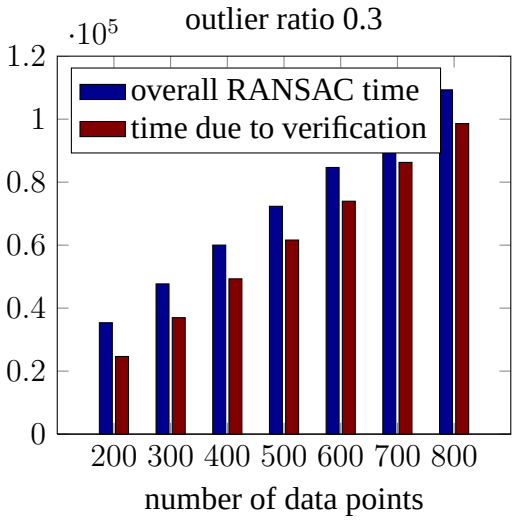


Figure 4.6: Verification time and overall RANSAC computation time when the outlier ratio is 30%.

Preemptive RANSAC [76] is an algorithm designed for real time implementations. Here the best hypothesis is selected from a fixed number of generated hypotheses. In other words, the algorithm is independent of the inlier ratio and returns the final result within a scheduled pre-allotted time. When the fraction of inliers is too low for the predetermined number of hypotheses, the method fails.

Randomized RANSAC with $T_{d,d}$ is another method that uses the $T_{d,d}$ test. This test requires that for a single hypothesis to be verified, it must be consistent with d data points out of d selected data. The optimal value of d is mathematically derived to be one point. Therefore, during the hypothesis verification step, if the first randomly chosen point is not consistent with the model, the model is not verified further. As with other randomized RANSAC algorithms, this means that some good models will also be rejected. However, the algorithm achieves an overall computation time improvement by an early rejection of bad hypotheses.

Randomized RANSAC with Sequential Probability Ratio Test or SPRT [20] is another randomized verification algorithm whereby the speed of RANSAC is improved by finding the optimal number of data points to verify given the current estimate of the inlier ratio. This algorithm uses Wald's decision theory [127] to perform an optimization to find the optimal number of data points to verify based on the current estimate of the inlier ratio. Since this method is combined with the proposed algorithms in order to achieve a lower computational cost as discussed in Section 4.7, it will be briefly expanded upon.

SPRT-based RANSAC essentially reduces the RANSAC computational cost by approach-

ing the verification of hypothesized models as a sequential testing similar to those performed in industrial inspection. The problem is to decide whether a batch is "good" or "bad" by making the smallest number of observations while minimizing the possibility of rejecting a good model or accepting a bad model. This is based on Wald's sequential probability test which calculates a likelihood ratio as:

$$\lambda_i = \prod_{j=1}^{i} \frac{p(z_j|H_b)}{p(z_j|H_g)} = \lambda_{i-1} \times \frac{p(z_j|H_b)}{p(z_j|H_g)} \tag{4.9}$$

where H_b is the hypothesis of having chosen a bad model and H_g is that of a good model. When z_j is consistent with a model it takes the value of 1 and zero otherwise where this consistency is based on the Sampson error being less than a threshold. The probability of a random points being consistent with a good model or $p(1|H_g)$ is simply the inlier ratio and the probability of a data point being consistent with a bad hypothesis is modeled as a Bernoulli distribution whose value is analytically derived. This ratio is sequentially updated during the verification of a hypothesized model and once it exceeds some threshold A the model under consideration is rejected. This value is the only parameter that needs to be optimized based on the latest estimate of the inlier ratio and the desired probability of rejecting a good hypothesis. In fact, the value of A is recalculated based on an analytical formula whenever a new estimate of the inlier ratio is obtained by finding a new model with the highest score.

For a more complete survey of RANSAC algorithms for improving the computation time and real-time performance the interested readers can refer to [87].

4.4.5 Methods Based on an Improved Termination Criterion

Another important question in RANSAC is the problem of the number of iterations, or the termination criterion. The traditional RANSAC algorithm proposes that given a confidence value, η, the number of required samples is $I_{max} = \frac{\log(1-\eta)}{\log(1-p(v)^7)}$ for RANSAC. This formula is a statistical measure of how many iterations it will take before RANSAC takes a minimal sample consisting only of inliers. This is however a very overoptimistic approximation. As observed in several works, including [115], it is not enough to only take inlier samples, but the samples should span the manifold well enough. This means that often the actual required number of iterations required is higher than the one in traditional RANSAC. As a result, some modifications to this termination criterion have been proposed.

Ensemble RANSAC [130] and Preemptive RANSAC [76] uses a predetermined number of iterations for their methods. The advantage of this is that it guarantees a certain amount of

time for the RANSAC procedure, but the number of iterations can become independent of the actual number of inliers. Also, the methods do not extend to higher outlier ratios.

Randomized RANSAC with Sequential Probability Ratio Test or SPRT [20] offers a modified termination criterion. This algorithm modifies the traditional RANSAC's formula in order to incorporate their proposed stochastic hypothesis verification. This is due to the fact that this algorithm does not verify a hypothesis against all data points and instead uses a stochastic method of deciding how many data points to verify against. Therefore, in this case it is possible to have a good sample but reject it due to an error in the randomized verification. As a result the termination criterion has been modified to take this into account.

LEV-RANSAC as described in Section 4.7 offers a termination criterion based on the convergence of the proposed *a priori* information as will be explained.

4.5 Proposed Algorithms

In spite of the large number of modifications offered to the traditional RANSAC, there are still open challenges in the case of robust motion estimation. Some of these are:

- Inadequacy of existing methods in dealing with higher error ratios.
- Challenge of multiple structures in images due to local motion.
- Computation time and the requirement of real-time applications for faster algorithms.
- Accuracy: the impact of poor fundamental matrix estimation on self-calibration and scene reconstruction.
- Degeneracies, their detection and handling.

The following are the two proposed algorithms that aim to improve the accuracy and speed of RANSAC. The first algorithm, RES-RANSAC [90] is a residual-based RANSAC algorithm and the second is LEV-RANSAC [89] which is based on the regression diagnostics. The two algorithms are presented and the experimental results are shown afterwards.

4.6 RES-RANSAC

RES-RANSAC is an extension to the MLESAC algorithm that aims to improve the sampling process. This is carried out by incorporating attributes of the residuals from the hypotheses

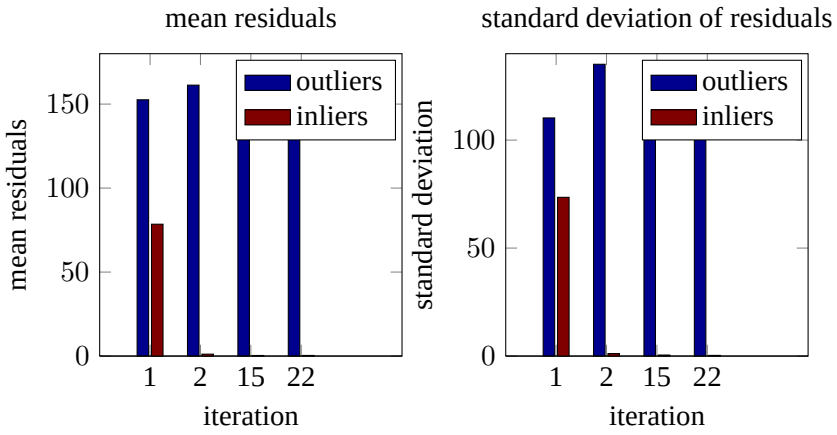
estimations to the minimal samples and using them as prior validity of correspondences and refining these probabilities during the iterations of RANSAC. This algorithm is based on the idea that residuals belonging to inliers often have lower magnitudes than the outlier residuals. Also the inliers tend to cluster in one area and have a lower dispersion. In order to illustrate this fact Figure 4.7 shows two instances of the RANSAC estimation process for two frame pairs from the Wadham And Merton College sequences with different magnitudes of outliers inserted into the set of correspondences. The plots contain the mean of the residuals and their standard deviation for those iterations where a new model is acquired. Since ground truth correspondences are available for these sequences and the outliers are artificially added, it is possible to record the residuals of the actual inliers versus outlier during the iterations of RANSAC. It is shown that the mean of the residuals are indicative of their status as an inlier or an outlier. Also as the plots for the standard deviations of the residuals show, the inliers often have a much lower dispersion and cluster together. This is used to incorporate a histogram of residuals and their magnitude in calculating the data *a priori* measures. These *a priori*s, denoted by $p(v_i)$ are used to improve the sampling by allowing it to pick points that are more likely to be inliers. Experimental results show that at various outlier ratios, the RES-RANSAC algorithm reduces the Sampson error and is also faster (in terms of the number of trials) in comparison with existing algorithms.

4.6.1 Validity of Correspondences

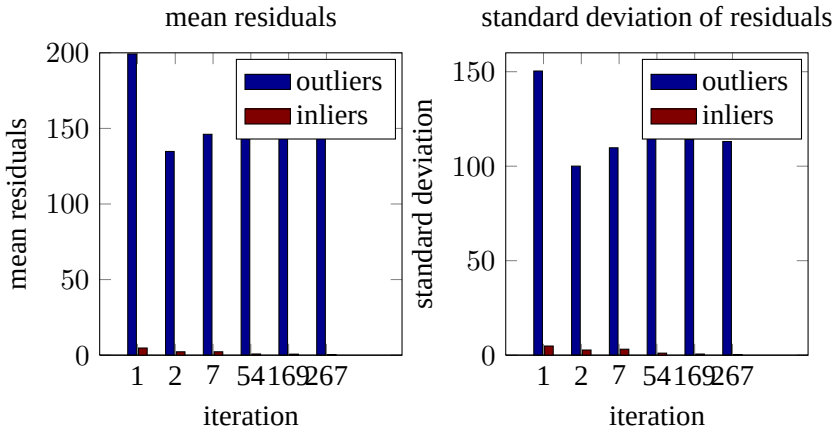
We evaluate the validity of correspondence i using the conditional probability $p(v_i|r_i)$ when the error r_i is observed. Error here being defined as the square root of the Sampson error as explained. We approximate the conditional probability $p(v_i|r_i)$ as:

$$p(v_i|r_i) \propto 1 - p(0 \leq r \leq r_i) \tag{4.10}$$

where $p(0 \leq r \leq r_i)$ is the probability of an error value r, lying in the interval $[0, r_i]$. The larger the conditional probability $p(v_i|r_i)$ is, the more valid the corresponding pair will be. The justification for this formulation can be seen by noting that if a given correspondence i is associated with the manifold currently being estimated, the error r_i will be small as shown in Figure 4.7b. With a small error, the length of the interval $[0, r_i]$ is reduced. This makes the probability of the error being inside the interval very small. Consequently, the conditional probability $p(v_i|r_i)$ will be large. On the other hand, when the error is large, the length of the interval is also large. This causes the probability of the error being inside the interval large, making the conditional probability $p(v_i|r_i)$ small.



(a) Merton College sequence residuals, 30% outlier ratio.



(b) Wadham College sequence residuals, 50% outlier ratio.

Figure 4.7: Mean and standard deviations of the residuals of inliers and outliers for two frame pairs from two sequences with varying levels of artificial outliers in the correspondences. The residuals are those from the iterations of RANSAC where a best model is estimated.

A practical question is how to model the probability $p(0 \leq r \leq r_i)$. A straightforward solution is to employ a Gaussian function to model the probability distribution:

$$p(r) \propto e^{\frac{-r}{2\sigma_r}} \tag{4.11}$$

with σ_r being the standard deviation calculated from the overall set of errors. However, the evaluation of this formula is time-consuming and the probability $p(0 \leq r \leq r_i)$ is sensitive to the variation of the error r_i. To avoid this, we quantize possible values of r into a finite number of intervals with each interval having the same size. In this implementation, we are using 100 bins with every bin being $\frac{\text{median}(r)}{100}$ pixels in size. Errors outside this range will be assigned to the last interval. In order to ensure the correctness of our parameters we verified the feasibility of this approach experimentally against various other bin assignments and found that using this value guarantees the highest likelihood of convergence. Let $Q_{l,c}$ be the midpoint of the interval (Q_l, Q_{l+1}) then the probability distribution is approximated as:

$$p(Q_l < r \leq Q_{l+1}) \propto e^{\frac{-Q_{l,c}}{2\sigma_r}}. \tag{4.12}$$

Also in order to utilize the fact that inliers cluster closely together, a measure of how frequently the residuals occur in each bin of the histogram has been utilized. Let $H(Q_l, Q_{l+1})$, be the proportion of errors r_i in the interval (Q_l, Q_{l+1}), then we can argue that the higher this value, the more likely the points in that interval are to be inliers. Therefore, the probability distribution within the interval (Q_l, Q_{l+1}) is modified as:

$$p(Q_l < r \leq Q_{l+1}) \propto H(Q_l, Q_{l+1}) \times e^{\frac{-Q_{l,c}}{2\sigma_r}}. \tag{4.13}$$

Normalizing $p(Q_l < r \leq Q_{l+1})$ results in:

$$p(Q_l < r \leq Q_{l+1}) = \frac{H(Q_l, Q_{l+1}) \times e^{\frac{-Q_{l,c}}{2\sigma_r}}}{\sum_{l=1}^{100} H(Q_l, Q_{l+1}) \times e^{\frac{-Q_{l,c}}{2\sigma_r}}}. \tag{4.14}$$

Consequently, the probability $p(0 \leq r \leq r_i)$ is represented by:

$$p(0 \leq r \leq r_i) = \sum_{l=0}^{L} p(Q_l < r \leq Q_{l+1}) \tag{4.15}$$

with L determined by $Q_L < r_i \leq Q_{L+1}$. To summarize, the probability of a data point being an inlier is assigned by the residuals of the said data point with respect to the latest hypothesis. In addition, this probability is modeled using a Gaussian combined with a binning algorithm that assigns a higher score to bins with higher number of residuals.

4.6.2 Estimation of the Prior Validity

As described in Section 4.4.4, the prior probability $p(v_i)$ is a measure of the validity of a correspondence before a random sampling trial is performed. As an estimate of the prior validity of a correspondence, we propose that the conditional probability $p(v_i|r_i)$ becomes the prior:

$$p(v_i) \leftarrow p(v_i|r_i). \tag{4.16}$$

Letting $p(v_i^{(j)})$ be the *a priori* of the i-th point at iteration j, we can use the above formulation to assign its value. However, the value of $p(v_i^{(j)})$ is unknown since the $p(r_i^{(j)})$ is derived from the current model at the j-th iteration, and is not known at this stage. Letting M represent a model being estimated, and M^* denote the best model found so far in the iterations before the j-th trial, we can calculate the probability $p(v_i^*|r_i(M^*))$ using the data set associated with the best model M^* that has been attained so far. In our formulation, we use the conditional probability $p(v_i^*|r_i(M^*))$ to approximate the prior validity $p(v_i^{(j)})$ at the j-th trial, i.e. $p(v_i^{(j)}) = p(v_i^*|r_i(M^*))$. This approximation is reasonable when we consider that the model to be estimated is the one that is associated with the dominant data in the complete set of putative correspondences.

After the current model $M^{(j)}$ is derived, it is determined whether or not it is better than the best model found so far. If so, the best model M^* and the conditional probability $p(v_i^*|r_i(M^*))$ will be updated, respectively, by $M^* = M_j$ and $p(v_i^*|r_i(M^*)) = p(v_i^{(j)}|r_i(M^{(j)}))$. Otherwise, the current model $M^{(j)}$ is discarded and no update on the estimates of the conditional probability $p(v_i^*|r_i(M^*))$ will be carried out. It is possible that an early unreliable estimate of the prior probability can make the final estimates of our algorithm diverge from the true model. This is due to the fact that if a decision is made too early about the value of the prior validities, it can trap the algorithm in an incorrect solution. To reduce this possibility, the estimation of the conditional probability $p(v_i^*|r_i(M^*))$ is performed only after at least 10% of the total number of trials have been completed. The feasibility of this approached was verified experimentally.

The prior probabilities are used in a Monte Carlo sampling strategy where points with higher priors are likely to be picked more often. This is similar to the sampling strategy presented in [115]. The experimental set up and the results showing the effectiveness of this strategy are presented in Section 4.8.

4.7 LEV-RANSAC

LEV-RANSAC presents an improvement to RANSAC where regression diagnostics information is incorporated in RANSAC in order to improve accuracy and speed. Similar to RES-RANSAC presented in Section 4.6 this algorithm aims to estimate correspondence *a priori* values in order to improve the sampling process. This improvement in the sampling stage speeds up the estimation by taking advantage of *a priori* information derived from regression diagnostics calculated from the residuals during the iterations of the minimal sampling. Also, a new stopping criterion is presented that enables a much higher speed in the estimation process and guarantees that the iterative estimation is stopped as soon as the best model is found. In addition, LEV-RANSAC can easily be combined with existing randomized hypothesis verification techniques, such as SPRT which was discussed earlier, to achieve an even higher accuracy and speed. The algorithm shows a marked improvement in the accuracy of the estimation and a speed up of more than three times over the traditional RANSAC.

4.7.1 Overview

As argued, there is readily available information from the iterations of the RANSAC process that can be used to estimate *a priori* information that will guide the sampling process. The *a priori*s estimated by LEV-RANSAC are based on a more statistically meaningful measure than pure residuals (i.e., RES-RANSAC) and a more general one than matching scores (i.e., Guided MLESAC). In addition, the method works well independently of the number of motions present in the scene since it does not rely on matching scores.

4.7.2 Regression Diagnostics

In LEV-RANSAC, the values of the *a priori* probabilities are derived from the measure of each correspondence's *influence*. In other words, the amount by which a single correspondence influences the estimation of the fundamental matrix is used as a measure of its probability to be an inlier. Influential points tend to "pull" the regression coefficients in their direction and are often outliers as reviewed in Chapter 3. In order to assess the influence of the correspondences, the value of Cook's distance is adopted in the proposed algorithm since the fundamental matrix is a linear model and the same principles used in standard linear regression apply here. This quantity was explained in Section 3.3.4 as is stated again as:

$$D_i = \frac{d_i}{p} \frac{h_{ii}}{1 - h_{ii}}. \tag{4.17}$$

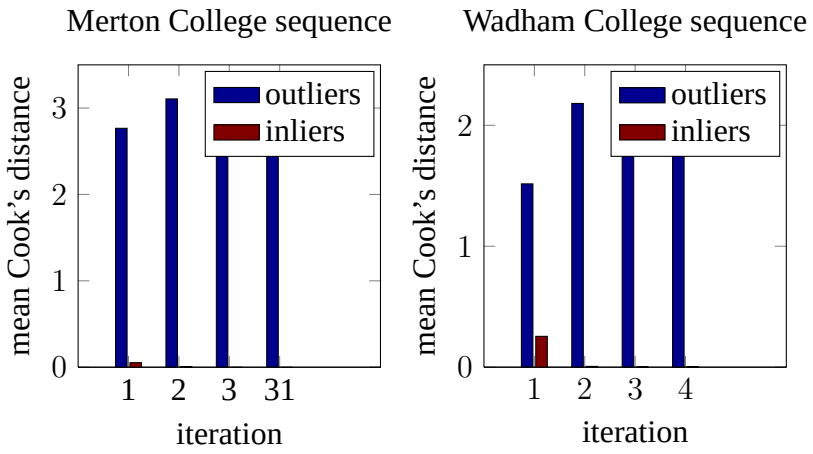


Figure 4.8: Example of the mean Cook distances for outliers versus inliers in the iterations of the proposed algorithm in the Merton College sequence residuals (30% outlier ratio) and Wadham sequence (50% outlier ratio).

where in this case d_i is the studentized residuals of a correspondence with respect to the fundamental matrix and h_{ii} is the leverage of said point. In the proposed algorithm, Cook's distance for all data points is iteratively updated and the *a priori* probabilities are estimated based on this distance. Since Cook's distance is more meaningful than mere residuals, the algorithm is able to discern between inliers and outliers more effectively than the RES-RANSAC algorithm. Figure 4.8 shows an example of the Cook's distance measure for the inliers and outliers for two different frame pairs with artificial outlier noise. The measure of Cook's distance is highly indicative of the inliers and outliers and is a more effective means of detecting them.

4.7.3 Sampling with Regression Information

In order to use Cook's distance as a measure of a probability, a density function has to be defined that maps Cook's distance to a probability value. In this case a probability density function is chosen to be a Gaussian function defined by: $p(D_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(D_i)^2}{2\sigma^2}}$. The mean of the Cook distances is assumed to be zero, and the standard deviation is calculated using the median absolute deviation [64]. In other words, a smooth function of the Cook distance measures is used to denote *a priori* probabilities, and so: $p(v_i) \leftarrow p(D_i)$.

4.7.4 Termination Criterion

Normally the number of iterations that the RANSAC algorithm has to run through is calculated statistically as shown in Eq. 3.19. Here we propose a new stopping criterion where the iterations are stopped when the *a priori* probabilities converge. The convergence is detected when the mean absolute difference between the *a priori* values between two consecutive updates is below a threshold, G. Similar to the traditional method of stopping the RANSAC iterations, the user has to specify this threshold. In other words, the iterations of LEV-RANSAC are terminated when:

$$\sum_{i=0}^{n} |p(v_i^{(j)}) - p(v_i^{(j-1)})| < G \tag{4.18}$$

where $p(v_i^{(j)})$ is the *a priori* probability assigned to i-th data point during the j-th update, and G is set to 0.05 in the experiments. This value was found experimentally; higher threshold values will generally sacrifice accuracy for speed and lower values will achieve a lower error at a higher computational cost. This is true in a probabilistic sense; in other words, a lower threshold will *on average*, produce more accurate results. Note that unlike the RANSAC termination criterion, the proposed criterion does not explicitly consider the probability of choosing an inlier set. Rather, the proposed criterion ensures that no additional iterations are performed once the values for the probabilities come to a point of convergence, since a better sample is highly unlikely to be picked at this point. Also it is unlikely for the probabilities to converge when a good model has not been found yet, since fitting to outlier sets leads to highly variable errors which do not tend to the same values in subsequent iterations. Figure 4.9 shows an example of the *a priori* values for the inliers versus outliers for two frame pairs. The mean of the probabilities tend to be significantly higher for the actual inliers, indicating the improvement in the sampling process.

4.7.5 Overall Algorithm

Algorithm 1 outlines the various stages of the LEV-RANSAC method. Here, j denotes the number of the current iteration, h_{ii} and r_i denote the values of leverage and error residual for the i-th point. $p(v_i)$ denotes the measure of the validity of point i calculated using Cook's distance. $F^{(j)}$ denotes the fundamental matrix that is estimated at iteration j using the minimal sample that is found in this iteration, and F^* denotes the best model found so far. $Q^{(j)}$ denotes the score at iteration j and Q^* denotes the best score found so far. Note that the score used in LEV-RANSAC is the MSAC score [120] due to its robustness and simplicity. Also, the error

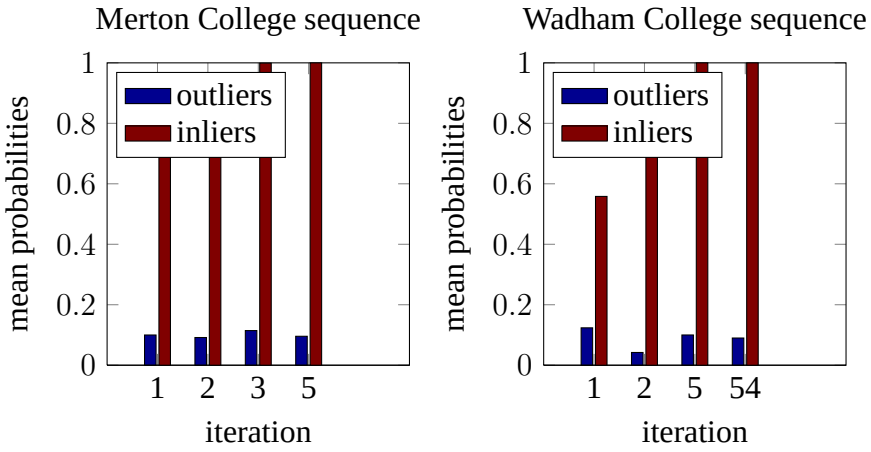


Figure 4.9: Example of the *a priori*s for outliers versus inliers in the iterations of the proposed algorithm in the Merton College sequence residuals (30% outlier ratio) and Wadham sequence (50% outlier ratio).

residual r_i is based on the square root of the Sampson error of correspondence i with respect to the hypothesized fundamental matrix.

The algorithm proceeds by updating the probabilities when a best new sample is found using the new values of leverage and the new residuals calculated from this best model. From this point on, the sampling process is guided based on the new probabilities. The probabilities, $p(v_i)$ are updated only when a new best model is found. When this occurs, the set of inliers with respect to the new model are found based on some threshold, T which is set to be $1.96 * \sigma$ and where σ is often taken to be 1 pixel [119].

Finally the algorithm ends when one of two termination criteria is met. The first termination criterion is the traditional one used in RANSAC and the second is the proposed criterion 4.18. However, almost in all cases, the proposed criterion is met first since the presented guided sampling improves the quality of the samples and so almost always a lower number of iterations is needed than plain RANSAC.

4.8 Experimental Results

In order to test the proposed algorithms, a synthetic correspondence generation framework is devised. For every single test, a random camera pair set up is generated where the two cameras have an arbitrary rotation and translation (non-convergent setups are rejected). Once this setup is created the camera intrinsic parameters are set to some initial values (image size is 512x512,

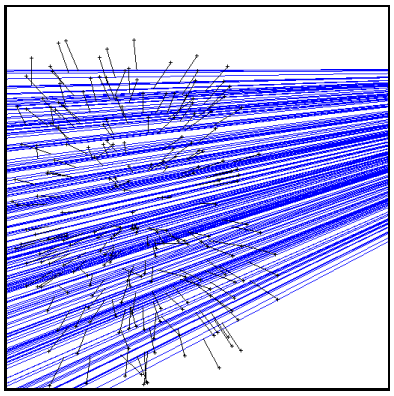
Algorithm 1 Guided Sampling with Regression Diagnostics

Detect image matches
Initialize: $p(v_i) \leftarrow 1$, $j = 0$, $Q^* = 0$
while $j < j_{max}$ **do**
  take minimal sample S_{min} using $p(v_i^{(j)})$
  fit model $F^{(j)}$, find errors, r_i w.r.t. $F^{(j)}$, find score $Q^{(j)}$
  if $Q^{(j)} < Q^*$ **then**
    update: $Q^* = Q^{(j)}$ and $F^* = F^{(j)}$
    form inlier set $S_{in} = \{i | r_j < T\}$
    update leverage of inliers from design matrix: $X(S_{in})$
    find Cook distances D_i and $p(v_i)$ for all the data
    if $\sum_{i=0}^{n} |p(v_i^{(j)}) - p(v_i^{(j-1)})| < G$ **then**
      Terminate and Return F^*
    end if
  end if
end while
return F^*

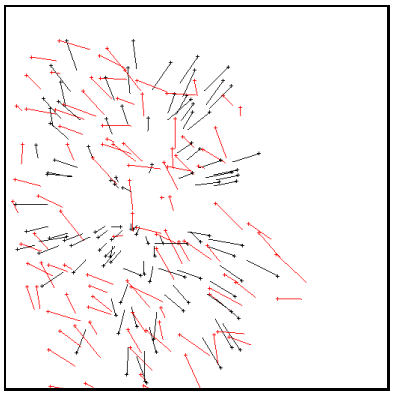
focal length is 700). Then a set of random points in space that are seen by the two cameras are projected into both images and the set of image correspondences are created. This data is then corrupted by two separate noise processes. The locations of the feature matches are affected by a Gaussian noise with a one pixel standard deviation to mimic the process of feature detection and its inherent localization errors. The outliers are also generated by adding an arbitrary value from a uniform distribution to the locations of matches in one image. For any given outlier ratio, 200 such tests are created and the results are averaged over all the tests. Each test configuration consists of 1200 points projected in each of the cameras. Figure 4.10 shows a sample of an instance of a generated synthetic image pair with their correspondences. While the synthetic noise-free matches and their corresponding epipolar lines are shown in Figure 4.10a, Figure 4.10b shows the corrupted correspondences with the outliers marked in red.

The comparisons were made between MSAC, MLESAC, RANSAC, RES-RANSAC and LEV-RANSAC, and plain RANSAC with SPRT and LEV-RANSAC combined with SPRT for accelerated verification. Guided MLESAC was omitted due to absence of correlation values since the tests are synthetic and so no texture information exists.

The first set of tests results, as shown in Figure 4.11a presents the comparison of the mean error averaged over the number of tests for varying levels of noise contamination. In other



(a) Ground truth matches and epipolar lines for a synthetic frame.



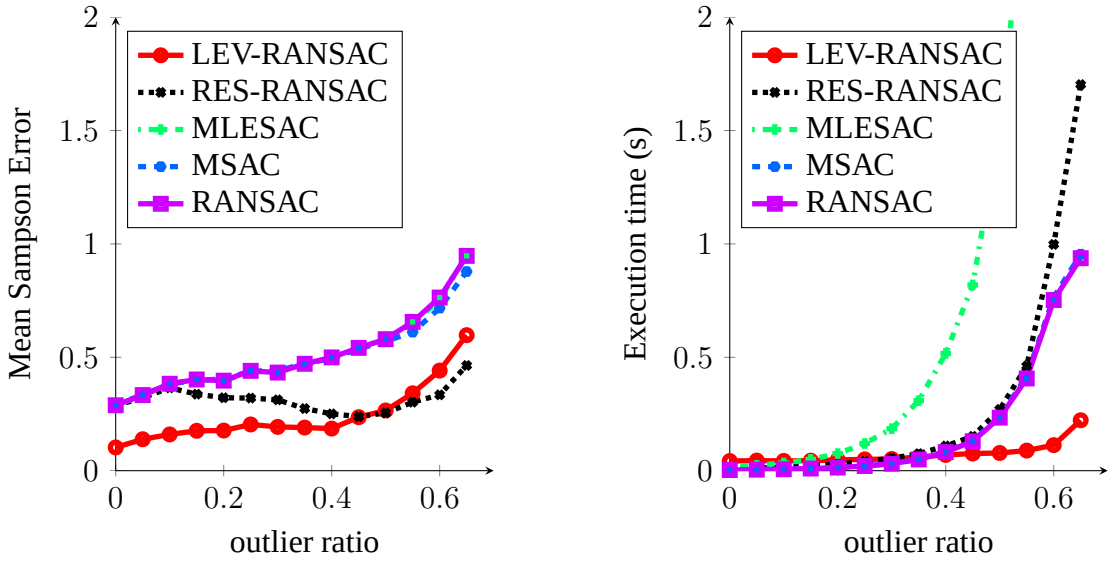
(b) Matches corrupted by noise to form artificial outliers.

Figure 4.10: Synthetic correspondence generation with artificially added outliers and Gaussian noise.

words, for every contamination level, 200 tests are performed. The results of a given test for each algorithm is a final estimation of the fundamental matrix. Therefore, the mean error for every algorithm is defined as the mean error of its hypothesized fundamental matrix with respect to the ground truth matches, averaged over the number of trials. As it is shown, the error of the fundamental matrices found using LEV-RANSAC is significantly lower than other algorithms except for higher outlier ratios compared with RES-RANSAC. This performance drop is negligible when considering the fact that LEV-RANSAC finds this solution in a significantly lesser amount of time.

The second set of test results, as shown in Figure 4.11b, presents the computation time comparison between the tested algorithms. Note that MLESAC has a much higher computation time because of the expectation maximization step [120]. It is clear that the LEV-RANSAC method has a much lower computation time than the competing algorithms. Even though the leverage values need to be calculated, this only happens when a best model is encountered which is only $\log(k)$ times where k is the number of iterations [20]. This additional cost incurred for calculating the leverage is well compensated by the lower number of iterations due to the higher quality samples.

The third and fourth sets of results demonstrate the effectiveness of combining LEV-RANSAC with two existing hypothesis verification methods. Figures 4.12a and 4.12b show the error and time comparison between the $T_{d,d}$ method with LEV-RANSAC combined with $T_{d,d}$. The combination of LEV-RANSAC with randomized verification shows a lower error for all outlier ratios due to the higher quality of the samples. Although the computation time of



(a) Mean Sampson error for the estimated fundamental matrices.

(b) Average computation time taken to estimate the fundamental matrix.

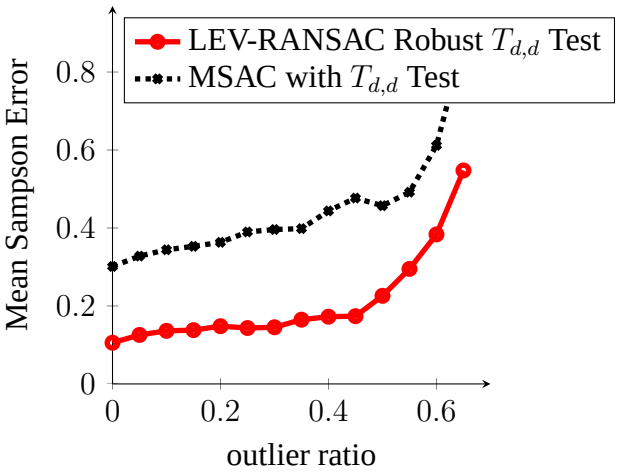
Figure 4.11: Comparison of the performance of the proposed algorithms.

the plain $T_{d,d}$ is better for lower error ratios, the combined method maintains its steady computation time for all outlier ratios, whereas $T_{d,d}$ degrades quickly as the contamination levels increase.

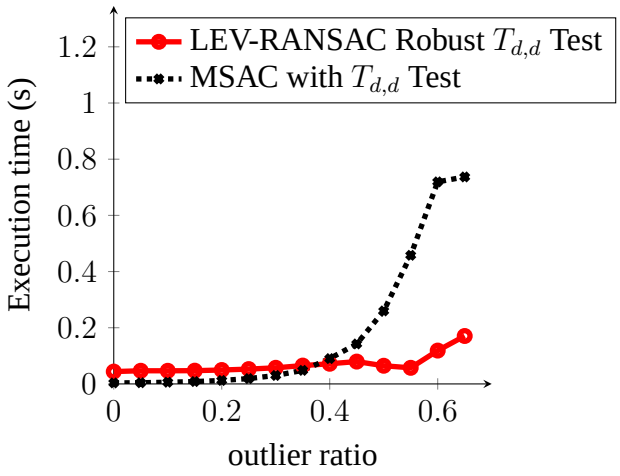
Finally, the last set of experiments, as shown in Figures 4.13a and 4.13b show the accuracy and timing comparisons of SPRT versus LEV-RANSAC with SPRT. Similar to the $T_{d,d}$ case, this combination yields an even faster method than the pure SPRT case. Here again, the combination of SPRT with LEV-RANSAC method has a lower estimation error for all outlier ratios. In the case of the computation time, LEV-RANSAC performs worse than plain SPRT in lower ratios, but the timing remains steady in the higher error ratios, whereas plain SPRT's speed degrades very quickly in these higher error ratios.

4.9 Summary

The robust estimation of multiple view geometry, mainly the fundamental matrix via the RANSAC framework has been discussed in this chapter. Several methods that have built upon the traditional RANSAC algorithm have been reviewed. Even though various improved sampling techniques exist, many rely on information that is not accurate in all cases. Two algo-

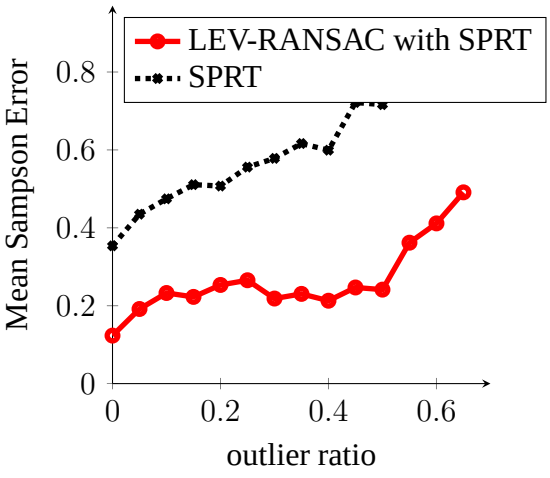


(a) Mean Sampson error.

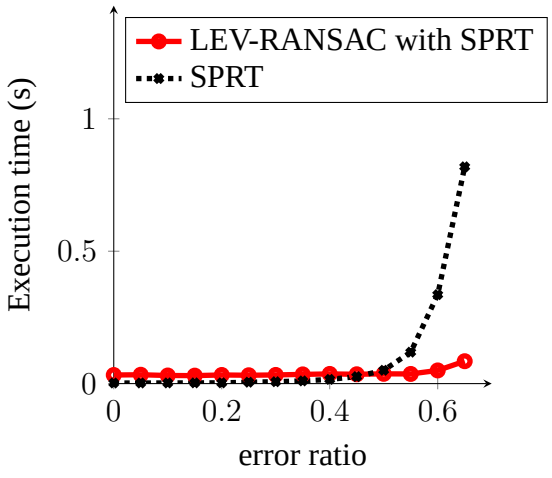


(b) Computation time.

Figure 4.12: Comparison of LEV-RANSAC combined with $T_{d,d}$ test with plain $T_{d,d}$.



(a) Mean Sampson error.



(b) Computation time.

Figure 4.13: Comparison of LEV-RANSAC combined with SPRT with plain SPRT.

rithms have been presented, namely LEV-RANSAC and RES-RANSAC, that do not depend on any domain knowledge such as matching scores or geometric proximity of correspondences to form sampling probabilities. The results show the improved speed and accuracy of the proposed methods. In addition, both methods can be combined with randomized verification algorithms. This has been in fact tested with LEV-RANSAC and the results show the time improvement of this algorithm when combined with randomized verification and speedups of up to three times over RANSAC.

Chapter 5

Robust Self-calibration

5.1 Introduction

Camera self-calibration is the process of estimating the camera parameters using only the information available through an image sequence [46]. It enables 3D reconstructions from a sequence of uncalibrated images without having to rely on a formal calibration process. Such formal calibration processes are often cumbersome as discussed in Section 2.3.2. Also, access to the original acquisition device is not available in many cases such as applications involving Internet image databases. Therefore, the idea of self-calibration, or auto-calibration, is highly attractive for the computer vision community, offering the possibility to infer camera parameters using only the images themselves. Furthermore, accurate and robust self-calibration offers the possibility of a consumer level 3D reconstruction toolkit with no need for any external information other than the images.

There are several widely used categories of self-calibration methods that will be reviewed. The first category uses the projective geometry of a scene and the absolute quadric, as described in Section 2.2.8 to estimate the camera parameters [82]. An initial projective reconstruction has to be available before self-calibration can be achieved; this is a computationally intensive step since bundle adjustment needs to run through the projective reconstruction to ensure an accurate estimation of the parameters.

The second category is based on the Kruppa equations that use an imaginary conic with complex points, namely the absolute conic, as described in Section 2.2.8 [112]. Its performance relies on precise localization of the epipoles, which is not always possible since the locations of the epipoles are highly sensitive to noise.

The third category uses algebraic properties of the essential matrix to provide camera pa-

rameter estimates [128, 70, 36]. However, the methods in this category have to deal with finding a global minimum from a difficult objective function. The methods offered in this thesis belong to the last category, since this set of methods only relies on an accurate estimation of the fundamental matrix. As described in Chapter 4, it is possible to estimate the fundamental matrix accurately in very high-noise situations. Also, this set of methods does not require localization of the epipoles or the computationally expensive projective reconstruction. However, as mentioned, the challenge with this set of methods is the minimization of a difficult objective function that contains many local minima. In addition, the contributed methods address the problem of robustness in self-calibration, defined as immunity to certain inherent degenerate configurations and noisy input data (i.e., fundamental matrices). Robust self-calibration is an idea that is often neglected in the literature but is of great importance, considering the sensitivity of self-calibration to input noise and degenerate and near-degenerate configurations.

To this end, three methods have been developed that will be discussed in this chapter. Before that, the basics of the aforementioned three existing self-calibration methodologies will be reviewed. Due to the nature of self-calibration algorithms, the review of existing methods will be divided into a review of existing constraints that can be used for self-calibration and those methods that can effectively obtain solutions using these constraints. Subsequently, the proposed methods are outlined, and finally the experimental results are presented.

5.2 Existing Constraints

There are several constraints for finding the camera parameters in a self-calibration framework, some of which make assumptions about the scene (e.g., planes [123]) or the camera motion (e.g., pure rotation [42]). However, the most general algorithms depend only on correspondences between the images in a sequence and are independent of special motion or structure in the scene. Four main categories of self-calibration constraints will be discussed in the following section.

5.2.1 Estimation of the Absolute Quadric from Projective Reconstruction

Originally proposed in [122] and [82], these methods perform an upgrade from a projective space to a Euclidean one by using the location of the absolute quadric as described in Section 2.2.8. As reviewed in Chapter 2, the projection of the absolute quadric to the image of the absolute conic can be written as:

$$KK^T = \omega^* = PQ^*_\infty P^T \tag{5.1}$$

where K is the matrix of the intrinsic camera parameters defined as:

$$K = \begin{bmatrix} f_x & s & u_c \\ 0 & f_y & v_c \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

and Q^*_∞ denotes the matrix containing the coefficients of the absolute quadric and P denotes a projection matrix. In order to estimate the camera parameters K, the coefficients of the absolute quadric have to be found first. The method in [82] proposes a minimization of the objective function:

$$\min \sum_{i=1}^{n} \left\| \frac{KK^T}{\|KK^T\|} - \frac{P_i Q^*_\infty P_i^T}{\|P_i Q^*_\infty P_i^T\|} \right\|^2 \tag{5.3}$$

where the subscript $\|.\|$ denotes the Frobenius norm and each P_i is the projection matrix of a camera in a projective space. This objective function is minimized using a nonlinear optimization method. As mentioned previously, a projective reconstruction is required to instantiate the above objective functions and the resolution of the parameters using this constraint requires a good initial estimate. Also, the objective function suffers from numerical instabilities which have been remedied in a newer edition of the algorithm proposed in [85].

5.2.2 Modulus Constraint

Similar to the case of the absolute quadric, the use of the modulus constraint requires an initial projective reconstruction. However, the upgrade from projective to metric reconstruction is now done in two separate steps. This is also referred to as a stratified reconstruction where the plane at infinity $\boldsymbol{\pi_\infty}$ (discussed in Section 2.2.7) has to be found first. This helps upgrade the projective reconstruction to an affine space. Following this, the camera parameters are estimated which helps to upgrade the reconstruction fully to the metric space. This algorithm was initially proposed in [84]. Letting the plane at infinity be defined as: $\boldsymbol{\pi_\infty} = (\mathbf{p}, 1)^T$ and a camera projection matrix defined in a projective space as: $P_i = [A_i|\mathbf{a}_i]$ it is known that:

$$A_i - \mathbf{a}_i \mathbf{p}^T = KR_iK^{-1}. \tag{5.4}$$

where R_i is the relative rotation of the i-th frame with respect to the reference coordinate system. Since KRK^{-1} is conjugate to a rotation, it has a set of known eigenvalues [81]. As

a result we can assume the eigenvalues of the left side of the equation are known and this gives a set of polynomials in terms of the characteristic equation of the $(A_i - \mathbf{a}_i\mathbf{p}^T)$ matrix. Solving these equations using a continuation method gives a set of answers that can be filtered through using additional frames in the sequence. Once the equation for the plane at infinity is resolved the reconstruction can be upgraded to an affine reconstruction. Subsequently, the camera intrinsic parameters are found using a linear method and refined through a nonlinear minimization.

In addition to having the same issues as the method using the absolute quadric, this method depends on localizing the plane at infinity which is very sensitive to noise [44]. Several methods have been devised to incorporate scene constraints in order to improve the localization of the plane at infinity such as the method proposed in [56].

5.2.3 Kruppa Equations

Another group of self-calibration constraints are based on the Kruppa equations [30, 112]. This is one of the earlier self-calibration methods that use the fundamental matrix as input. The Kruppa equations are algebraic representations of the correspondences of epipolar lines tangent to the absolute conic. The complete geometric explanation of this equation is beyond the scope of this thesis, however, a brief description is provided.

The Kruppa equations relate the absolute conic to the fundamental matrix and its epipoles by:

$$[\mathbf{e}']_\times^T \omega^* [\mathbf{e}']_\times = \lambda F \omega^* F^T. \tag{5.5}$$

where λ is an arbitrary scale factor and $\mathbf{e}'$ is the epipole. Methods utilizing this constraint either attempt to explicitly calculate this scale factor [39] or eliminate it altogether [62]. One way of eliminating this scale factor is by using the ratio of the corresponding elements from either side of the above equation leading to six constraints as:

$$\begin{aligned}\frac{([\mathbf{e}']_\times^T \omega^* [\mathbf{e}'])_{11}}{(F\omega^* F^T)_{11}} = \frac{([\mathbf{e}']_\times^T \omega^* [\mathbf{e}'])_{12}}{(F\omega^* F^T)_{12}} = \frac{([\mathbf{e}']_\times^T \omega^* [\mathbf{e}'])_{22}}{(F\omega^* F^T)_{22}} = \\ \frac{([\mathbf{e}']_\times^T \omega^* [\mathbf{e}'])_{13}}{(F\omega^* F^T)_{13}} = \frac{([\mathbf{e}']_\times^T \omega^* [\mathbf{e}'])_{23}}{(F\omega^* F^T)_{23}} = \frac{([\mathbf{e}']_\times^T \omega^* [\mathbf{e}'])_{33}}{(F\omega^* F^T)_{33}}.\end{aligned} \tag{5.6}$$

This leads to six polynomials in the unknowns of the calibration matrix K. It was shown in Section 2.5.4 that dual of the image of the absolute conic ω^* is defined as: $\omega^* = KK^T$ and so solving the above equations leads to solutions in the intrinsic parameters. However, from the above six equations only two are independent, and therefore the task of choosing which

ones to use or how to combine the results together can be a challenging problem on its own [59]. In fact, much research has been carried out on improving the performance and numerical stability of the Kruppa equations. One variant of the Kruppa equations that is designed to improve its reliability is presented in [112] and used later on in this thesis in order to initialize an optimization for one of the proposed algorithms in Section 5.6.

5.2.4 Huang-Faugeras constraint

This constraint is based on properties of the essential matrix [46], and is commonly referred to as the Huang-Faugeras constraint. As discussed previously, the essential matrix can be written in terms of the fundamental matrix as:

$$E = K^T F K = [\mathbf{t}]_\times R. \tag{5.7}$$

Note that the intrinsic parameters of the two cameras that are used to calculate F are assumed to be identical as mentioned in the introduction.

Given Eq. 5.8, one can deduce that the essential matrix has two identical singular values and a third singular value that is zero [46]. The fact that the fundamental matrix has rank two enforces the constraint on the third singular value, but the constraint that the other two singular values are equal is only met when the essential matrix is calculated from the correct camera intrinsic parameters. This in fact provides two constraints over the intrinsic parameters [70]. Therefore, given N fundamental matrices, $F_1, \ldots, F_N$ the objective function over the intrinsic parameters can be formulated as:

$$\min \sum_{i=1}^{N} \left(\frac{w_i}{\sum_{m=1}^{N} w_m} \frac{\sigma_{(1,i)} - \sigma_{(2,i)}}{\sigma_{(2,i)}} \right) \tag{5.8}$$

where $\sigma_{(1,i)}$ is defined as the first singular value of the i-th essential matrix obtained from the i-th fundamental matrix using Eq. 5.8 with the parameterized camera intrinsic parameters K. Similarly $\sigma_{(2,i)}$ is the second singular value of the i-th essential matrix. Also, the weights are denoted by w_i and represent the confidence in the estimation of the fundamental matrix. Alternatively, they can be set according to the number of inliers, or a function of the residuals for the fundamental matrix as detailed in [70]. An equivalent but algebraically different objective function that can be derived from the above formulation was proposed in [36]. This objective function can be written as:

$$\min \| 2EE^T E - \text{tr}(EE^T)E \|_F. \tag{5.9}$$

Using a sufficient number of fundamental matrices, the intrinsic parameters of the camera can be found by minimizing the above objective functions. However, due to various degeneracies and numerical instabilities and noise affecting the fundamental matrices, this is a challenging problem. The following section discusses the issue of robustness in the context of self-calibration and the subsequent sections offer the three contributed methods towards this end.

5.2.5 Trivedi Constraint

Given that the essential matrix E has only six degrees of freedom, $S = EE^T$ also has six degrees of freedom (refer to Section 2.4.3). In fact:

$$S = EE^T = ([\mathbf{t}]_\times R)([\mathbf{t}]_\times R)^T = [\mathbf{t}]_\times [\mathbf{t}]_\times' = \begin{bmatrix} t_2^2 + t_3^2 & -t_1 t_2 & -t_1 t_3 \\ -t_2 t_1 & t_3^2 + t_1^2 & -t_2 t_3 \\ -t_1 t_3 & -t_2 t_3 & t_1^2 + t_2^2 \end{bmatrix}. \tag{5.10}$$

Since the above formulation decimates the rotation matrix, the resulting matrix only has three degrees of freedom belonging to the displacement matrix. As a result, it can be shown that:

$$4S_{ij}^2 - (\text{tr}(S) - 2S_{ii}) * (\text{tr}(S) - 2S_{jj}) = 0 \tag{5.11}$$

where S_{ij} is the ij-th element of matrix S. The above can be used as an objective function to be minimized in search of a set of intrinsic values K, in the same way as the Huang-Faugeras constraint. In fact, the above formulation has been shown to be equivalent to the Huang-Faugeras equations [61].

5.2.6 Special Cameras and Camera Motions

The attempt in this thesis has been to solve for the set of intrinsic parameters of a general image sequence when assuming camera parameters are fixed. However, for specific types of cameras and camera motions, simplified self-calibration can be derived. Two of these methods will be briefly mentioned for completeness.

One such specialized camera motion is that of rotating cameras. This is a commonly occurring situation since pan-tilt cameras are a common imaging device. These methods often rely on the homography relating the images of a scene taken by the camera undergoing rotation [8]. Constraints over the calibration matrix K can be derived from these homographies.

An example of a special camera is that of a semi-calibrated camera. Such a camera is fully calibrated except for an unknown focal length. This can be due to known assumptions over

the camera. For instance, if a camera is known to have aspect ratio of unity and an optical center which coincides with its image center, the only parameter that is left to search for is a single value of the focal length. An example of an algorithm based on this assumption is presented in [54] where the fundamental matrix and the unknown focal length is estimated from the correspondences between a frame pair in a RANSAC framework.

5.3 Existing Robust Methods

Robust self-calibration is an area that has received little attention in the research community. There are very few methods that have proposed ideas that improve the robustness in the self-calibration process. One of these methods relies on the confidence in the fundamental matrix in order to introduce robustness to the underlying process. The algorithm presented in [70] uses a method of weighing the fundamental matrices in the optimization used to find the calibration parameters according to their confidence levels, based on the regression characteristics of the estimation of the fundamental matrices. The proposed weighing scheme uses either the number of inliers used to calculate a fundamental matrix or the mean of its residuals as a measure of confidence. This is an effective robust strategy to obtain a measure of robustness against error in the fundamental matrix caused by outlying matches; however, as will be discussed, there are other sources of error involved in self-calibration which this method does not address. Comparisons have been made against this weighing scheme in the experimental results sections of this chapter where this method is referred to as the Geometric algorithm, referring to the use of geometric error in estimating the weights.

Another attempt at producing robustness in self-calibration was made by the work presented in [59] where the covariance of the fundamental matrix estimates are used to compute weights for each constraint. The covariance of the fundamental matrix is obtained as a byproduct of the estimation of the fundamental matrix as proposed in [25]. The use of the covariance of the fundamental matrix follows the same strategy as the previously mentioned method of using the geometric error, since both algorithms use measures of numerical stability of the fundamental matrix for weighing the fundamental matrices. This method achieves mixed success since the results for higher noise ratios show oscillatory behavior as mentioned by the authors.

The self-calibration method proposed in [128] uniformly samples the parameter space and uses a gradient descent in conjunction with a Genetic Algorithm iteratively. Due to the stochastic nature of this optimization, the algorithm is naturally able to avoid the problem of local minima. However, they ignore the estimation of the optical center. Furthermore, since no prior distribution is assumed over the parameter space, this method suffers from redundant

computations due to over-sampling.

The proposed numerical normalization in [85] aims to improve the robustness of self-calibration with respect to numerical instabilities. This relies on making assumptions about the range of possible values for the intrinsic parameters in order to condition the projection matrices to have a uniform range of values which would lead to better numerical stability.

A method offered in [82] uses a stability analysis method to detect which sequences are suitable for self-calibration. This method aims to find degenerate or near degenerate image sequences with respect to self-calibration. This is carried out by examining the singular values of the Jacobian matrix in the optimization for the image of the absolute conic. When a near-zero singular value is detected the problem is indicated as unstable and the sequence flagged as degenerate.

A preliminary work on the sensitivity of self-calibration to noise was carried out in [45]. The author presents a Least of Median squares (LMedS) approach, briefly explained in Section 4.4.3. This simple statistical approach to robust estimation has a breakdown point at 50% outlier ratio. Also the proposed method is based on the assumption that the motion between cameras is purely rotational. The presented algorithms in this chapter are tested in significantly higher error ratios and apply to general camera motions.

5.4 Robust Self-calibration Strategy

There are multiple sources of error for self-calibration algorithms. Often more than one of these sources of error affects the results of self-calibration. These four noise generation processes are:

- Error in the fundamental matrix.
- Degeneracy or near degeneracy.
- Incorrect assumptions over the intrinsic parameters.
- The inherent difficulties in finding a global minimum in the objective function for self-calibration.

First and foremost, in spite of using robust methods, often a few outliers are used in the final estimation of the fundamental matrix. The results of self-calibration are highly sensitive to this noise. Even though this noise might not visibly change the epipolar geometry, the effects on the results of self calibration are pronounced. Figure 5.1 shows the effects of a single outlier

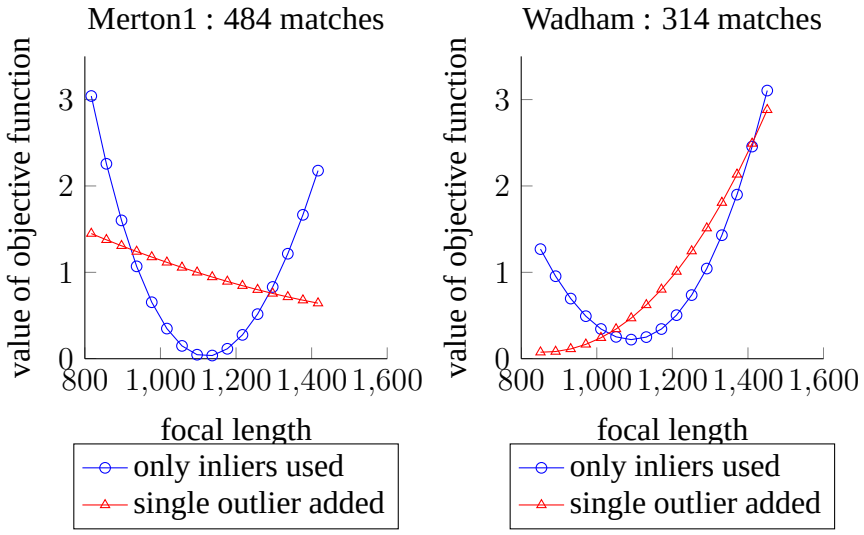


Figure 5.1: Effects of a single outlier on the objective function.

on the results of self-calibration for two different image sequences. These are the Wadham and Merton College sequences obtained from [6]. The number of correspondences are in the order of hundreds as displayed in the diagram, but as it is shown, a single outlier can significantly change the shape of the objective function for the estimation of the focal length. This single outlier has been artificially added to the set of inlying correspondences and this in turn used to calculated the fundamental matrices involved in calculating the objective function. This single outlier is created randomly within the image dimensions. The objective function in this case has been shown in terms of a single variable, namely the focal length, for illustration purposes.

The second issue affecting self-calibration is the presence of degeneracies [111]. Degenerate configuration or critical motions are specific motions of the camera that cause the self-calibration constraint to fail to provide a stable solution. There are several known degeneracies; however a prominent degeneracy that effects all self-calibration methods is the case of pure translation. Under these circumstances it is not possible to use self-calibration techniques to estimate the parameters of the camera. Figure 5.2 shows the effects of degeneracy on the shape of a 1D slice of the self-calibration objective function when only the focal length is varied. The objective functions shown are generated from synthetic image sequences. Each set contains three images where two are parallel and a third is non-parallel. The figure shows the superposition of the objective function of the parallel (degenerate) camera pair and the non-parallel camera pair from the same sequence of three images. It is clear that the objective function is invalid in cases of pure translation. In addition, near-degenerate cases can also cause numerical instability.

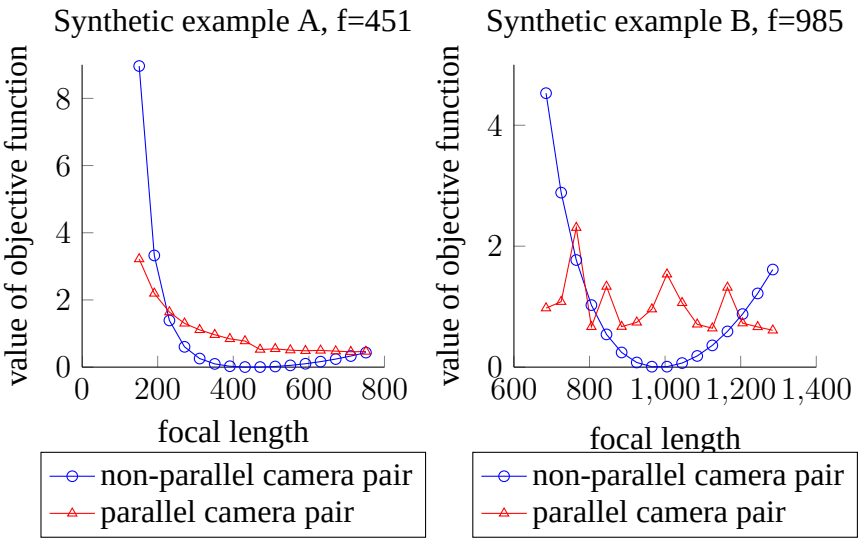


Figure 5.2: Influence of degeneracies on the objective function.

Another source of errors occurs when assumptions on the intrinsic parameters are violated. Oftentimes algorithms in self-calibration ignore the estimation of the optical center, arguing that it is very close to the image center and so it can be assumed predefined. Also, some argue that since the value of the optical center does not influence the reconstruction significantly, it can be ignored [13] (for a review of the definition of the optical center, refer to Section 2.3). While the first assumption is not always true, one thing that is certain is that assuming that the optical center is predetermined can lead to inaccuracies in the estimation of the focal length. Figure 5.3 shows the effects of making an incorrect assumption about the optical center on the objective function. This set of experiments have also been performed on the Wadham and Merton college sequences. In both cases the shape of the objective function based on using the correct set of values for the optical center and the ones based on using an incorrect value for this quantity have been shown. The erroneous objective functions have been estimated when making a 100 pixel error on the location of the optical center. Considering the image widths are 1024 pixels, this constitutes approximately 10% error. This has a very large impact on the location of the minimum of the objective function. As a result, particular attention has been paid in this thesis to the estimation of the optical center.

Incorrect assumptions can also come into play when assuming an aspect ratio of unity or a skew of zero when these assumptions are not valid. Also, when assuming the camera parameters are constant across a sequence when in fact they vary, one can incur further errors. The latter issues have been addresses in this thesis and experimental results will be presented

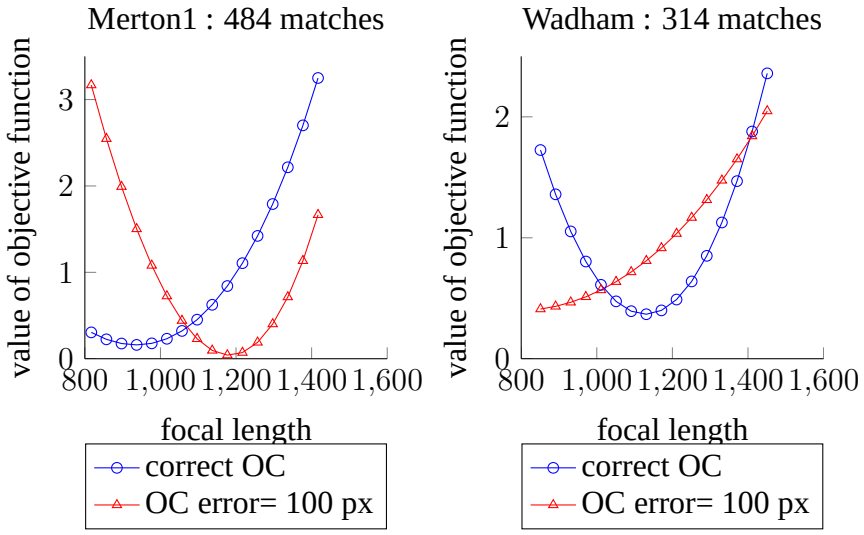


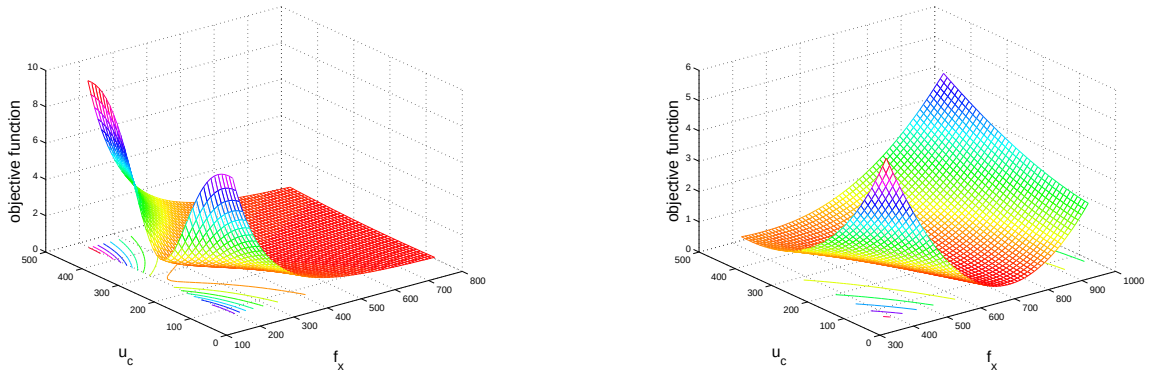
Figure 5.3: Influence of making incorrect assumptions about the optical center (abbreviated to "OC").

which demonstrate the robustness of the proposed methods with respect to this type of error. However, for all experiments it has been assumed that skew is zero and the search for the self-calibration parameters has been performed in a 4D space in this thesis as mentioned in the introduction.

Another challenge in the accurate estimation of the camera parameters through self-calibration is the fact that oftentimes the objective function is difficult to minimize. This is due to local minima and flat regions where the gradient approaches zero and thus causes either a termination in the nonlinear optimizer or a change in direction to an erroneous location on the surface of the objective function. Figure 5.4 shows the objective function of two different image sequences, namely the Corridor and the Valbonne sequences. Here, only focal length in the x direction, f_x and the x-axis component of the optical center, u_c are varied and the other three parameters of the intrinsics matrix are set to their ground truth value. It is difficult to observe the local minima from this 3D slice of the 5D objective function; however it is clear that the global minimum is difficult to locate since a clear basin is absent in both cases.

5.5 Proposed Algorithms

Various problems involving self-calibration have been addressed in the previous section. As a result of these issues, it is clear that a naive approach to self-calibration without consider-



(a) Corridor sequence.

(b) Valbonne sequence.

Figure 5.4: Objective function for focal length and x-axis of optical center.

ation for the these sources of error is bound to produce erroneous results. As a result, three algorithms have been proposed to alleviate the issues involved with self-calibration. The next sections provides a brief overview of the adopted self-calibration strategy, the proposed algorithms and following this the experimental framework used to assess the proposed algorithms is outlined. Subsequently, the proposed algorithms are presented in their full detail.

5.5.1 Self-calibration Strategy

The proposed methods in this chapter aim to improve the robustness of self-calibration from fundamental matrices of an image sequence whose frames have constant, or near-constant intrinsic calibration parameters. Self-calibration has been defined as estimating this set of intrinsic values which contain all parameters except skew and without using any scene geometric constraints, assuming specific camera motions or known points in the scene. As explained, there are different categories of constraints for self-calibration of a general image sequence. These can be mainly categorized into the algorithms that use fundamental matrices, and those that use an initial projective reconstruction. The focus on the proposed thesis has been that of self-calibration from the fundamental matrix due to several reasons. First and foremost, since the goal of this thesis is to present a generic self-calibration algorithm which is independent of a particular application it is desirable to avoid having to perform a projective reconstruction if possible. Also, using a fundamental matrix-based self-calibration approach is more general since it is not always possible to create an initial projective transformation in cases where it is hard to relate all the images into a single projective frame [81]. Moreover, it is shown in [79, 77, 78] that projective reconstruction suffers from more than merely the projective ambi-

guity. In fact, the error surface for the Euclidean reconstruction has better numerical properties than the projective one. As a result, it has been the aim of this thesis to provide a framework for obtaining camera calibration first and then carrying out a Euclidean reconstruction.

Within the self-calibration techniques relying on the fundamental matrix, we have chosen to utilize the properties of the essential matrix presented in Section 5.2.4 rather than the Kruppa equations. There are several reasons that make the Kruppa equations less suitable for devising a robust method. These issues are:

1. The known difficulty in estimating the scale factors in the Kruppa equations [39].
2. Difficulty in choosing which of the six dependent equations per fundamental matrix to use [59].
3. Additional degenerate configurations than other self-calibration methods [110].
4. The sensitivity of the solutions to the positions of the epipoles (often numerically unstable) [44].

The last item mentioned renders the results of the Kruppa equations particularly sensitive to any inaccuracy in the estimation of the fundamental matrix. To illustrate this point Figure 5.4 shows the performance degradation of both self-calibration constraints when the fundamental matrix is perturbed with a noise matrix with variance of $10^{-6}\|F\|$ where $\|F\|$ is the Frobenius norm of the fundamental matrix. The figure shows the one dimensional objective function for varying focal length where all other parameters are set to their ground truth value after and before noise is introduced. As shown in the diagram, the Kruppa equations are affected by this noise to a much higher degree than the equations based on the Huang-Faugeras constraint. As a result, the focus of the proposed thesis has been on incorporating robustness into this particular self-calibration constraint.

5.5.2 Overview of Proposed Algorithms

The first proposed algorithm for robust self-calibration based on the Huang-Faugeras constraint is the Randomized Multi-Start optimization that uses a simplified Kruppa constraint to initialize the optimization process and then performs a guided search in the parameter space. This algorithm has the ability to overcome the difficulties of finding the global minimum when the objective function is a difficult one to minimize in addition to performing a random selection of the fundamental matrices which leads to a reasonable degree of robustness with respect to outliers. The second algorithm is based on perturbation theory and uses a weighted nonlinear

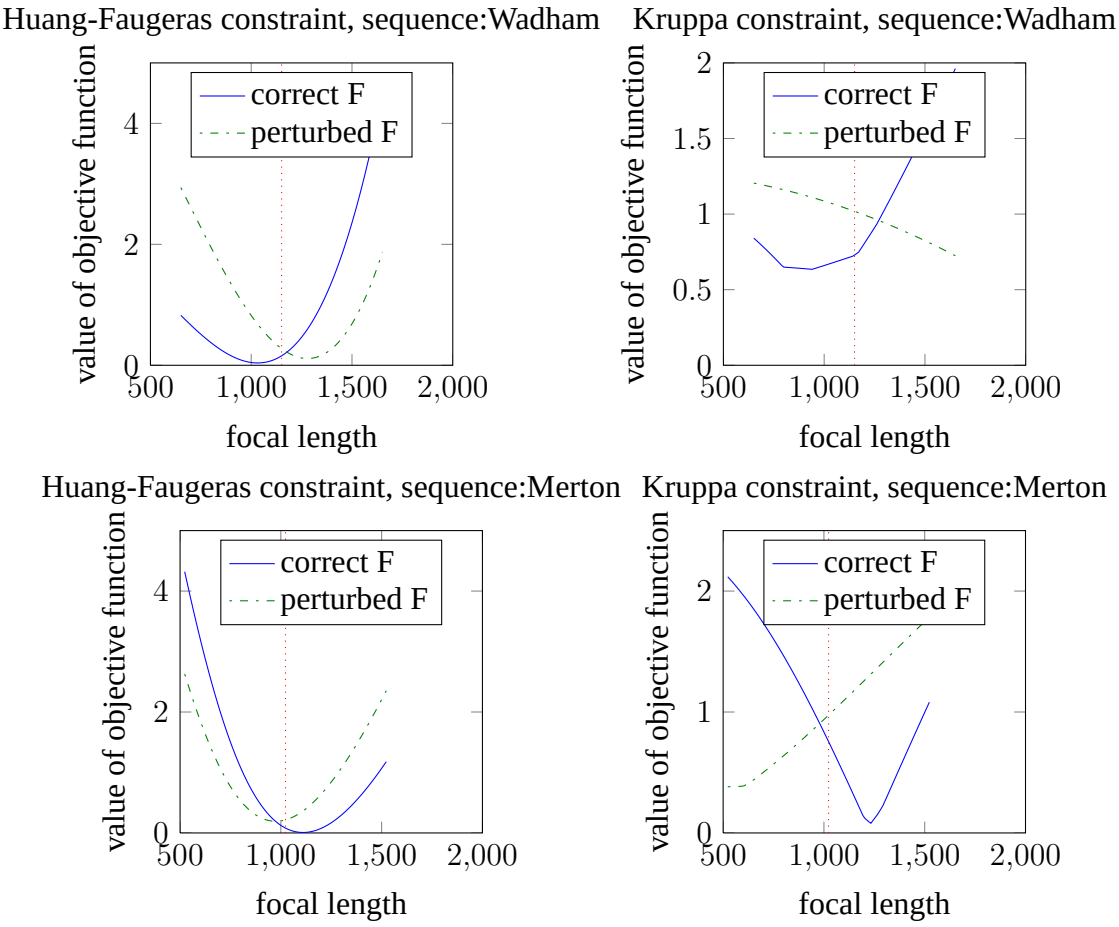


Figure 5.5: Performance degradation of self-calibration constraints in the presence of noise for the Kruppa equations and the Huang-Faugeras constraints. The correct value of the focal length is indicate by the dashed red vertical lines.

optimization framework, where the weights are iteratively refined based on a measure of the expected error of each fundamental matrix with respect to self-calibration as determined by results from perturbation theory. This algorithm is simply referred to as "Perturbation" in the experimental comparisons. Finally an algorithm is proposed based on the localization of the convergence of a set of roots of bivariate polynomials. The set of roots (or zero curves) are used in a geometric optimization framework to find the hypothetical point of convergence. This algorithm is referred to as the Zero Curves algorithm. At the core of the proposed algorithm is the Huang-Faugeras constraint based on the fundamental matrix.

The algorithms have been applied to the problem of retrieving the constant intrinsic parameters of an image sequence. The methods extend just as well to varying parameters; however in such cases more images are needed to meet the required constraints. Also in all subsequent performance comparisons, the full set of parameters belonging to the camera intrinsic parameters have been retrieved except the skew factor which can be safely assumed to be zero in modern cameras.

5.5.3 Experimental Framework

In order to assess the performance of the proposed algorithms, they have been tested in two different segments. In the first set of tests, each individual algorithm is compared with two benchmarks in order to show its merit on its own. This is carried out in the final section of each individual proposed algorithm and it involves assessing the robustness of the algorithm being discussed under a synthetic test framework. In the second stage of testing presented in a final Section 5.9 of this chapter, all the algorithms have been compared to one another in a more comprehensive framework. This includes evaluating the performance of the proposed algorithms with respect to varying levels of matching outliers, varying numbers of corrupted fundamental matrices, increasing variance in Gaussian noise, timing comparisons and scalability assessment. These have been performed on real and synthetic image sequences and will be fully explained in Chapter 5.9.

However, the synthetic results provided within the section of each algorithm will be explained here in order to avoid having to duplicate this information while presenting the proposed algorithms. In the individual comparisons sections, in order to show the effectiveness of each proposed algorithm two types of synthetic experiments have been devised. In each set, synthetic data has been created consisting of five cameras, positioned in a random arrangement in space. The cameras all start out with the same orientation and position, and then their locations are perturbed with a random amount of change within a pre-determined range. The

non-convergent geometries are rejected. Each camera setup has a different set of intrinsic parameters. In other words, each camera has different locations for its optical center, different focal length and the aspect ratios have been randomly selected in a wide range between 0.8 and 1.2. Subsequently, 200 points are randomly created in 3D space and projected to all cameras. Those points that are visible in all cameras are kept, otherwise a new point is created. Following this, fundamental matrices are fit to the point matches between these cameras. The input fundamental matrices used in the self-calibration algorithms have all been estimated using the LEV-RANSAC method proposed in Section 4.7. This is a similar synthetic testing framework as used in Section 4.8 for testing the proposed robust fundamental matrix estimation strategies. However, in the current context, the framework consists of a series of images and their feature tracks, rather than only a pair of cameras. Also, in order to create outliers, two different methods are utilized leading to two sets of synthetic experiments. These two synthetic experiments differ in the way outliers are created. In the first set of experiments, outlying fundamental matrices are created by adding outlying matches to the point correspondences used in calculating the fundamental matrices. In this case, two fundamental matrices (out of a total of ten) have been corrupted by varying degrees of outliers. The resulting fundamental matrices are then used as input to the self-calibration algorithms. This effectively tests the robustness of the proposed algorithms with respect to matching errors which lead to poor estimation of the fundamental matrix. The second set of synthetic experimental results also consist of measuring the performance of the algorithms with respect to outlying fundamental matrices. However, in this experiment outlying fundamental matrices are created via creating a deviation from the assumption that the intrinsic parameters, K are constant across all the frames. These experiments are also carried out using synthetic data containing five frames where one frame has a different focal length than the others. Therefore, four fundamental matrices in the set of the total of ten fundamental matrices will produce erroneous results when a self-calibration method assumes the camera parameters are constant. Such outlying fundamental matrices are clearly inliers with respect to their feature correspondences and yet act as outliers with respect to self-calibration with constant parameters assumption. This scenario often arise when a sequence is obtained from a zooming camera or when images in a sequence are obtained with multiple acquisition devices.

Each experiment for a given outlier magnitude is repeated 100 times for each outlier level and the performance is averaged over all these experiments. The error reported is the error in the estimated focal length in terms of the percentage error over the ground truth focal length. The estimation of the optical center has been ignored until the experimental results Section 5.9. This error value has also been reported as a percentage over the correct value for the

optical center. Since both the focal length and the optical center have two components, in x and y directions, the error reported for both quantities is the average percentage error in both components. In other words:

$$e_{focal} = \frac{1}{2}(|\frac{\widetilde{f}_x - f_x}{f_x}| + |\frac{\widetilde{f}_y - f_y}{f_y}|) \times 100$$
$$e_{OC} = \frac{1}{2}(|\frac{\widetilde{u_c} - u_c}{u_c}| + |\frac{\widetilde{v_c} - v_c}{v_c}|) \times 100$$

where e_{focal} is the reported focal length error and e_{OC} is the optical center error. $\widetilde{f}_x$ and $\widetilde{f}_y$ are the estimated values for the focal length in the x and y directions and f_x and f_y are the true values for these quantities. Similarly, $\widetilde{u_c}$ and $\widetilde{v_c}$ are the estimated values for the optical center in the x and y directions and u_c and v_c are the true values for these quantities.

In each individual section, three algorithms are compared using the above method. These consist of the robust method under consideration, a non-robust algorithm which uses identical weights for all fundamental matrices, and the method proposed in [70] which uses the geometric error residuals of the fit for the fundamental matrices as a measure of the reliability of the fundamental matrix, presented under the Geometric algorithm as mentioned in the introduction.

The following sections present the details of the proposed robust self-calibration algorithms accompanied by the synthetic experiments as outlined above. Following this, Section 5.9 presents a more detailed comparison of all the proposed algorithms with other algorithms which includes real image data and analysis of the results.

5.6 Randomized Multi-Start Optimization

This method presents a random sampling framework for camera self-calibration with modeling of the camera intrinsic parameter space. The focal length is modeled using a Gaussian distribution derived from the results of the Kruppa equations, while the optical center is modeled based on the assumption that the optical center is close to the image center but deviates from it due to some manufacturing imprecision. This model enables us to narrow the search range of the parameter space and therefore reduce the computational cost. In addition, a random sampling strategy is utilized in order to avoid local optima, where the samples are drawn according to this model. The algorithm involves randomly selecting a minimal number of fundamental matrices as inputs for the optimization in order to guarantee outlying fundamental matrices do not affect the final results. In brief, the proposed algorithm is a randomized multi-start optimization algorithm where the input data to the optimization is randomized and also

the parameter space is sampled via a calculated distribution. The result is a robust algorithm which is able to find a near-global minimum and is also able to perform accurately in spite of outlying fundamental matrices. The optimization framework is based on minimizing the difference between the singular values of the essential matrix, or the Huang-Faugeras constraint [46]. The Levenberg-Marquardt optimization routine is used in the core of the algorithm to produce candidate solutions based on the sampled initial starting points and using the selected minimal sets of fundamental matrices. The proposed sampling method shares similarities with the random sampling method in [128] where the search space is randomly sampled in a Hill Climbing algorithm based on Genetic Algorithms. However, the proposed strategy estimates distributions over the parameter space in order to avoid oversampling the large parameter space by emphasizing the more likely regions of this space. In addition, fundamental matrices are also sampled as well as points in the parameter space. In summary, the proposed methodology is able to robustly obtain accurate solutions by avoiding local minima and by reducing the effects of outlying fundamental matrices. Experimental results show the performance of the algorithm compared with two benchmark methods (i.e., the Geometric and the non-robust algorithms).

5.6.1 Modeling of Focal Length

We use a Gaussian distribution to model the search range of camera focal length. The parameters of this Gaussian distribution are determined from the solution a linear variation of the Kruppa equations that only aims to estimate the focal length as proposed in [112]. The Kruppa equations provide a rough approximation for the focal length of each frame. These results are often close to the true solution but not sufficiently accurate. From the results of the Kruppa equations, we determine the mean and variance of this Gaussian distribution as:

$$\mu_{\text{focal}} = \sum_{i=1}^{N} w_i f_{\text{(i,Kruppa)}} \tag{5.12}$$

$$\sigma^2_{\text{focal}} = \sum_{i=1}^{N} w_i (f_{\text{(i,Kruppa)}} - \mu_{\text{focal}})^2 \tag{5.13}$$

where $f_{\text{(i,Kruppa)}}$ is the focal length estimate as calculated from the i-th fundamental matrix in the sequence according to the Kruppa equations, μ_{focal} is the weighted mean of the focal length values and N represents the number of fundamental matrices in the sequence. The weights, w_i can be set according to the confidence in the estimation of the given fundamental matrix as in

[70] and are assumed to have been normalized (i.e., $\sum_{i=1}^{N} w_i = 1$). In our experiments, we set these weights equal to $\frac{1}{N}$ since all the fundamental matrices were estimated from the ground truth data provided with the image sets.

5.6.2 Modeling of the Optical Center

The model for the optical center is also chosen as a Gaussian distribution centered on the image center. This strategy was adopted by noting that the optical center is usually close to the image center but deviates from it due to some manufacturing imprecision that can be modeled as a Gaussian distribution. In the experiments we model the x and y components of the optical center as individual Gaussian distributions whose mean is the center of the image and whose 3σ point coincides with the limits of the width and the height. For example, by setting the mean of the distribution of the optical center in the x direction of the image at the center of the image, and the 3σ to be half of the image width, we ensure that more than 99% of the points generated by this model fall inside the image but are mostly centered around the image center. This leads to a standard deviation of $\frac{width}{6}$ and $\frac{height}{6}$ for the distributions of the optical center in the x and y directions.

5.6.3 Sampling of the Fundamental Matrices

Since the goal of this chapter is the robust estimation of the camera via self-calibration, outlying fundamental matrices must be considered. As explained in Section 5.5, various sources of noise can affect a fundamental matrix with respect to self-calibration. In order to obtain results that are accurate, we adopt a similar strategy to RANSAC for selecting fundamental matrices to be used in self-calibration. In other words, a random minimal set of fundamental matrices is used at every iteration of the algorithm. One important consideration is the number of required fundamental matrices in a minimal sample. Depending on the number of parameters that are being searched for, different numbers of fundamental matrices are required in a minimal sample.

It was shown in [82] that given N frames, assuming constant intrinsic parameters, we can estimate the number of frames required according to:

$$N \times (5 - k) + (N - 1) \times k \geq 8 \tag{5.14}$$

where k is the number of parameters that are searched for. According to this formula, when more than two parameters are assumed unknown in the intrinsic parameters of the sequence

at least three frames are required. Since we are looking for four parameters in the intrinsics matrix, the value of N is set to three in the proposed algorithm.

The stoping criterion of the proposed algorithm is also based on this number of minimal samples. In other words, the variable "MaxIter" which represents the maximum number of iterations, as shown in the algorithm outline 2 is set to ensure every possible combination of minimal sets is chosen. Unlike RANSAC, the number of inliers in the above random sampling technique cannot be determined since the errors do not correspond to physical quantities (as is the case with fundamental matrix errors). As a result, an exhaustive search strategy is chosen that ensures every possible minimal sample is picked. The value of MaxIter is set according to: MaxIter $= \frac{\log(1-0.95)}{\log(1-\frac{1}{\binom{N}{3}})}$ which is the probability of picking every arrangement of three fundamental matrices at least once with a probability of 0.95. This formula is similar to the RANSAC equation Eq. 3.19 where the desired outcome is adjusted to picking all possible minimal sets rather than picking a set of all inliers.

5.6.4 Complete Random Sampling Framework

In order to guarantee that we have not chosen a local minimum, we perform the nonlinear optimization combined with a guided sampling technique. In addition, the fundamental matrices that are used for each sample are randomly selected. In other words, the optimization is run multiple times where each instance of the optimization is initiated with a sampled point in the parameter space according to the distributions over these parameters as defined previously and with a different minimal set of fundamental matrices. The final result is chosen as the solution with the lowest error. In order to guarantee that the scores have similar scale, the fundamental matrices are normalized in a preprocessing stage. This involves setting the Frobenius norm of each fundamental matrix to unity.

Algorithm 2 details the summary of the proposed random sampling nonlinear optimization framework. The lines 1-3 show the distributions defined over each of the variables in the parameter space. Initial points for the focal length parameter used in the nonlinear optimization will be chosen from the distribution $p(f)$ where μ_{focal} and σ_{focal} are defined in Eq. 5.13, and initial points for the camera center are chosen from $p(c_x)$ and $p(c_y)$ where the distributions have been previously estimated as mentioned. "Residual(s)" returns the error of a given solution, s. "RandomSelection" performs the random selection of the initial solution s_{init} according to the models of parameter space. Inside the loop of the iterative sampling, random minimal sets of fundamental matrices are used as denoted by $S_{minsample}$. "Optimize$(s_{init}, S_{minsample})$" shows the optimization step from starting point s_{init} and using an objective function over the

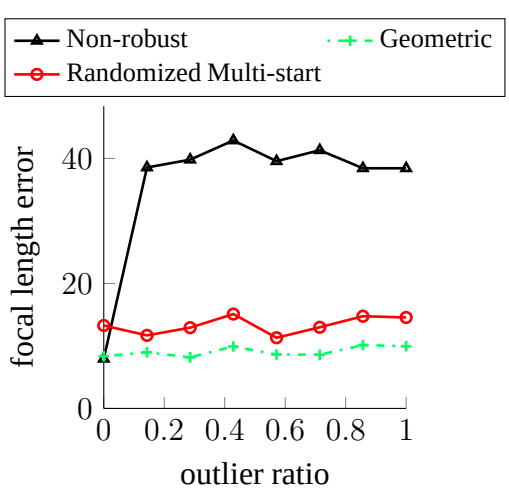
Algorithm 2 Random Sampling Least Squares Self-calibration

$p(f) = N(\mu_{\text{focal}}, \sigma^2_{\text{focal}})$
$p(c_x) = N(\frac{w}{2}, \frac{w}{6})$
$p(c_y) = N(\frac{h}{2}, \frac{h}{6})$
while numSample $<$ MaxIter **do**
  $s_{init} =$ RandomSelection$[f_0 \leftarrow p(f), x_0 \leftarrow p(c_x), y_0 \leftarrow p(c_y)]$
  Select random minimal set $S_{minsample}$
  $s_{numSample} =$ Optimize$(s_{init}, S_{minsample})$
  if Residual$(s_{numSample}) <$ minResidual **then**
    $s_{Best} = s_{numSample}$
    minResidual $=$ Residual$(s_{numSample})$
  end if
  numSample $=$ numSample $+ 1$
end while
set $s_{optimal} = s_{Best}$

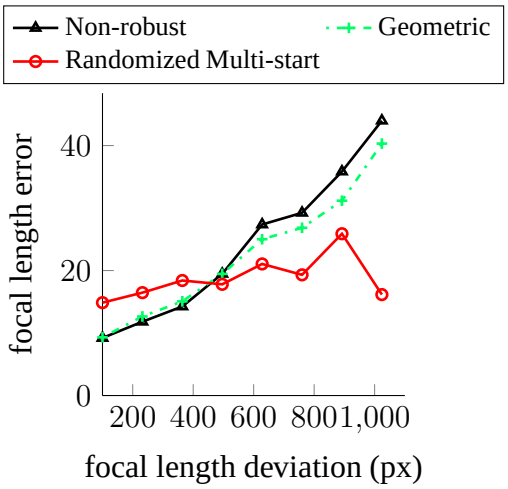
fundamental matrices in the random minimal set denoted by $S_{minsample}$. The "MaxIter" variable which determined the stopping criterion is set according to the proposed formula and the number of unknown parameters. After the maximum number of random samplings has been reached, the optimization stops and sets the kept solution, s_{Best} as the final optimal estimates $s_{optimal}$.

5.6.5 Experiments

To illustrate the robustness and the accuracy of the proposed self-calibration technique, the two synthetic tests outlined in Section 5.5.3 have been presented in Figure 5.6. In the first set of experiments shown in Figure 5.6a, the proposed Randomized Multi-start optimization method has been compared to the two benchmark algorithms. The input to the self-calibration algorithms consist of ten fundamental matrices, two of which have been contaminated by increasing levels of matching outliers. The proposed method maintains a reasonable performance for all levels of matching error. The non-robust method's performance degrades quickly as soon as the error ratio goes above zero. The Geometric method performs best when the outlying fundamental matrix is affected by matching error, since it uses the epipolar geometric error as the inverse weighting factor in self-calibration. In the second set of experiments shown in Figure 5.6b, the error is due to having one frame whose focal length is different from the rest



(a) Performance vs varying levels of matching outliers.



(b) Performance vs focal length deviation.

Figure 5.6: Synthetic performance evaluation of the Randomized Multi-start method compared with two standard self-calibration techniques.

and so the Geometric algorithm's performance degrades in a similar fashion to the non-robust algorithm. However, the Multi-start Optimization method maintains a roughly 18% error in the estimation of the focal length. The reason why the proposed algorithm performs worse than the competitors in a lower noise ratio is due to the fact that only minimal numbers of fundamental matrices are used in the optimization and so not all available constraints are taken advantage of. Unlike the application of RANSAC to the estimation of the fundamental matrix, as detailed in Chapter 4, where a final estimation is carried out using the best inliers in the iterations of RANSAC, no final estimation using an all inliers set is carried out in the proposed algorithm. This is due to the fact that unlike the fitting of epipolar geometry, the error in self-calibration does not have a physical meaning and so a threshold cannot be determined for discerning inliers from outliers. Therefore, the optimization is only carried out with minimal sets of fundamental matrices, which is not optimal when outliers are not present. The next sections outline two algorithms that are able to maintain a steady performance advantage over non-robust methods in low noise situation in addition to being robust against outliers.

5.7 Self-calibration using Perturbation Theory

As outlined in the previous sections, the task of self-calibration is fraught with multiple sources of error. As a result, performing an optimization using all fundamental matrices in a sequence often leads to erroneous results as shown in Figure 5.1. Therefore, an optimal approach to self-calibration should take into consideration the underlying errors affecting the fundamental matrices. Existing methods such as ones proposed in [70], that use the geometric error or the number of inliers are not adequate for coping with all sources of error in self-calibration, as be demonstrated.

The algorithm proposed in this section uses ideas from perturbation theory as applied to the singular value decomposition to create a robust weighted self-calibration optimization method. The proposed algorithm is an iterative estimation method that indirectly estimates the noise in each fundamental matrix while refining the estimate of the intrinsic parameters, K. The estimated magnitude of the noise affecting each fundamental matrix is used to inversely weigh a given fundamental matrix in the subsequent iterations within a weighted nonlinear minimization framework. This effectively reduces the impact of those fundamental matrices that are corrupted by noise while increasing the weights of the fundamental matrices with lower levels of noise. Noise is defined previously in Section 5.5 and can be caused by erroneous matches used in calculating a fundamental matrix, self-calibration degeneracies or violations in assumptions on the intrinsic parameters. The iterations of the proposed algorithm consist of the estimation of the weights, followed by a nonlinear weighted regression using the most recent estimate of the weights.

The proposed algorithm proceeds by first defining an error expansion for the self-calibration objective function. This error expansion will contain terms for the error in the intrinsic parameters and error in the fundamental matrix. We then derive an alternate error expansion which focuses on the noise in the intrinsic parameters rather than the noise in the fundamental matrix part of the equation. Subsequently, an approximate value for the error bound for the objective function is derived based on perturbation theory of singular values to account for errors in the estimation of the intrinsic parameters. Once this is estimated, we can find how much of the error residual in the self-calibration optimization with respect to a single fundamental matrix is due to the error in the estimated value of K, and how much is due to the error in the underlying fundamental matrix. This will then be used to derive a set of weights that effectively filter out noisy fundamental matrices by reducing their impact in the overall optimization. The algorithm proposed in this section is simply referred to as the Perturbation method in the experimental results section.

5.7.1 Perturbation of the Singular Values of the Essential Matrix

Let matrix $\widetilde{A}$ be some estimated value for matrix A consisting of an error term Δ_A such that:

$$\widetilde{A} = \Delta_A + A. \tag{5.15}$$

Using the above definition, matrix perturbation theory provides various insights on how functions of matrices $F(A)$ are related to their noisy estimates: $F(A+\Delta_A)$. Note that in this thesis all estimated values of various quantities have been indicated *without* a tilde. However, in this section the difference between noise-free and estimates needs to be drawn and so all estimates are denoted by the tilde sign whereas noise-free quantities are denoted simply by the symbol.

The specific domain of perturbation theory that is of interest in the proposed algorithm is the perturbation of the singular values of a matrix. Since this is how the objective function of self-calibration is defined, we will exclusively focus on the perturbation theory as it applies to the singular values of a matrix. In fact, a simple relationship exists between the singular values of a matrix and its noisy counterpart. According to [106]:

$$|\widetilde{\sigma}_i - \sigma_i| \leq \|\Delta_A\|_2, \quad i = 1, \ldots, n. \tag{5.16}$$

Here the standard definition of the matrix Euclidean norm $\|\Delta_A\|_2$ is used as: $\|\Delta_A\|_2 = \max_{x \neq 0} \frac{\|\Delta_A x\|_2}{\|x\|_2}$ and the Euclidean vector norm $\|x\|_2$ is defined as: $\|x\|_2 = (x^T x)^{\frac{1}{2}}$. The matrix Euclidean norm can be calculated from the definition to be the largest singular value of a matrix [37]. Therefore, the difference between the i-th singular value of a noise-free matrix and the i-th singular value of the noisy version of that matrix must be less than the Euclidean norm of the noise matrix, Δ_A, or its largest singular value. Before this theory can be applied to the objective function for self-calibration, we need to reexamine the self-calibration equations.

Our objective function Eq. 5.17 is defined in terms of the difference between the first and the second singular values of the essential matrix and is restated here, without the normalizing factor, as:

$$\min \sum_{i=1}^{n} \left(\frac{w_i}{\sum_{m=1}^{n} w_m} \left(\sigma_{(1,i)} - \sigma_{(2,i)} \right) \right) \tag{5.17}$$

where the value $\sigma_{(k,i)}$ is the k-th singular value of the i-th essential matrix and the w_is are the weighting factors for each fundamental matrix. The reason why the denominator of Eq. 5.8 is removed is that a different normalization will be adopted in this section. In order to create a uniform set of values for this objective function across all fundamental matrices, we ensure their largest singular value is one by dividing a fundamental matrix with the appropriate factor.

The estimate of the essential matrix can be written in terms of the noise-free and the error values of the fundamental matrix and the intrinsic parameters, using the same notation as Eq. 5.15 as:

$$\widetilde{E} = (K^T + \Delta_K^T)(F + \Delta_F)(K + \Delta_K). \tag{5.18}$$

Expanding the above terms, one obtains the error term of the resulting essential matrix, Δ_E as:

$$\Delta_E = \Delta_K^T F K + K^T F \Delta_K + \Delta_K^T F \Delta_K + \Delta_K^T \Delta_F K + K^T \Delta_F \Delta_K + \Delta_K^T \Delta_F \Delta_K + K^T \Delta_F K. \tag{5.19}$$

A bound on the value of the above error term can be found from Eq. 5.16. However, we wish to set a bound for $|\widetilde{\sigma}_1 - \widetilde{\sigma}_2|$ rather than $|\widetilde{\sigma}_i - \sigma_i|$ as in Eq. 5.16. Rearranging Eq. 5.16 by: $|\widetilde{\sigma}_1 - \widetilde{\sigma}_2| = |(\widetilde{\sigma}_1 - \sigma_1) - (\widetilde{\sigma}_2 - \sigma_2) + (\sigma_1 - \sigma_2)| \leq |\sigma_1 - \sigma_2| + 2\|\Delta_E\|_2$ we arrive at the desired bound:

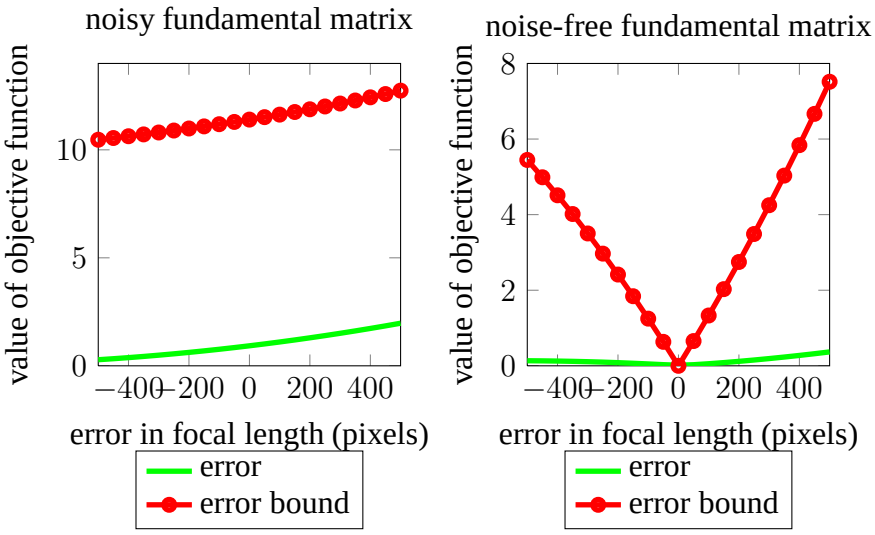
$$|\widetilde{\sigma}_1 - \widetilde{\sigma}_2| \leq |\sigma_1 - \sigma_2| + 2\|\Delta_E\|_2. \tag{5.20}$$

In other words, the distance between the first and second singular values of a noisy matrix, must be less than the distance between the first and second singular value of the noise-less matrix plus two times the Euclidean norm of the error matrix. It must be noted that this formula holds for an error of arbitrary size [106]. In addition, considering the fact that our objective function is merely the distance between the singular values of the essential matrix and that the noise-free essential matrix has identical first and second singular values (i.e., $\sigma_1 = \sigma_2$), the bound on our objective function can be simplified to:

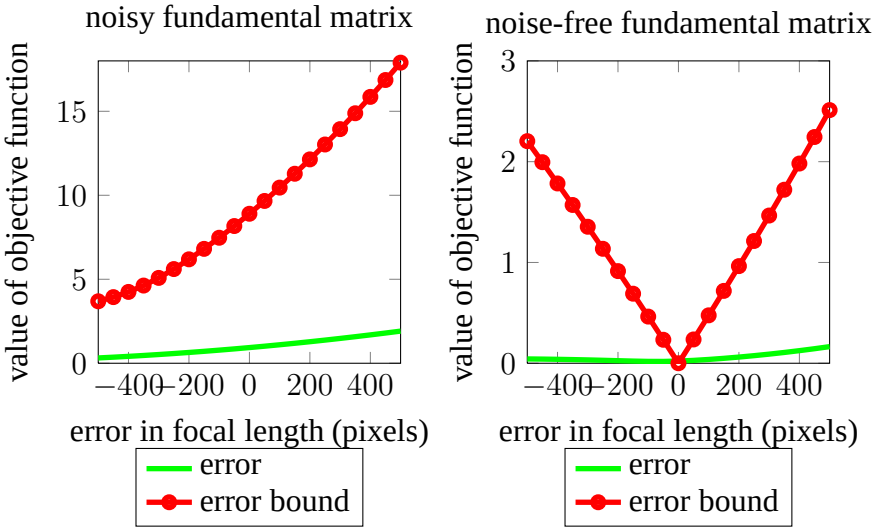
$$|\widetilde{\sigma}_1 - \widetilde{\sigma}_2| \leq 2\|\Delta_E\|_2. \tag{5.21}$$

Therefore, using the above formulation, a bound is estimated for our objective function, which is defined in terms of the difference between the singular values of an approximated essential matrix in terms of the error expansion, Δ_E of this estimated essential matrix. Note that the magnitude of this error depends on two error quantities, the error in the intrinsic parameters Δ_K and the error in the fundamental matrix Δ_F. If the fundamental matrix and the intrinsic parameters are exact, this bound approaches zero.

Figure 5.7 shows two examples of the estimated bound for two fundamental matrices from two different sequences where the ground truth fundamental matrix and intrinsic parameters are available and so the exact bound can be computed. Each sequence has two examples, one where noise is added to the fundamental matrix, and one with the noise-free fundamental matrix. The x-axis shows the error in the focal length of the intrinsic parameters used to estimate the essential matrix from the fundamental matrix.



(a) The value of the objective function and its bound for a single fundamental matrix in the Merton sequence.



(b) The value of the objective function and its bound for a single fundamental matrix in the Wadham sequence.

Figure 5.7: Examples of the objective function and its bound for noisy and noise-free fundamental matrices.

5.7.2 Robust Estimation Using Perturbation Bound

The above formulation of the bounds on the objective function for self-calibration using the fundamental matrix can be utilized effectively in designing a robust estimation method that is able to obtain calibration results even if a number of fundamental matrices used in the self-calibration process are noisy or violate certain basic assumptions of constant parameters. This method works via a modified form of the error definition shown in Eq. 5.19. This reduced error expansion does not account for errors due to the noise in the fundamental matrix but does account for errors in the intrinsic parameters. A bound is then derived from this error form using Eq. 5.21 and this bound is then used to calculate an Error Discrepancy (ED) value that indicates how much of the error is not accounted for by the modified bound and must therefore be due to noise in the fundamental matrix. This is then used in a weighing scheme which penalizes fundamental matrices in the optimization that have an error larger than their designated bound. This is due to the simple fact that any errors larger than our reduced bound must be due to errors in the fundamental matrix. As a result, fundamental matrices which contain sizable errors are effectively removed from the optimization. This process is repeated in an iterative scheme where the reduced bound and the values of the intrinsic parameters are both refined in each iteration.

Ideally, it would be desirable to find a modified form of Eq. 5.19 which only contains the terms that are due to the errors in the intrinsic parameters. This would then be an ideal bound for detecting errors that are too large and must be due to the additional terms which contain errors in the fundamental matrix. However, since no explicit knowledge of the error in the fundamental matrix Δ_F exists, an alternative way of finding this bound has to be devised. This is done by using the following form of error, designated as Δ'_E, which does not depend on explicit knowledge of the error in the fundamental matrix Δ_F or the ground truth fundamental matrix F :

$$\Delta'_E = \Delta_K^T \widetilde{F} K + K \widetilde{F} \Delta_K^T + \Delta_K^T \widetilde{F} \Delta_K^T \tag{5.22}$$

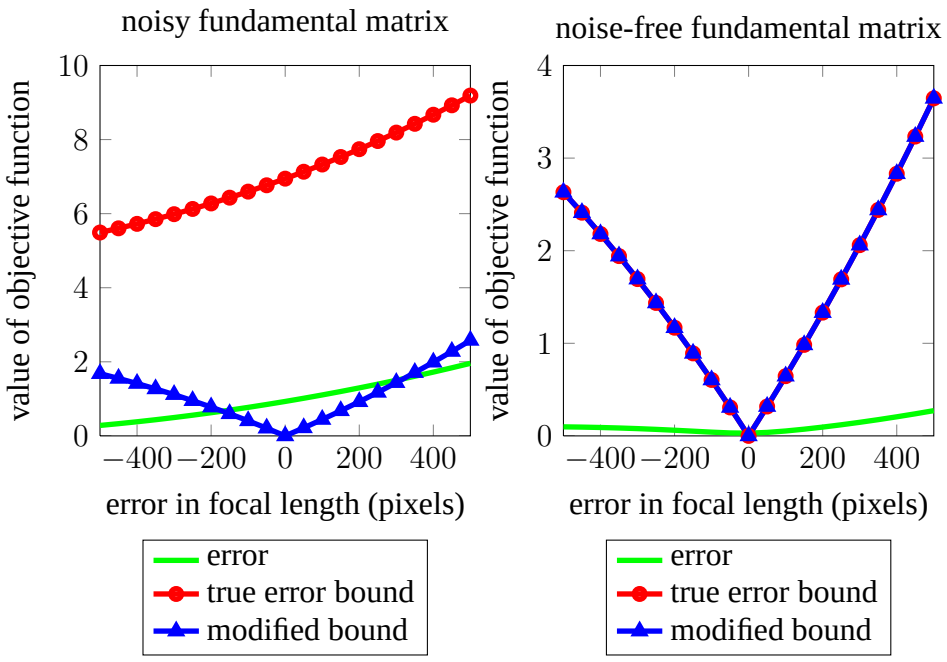
where $\widetilde{F} = F + \Delta_F$. This formulation of the error in the essential matrix expands to the full error in Eq. 5.19, except for the term $K^T \Delta_F K$. This term is in fact the most dominant term among the error terms since the largest norms of the camera intrinsic parameters K are far larger than the ones in the fundamental matrices $\widetilde{F}$ or the error in the intrinsic parameters matrix Δ_K. Note that this is always guaranteed since in our algorithm every fundamental matrix is divided by its Euclidean norm in a pre-processing stage, thus setting their norm to unity, whereas the norm of the intrinsics matrix is in the order of the focal length.

An important property of the new error definition Δ'_E is that it tends to approximate the full

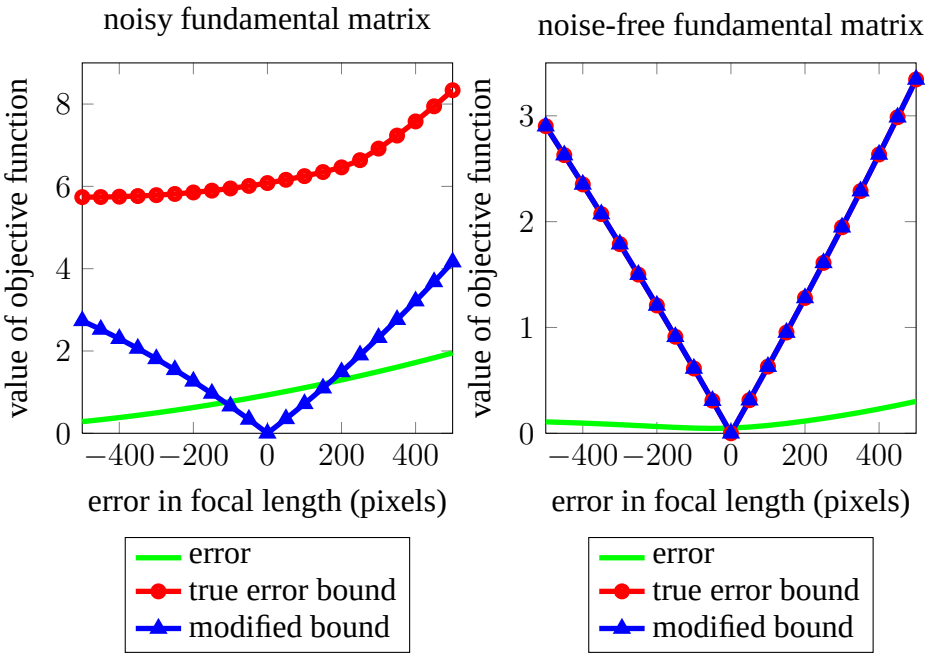
error expansion Δ_E for cases where the error in the fundamental matrix approaches zero, since in this case $K^T \Delta_F K$ becomes zero. Thus using the modified error, we can devise a bound that tends be higher than the actual error for the cases where the errors in the fundamental matrix are small, and lower for cases where the error in the fundamental matrix increases.

The new error bound is then used in Eq. 5.21 as $2\|\Delta'_E\|_2$. Figure 5.8 shows an example of the new error bound. Two sequences are used here, similar to Figure 5.7, showing the value of the objective function for varying levels of noise in the estimation of the focal length, with the addition of the new error bound. As it is shown, while the error bound derived from the full error terms is clearly larger than the error values for all error magnitudes, the modified error is less than the actual error for most values of the error in the focal length in the case of the noisy fundamental matrix. However, for the noise-free fundamental matrix, the error bound is identical to the full error bound and these two coincide. As a result, the modified error bound is always larger than the actual error for the case of the noise-free fundamental matrix. As explained, this is because the modified error bound differs from the full error bound by a single term that disappears in the case of noise-free fundamental matrices.

This idea is used to devise an iterative strategy to find a set of weights for the set of fundamental matrices available for self-calibration in order to reduce the impact of the outlying fundamental matrices. In the following discussion of the iterative approach we denote the estimate of the intrinsic parameters at the j-th iteration by $\widetilde{K}^{(j)}$, the i-th fundamental matrix as $\widetilde{F}_i$. Also, the error of the essential matrix calculated from the i-th fundamental matrix at iteration j is denoted as $\widetilde{E}_i^{(j)}$ and the estimate of the error in the intrinsic parameters at iteration j is denoted by $\Delta_K^{(j)}$. The algorithm starts by a weighted minimization of the objective function where all fundamental matrices are weighted equally. The nonlinear minimization then finds a set of values for the intrinsic parameters $\widetilde{K}^{(1)}$, starting from some initial value $\widetilde{K}^{(0)}$. At this point, the error of each fundamental matrix, $\widetilde{F}_i$, with respect to the estimated solution, $K^{(1)}$ is estimated by simply taking the difference between the first and the second singular value of the given essential matrix found using the estimated solution. The modified bound, $\Delta'_{E_i^{(j)}}$ is also calculated for every fundamental matrix at every iteration. This is done by approximating the value of the intrinsic parameters K by the latest estimate of the intrinsic parameters, in this case $K = \widetilde{K}^{(1)}$ and by approximating the error in the intrinsic parameters matrix with its change since the last iteration: $\Delta_K^{(j)} = \widetilde{K}^{(j)} - \widetilde{K}^{(j-1)}$. The amount of change in the estimate of the intrinsic parameters in two consecutive iterations is indicative of how stable the results are and is a reasonable estimate of the error in the underlying solution. Note that for estimating the error bound, the actual value of the error parameters or the intrinsic parameters is not of consequence. It is the norm of these parameters that is of interest, and as long as the norms



(a) Modified bound for a single fundamental matrix in the Merton sequence.



(b) Modified bound for a single fundamental matrix in the Wadham sequence.

Figure 5.8: Modified perturbation bound for the objective function for self-calibration for two sequences.

of the estimates are within reasonable proximity of the true values, the bound will serve its purpose in helping determine the appropriate weights for each fundamental matrix.

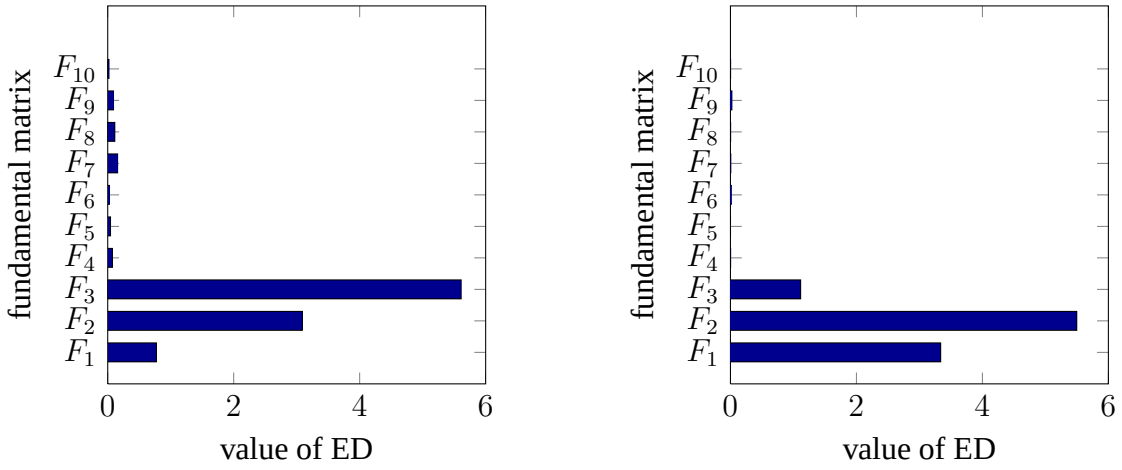
Once the modified bound, $2\|\Delta'_{E_i^{(j)}}\|_2$ for each fundamental matrix i is found at iteration j, a measure of how much the actual errors fall within this bound is found using an estimate of the Error Discrepancy (ED). The ED for the i-th fundamental matrix at the j-th iteration is defined as:

$$ED_i^{(j)} = \left(\frac{\sigma^{(1,i)} - \sigma^{(2,i)}}{2\|\Delta'_{E_j^i}\|_2} \right)^2 . \tag{5.23}$$

In other words, the error of a fundamental matrix with respect to the latest estimate of the intrinsic parameters via self-calibration, over the modified error bound for that given iteration is used as an indicator of how likely a fundamental matrix is to be erroneous. To see this, note that when the modified bound exceeds the error in the numerator, the value of ED will be small, otherwise this value will increase. Also, errors that result from poor estimates of the intrinsic parameters will be common to all fundamental matrices in the optimization, but when the source of the error is a particularly noisy fundamental matrix, the measure of the ED will be disproportionately large for that fundamental matrix. Figure 5.9 shows an example of finding the ED measure for ten fundamental matrices from two frames from two separate sequences. All fundamental matrices in these sets are fit to ground truth correspondences, except the first three fundamental matrices (i.e., F_1,F_2 and F_3) which are fit to a set of correspondences containing 10% outliers. The indices of the y-axis show the number of the fundamental matrix, the first three of which are outlying fundamental matrices. In addition, the self-calibration objective function is estimated to an incorrect set of intrinsic parameters where the focal length is changed by 10% of its magnitude for both sequences. Therefore, all objective functions are deviating from their optimum value via errors in the calibration parameters and some due to noise in the fundamental matrix in addition to this. As it is shown, although all objective functions are estimated for an incorrect value for the focal length, the ED measures are significantly higher for the noisy fundamental matrices.

As a result one can use the outlying values of ED measures to find outlying fundamental matrices. In order to do this, the ED measures are used to derive a set of weights used in the subsequent iteration of the nonlinear minimization. This is done by mapping the $ED_i^{(j)}$ measures, to a set of smooth values via an exponential function. The weight for the i-th fundamental matrix, at the j-th iteration is found via:

$$w_i^{(j)} = \exp(\frac{-ED_i^{(j)}}{s_j}) \tag{5.24}$$



(a) Values of the ED index for Corridor sequence. (b) Values of the ED index for Wadham sequence.

Figure 5.9: Using the Error Discrepancy index to discern outliers in a set of 10 fundamental matrices fit to the first five frames of the Corridor and Wadham sequences where the first three fundamental matrices are corrupted by 10% outliers noise and all focal length parameters corrupted with 10% error.

Here, the s_j is a robust measure of the standard deviation of the ED values at iteration j which is found using the median absolute deviation (MAD) of ED values according to $s_j = 1.4826\text{MAD}$ [64]. These weights are then used for the next iterations of the algorithm. The iterations are stopped once the estimate of the error in the intrinsic parameters $\Delta_K^{(j)}$ approaches some value T. This value is simply the tolerance in the accuracy of the estimated results desired for the algorithm and is set by the user. Also, the iterations can be stopped via a maximum iteration cap.

Algorithm 3 shows the outline of the robust self-calibration algorithm.

Here, K_{init} is defined as $K_{init} = \begin{bmatrix} width & 0 & \frac{width}{2} \\ 0 & width & \frac{height}{2} \\ 0 & 0 & 1 \end{bmatrix}$ which is our starting point for the calibration parameters for a wide-angle camera where no other prior information is provided. This constitutes a 90° field of view camera. The values of $width$ and $height$ are the width and height of the image respectively and the sequence contains N fundamental matrices. The weighted nonlinear minimization is carried out via the Levenberg-Marquardt algorithm. In our experiments the maximum number of iterations is set to 4 which was found to be a reasonable value, balancing accuracy in the solution and computation time.

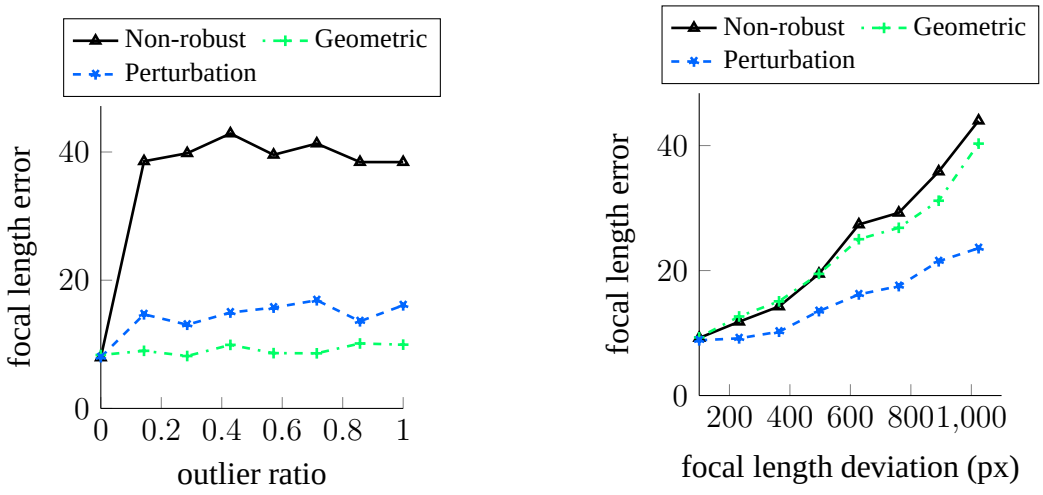
Algorithm 3 Robust self-calibration via perturbation bound

1: Estimate normalized fundamental matrices in the sequence $\frac{\widetilde{F}_i}{\|\widetilde{F}_i\|_2}$
2: Initialize: $j = 1, w_i^{(1)} \leftarrow 1$, for all i and $\widetilde{K}^{(0)} = K_{init}$
3: **while** $\Delta_K^{(j)} > T$ and $j <$ maxIter **do**
4: Set starting point for optimization to $\widetilde{K}^{(j-1)}$
5: $\widetilde{K}^{(j)} = \arg\min_K \left(\sum_{i=1}^{n} \left(\frac{w_i^{(j)}}{\sum_{m=1}^{N} w_m^{(j)}} \frac{\sigma^{1,i} - \sigma^{2,i}}{\sigma^{2,i}} \right) \right)$
6: Set $\Delta_K^{(j)} \leftarrow \widetilde{K}^{(j)} - \widetilde{K}^{(j-1)}$
7: Estimate error $\Delta'_{E_i^{(j)}}$ for all fundamental matrices via Eq. 5.22
8: Estimate Error Discrepancy $ED_i^{(j)}$ for all fundamental matrices via Eq. 5.23
9: Set weights for next iteration: $w_i^{(j)}$ via Eq. 5.24
10: j=j+1
11: **end while**
12: **return** $\widetilde{K}^{(j)}$

5.7.3 Experimental Results

In the first set of synthetic experimental results shown in Figure 5.10b outlying fundamental matrices are created by adding outlying matches to the point correspondences used in calculating the fundamental matrices, as explained previously in Section 5.5.3. In this case two fundamental matrices have been corrupted by varying degrees of outliers. As shown in the graph, the non-robust method performs worst by producing an estimate that is at most within 50% of the ground truth. On the other hand, the Geometric method produces the best results, with an average performance of about 7% error. The proposed algorithm is able to produce results that are independent of the magnitude of the error. However the performance is slightly worse than the Geometric algorithm. This is due to the fact that in this particular case, the error was due to errors in the matching stage in the self-calibration pipeline, which is exactly what is used as a measure of confidence in the fundamental matrices by the Geometric method. Note that the levels of confidence in the fundamental matrices calculated via the Geometric method and the perturbation-based method can in fact be combined via a mixing parameter in a hybrid weighing scheme.

The second set of experimental results also consists of measuring the performance of the algorithms with respect to outlying fundamental matrices. As explained, in this experiment outlying fundamental matrices are created via deviation from the assumption that the intrinsic parameters are constant across all frames. This is done by creating a synthetic image sequence



(a) Performance vs varying levels of matching outliers.

(b) Performance vs focal length deviation.

Figure 5.10: Synthetic performance evaluation of the Perturbation-based method compared with two standard self-calibration techniques.

of five frames with fixed parameters and then changing the focal length of one of the frames. As it is shown in Figure 5.10b, in this case the perturbation-based method is able to produce significantly better results than the other two methods. This is due to the fact that the weights calculated in this algorithm are dependent on the error with respect to the self-calibration constraint, unlike the Geometric method which relies on the confidence in the epipolar geometry. The Geometric method performs better in the first test scenario due to the fact that the errors of the fundamental matrices with respect to the correspondences is the method by which the outliers in this experiment are created. However in the second scenario, the Geometric error is no longer indicative of the outlying nature of the fundamental matrices and so the proposed algorithm which does not rely on this information performs best.

In summary, perturbation theory offers an insight into the behavior of the objective function used in self-calibration and this fact can be used in devising a robust estimator for self-calibration. The performance of this algorithm is competitive with the Geometric method that uses information from fitting the fundamental matrices when the source of error is solely in the correspondences used to fit the multi-camera relations. However, the proposed algorithm outperforms the Geometric method in cases where the source of errors is violation of the constancy of the intrinsic parameters. Therefore, the proposed method is more general and does not rely on matching information and can solely perform on the provided fundamental matri-

ces for an image sequence and provide accurate estimates of intrinsic parameters in spite of outlying fundamental matrices.

5.8 Self-calibration by Localization of the Convergence of Zero Curves

5.8.1 Introduction

The algorithm proposed in this section falls within the same robust method of retrieving the camera parameters as the Perturbation method and the Randomized Multi-start method presented previously in Section 5.7. However, unlike the previous algorithms, the proposed method in this section does not rely on a nonlinear optimization routine to solve for the self-calibration equations. In fact, only a root finding algorithm is required to use this method. Unlike the previous algorithms that attempt to use the singular values of the essential matrix to solve the self-calibration problem, a polynomial objective function is used which is based on the norm of the essential matrix, as proposed in [36]. In addition, the proposed method can handle multiple acquisition devices and degeneracies and can handle large errors in the underlying camera relations used for self-calibration. As in the case of previous algorithms, the emphasis has been placed on the robustness of the algorithm with respect to outliers as defined in Section 5.5. As explained previously, the issue of outliers has a dramatic impact on self-calibration. As a result the goal of this algorithm is overcoming the sensitivity of self-calibration to the various outliers that have been presented.

Unlike similar methods of finding the solutions to a set of polynomials, the proposed method can handle significant outliers and does not rely on computationally expensive methods of finding solutions to sets of polynomials. In fact, only the roots to single univariate polynomials are estimated using the Companion Matrix method [27] which only involves finding the eigenvalues of an 8×8 or 4×4 matrix in our algorithm. Therefore, the only mathematical library required is one for calculating eigenvalues.

Unlike the method presented in [36] where the global optimization method of Interval Analysis is utilized, here a robust method of solving the set of eight degree polynomials based on the norm of the essential matrix is adopted. The advantage of the proposed method over Interval Analysis is computation time, robustness and simplicity of the implementation. The runtime of the Interval Analysis self-calibration method is in the order of 30 minutes for a sequence of five images, where an un-optimized implementation of the proposed algorithm

runs in microseconds. Most importantly, as with the other proposed self-calibration methods, the presented algorithm is able to deal with high levels of outliers as explained in Section 5.5.

Solving systems of multivariate polynomial equations is generally an NP-hard problem [24]. Generally the solution involves constructing a Gröbner basis [10]. This approach, while being computationally expensive has already been explored in the computer vision literature for the pose estimation problem [107]. A more established method in the computer vision community is the homotopy continuation method which has also has been used in self-calibration in solving the Kruppa equations. Initially, the method presented in [61] used an earlier implementation of this method to solve the Kruppa equations. Since over-determined sets of polynomial equations are not expected to have any solutions [24], the authors choose to use a minimal set of such equations and find all solutions of such subsets and then consolidate all the non-imaginary results and find the one with the lowest error. However, the number of solutions derived from these minimal sets is rather high, since according to Bezout's theorem [55] the number of possible solutions is d^N where d is the degree of the polynomial and N is the number of equations in the system. In the case of the algorithm proposed in [61] this amounts to 2^5 possible solutions for each subset. Another technique for solving polynomial objective functions inspired by Algebraic Geometry is one proposed in [54] using the Hidden Variable Method. This method for estimating the motion and the focal length is highly effective for the six-point semi-calibration scenario where the only unknown is the focal length. If the number of unknowns increases from one to four in the case of full calibration, this method would also have to deal with a very large number of possible solutions. Unlike such methods that need to prune a large number of solutions for minimal subsets of polynomials, the proposed method can seamlessly accommodate an arbitrary number of equations and solve an over-determined set of multivariate polynomials without any ambiguities in the solution. Most importantly, the proposed method can accommodate polynomials that are derived from outlying fundamental matrices and share no common roots with other polynomials in the system.

Another self-calibration method that also utilizes systems of multivariate polynomials is proposed in [39] where the authors use homotopy continuation to solve for the scale factor in the Kruppa equations. Similar to the previous case, this method requires filtering through a large number of solutions and does not handle outliers.

Section 5.8.2 defines the objective function and the resulting polynomial constraints. The method of finding families of solutions is proposed in Section 5.8.3. The localization of the unique calibration solution from these families of solutions is proposed in Section 5.8.4 and some experimental results pertaining to degeneracies and multiple acquisition devices are also presented in Section 5.8.4. Similar to the other self-calibration algorithms presented, the com-

parisons with other methods have been presented in Section 5.9.

5.8.2 The Polynomial Objective Function

Another form of the Huang-Faugeras constraint was presented in [61]. This is the objective function that is used in the proposed algorithm and is stated as:

$$2\text{tr}((EE^T)^2) - (\text{tr}(EE^T))^2 = 0. \tag{5.25}$$

Since $E = K^T F K$ then the same constraint can be restated as:

$$2\text{tr}((K^T F K K^T F^T K)^2) - (\text{tr}(K^T F K K^T F^T K))^2 = 0. \tag{5.26}$$

Expanding this constraint one obtains an eighth degree polynomial in the unknowns of the intrinsic camera parameters. This polynomial is denoted as : $\phi(f_x, f_y, u_c, v_c) = 0$. In total there are as many such equations as there are fundamental matrices in the image sequence. As explained before, previous works have attempted to solve this equation using minimal sets using homotopy continuation. However, in order to achieve robustness and a reduction in computation time, it is important to ensure that outlying fundamental matrices do not interfere with the estimation. As a result, in this algorithm the equation is solved for each fundamental matrix separately. However, unlike the minimal sets, this means that the solution is under-constrained and there are infinitely many solutions. These solutions generally form a connected differentiable curve as will be discussed in the next section. Once these solution curves are estimated for each single fundamental matrix, they will be aggregated and a solution is estimated via a consensus method that localizes the most likely solution to such a system. To see how this differs from other possible approaches, three categories of methods of solving systems of multivariate polynomial equations will be reiterated. Essentially, given m polynomials of n unknowns, assuming $m > n$, there are three methods of solving such a system. The first method, is to use all m equations and solve this problem at once. For example this can be achieved via a nonlinear optimization routine that minimizes some measure of the overall error in all m equations, such as least squares. However, this idea suffers from the fact that the summation of errors cancels many of the convexities of the individual polynomials which means the final outcome is likely to diverge from the global minimum [54]. In addition solving a polynomial system by the general nonlinear minimization approach of using an initial starting point iteratively minimizing the error will yield trajectories that do not converge and many other trajectories will converge to the wrong solution [61]. Using an algebraic geometry approach would also fail when using all m equations together since when using more polynomials than

category	generation	# solutions	aggregation
minimization	$\min \sum_{i=1}^{n} \lvert g_i(\mathbf{x}) \rvert$	1	none
algebraic geometry	$Z := \{\mathbf{x} \in \mathbb{C}^m \lvert g_{t_1}(\mathbf{x}) = g_{t_2}(\mathbf{x}) \cdots = g_{t_m}(\mathbf{x}) = 0\}$ **t** selects m subset of eqs	$\binom{m}{n} \times d^n$	Kernel voting Score minimization
proposed	$Z_k := \{\mathbf{x} \in \mathbb{C}^m \lvert g_k(\mathbf{x}) = 0\}$ for all k	∞	$\bigcap_{i=1}^{n} Z_i$

Table 5.1: Finding the solution $\mathbf{x}^*$ to a system of n multivariate polynomials, $P(\mathbf{x}) = ((g_1(\mathbf{x}) = 0, g_2(\mathbf{x}) = 0 \ldots g_n(\mathbf{x}) = 0)$ of degree d with m unknowns (i.e., $\mathbf{x} \in \mathbb{R}^m$). The polynomials are defined as: $g(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{Z}^m, \alpha_1 + \alpha_2 + \cdots + \alpha_m \leq d} p_{\boldsymbol{\alpha}} x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_m^{\alpha_m}$.

unknowns, the system is often inconsistent and no solutions exist. The second method of finding the solution to this system is by using minimal sets, i.e., using n equations at a time, as explained in the introduction. This approach requires computationally expensive methods and the number of generated solutions is $\binom{m}{n} \times d^n$, where d is the degree of the polynomial. These solutions need to somehow be filtered which is also an ambiguous process.

Since the goal of this thesis is robustness, an approach that does not "bundle" more than one fundamental matrix at a time is required. As a result of this, for each fundamental matrix F_i, the polynomial $\phi_i(f_x, f_y, u_c, v_c) = 0$ is solved separately. As mentioned, this leads to infinitely many solutions. However, these solution sets have particular structural properties that can be taken advantage of in designing a consensus-based algorithm to find the best solution which is free of any of the noisy fundamental matrices in the sequence. Table 5.1 summarizes these three categories of solving multivariate polynomial systems of equations. The second column labeled "generation" denotes the method used to generate the set of solutions. The "aggregation" column denotes the method used to isolate a unique solution from the generated solutions.

As it is shown, the proposed method differs from the other main approaches in that polynomials are solved one at a time, therefore making it possible to detect and prune out outlying fundamental matrices. However, the challenge in this case is the aggregation of the results. Essentially, in the proposed method, the aggregation step is finding some intersection among the series of solution curves as shown in the table. Although not specifically an intersection operation between the solution sets, the goal is to find a location in the solution space that minimizes the distance of the curves to this location, or maximizes some measure of consensus.

This will be discussed in detail in Section 5.8.4. Also, note that the solutions are shown to belong to the complex domain. Generally the roots of the equations used for self-calibration are complex. Dealing with spurious solutions and the complex components will be addressed in Section 5.8.4.

5.8.3 Finding Solution Curves

Before discussing the localization of the unique solution from the zero curves (or solution curves) provided by each individual polynomial in the next section, the method of producing these curves will be discussed. In order to reduce the computation time, the polynomials from the objective function, $\phi(f_x, f_y, u_c, v_c) = 0$ are separated into polynomials, $\phi(f_x, f_y, u_c = u_c^*, v_c = v_c^*) = 0$ and $\phi(f_x = f_x^*, f_y = f_y^*, u_c, v_c) = 0$. For brevity, the polynomial in the focal length is denoted as: $Z_i^f = \phi_i^f(f_x, f_y) = 0$ and the polynomial in the optical center is denoted by $Z_i^{oc} = \phi_i^{oc}(u_c, v_c) = 0$. Here f_x^*, f_y^* represents the latest estimate of the focal length in horizontal and vertical pixels and u_c^*, v_c^* represents the latest estimate of the location of the optical center. In other words, this polynomial in four dimensions has been interchanged with two polynomials in two dimensions. Although not generally an acceptable method of approaching an optimization problem, the aggregation method is robust to reasonable deviations in the assumed parameters when evaluating the minimum. In other words, the estimates of the focal length from $\phi(f_x, f_y, u_c, v_c) = 0$ and $\phi(f_x, f_y, u_c + \Delta_{u_c}, v_c + \Delta_{v_c}) = 0$ are sufficiently close, given that the errors in the components of the optical center Δ_{u_c} and Δ_{v_c} are reasonably small. This is in fact shown from the experiments where the robustness of the proposed algorithm is tested against incorrect assumptions about the assumed fixed parameters.

Before the above equations are analyzed, the fundamental matrices are computed between all pairs in the given image sequence. Having obtained N fundamental matrices, there are now N number of the above equations. So for each fundamental matrix, F_i, we form:

$$\phi_i(f_x, f_y, u_c = u_c^*, v_c = v_c^*) = 0 \tag{5.27}$$

$$\phi_i(f_x = f_x^*, f_y = f_y^*, u_c, v_c) = 0. \tag{5.28}$$

The solving of the above two equations is done in two separate steps. The equation in the focal length is solved first using the image center as the best estimate of the optical center. Following this, the estimated overall solution in the focal lengths is used in the equation for the optical center.

In order to generate solutions from each polynomial, the zero curves of each will need to be resolved in some predetermined interval for both the focal lengths and the optical centers.

Zero curves in this context refer to the set of all the roots of the bivariate polynomials under consideration. The zero curves for the focal lengths, Z_i^f and the optical center Z_i^{oc} for the fundamental matrix F_i are generated as follows:

$$Z_i^f := \{(f_x, f_y) \in \mathbb{C}^2 | \phi_i(f_x, f_y, u_c = u_c^*, v_c = v_c^*) = 0, \epsilon \leq f_x \leq 3w, \epsilon \leq f_y \leq 3w\} \quad (5.29)$$

$$Z_i^{oc} := \{(u_c, v_c) \in \mathbb{C}^2 | \phi_i(f_x = f_x^*, f_y = f_y^*, u_c, v_c) = 0, \epsilon \leq u_c \leq w, \epsilon \leq v_c \leq h\} \quad (5.30)$$

where w is the width and h is the height of the image. Bounding the focal length to be less than three times the width limits the camera focal length to those belonging to wide angle and mid-range focal lengths, ruling out telephoto lenses which are not commonly used for capturing an environment for the purpose of visualization. Also, ϵ denotes some minimum bound on the value of the focal length. This is generally set very low to ensure against zero values which can occur for degenerate scenarios. This will be addressed in Section 5.8.4. The zero curves are essentially numerical representations of a quantized segment of the roots of the polynomials in the parameter space. These form sets of tuples where in the case of the focal length and the adopted convention the first component of the tuple is f_x and the second is f_y.

The family of solutions, or the zero curves, Z_i^f and Z_i^{oc} are found by solving for the roots of the bivariate equations in a specific interval. The method of calculating these points is by simply fixing one parameter and solving the roots of the univariate polynomial and sweeping the parameter space, similar to the earlier methods of stability analysis of 2D FIR filters [98]. Specifically, the focal length in the horizontal pixels, f_x is varied uniformly in the range, so that the solution curve, Z_i^f contains N elements, where the j-th element is calculated as:

$$Z_i^f(j) = (f_x = j\delta, f_y = \beta) \quad \text{where } \phi_i^f(j\delta, \beta) = 0. \quad (5.31)$$

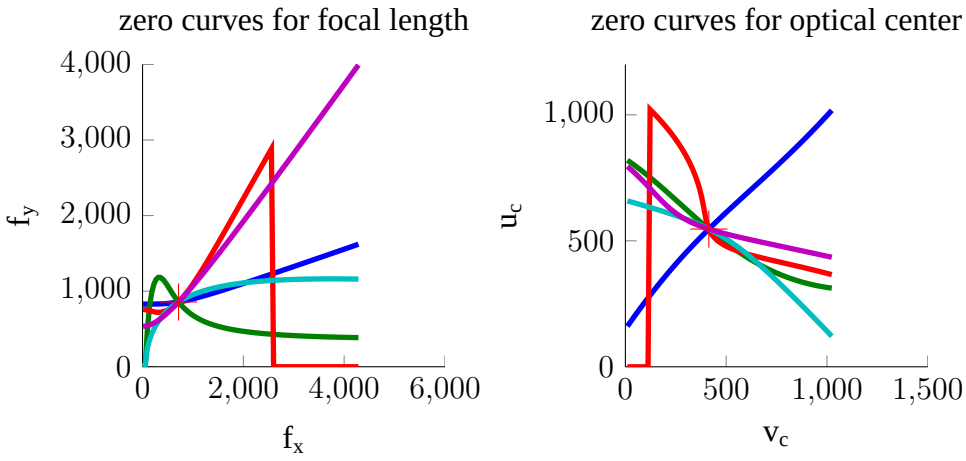
Note that the space of the focal length in the x direction is divided into N equal sized elements of length δ and for each value of the focal length in the x direction the value of the focal length in the y direction is calculated using the matrix companion method [27]. This amounts to finding the roots of a 4-th degree polynomial by finding the eigenvalues of a 4×4 matrix which is a very fast operation. The zero curve of the optical center is calculated similarly; however in this case the resulting univariate polynomial is of 8-th degree and so it requires finding the eigenvalues of an 8×8 matrix. The reason for the difference between the degrees of the polynomial in the optical center and the focal length is that the polynomial in the focal length is sparse and can be reduced to a 4-th degree polynomial by simple substitution, which is not the case for the polynomial in the optical center. Another important step in

calculating the zero curves is filtering out the spurious solutions. Clearly, there will be four, or eight roots depending on which zero curve is being estimated. The correct root is picked as the one with a non-zero real part and an approximately zero imaginary part. In the ideal case, this root would contain a zero imaginary part. However, due to noise and numerical imprecisions the imaginary part of the chosen root will not always be non-zero. When this is encountered, the imaginary part is discarded and the real part of the root used as the final value.

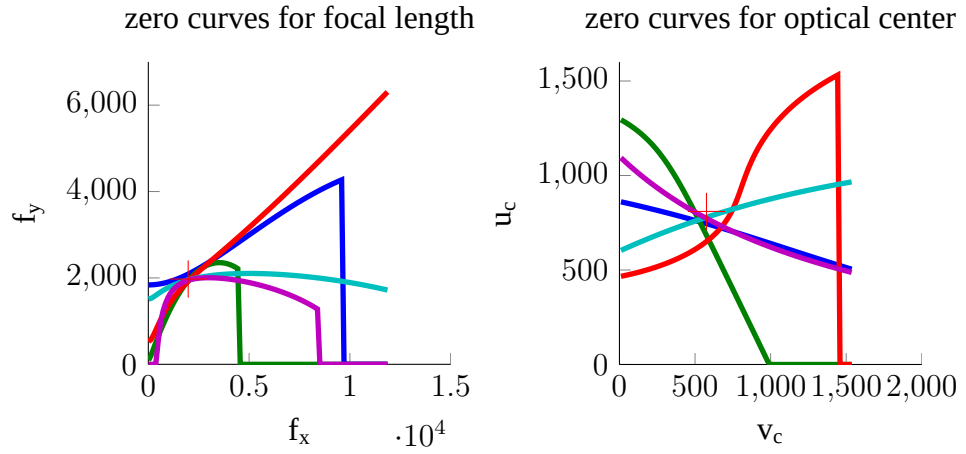
An important property of the zero curves of the objective polynomials is that they always form analytic curves in the parameter space except for cases where the zeros split into multiple components [32]. Another way to see that roots often form smooth varying curves is via the fact that the roots of the polynomial are eigenvalues of a matrix formed with the coefficients of the polynomial according to the matrix companion method. Since it is known that eigenvalues of a matrix are differentiable with respect to its entries, the same idea can be applied to the roots of a polynomial. In fact this has been used to set bounds for the roots of polynomials in [74]. This property is an important assumption in the proposed algorithm as will become apparent when the method of finding a unique solution from the zero curves is presented.

To see how the zero curves form smooth curves in the parameter space, several examples are shown in Figure 5.11. Figure 5.11a shows the zero curves for the polynomials derived from four fundamental matrices of a synthetic sequence. The left diagram shows the zero curves for the focal length and the right shows the zero curves for the optical center. The correct solution is marked with a "plus" sign in the graph. The same process is repeated below, in Figure 5.11b for the Rathaus image sequence [2]. As it is shown, the zero curves form smoothly varying curves and approach the real solution depending on the amount of noise in the fundamental matrix. In fact, ideally, all the zero curves should go through the correct solution; however, due to noise this does not always occur. Since the curves are generally smooth varying, the number of samples does not exert a large amount of influence in the overall results. A reasonable number of samples used in this thesis is 100. However, depending on how much accuracy is required this amount can be adjusted.

During the self-calibration of an image sequence containing N fundamental matrices, the above procedure is carried out for all fundamental matrices. In certain cases, such as when encountering degeneracies, some of the zero curves will collapse to zero. These will be discarded from the set of zero curves. The solution is calculated by first finding the point in the 2D parameter space, (f_x, f_y), with the highest "consensus" from all the zero curves. This value is used in the next step where the same procedure is repeated to find the optical center. The details are presented in the next section.



(a) Zero curves for five fundamental matrices from a noise-free synthetic sequence.



(b) Zero curves for five fundamental matrices for the Rathaus sequence.

Figure 5.11: Examples of four zero curves for a real and a synthetic image set. The true solution is marked with a plus. Each color-coded curve is obtained from a fundamental matrix.

5.8.4 Localizing Unique Solution

When all the zero curves have been calculated for the focal length and optical center, in other words when $Z_1^f, Z_2^f, \ldots Z_N^f$ and $Z_1^{oc}, Z_2^{oc}, \ldots Z_N^{oc}$ have been estimated, the camera parameters can be estimated. The numerical estimation of these zero curves is a very fast operation. For a sequence of 10 images, calculating all of the zero curves takes approximately 1.3s for a total of 45 pairs of zero curves. The computational complexity of estimating the zero curves is linear in the number of the fundamental matrices.

When the numerical estimates of the zero curves have been collected, the task is finding a point that would lie on all the curves. This would be the one point that is in common between all the zero curves (i.e., an intersection). Effectively this would be finding the intersection of the zero curves or: $(f_x^*, f_y^*) = \bigcap_{i=1}^N Z_i^f$ and $(u_c^*, v_c^*) = \bigcap_{i=1}^N Z_i^{oc}$. This would be a reasonable method of localizing the solution of the system if noise was not present in the underlying fundamental matrices as in the case of Figure 5.11a. Even though the zero curves of the polynomials of a noise-free intersect, such an assumption cannot be made for arbitrary sequences as shown in Figure 5.11b.

In order to find what is the most likely intersection of the noise-free version of the zero curves, several strategies can be adopted. Since the zero curves are merely possible solutions in the solution space, an approach based on Hough transform can potentially be adopted [48, 131]. One issue with using this approach is that the density of the data points is not necessarily sufficient for Hough transform to provide a clear solution (depending on the number of images in the sequence). Moreover, there is no clear method for deciding the quantization interval. Another possible approach is using a clustering method such as k-means clustering to find the cluster with the highest number of members from all the zero curves. This method has also been attempted. However, there is no way of fixing the number of clusters. Another method that has also been attempted is mode seeking algorithms, specifically the Mean Shift algorithm [23]. This approach also has the challenge of setting an appropriate bandwidth and poor localization when insufficient samples are used, such as in cases with fewer images. However, the fundamental issue with all the mentioned algorithms is their approach to treating every single sample as independent from all other samples. In fact, the data segments form highly dependent structures in the parameter space (smooth curves). Treating every point in the curves as independent from every other point in the same curve is ignoring the underlying structure of the data.

The approach adopted in this algorithm is to solve this from the perspective of a geometric optimization problem [14]. In other words, the goal is to find a solution whose distance from

all zero curves is minimized. In other words, for the case of the focal length we define the solution $\mathbf{f}^* = (f_x^*, f_y^*)$ as:

$$\mathbf{f}^* = \arg\min_{\mathbf{f}} \sum_{i=1}^{N} \mathrm{d}(\mathbf{f}^*, Z_i^f). \tag{5.32}$$

Here, $\mathrm{d}(\mathbf{f}^*, Z_i^f)$ is a distance between a point in the parameter space and one of the zero curves, defined as $\mathrm{d}(\mathbf{f}^*, Z_i^f) = \inf\{\parallel \mathbf{f}^* - \mathbf{x} \parallel \; | \mathbf{x} \in Z_i^f\}$. Clearly, this would be the intersection of the curves if such a point did in fact exist, as in the ideal synthetic case. However, in the presence of noise, this point would approach a point where the zero curves are most likely to pass through and effectively find a likely location for the solution of the problem.

One modification to the formulation in Eq. 5.32 that has been adopted is using a function of the distance that favors intersections and closer proximities. In other words, we have adopted to maximize $\max \sum_{i=1}^{N} \rho(\mathrm{d}(\mathbf{f}^*, Z_i^f))$ where ρ is a smooth penalty function that heavily favors intersections and closer proximities. This function is simply the exponential function: $\rho(d) = \exp^{(-d/\sigma_f)}$ which has its highest value at $d = 0$. The normalizing factor, σ_f is used to create a uniform set of values for the distance, regardless of camera geometry. This is a function of the assumed interval for the possible values of the focal length and the optical center and is set to be one tenth of the allowable range for the parameter that is being searched for . This has the effect of producing the same distribution of distances regardless of the camera parameters.

When maximizing this function of the distances, $\rho(d)$, the estimated point would be one that is most likely to be on an intersection of a set of curves. Rather than merely minimizing distance of the curves, this means that now we are searching for a point that is most likely to be an intersection. Although in synthetic cases the two approaches would produce identical results, in cases where noise is present this formulation leads to a solution that is more likely to be formed by intersections of the zero curves. To illustrate, Figure 5.12 shows a set of zero curves and the resulting objective function formed by a synthetic sequence with two outlying fundamental matrices. The parameter space in this example is the focal length and the objective function is defined as the negative exponent of the sum of distances of a point to all zero curves as defined above.

This problem is a nonlinear optimization which can be solved using a standard optimization routine. However, since a nonlinear optimization has the issues of convergence in cases where the starting point is not close enough to the global minimum, an alternative solution is sought. This method is based on assuming that the solution belongs to one of the zero curves. This is a reasonable assumption since ideally the solution has to be a member of at least one of the zero curves. This has the effect of reducing the search space to points that are only on the zero curves. This will change the optimization formulation as (removing the superscript which

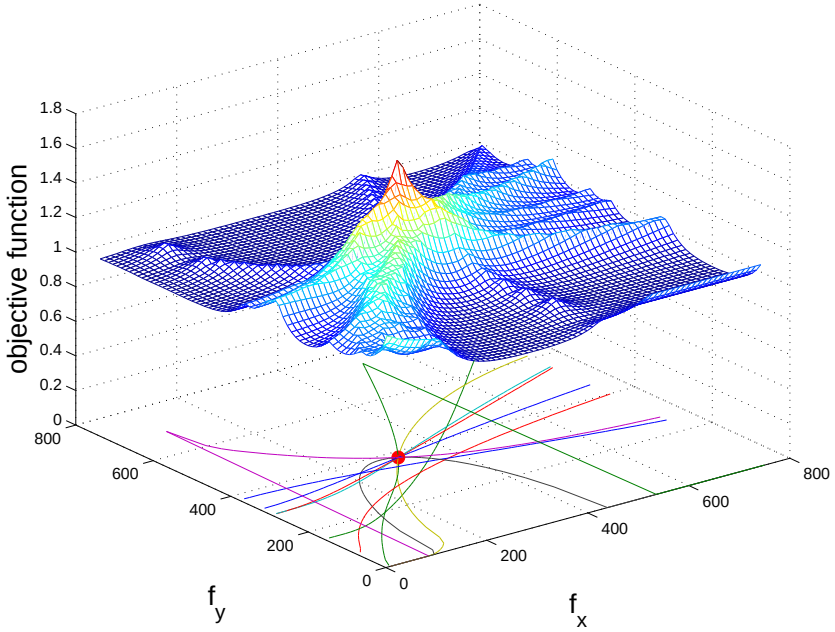


Figure 5.12: Zero curves and the objective function for a synthetic sequence consisting of five frames (10 fundamental matrices) with two outlying fundamental matrices. The correct solution is marked with a red circle.

denotes focal length):

$$\mathbf{f}^* = Z_{k^*}(j^*) \quad \text{where} \quad k^*, j^* = \arg\min_{k,j} \sum_{i=1, i\neq k}^{N} \rho(\mathrm{d}(Z_k(j), Z_i)) \tag{5.33}$$

where the optimal solution is now defined as $Z_{k^*}(j^*)$ the j^*-th point on the k^*-th zero curve having the highest score as defined by $\rho(d)$ as explained above. Also, $Z_{k^*}(j^*)$ has to be higher than zero to avoid picking one of the spurious solutions generated by the root finding method.

In addition, the distance between a point on one zero curve and another zero curve will be defined as:

$$\mathrm{d}(Z_k(j), Z_i) = Z_k(j) - Z_i(j). \tag{5.34}$$

This is a simplification that significantly reduces the computation cost and is a reasonable approximation since the zero curves are generally slowly varying and are defined on the same range of values. The search parameters in the optimization formulation above are simply the two indices that identify a particular zero curve and a specific element in its set. The advantage of this formulation is that now a global optimum to this modified problem can be found without using a nonlinear optimization routine. Since the parameter space is now reduced to a number of zero curves and the difference operation simply involves a subtraction, an exhaustive search can be performed in a reasonable amount of time to find the optimum solution. In addition, the computational complexity of this search increases linearly with the number of fundamental matrices.

An example of the application of the proposed method in finding the camera parameters of a real sequence (Rathaus [2]) is shown in Figure 5.13. The objective function for the focal length for a total of 21 fundamental matrices calculated between the frames in this sequence have been used in the optimization. The solution of a nonlinear search versus the proposed method of finding the global solution have been marked in addition to the ground truth solution. It can be seen that the proposed method finds a solution very close to the global minimum whereas the nonlinear search converges to one of the many local minima in one of the creases of the objective function.

In addition, Figure 5.14 shows the timing graph for the nonlinear exhaustive search versus a nonlinear optimization method. The timing diagram in Figure 5.14a shows that both methods increase in computation time linearly with the number of fundamental matrices. The next diagram shows the difference between the computation cost of the global reduced search versus a nonlinear optimization when the number of samples change for the zero curves. Figure 5.14b shows that when the number of samples in the zero curves exceed 370, the computation time starts to change in the favor of nonlinear minimization. However, this is considerably higher

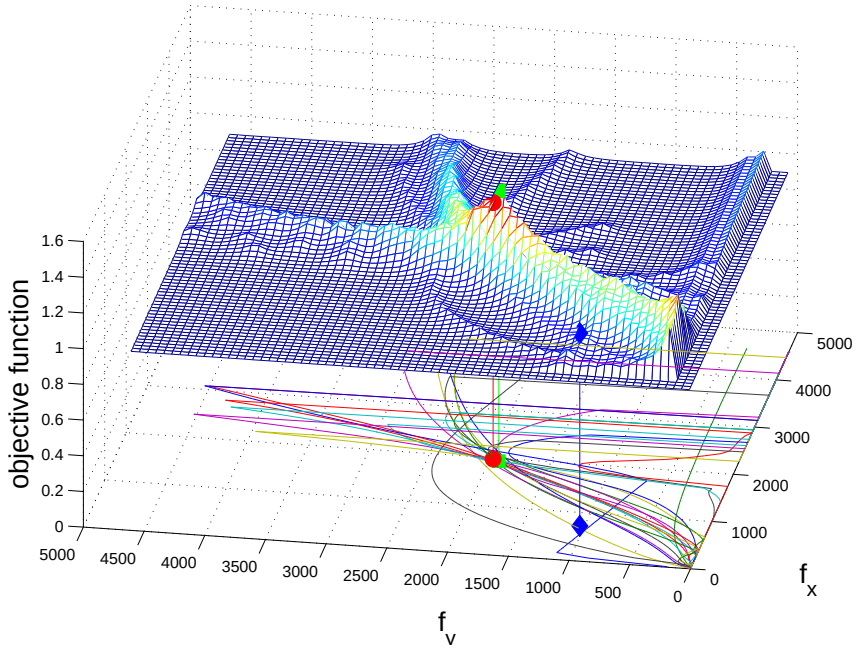
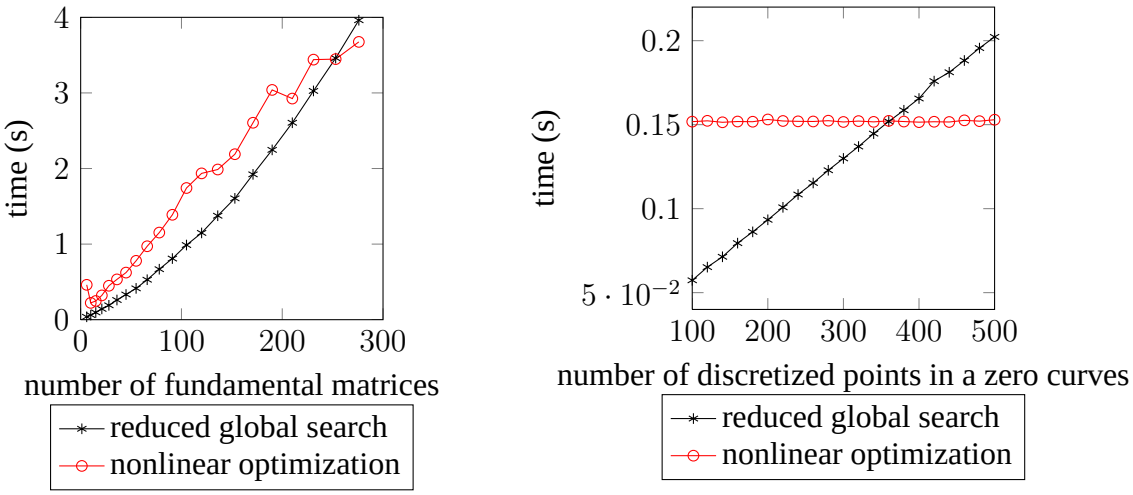


Figure 5.13: The objective function and the zero curves for the focal length estimation of the Rathaus sequence. The red circle marks the ground truth focal lengths and the green circle marks the solution of the proposed method. The blue diamond marks the result of using a nonlinear optimization routine.



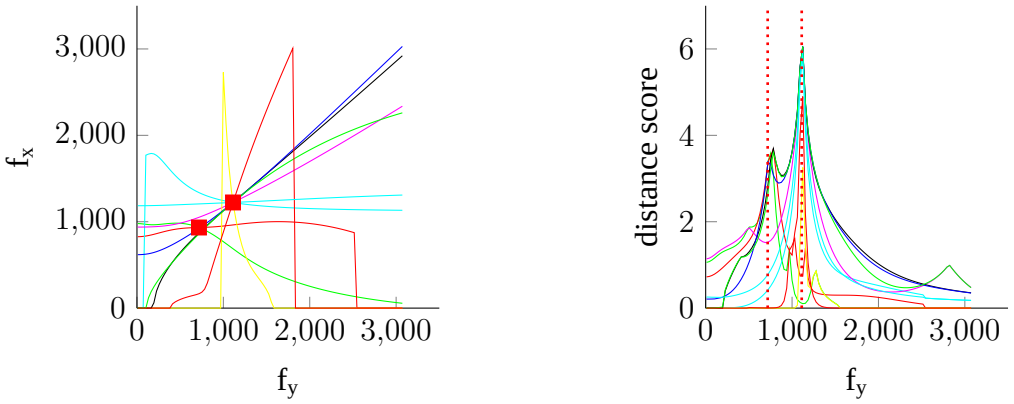
(a) Computation time vs number of fundamental matrices.

(b) Computation time vs number of samples in the zero curves.

Figure 5.14: The left diagram shows the computation time of the proposed search method and nonlinear optimization for different number of fundamental matrices. The right diagram shows the computation time of the two methods for varying number of samples of the zero curves.

than the required number of samples for a reasonable degree of accuracy in the estimated camera parameters.

Overall, the algorithm is able to find the optimal solution without resorting to a nonlinear optimization. This avoids the problem of local minima while maintaining a reasonable computation time. In addition, the algorithm can accommodate multiple acquisition devices in the scene, which is a common scenario when performing 3D reconstruction from large image databases retrieved from the Internet for a particular scene [103]. Since the solution is the cross section of the zero curves, multiple acquisition devices in the scene will simply manifest as multiple peaks in the search for points with highest scores. This would mean that the algorithm simply has to find peaks rather than the maximum in the search space. Even though this has not been implemented in the final algorithm, Figure 5.15 shows the search space and its ability to highlight multiple underlying camera parameters. Figure 5.15a simply shows the zero curves for a synthetic sequence example where two different camera parameters have been used in generating the fundamental matrices. Clearly there are two different points of intersection of the zero curves. Figure 5.15b is the resulting search space and the associated scores for the points sampled on the zero curves. All the resulting score points have been plotted with their original x-axis and the results superimposed. There are two lines marking the location



(a) zero curves for the focal length.

(b) scores for points on zero curves.

Figure 5.15: The left diagram shows the zero curves for focal length for a synthetic sequence with ten fundamental matrices formed by two different camera geometries. The ground truth solutions are marked. The right diagram shows the scores of the sample points for the same test. Vertical lines denote the location of the solution for the underlying camera focal lengths.

where the actual camera parameters would be. There are two discernable peaks along these two lines as shown.

In addition, the method offers a simple numerical method of detecting degeneracies. The issue of degeneracies is a challenging problem for all self-calibration algorithms. Even though an extensive theoretical background on this topic has been presented in [110, 111, 109, 50], simple numerical methods of detecting them have not been offered. The only work to the author's knowledge is the method of detecting the stability of the solution offered in [82] as explained in Section 5.3. The proposed method leads to a simple numerical method of deciding whether or not a zero curve is the result of a degenerate camera geometry. This is based on the observation that theoretically the objective function collapses for a degenerate configuration and so all values of the camera parameter lead to a zero in the objective function. This translates to zero curves whose values for the estimated roots amount to zero. However in other cases a noisy near-degenerate configuration can lead to a polynomial whose roots are highly uncharacteristic of the expected value of the input problem. Therefore, we use our predefined bound in Eq. 5.30 to simply find the percentage of out of bound values for the zero curves. A measure of the likelihood of the degeneracy of a particular camera arrangement is denoted by degen(Z). For instance, the degeneracy likelihood of the k-th fundamental matrix

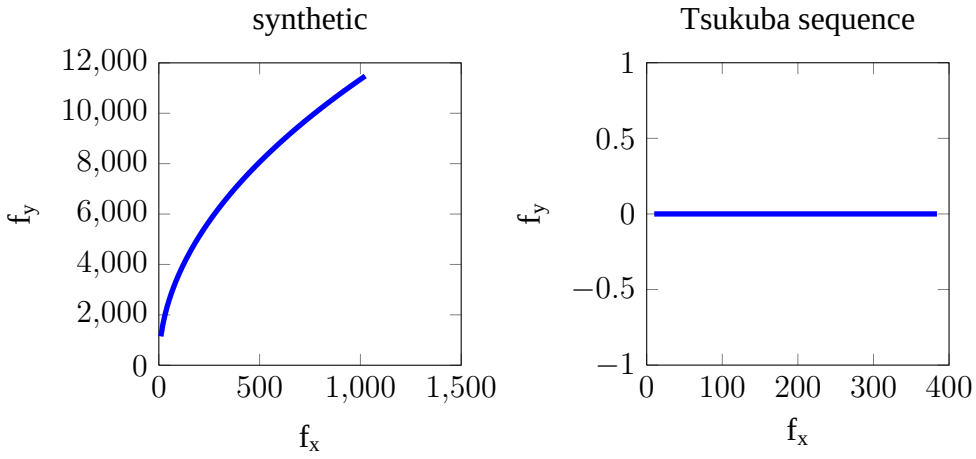


Figure 5.16: Zero curves for two fundamental matrices from two different sequences. Both cases are degenerate with respect to self-calibration. The synthetic camera pair has image size of 1024×768

is denoted by:

$$\text{degen}(Z_k^f) = \frac{|\{(f_x, f_y) \in Z_k^f | f_y \leq \epsilon \text{ or } 3w \leq f_y\}|}{|Z_k^f|}. \tag{5.35}$$

In other words, the proportion of the out of bound elements for a zero curve can be used as an indicator of the likelihood of its degeneracy. In the above formulation, only the f_y elements in each solution sample is checked since the f_x values are predetermined in the zero curves. This is due to the fact that the f_x components are swept in a predetermined interval while the f_y values are estimated. In the proposed method the threshold of this ratio is set to 50%. Zero curves that contain a higher ratio of out of bound elements, indicated by degen(Z_k^f) are simply rejected and not used in the final calculation. Figure 5.16 shows two examples of zero curves derived from camera pairs that are degenerate with respect to self-calibration. The figure on the left is a synthetic camera pair with parallel optical axis and the right is a fundamental matrix from the Tsukuba sequence [95] where images are rectified to adhere to parallel camera geometry. In the synthetic example most of the points in the zero curves exceed the threshold significantly and in the Tsukuba case the zero curve is simply collapsed to a set of zeros. As shown, the exact magnitude of the bound is not very important since most such examples either fall to zero or very large values.

Even though this method of detecting degeneracies does not necessarily identify the type of degeneracy or even discern degeneracies from cases where the root finding method simply fails for a large number of values, it offers the possibility of pruning out camera pairs from

the sequence that would adversely effect the self-calibration process. It is possible to speed up this process by using methods that find bounds for the roots of multivariate polynomial [80]; however generally estimating the zero curves is a reasonably fast operation. Algorithm 4 shows the outline of the Zero Curves algorithm.

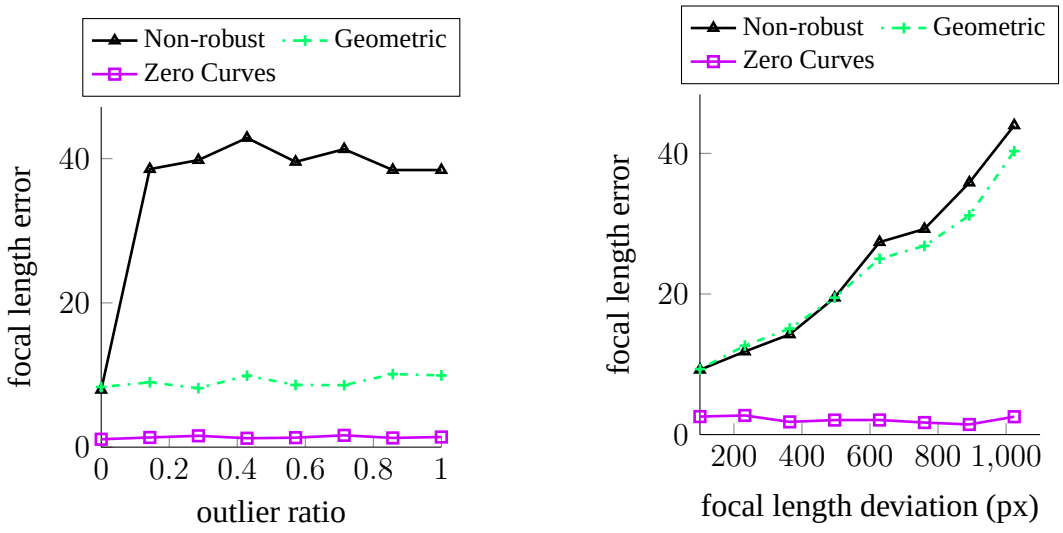
Algorithm 4 Robust self-calibration via Zero Curves

1: Estimate fundamental matrices in the sequence $F_1 \dots F_N$
2: Estimate focal length zero curves: $Z_1^f \dots Z_N^f$
3: Find focal length $\mathbf{f}^* = (f_x^*, f_y^*)$ by minimizing: $\sum_{i=1}^{N} \rho(\mathrm{d}(\mathbf{f}^*, Z_i^f))$
4: Estimate optical center zero curves: $Z_1^{oc} \dots Z_N^{oc}$
5: Find optical center $\mathbf{u}^* = (u_c^*, v_c^*)$ by minimizing: $\sum_{i=1}^{N} \rho(\mathrm{d}(\mathbf{u}^*, Z_i^{oc}))$
6: **return** $\mathbf{f}^*$ and $\mathbf{u}^*$

5.8.5 Experiments

The proposed algorithm is able to localize an accurate solution in spite of significant outliers. This is due to the fact that rather than minimizing the error of the objective function, the proposed algorithm minimizes a geometric objective function in the space of a set of hypothesized solutions for each individual fundamental matrix. These hypothesized solutions, or "zero curves" are essentially the set of roots of bivariate polynomials derived from the objective function of Huang-Faugeras constraint. These continuous curves in the parameter space ideally form an intersection where the optimal solution resides. However, due to noise such a clear intersection is often not found and so the problem is approached as a minimization of a distance score which finds the closest point to an intersection. The method is robust since only the polynomials that lead to a unique solution form a cluster in the parameter space. The noisy or degenerate objective functions often form independent segments in space and do not form a coherent cluster. As a result, when minimizing the score metric that was defined, the solution approaches the intersection of the inlying fundamental matrices and is not affected by the outliers. This method in fact has a high breakdown point, since even if the majority of the zero curves are outliers, the score is still minimum at the remaining few that form an intersection. Degenerate and erroneous cases can be discarded by defining a broad interval of expected values. This means that rather than minimizing errors in a noisy or degenerate fundamental matrix in a bundled optimization, which often leads to incorrect results, we can discard such camera pairs before the final optimization.

In order to test the algorithm, two sets of tests on synthetic sequences have been carried



(a) Performance vs varying levels of matching outliers.

(b) Performance vs focal length deviation.

Figure 5.17: Synthetic performance evaluation of the Zero Curves method compared with two standard self-calibration techniques.

out as outlined in section Section 5.5.3. Further results on real image sequences will be presented in Section 5.9. The synthetic results have been presented in Figure 5.17. The proposed algorithm is referred to in the experimental figures as the "Zero Curves" method.

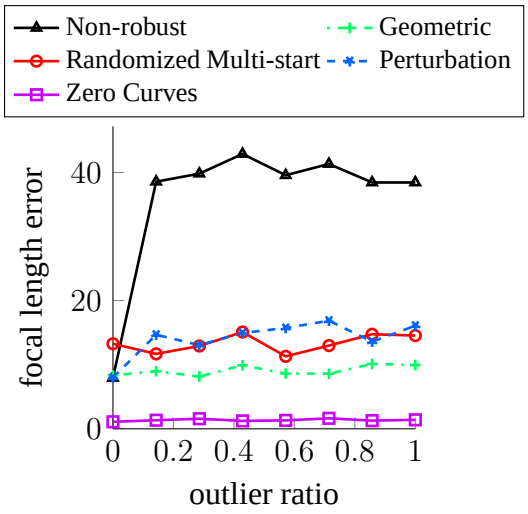
The first set of experiments shown in Figure 5.17a measures the error percentage in the estimated focal length for the proposed algorithm and the two standard benchmark algorithms used previously. As it is shown, the proposed algorithm maintains a very low level of error in the estimated focal length for all levels of noise. The second set of experiments shown in Figure 5.17b shows the percentage error of the estimated focal length while varying the difference between the focal length of one outlying frame from the rest of the focal lengths. As it is shown, the proposed algorithm is able to estimate a rather accurate set of focal lengths for all levels of the focal length deviation.

5.9 Experimental Results

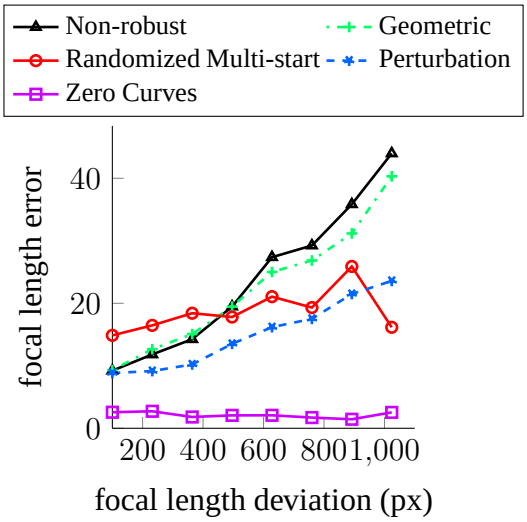
This section contains comparative results pertaining to the proposed algorithms. Before presenting the additional experiments, a complete comparison between the proposed algorithms and the two benchmark methods is presented. This is a summary of the previously used syn-

thetic experiments containing all the algorithms so far and including the estimation of the optical center. Figure 5.18a contains the experiments involving the evaluation of the estimated focal length for varying levels of matching outliers for to all the algorithms. The Zero Curves method has the strongest performance of all the proposed algorithms. The Perturbation and the Multi-start algorithms maintain a reasonable accuracy, slightly lower than the Geometric error for all outlier ratios. Figure 5.18b shows the synthetic experimental results in the case where outliers are formed by deviating the intrinsics of one of the frames in the sequence of five frames from the rest of the cameras. In this case the Perturbation method and the Zero Curves method performs the best among all the proposed algorithms. As expected, the Geometric algorithms performs as poorly as the non-robust algorithms. Overall, the zero curves and the Perturbation method maintain the strongest accuracy in the estimation of the focal length while maintaining a low computation time. In addition, in this set of experiments the optical center errors have also been included. Figure 5.18c shows the performance of the five algorithms in terms of the accuracy of the estimated optical center for varying levels of matching outliers included in the estimation of the fundamental matrix (same experiment as in Figure 5.18a). Normally one would expect the same trends in the estimation of the optical center as the focal length. However, in this diagram the Randomized Multi-start method performs worse than all other algorithms even though it has a better performance than the non-robust method and the Geometric method for the focal length. This can be explained due to the fact that the self-calibration constraints are poorly conditioned in the localization of the optical center and the Randomized Multi-start uses less constraints than all other algorithms since it uses the minimal samples. Figure 5.18d shows the errors in the estimation of the optical center. This diagram contains the results when the outliers are created by varying the focal length of a single frame from the rest as in Figure 5.18b. The optical center estimation in this case also follows the same trends as the estimation of the focal length in the same experiment, but with the except of the Randomized Multi-start which is performing poorly as explained.

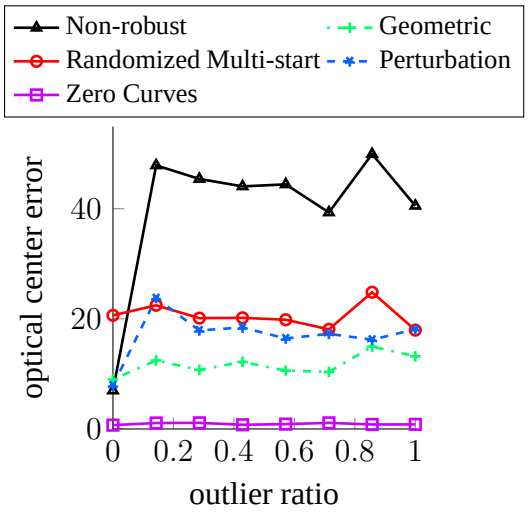
The next set of results presented in Figure 5.19 contain the performance evaluation of the self-calibration algorithms with respect to varying number of outlying fundamental matrices on real and synthetic sequences. In these experiments, the robustness of the proposed algorithms are measured with respect to varying *number* of outlying fundamental matrices each with a fixed outlier ratio in its correspondences. These outlying fundamental matrices have been estimated from a set of matches that consist of 50% outlying correspondences. In other words, for a sequence of five frames, consisting of ten fundamental matrices we increase the number of corrupted fundamental matrices in order to evaluate the performance degradation of the given algorithms. Figure 5.19a shows the synthetic experiments where the x-axis is



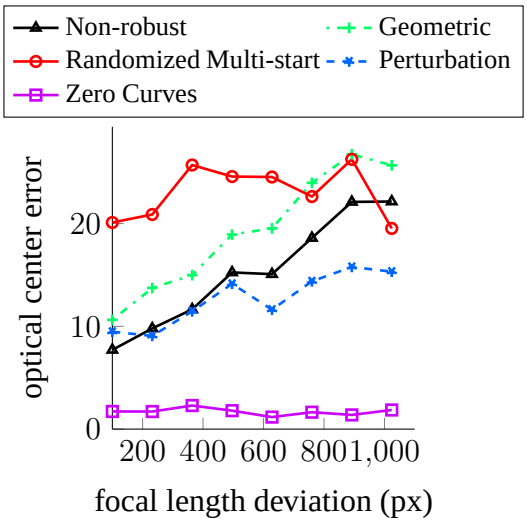
(a) Focal length percentage error vs varying levels of matching outliers.



(b) Focal length percentage error vs focal length deviation.

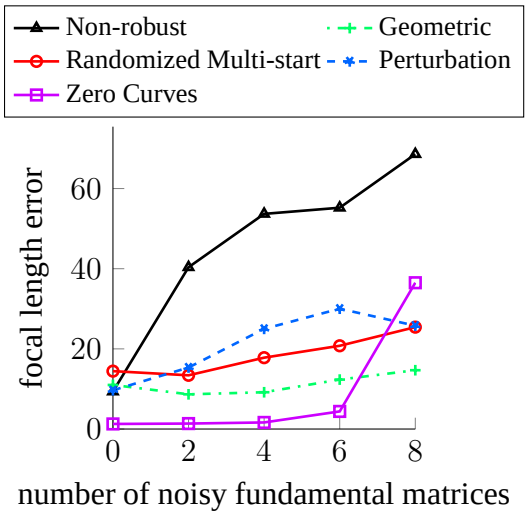


(c) Optical center percentage error vs varying levels of matching outliers.

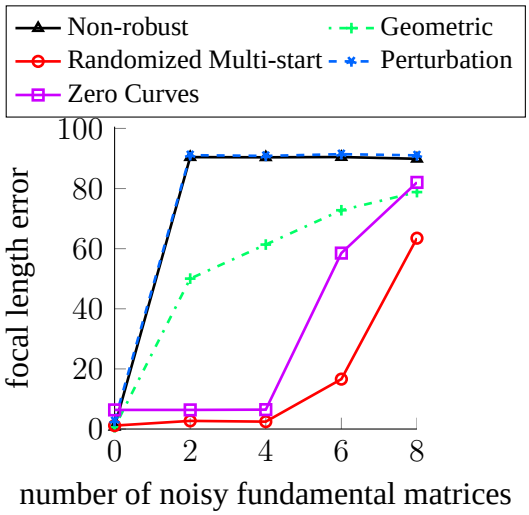


(d) Optical center percentage error vs focal length deviation.

Figure 5.18: Complete synthetic performance evaluation for assessing robustness of the three proposed self-calibration algorithms and two benchmark methods. Evaluations include focal length percentage error and optical center percentage error.



(a) Focal length error versus number of outlying fundamental matrices for a synthetic sequence of five frames.



(b) Focal length error versus number of outlying fundamental matrices for Wadham college sequence.

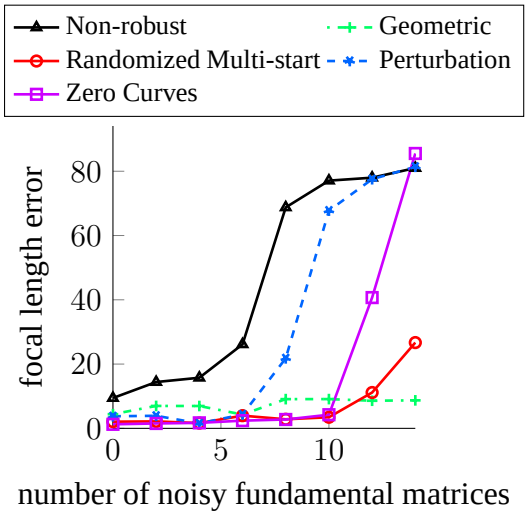
Figure 5.19: Performance comparison when the variable is the number of fundamental matrices that have been contaminated by 50% matching outlier.

the number of corrupted fundamental matrices. For all levels of noise contamination the Zero Curves algorithm performs best except for the final set of outlier ratio which consist of eight corrupted fundamental matrices out of a total of ten. The Geometric method maintains a steady performance for all levels of contamination. This is again due to the fact that the nature of the artificial noise used in this experiment is related to the type of metric that this algorithm uses in its weighted estimation scheme. The next set of experiments shown in Figure 5.19b are performed on the five frames of the Wadham College sequence [6] which contains ground truth correspondence information. During this set of tests the matching information provided with this sequence was used to estimated the fundamental matrices. In order to create the number of outlying fundamental matrices 50% of the correspondences were contaminated by outlying noise. The performance evaluations show that in this case the Randomized Multi-start optimization algorithm and the Zero Curves algorithm perform the best among other methods.

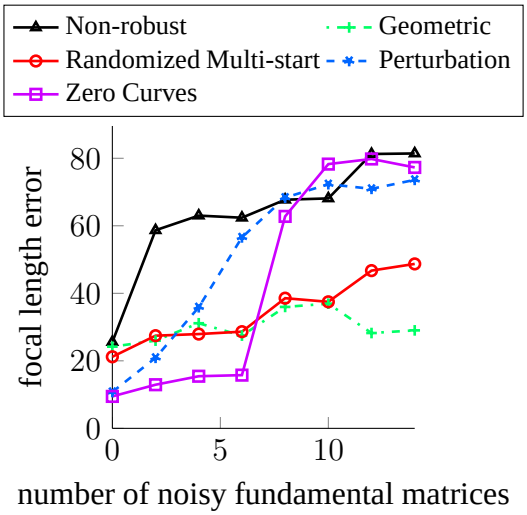
The next set of experiments shown in Figure 5.20 evaluate the robustness of the algorithms with respect to matching errors in real sequences. In this set of experiments, four real image sequences have been tested. The performance has been evaluated for increasing number of outlier fundamental matrices. Each sequence consists of seven frames (leading to 21 fundamental matrices), or has been reduced to contain only seven frames in order to create a uniform

scenario across all four sequences. The correspondence information has been found using the SIFT matcher [60]. Outlying fundamental matrices in this setup are created by including all the raw SIFT matches in its calculation. A noise-free (i.e., non-outlier) fundamental matrix in this experiment is obtained by using the robust LEV-RANSAC algorithm proposed in Section 4.7 but a fundamental matrix marked for being set an outlier is simply fit to ALL the correspondences found by SIFT, including the outliers, via the 8-point algorithm [44]. The utilized sequences are: Valbonne [6], Rathaus, Castle-P19 and Fountain-P11 sequences [108]. From the diagrams one thing that stands out is that the Perturbation algorithm performs the best when no error is deliberately injected into the data, at the lowest level of contamination on the x-axis. The Randomized Multi-start algorithm performs reasonably consistently across the outlier ratios. In other words, the Randomized Multi-start method maintains the same level of accuracy in spite of the outliers. This is often not the best level of accuracy among the methods; however outliers do not have an impact on the performance of this algorithm due to its ability to prune through outliers. However, since the inliers are not all fully taken advantage of, this consistent performance is often not comparable to the best performing algorithm. This happens to be the Zero Curves method which for reasonable contamination levels maintains a very high level of accuracy. However, when the number of contaminated fundamental matrices increases, spurious peaks can be found in the objective function used by this algorithm as explained in Section 5.8.4. The Perturbation algorithm performs poorly in moderate levels of noise, which can be attributed to the fact that the overall Error Discrepancy scale is found using a robust method with a relatively high breakdown point. Therefore, this method is best utilized at low error levels where it performs relatively better than the others. The Geometric algorithm is also relatively constant in its performance expressed as the focal length error. As explained, this algorithm performs poorly when the source of the error is other than noise in the fundamental matrix and so this algorithm does not extend to all scenarios. Furthermore, the performance of the Zero Curves method is overall stronger in lower error levels than the Geometric method.

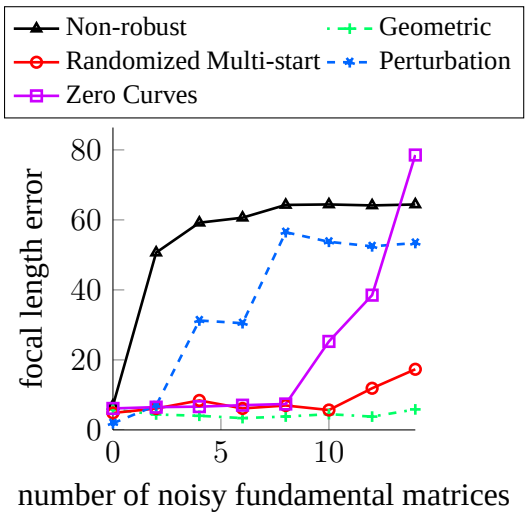
Outliers are generally assumed to contain error values that cannot be attributed to a Gaussian distribution. However, the performance of the robust algorithms have been compared when various levels of Gaussian noise are present. In the following experiment, shown in Figure 5.21 Gaussian noise has been added to the correspondences of all the frames of the synthetic sequence of five images. As before, random geometry of cameras has been assumed for each level of noise and the results averaged over 100 experiments. However, unlike the previous cases the amount of noise is normally distributed with the varying standard deviations as shown on the x-axis. In this case, none of the fundamental matrices are outliers per



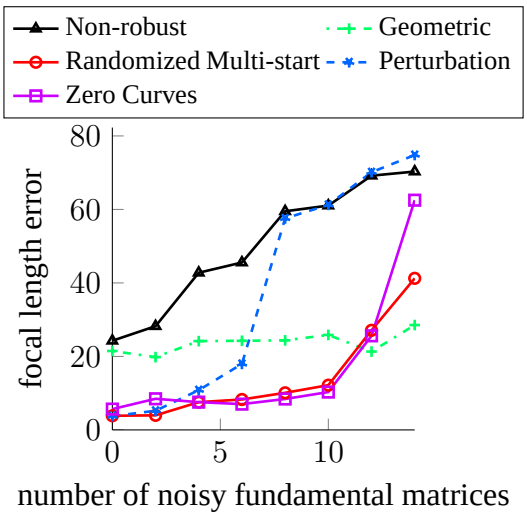
(a) Rathaus sequence.



(b) Valbonne sequence.

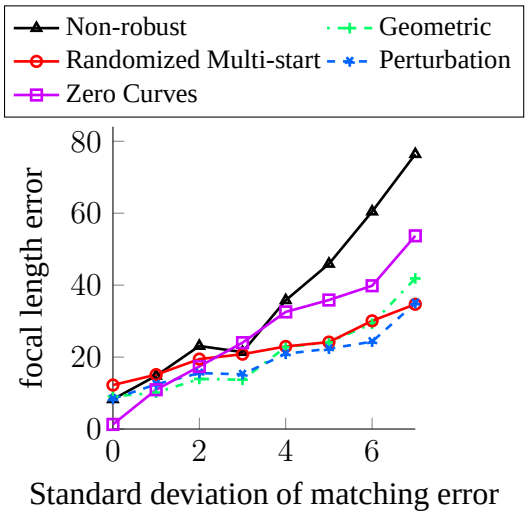


(c) Castle-P19 sequence.

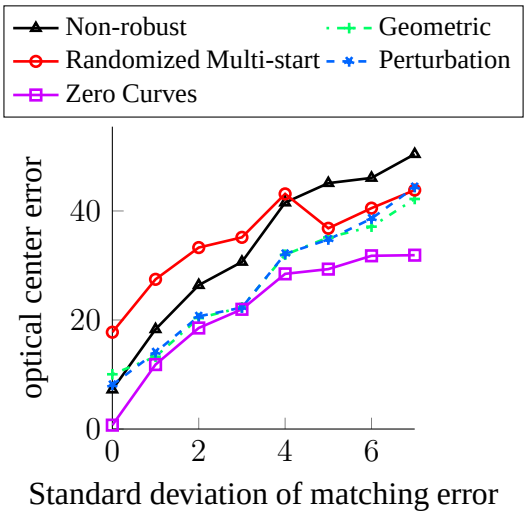


(d) Fountain-P11 sequence.

Figure 5.20: This experiment measures the accuracy of the estimated focal lengths for different number of fundamental matrices that have been estimated to raw image correspondences without the use of a robust fundamental matrix estimator and are therefore outliers. The x-axis shows the number of such erroneously calculated fundamental matrices. All sequences contain seven images and 21 fundamental matrices.



(a) Focal length estimation error versus standard deviation of error added to all correspondence data.

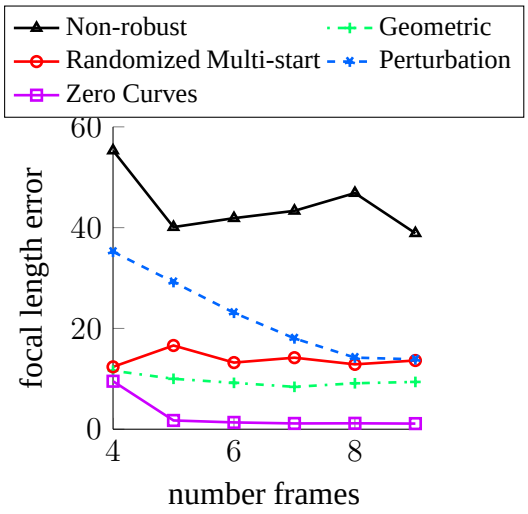


(b) Optical center estimation error versus standard deviation of error added to all correspondence data.

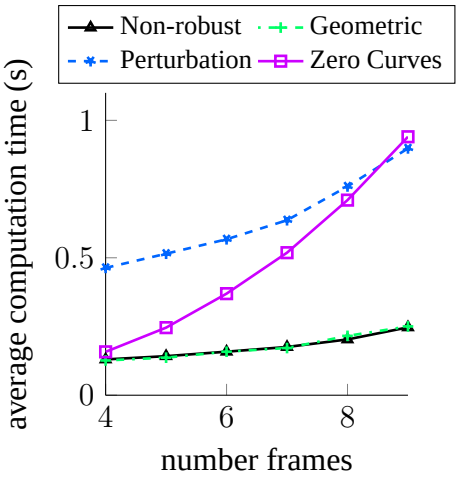
Figure 5.21: Synthetic evaluation of self-calibration with respect to normally distributed noise added to all the image correspondences used to calculate the fundamental matrix.

se and so one expects the non-robust algorithm to perform the best. Although the differences between the performances are not very significant, the Perturbation algorithm performs better than other algorithms for higher levels of noise level. This is due to the fact that the Error Discrepancy measure defined in this algorithm is able to prune out noisy fundamental matrices, even when the noise is small. However, the Zero Curves algorithm performs poorly in this case, most likely due to the fact that at higher levels of normally distributed noise the localization of the convergence point gets less precise. The surprising results are the results for the error in the estimation of the optical center shown in Figure 5.21b. In this diagram the Zero Curves method achieves the highest accuracy in the estimation of the optical center across all levels of normally distrusted noise. This is very likely due to the fact that a different mechanism is used for finding the solutions which might be more suitable for finding the optical center. As mentioned, the estimation of the optical center is inherently more difficult than the focal length when using typical gradient based optimization methods.

The next set of experiments examine the scalability of the given algorithms in terms of performance and computation time, shown in Figure 5.22. Figure 5.22a shows the accuracy of the estimated focal lengths for increasing number of frames when one frame contains outliers. In this set of experiments, the image features of a single frame have been corrupted by noise



(a) Focal length error versus number of frames.



(b) Computation time versus number of number of frames.

Figure 5.22: Performance and computation time comparison for increasing number of frames.

and so all of the fundamental matrices involving this frame are contaminated. For example, if there are five frames, four fundamental matrices relating the noisy frame to others will be contaminated by 50% outliers. It can be observed that the accuracy of all robust algorithms slowly increases with the number of frames for most methods since there are more constraints available as a percentage of the total number of fundamental matrices increases. Figure 5.22b shows the computation time for self-calibration based on varying number of frames with no outliers. The Randomized Multi-start algorithm has been removed since it is significantly more time consuming than the other methods. In fact, it is ten times more computationally costly on average than Zero Curves method.

The final set of presented experiments is the set of results of all the algorithms as tested on a set of benchmark sequences shown in Table 5.2. This table contains the estimated results of self-calibration for our compared method, whereas the errors for these experiments is provided in the accompanying Table 5.3. The fundamental matrices of these algorithms are estimated via the LEV-RANSAC method proposed in Section 4.7 from matching data found using the SIFT matcher. No artificial noise has been added to the fundamental matrices of these sequences. The experiments have been repeated 50 times for each sequence and the results averaged over all the trials. Due to space limitations, the Zero Curves method is abbreviated to “ZC”, the Randomized Multi-Start method to “RMS” and the Perturbation method to “PBW”. Averaging over all the focal length errors for all the sequences, the Zero Curves method achieves

the highest accuracy by a 14.8% error in the estimation of the focal length. The second most accurate method is the Randomized Multi-start method with a 17.1% accuracy and the third is the Perturbation method with a 20.1% accuracy level. The worst performing algorithms are the Geometric method with a 25% error in focal length estimation and the non-robust with 29%. However, the results for the optical center indicate that the Perturbation algorithm performs best with 16% error and the rest of the algorithms average in a close range between 20% and 27%. However, since the estimation of the focal length is of primary interest, it is clear that the Zero Curves algorithms has the highest performance rate. The Randomized Multi-start algorithm can not be recommended in a typical SfM scenario due to its high computation time. The Perturbation theory, while moderately accurate still lags behind the Zero Curves method. In terms of a weighted optimization framework; however, it is superior to the Geometric method since the weights rely on the properties of each fundamental matrix as it related to self-calibration. On a final note, the result with respect to the Corridor sequence are quite poor for all the given algorithms. This is due to the fact that this sequence is nearly degenerate due to the motion of the camera being purely translational (inside and sideways out of the corridor).

5.10 Discussion

Three robust self-calibration algorithms have been proposed and their performance tested against a number of synthetic and real image sequences. These tests included robustness assessment with respect to varying levels of outliers in the correspondences used to estimate the fundamental matrix, Gaussian noise, number of noisy fundamental matrices, changing frame parameters with respect to the sequence and also computational time evaluation. The first algorithm, the Randomized Multi-start algorithm is a computationally costly search in the space of parameters and in the input fundamental matrices. The algorithm is able to overcome local minima and outliers at the expense of large computation time. Also, since the self-calibration constraints are only enforced on minimal sets of fundamental matrices, very accurate results are not obtained. When off-line self-calibration is the context, this algorithm is more applicable due to its high computation time. The second algorithm uses perturbation theory to obtain a set of weights for the fundamental matrices at each iteration. The weights are based on the likelihood of a fundamental matrix to be an outlier based on some measure of how much its error in the objective function deviates from some expected threshold derived from perturbation theory. This algorithm performs moderately well at a lower computation time than the Randomized Multi-start algorithm. As demonstrated, the performance of the Perturbation al-

gorithm is superior to that of the Geometric algorithm for the real image sequences. Also, no information about the correspondences are needed in order to perform this iterative weighted optimization. The last algorithm proposed is the Zero Curves algorithm which uses a geometric optimization in the space of a set of zero curves of the polynomials derived from the objective functions based on the fundamental matrices. This algorithm is computationally feasible in addition to being accurate for real scenarios. Also, in synthetic experiments it provides very accurate self-calibration results and is robust against outlying fundamental matrices. Also it avoids local minima in the objective function since a global search is carried out.

Table 5.2: Self-calibration comparison results on real image sequences. Best results based on focal length accuracy are highlighted. The ground truth intrinsic parameters provided along with the image sequences are shown in the "quantity" column.

sequence	quantity	ZC	RMS	non-robust	Geometric	PBW
Corridor 	$f_x = 495.23$	333.39	462.72	959.20	824.67	663.76
	$f_y = 496.92$	669.34	600.30	1266.36	1151.28	1038.77
	$u_c = 272.50$	207.76	253.98	335.89	304.05	272.99
	$v_c = 279.98$	245.35	213.23	257.61	263.78	281.79
Wadham 	$f_x = 1150.86$	1101.80	1150.35	303.53	447.42	284.72
	$f_y = 1138.18$	1111.36	1270.83	968.12	1003.73	1042.63
	$u_c = 534.65$	296.05	542.16	622.05	607.41	622.21
	$v_c = 388.79$	550.29	324.97	392.71	359.97	399.17
Merton1 	$f_x = 1116.02$	1279.20	1123.73	1139.52	1129.69	1137.04
	$f_y = 1101.77$	477.06	1333.98	1327.88	1318.21	1353.69
	$u_c = 435.65$	441.46	519.74	535.55	531.51	537.32
	$v_c = 351.58$	436.43	464.47	458.22	458.64	447.74
Merton2 	$f_x = 1161.37$	919.32	1061.49	1051.36	1032.73	1037.69
	$f_y = 1167.33$	1017.76	1186.75	1227.04	1262.23	1211.61
	$u_c = 466.88$	588.68	537.95	526.04	526.86	526.25
	$v_c = 347.78$	476.12	373.39	374.94	329.06	392.02
Merton3 	$f_x = 1022.95$	1032.52	1101.30	1104.87	1107.70	1110.84
	$f_y = 1036.60$	998.97	1466.46	1033.45	1065.33	1051.35
	$u_c = 542.92$	116.43	404.33	442.77	444.63	438.88
	$v_c = 401.18$	371.34	220.84	380.84	370.49	382.16
Rathaus 	$f_x = 1977.35$	2126.01	1703.68	1562.46	1661.46	1981.87
	$f_y = 1974.00$	2022.65	1994.85	1758.98	1819.64	1945.61
	$u_c = 809.95$	869.75	768.31	713.15	744.86	827.40
	$v_c = 575.70$	573.77	553.18	304.69	327.63	573.66
Valbonne 	$f_x = 681.35$	691.00	468.05	360.06	380.17	524.13
	$f_y = 679.28$	660.90	516.31	554.03	561.66	654.66
	$u_c = 258.80$	491.60	193.70	203.87	211.97	168.18
	$v_c = 383.19$	483.47	375.21	82.43	100.08	268.43
Castle-P19 	$f_x = 1379.74$	1382.90	1318.75	1326.46	1339.04	1350.21
	$f_y = 1382.08$	1362.81	1352.31	1370.12	1374.93	1390.88
	$u_c = 760.35$	750.94	775.07	702.76	714.21	738.70
	$v_c = 503.40$	515.13	494.11	495.44	499.28	495.11
Fountain-P11 	$f_x = 1379.74$	1224.99	1375.76	283.56	344.97	362.87
	$f_y = 1382.08$	1364.75	1361.73	1395.35	1397.46	1385.33
	$u_c = 760.35$	746.02	740.90	907.14	898.23	876.34
	$v_c = 503.40$	560.05	536.45	582.97	601.48	659.75

Table 5.3: Self-calibration percentage errors on real image sequences. Best results based on focal length accuracy are highlighted.

sequence	quantity	ZC	RMS	non-robust	Geometric	PBW
Corridor	e_{focal}	32.98	39.64	109.62	92.65	68.14
	e_{OC}	19.64	25.11	28.16	16.06	10.20
Wadham	e_{focal}	15.06	11.34	44.18	36.37	43.96
	e_{OC}	46.96	19.88	12.91	12.66	13.97
Merton1	e_{focal}	47.08	15.62	12.95	11.52	13.21
	e_{OC}	18.67	24.70	24.91	24.51	23.65
Merton2	e_{focal}	19.87	13.64	10.97	10.65	10.70
	e_{OC}	46.65	30.03	31.22	23.75	31.74
Merton3	e_{focal}	4.64	25.82	7.24	6.13	7.62
	e_{OC}	44.13	38.30	13.52	14.22	15.16
Rathaus	e_{focal}	4.99	7.74	16.49	12.44	1.67
	e_{OC}	6.44	7.63	32.48	27.59	3.71
Valbonne	e_{focal}	11.11	33.13	32.93	30.77	14.38
	e_{OC}	70.59	33.16	49.87	45.98	33.65
Castle-P19	e_{focal}	1.99	9.80	3.92	2.76	1.84
	e_{OC}	2.13	20.26	6.29	4.87	2.87
Fountain-P11	e_{focal}	6.28	5.01	40.34	38.17	37.38
	e_{OC}	9.26	6.52	18.28	18.87	23.74

Chapter 6

Conclusion

The goal of this thesis has been to focus on improving the robustness of two of the most important stages in SfM, namely fundamental matrix estimation and self-calibration. In addition to being of utmost important to the SfM tool-chain, these two areas of computer vision are of use to many other problems. The introduction in Chapter 1 presented the challenges of visualizing the world through images and the inadequacies of existing methods, thus providing the motivation behind improving the robustness in the uncalibrated SfM pipeline. Overcoming some of the challenges of working towards a general and robust uncalibrated SfM framework were then presented as the goals of the thesis. The following provides a summary of the thesis and the algorithms proposed.

6.1 Summary

The thesis begins with an introduction to the field of 3D vision and the SfM pipeline. The issue of robustness is then discussed along with its importance in 3D reconstruction. Once the motivation for robust methods in the field is presented, the importance of the fundamental matrix in the field of 3D vision is outlined. The problem of robustness in the estimation of the fundamental matrix is then discussed followed by an introduction to self-calibration. It is then argued that robustness in self-calibration is essential for a real-world scenario. The thesis then proceeds by providing two background chapters on 3D vision and robust statistics. Following this, the first set of contributions in robust estimation of the fundamental matrix are detailed in Chapter 4. Subsequently the second set of contributions are outlined in Chapter 5 where robust self-calibration is discussed. A summary of the above chapters will now be presented.

Chapter 2 contains the preliminary ideas from the field of 3D vision that set the terminology

and the background for the thesis. This chapter starts by describing the pinhole camera model which is used throughout this thesis. Following this, multiple-view relations are described and their utility in the SfM is provided. These include the homography, the essential matrix and the fundamental matrix. The different levels of structure recovery from images are also enumerated: projective, affine and Euclidean and metric reconstruction. Camera calibration is also explained and several general methods of approaching this problem are stated. The preliminary ideas for self-calibration such as the absolute conic and quadric are also explained. The chapter ends with a brief overview of the topic of degeneracy.

Chapter 3 contains the preliminary ideas used from the area of robust statistics. This chapter begins by detailing the problem of linear regression and the issue of noise. Then the idea of robust regression is presented accompanied by several experiments showing the importance of robustness of parameter estimation and the effects of even small amounts of outliers on this process. The problem of noise is then examined and different types of noise are discussed. Subsequently, the issue of influence is presented along with the measure of leverage which is used later in developing a robust sampling method. Following this, RANSAC and M-estimators are presented as two general approaches to the problem of robust estimation.

Chapter 4 presents the first portion of the contributions, namely, robust estimation of the fundamental matrix. The problem of estimating the fundamental matrix from correspondences that can often contain a large number of outliers is reviewed. Various metrics for assessing the quality of the fundamental matrix are also provided. Following this, several categories of the innovations in this field are presented. Each category includes several recent methods in the literature which are reviewed and analyzed. It is pointed out that many of the methods that improve the sampling of RANSAC are not general and fail in the presence of local motions. Two algorithms were then proposed to alleviate these issues. The first algorithm, RES-RANSAC was described in Section 4.6, it relies on the analysis of the residuals and the idea that inlier residuals cluster together and are normally lower in magnitude than outlier residuals. Using this idea a guided sampling method is adopted to improve the robustness and the accuracy and speed of the RANSAC process. This work was published in the 2008 proceedings of the International Conference on Pattern Recognition [90]. An algorithm devised based on a similar residuals-based strategy as RES-RANSAC but using not just the last residual update, but all previous residuals is published in the 2009 proceedings of the International Symposium on Visual Computing [129]. Also a US Patent based on this strategy was filed [91]. The second algorithm in this chapter, LEV-RANSAC was introduced in Section 4.7. This algorithm also relies on residual analysis to steer the sampling process towards inliers. However, in this algorithm a superior quantity than pure raw residuals is utilized. Regression diagnostics in-

formation is derived from the iterations of RANSAC which are then used to estimate the *a priori* information for guiding the sampling. This work was published in the 2012 proceedings of the International Symposium on Visual Computing [89]. The experimental results are then presented comparing the performance of the two proposed algorithms with existing methods.

Chapter 5 presented the second set of contributed works in the context of self-calibration. The challenging problem of calibrating image sequences from purely image-based data was presented along with the challenges facing existing self-calibration methods. Several of the most important self-calibration constraints are then presented. It is then argued that the Huang-Faugeras constraint is the most suitable for use in a robust and general-purpose self-calibration context. Following this, the sensitivity of self-calibration to outliers was demonstrated using experimental results. Three methods were then proposed for improving the robustness of existing self-calibration methods. The first method used a guided sampling of the search space in order to avoid local minima, outlined in Section 5.6. This method was initially published in 2010 proceedings of the International Symposium on Visual Computing [88]. The second method proposed relies on perturbation theory to find an approximation for the upper bound for errors derived from the singular values of the essential matrix. This is then utilized to perform a weighted nonlinear optimization where the weights are derived from an Error Discrepancy measure based on the bounds for the error values. The last proposed algorithm for robust self-calibration is based on a geometric optimization where a solution is found based on its level of consensus from a number of zero curves. These zero curves are defined as the families of solutions derived from individual fundamental matrices and are effectively the set of roots for the resulting multivariate polynomial derived from each fundamental matrix. Finally experimental results based on synthetic and real images are presented and the proposed methods are compared and analyzed. The Perturbation algorithm and the Zero Curves method are currently in the process of being prepared for submission to a relevant journal.

6.2 Contributions of the Thesis

This thesis has achieved several contributions towards the goal of robust 3D vision. The contributions have been focused on the important areas of fundamental matrix estimation and self-calibration. Below a summary of the contributions is presented.

- RES-RANSAC which is a novel algorithm was proposed for improving the sampling for RANSAC. This algorithm shows that the residuals from the iterations of RANSAC can be used for more than merely measuring the level of consensus. From analysis of the

residuals, it is inferred that residuals for inliers are lower in magnitude regardless of the quality of the models, and that they tend to cluster together. Both of these ideas are utilized to improve the sampling of RANSAC via an algorithm that estimates *a priori*s for the correspondence data. This algorithm reduces the error of the estimated fundamental matrix by 9% compared to MLESAC and reduces the required number of iterations by an average of 16%.

- The LEV-RANSAC method was developed where regression diagnostics are incorporated in the sampling process of RANSAC. This is an algorithm that brings ideas from the field of robust statistics into the domain of computer vision. Using Cook's distance which is a well known quantity in the field of robust statistics, validity information is derived from the residuals of the RANSAC process. This is then used as an *a priori* measure that guides the sampling process. Also the stopping criterion of RANSAC is modified to depend on the convergence of the validity measures. The experimental results show that this method performs as much as seven times faster than traditional RANSAC while maintaining a higher accuracy in the estimate of the fundamental matrix. Note that this strategy is applicable to any model being estimated using the RANSAC framework including homography estimation and essential matrix estimation in calibrated sequences.

- A Randomized Multi-Start Optimization algorithm was presented to explore the self-calibration constraints in scenarios where noise is a major factor. This algorithm is based on using a distribution over the parameter space that improves the sampling of these parameters. This scheme also contains sampling of the fundamental matrices in order to reduce the impact of noise. We combined sampling strategies with the Kruppa equation and a distribution based on its solutions to help infer calibration data. The resulting algorithm can estimate the camera intrinsic parameters up to 60% more accurately than the non-robust method when the source of the error is poor estimation of the fundamental matrix and up to 50% more accurately than the non-robust method and the Geometric method in higher error ratios when the source of the error is incorrect assumptions over the camera parameters.

- A novel algorithm was presented which combines perturbation theory to create a bound on the errors from the self-calibration constraints on the essential matrix. Using these a weighted optimization strategy is devised that is robust and time efficient. The proposed algorithm achieves up to 60% more accuracy than the non-robust method when

the source of the error is noise in the estimation of the fundamental matrix and up to 50% higher accuracy when the source of error is incorrect assumptions about camera parameters.

- A robust strategy was presented for finding solutions to sets of multivariate polynomials. This is combined with self-calibration constraints to produce a robust method that is highly efficient and accurate. This method can also be applied to other applications where solutions of systems of polynomial equations are to be estimated. This algorithm is as much as 12 times more accurate than the non-robust method and three times more accurate than the robust Geometric method when the source of the error is the noise in the fundamental matrix. When the source of the errors is incorrect assumptions on the camera parameters, the Zero Curves method performs as much as 20 times more accurately than the competing methods. In addition, this method can potentially discern multiple acquisition devices in the self-calibration process. Degeneracy detection is another venue explored using this method and results show that degeneracy detection can be feasible using the proposed approach. The Zero Curves method only uses an eigen decomposition function for finding the roots of a polynomial and so unlike previous methods does not rely on a nonlinear optimization technique.

In summary, two areas have been addressed and improved upon during this work. The important problem of motion estimation has been addressed and two method that improve both the speed and the accuracy of the process have been presented. The method based on regression diagnostics has the ability to maintain a nearly constant computation-time over a long range of outlier ratios. This is a significant improvement over existing methods since their accuracy and speed degrade exponentially at higher outlier ratios. This improvement can have significant benefits for real-time vision applications where an upper bound on the computation time is required. Also the issue of the robustness of self-calibration has been addressed. Even though self-calibration has existed in the literature for a long time, most algorithms based on self-calibration are not reliable due to their high noise sensitivity. In order to address this, three algorithms have been presented that are able to overcome this shortcoming of existing self-calibration techniques. These algorithms provide solutions where a non-robust self-calibration algorithm completely fails.

6.3 Future Work

Due to its central place in many computer vision problems, the estimation of the fundamental matrix is of great importance. In this thesis we have provided various insights on the residuals derived during the iterations of RANSAC and how they can be used to derive the validity information. The properties of these residuals can be further investigated in order to devise more rigorous validity data. Also using the "history" of a point's residuals should be incorporated into this measure of validity. In addition, analysis of residuals can be used in order to not only discern inliers, but also the presence of local motion in the scene. Preliminary tests have shown that the histograms of scenes with local motions contain multiple peaks that can be used to perhaps estimate multiple motions at once or at least infer validity information for the independent motions during a single run of RANSAC to be used in subsequent iterations to estimate the local motions. The field of robust statistics has also be applied to robust estimation of the fundamental matrix. Additional ideas can be used in order to refine the validity measures. Regression diagnostics can be further used to infer degenerate configurations. Also the regression diagnostics information can be used as *a priori* information not just in the sampling process, but also in the reconstruction process and during bundle adjustments. Weighing certain points according to the level of their confidence that is estimated during LEV-RANSAC can be helpful in improving the final results of bundle adjustment. Also the applications of the two proposed motion estimation methods can be tested on the homography estimation.

With respect to self-calibration, robustness remains at very early stages. As a result, several directions exist within this research, for instance, the improvement of the estimation of the optical center. This has remained a difficult problem regardless of the constraints used for self-calibration. In addition, better modeling of the position of the optical center can be carried out. In our proposed sampling algorithm the deviation of the optical center from the image center is modeled as a Gaussian. This can be further refined using knowledge from lens manufacturing or by compiling specs from a large number of cameras. Finally, the Zero Curves method can be investigated as a general purpose solution estimator for large systems of equations. Detecting degeneracies is also another venue that can be explored further using the Zero Curves method. The experimental results showed that we were able to detect degeneracies for a simple example of a degenerate camera motion, however more complicated cases can also be examined.

One important idea that has not been explored during this thesis is the idea of detecting key-frames. Key-frame detection is finding reliable frames from within a sequence of images to initiate the SfM process from. Knowledge of the intrinsic parameters of this initial pair of key-frames means that we can find the calibration parameters of the rest of the images in the

sequence via resectioning [44]. Therefore, using the confidence in the estimation of the camera parameters during self-calibration can provide a good clue as to which frames to use to initiate the SfM process. For instance, using the Perturbation method one can use the estimated camera parameters during self-calibration to initiate the SfM process from the pair of frames having the highest weights. Similarly, the Zero Curves methods provides a measure of consensus for the estimated solution. This measure of consensus can be used to decide which frames have been calibrated with the highest level of confidence and then in turn used as key-frames in SfM. Although there are other important aspects of choosing key-frames [113], the above self-calibration criterion can be incorporated into existing methods that aim to find the most stable key-frames to start SfM from.

Appendix A

List of Symbols

C Conic

C^* Dual conic

$\mathbb{P}^3$ 3D space

$\mathbb{P}^2$ 2D space

$\mathbf{X}$ Point in 3D space

$\mathbf{X}_i$ i-th point in 3D space

$\mathbf{x}$ Point in 2D space (e.g., image plane)

$\mathbf{x}_i$ i-th point in image

$\mathbf{x'}_i$ i-th point in second image

$\mathbf{x} \leftrightarrow \mathbf{x}'$ Correspondence between two image points in two different views

H Homography

T Transformation matrix

$\boldsymbol{\pi_\infty}$ Plane at infinity

Ω_∞ Absolute conic

I Identity matrix

Q^*_∞ Absolute dual quadric

R Rotation matrix

$\mathbf{t}$ Translation vector

K Intrinsic camera parameters, or intrinsics matrix, or camera optical matrix

$\mathbf{C}$ Camera center

f_x Focal length in the x-direction

f_y Focal length in the y-direction

$\mathbf{f}$ Vector representation of the x and y components of the focal length, i.e., $\mathbf{f} = (f_x, f_y)$

f Focal length when the aspect ratio is unity (i.e. $f = f_x = f_y$)

u_c Optical center, (x-component)

v_c Optical center, (y-component)

OC Optical center

P Projection matrix

V Vanishing point

F Fundamental matrix

F_i i-th fundamental matrix

$\mathbf{e}$ Epipole

$\mathbf{e}'$ Epipole of second view

$[\mathbf{t}]_\times$ Skew symmetric matrix of a vector $\mathbf{t}$

E Essential matrix

$d(\mathbf{x}, \mathbf{y})$ Distance between $\mathbf{x}$ and $\mathbf{y}$

r_i Residual of the i-th data point

d_i Studentized residual of the i-th data point

h_{ii} or h_i Leverage of the i-th data point

D_i Cook's distance of the i-th data point

σ Standard deviation

$p(v_i)$ Probability that the i-th data point is an inlier

N Number of data points

$p(v_i^{(j)})$ Probability estimate during iteration j that the i-th data point is an inlier

$p(v_i^*)$ Probability of data point i-th being inlier estimated during the last iteration where a best model was found

$M*$ Best model found so far during the RANSAC iterations

M_j Model found at iteration j during the RANSAC iterations

$\widetilde{A}$ The estimated value of a matrix A

Δ_A The error in the estimate of matrix A

$\Delta_A^{(j)}$ Error in the estimate of matrix A during the j-th iteration

σ_i i-th singular value

$\|A\|_2$ Euclidean norm of matrix A

$\|A\|_F$ or $\|A\|$ Frobenius norm of matrix A

$ED_i^{(j)}$ Error Discrepancy measure during iteration j for the i-th data point

w_i Weight for data point i

Z_i Zero curve, or the connected components defining the zeros of a multivariate polynomial of the objective function, derived from the i-the fundamental matrix F_i

Z_i^f Zero curves for the focal length

Z_i^{oc} Zero curves for the optical center

Bibliography

[1] Autostitch. [online]. http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html. Accessed: 2012-09-30.

[2] EPFL CVLAB dataset. [online]. Available: http://cvlab.epfl.ch/data/strechamvs/. Accessed: 2012-09-30.

[3] KLA-Tencor. [online]. http://www.kla-tencor.com/. Accessed: 2012-09-30.

[4] Microsoft Bing Map. [online]. http://www.bing.com/maps/. Accessed: 2012-09-30.

[5] Microsoft kinect. [online]. Available: http://www.xbox.com/en-us/kinect/. Accessed: 2013-01-30.

[6] Oxford visual geometry group dataset. [online]. Available: http://www.robots.ox.ac.uk/~vgg/data/. Accessed: 2012-09-30.

[7] Sift keypoint detector. [online]. http://www.cs.ubc.ca/~lowe/keypoints/. Accessed: 2012-09-30.

[8] L. Agapito, E. Hayman, and I. Reid. Self-calibration of rotating and zooming cameras. *International Journal of Computer Vision*, 45(2):107–127, 2001.

[9] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski. Building rome in a day. In *IEEE International Conference on Computer Vision (ICCV)*, pages 72–79, 2009.

[10] I.A. Ajwa, Z. Liu, and P.S. Wang. Gröbner bases algorithm. Technical Report ICM-199502-00, Kent State University, Kent, Ohio, 1995.

[11] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3D reconstruction of buildings from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, pages 69–80, 1999.

[12] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.

[13] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *IEEE International Conference on Computer Vision (ICCV)*, pages 790–796, 1998.

[14] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[15] D. Capel. An effective bail-out test for RANSAC consensus scoring. In *British Machine Vision Conference (BMVC)*, 2005.

[16] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[17] S. Chen. QuickTime VR - an image-based approach to virtual environment navigation. In *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 29–38, 1995.

[18] T.J. Chin, J. Yu, and D. Suter. Accelerated hypothesis generation for multi-structure robust fitting. In *European Conference on Computer Vision (ECCV)*, pages 533–546, 2010.

[19] O. Chum and J. Matas. Matching with PROSAC - progressive sample consensus. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 220–226, 2005.

[20] O. Chum and J. Matas. Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, 2008.

[21] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. *Pattern Recognition*, 2781:236–243, 2003.

[22] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. In *British Machine Vision Conference (BMVC)*, 1999.

[23] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[24] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pages 392–407, 2000.

[25] G. Csurka, C. Zeller, Z. Zhang, and O.D. Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding*, 68(1):18–36, 1997.

[26] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[27] A. Edelman and H. Murakami. Polynomial roots from companion matrix eigenvalues. *Mathematics of Computation*, 64(210):763–776, 1995.

[28] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *European Conference on Computer Vision (ECCV)*, pages 563–578, 1992.

[29] O.D. Faugeras. *Three-dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[30] O.D. Faugeras, Q.-T. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision (ECCV)*, pages 321–334, 1992.

[31] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[32] M. Floater. On zero curves of bivariate polynomials. *Advances in Computational Mathematics*, 5(1):399–415, 1996.

[33] D.A. Forsyth and J. Ponce. *Computer Vision: a Modern Approach*. Prentice Hall, 2002.

[34] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *European Conference on Computer Vision (ECCV)*, pages 368–381, 2010.

[35] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[36] A. Fusiello, A. Benedetti, M. Farenzena, and A. Busti. Globally convergent autocalibration using Interval Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1633–1638, 2004.

[37] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[38] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The Lumigraph. In *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 43–54, 1996.

[39] A. Habed and B. Boufama. Camera self-calibration from bivariate polynomials derived from Kruppa's equations. *Pattern Recognition*, 41(8):2484 – 2492, 2008.

[40] M. J. Harker and P. L. O'Leary. First order geometric distance (The Myth of Sampsonus). In *British Machine Vision Conference (BMVC)*, 2006.

[41] R.I. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1036–1041, 1994.

[42] R.I. Hartley. Self-calibration from multiple views with a rotating camera. In *European Conference on Computer Vision (ECCV)*, pages 471–478, 1994.

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

[44] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.

[45] R. Hu. Robust camera self-calibration. Master's thesis, University of Nevada, 2001.

[46] T.S. Huang and O.D. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, 1989.

[47] P.J. Huber and E.M. Ronchetti. *Robust Statistics*. Wiley, 2nd edition, 2009.

[48] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87–116, 1988.

[49] R.A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):122–139, 1983.

[50] F. Kahl, B. Triggs, and K. Aström. Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision*, 13(2):131–146, 2000.

[51] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–517, 2004.

[52] K. Khoshelham. Accuracy analysis of Kinect depth data. In *ISPRS Workshop Laser Scanning*, volume 38, 2011.

[53] M. Levoy and P. Hanrahan. Light field rendering. In *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 31–42, 1996.

[54] H. Li. A simple solution to the six-point two-view focal-length problem. In *European Conference on Computer Vision (ECCV)*, pages 200–213, 2006.

[55] T.-Y. Li. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta Numerica*, 6(1):399–436, 1997.

[56] D. Liebowitz and A. Zisserman. Combining scene and auto-calibration constraints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 293–300, 1999.

[57] M.I.A. Lourakis. levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. [online]. `http://www.ics.forth.gr/~lourakis/levmar/`. Accessed: 2012-09-30.

[58] M.I.A. Lourakis and A.A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1):1–30, 2009.

[59] M.I.A. Lourakis and R. Deriche. Camera self-calibration using the singular value decomposition of the fundamental matrix: From point correspondences to 3D measurements. Technical Report RR-3748, ROBOTVIS - INRIA Sophia Antipolis, 1999.

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

[61] Q.-T. Luong and O.D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22:261–289, 1997.

[62] Y. Ma, R. Vidal, J. Koeck, and S. Sastry. Kruppa equation revisited: Its renormalization and degeneracy. In *European Conference on Computer Vision (ECCV)*, pages 561–577, 2000.

[63] J. Mallon and P. Whelan. Projective rectification from the fundamental matrix. *Image and Vision Computing*, 23(7):643–650, 2005.

[64] R.A. Maronna, R.D. Martin, and V.J. Yohai. *Robust Statistics*. Wiley, 2006.

[65] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London, B*, 204:301–328, 1979.

[66] L. McMillan and G. Bishop. Plenoptic modeling: an image-based rendering system. In *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 39–46, 1995.

[67] G. Medioni and S.B. Kang. *Emerging Topics in Computer Vision*. Prentice Hall, 2004.

[68] P. Meer, D. Mintz, A. Rosenfeld, and D.Y. Kim. Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70, 1991.

[69] P. Meer, C.V. Stewart, and D.E. Tyler. Robust computer vision: An interdisciplinary challenge. *Computer Vision and Image Understanding*, 78(1):1–7, 2000.

[70] P.R.S. Mendonca and R. Cipolla. A simple technique for self-calibration. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 500–505, 1999.

[71] R. Mohr and B. Triggs. Projective geometry for image analysis. In *International Symposium on Photogrammetry & Remote Sensing (ISPRS)*, 1996.

[72] D.C. Montgomery, E.A. Peck, and G.G. Vining. *Introduction to Linear Regression Analysis*. Wiley, 2007.

[73] D. R. Myatt, P.H.S. Torr, N. Slawomir, J. Bishop, and R. Craddock. NAPSAC: High Noise, High Dimensional Robust Estimation - it's in the Bag. In *British Machine Vision Conference (BMVC)*, 2002.

[74] A. Neumaier. Enclosing clusters of zeros of polynomials. *Journal of Computational and Applied Mathematics*, 156(2):389–401, 2003.

[75] K. Ni, H. Jin, and F. Dellaert. GroupSAC: Efficient consensus in the presence of groupings. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[76] D. Nistér. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.

[77] J. Oliensis. A new structure-from-motion ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):685–700, 2000.

[78] J. Oliensis. The least-squares error for structure from infinitesimal motion. *International Journal of Computer Vision*, 61(3):259–299, 2005.

[79] J. Oliensis and V. Govindu. An experimental study of projective structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(7):665–671, 1999.

[80] P. Pedersen. Multivariate sturm theory. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC)*, pages 318–332, 1991.

[81] M. Pollefeys. *Self-calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences*. PhD thesis, Katholieke Universiteit Leuven, 1999.

[82] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[83] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008.

[84] M. Pollefeys and L. Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):707–724, 1999.

[85] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[86] M. Pollefeys, F. Verbiest, and L. Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *European Conference on Computer Vision (ECCV)*, pages 837–851, 2002.

[87] R. Raguram, J.-M. Frahm, and M. Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision (ECCV)*, pages 500–513, 2008.

[88] H. Rastgar, E. Dubois, and L. Zhang. Random sampling nonlinear optimization for camera self-calibration with modeling of intrinsic parameter space. In *International Symposium on Visual Computing (ISVC)*, pages 189–198, 2010.

[89] H. Rastgar, E. Dubois, and L. Zhang. Guided sampling in multiple view robust motion estimation using regression diagnostics. In *International Symposium on Visual Computing (ISVC)*, pages 138–147, 2012.

[90] H. Rastgar, L. Zhang, D. Wang, and E. Dubois. Validation of correspondences in MLESAC robust estimation. In *International Conference on Pattern Recognition (ICPR)*, 2008.

[91] H. Rastgar, L. Zhang, D. Wang, and A. Vincent. Estimation of image relations from point correspondences between images. United States Patent 12/633,463, December 2009.

[92] G. Roth and A. Whitehead. Using projective vision to find camera positions in an image sequence. In *International Conference on Vision Interface (VI)*, pages 87–94, 2000.

[93] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*, volume 3rd. Wiley, 1987.

[94] T. Sattler, B. Leibe, and L. Kobbelt. SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2090–2097, 2009.

[95] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1–3):7–42, 2002.

[96] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[97] G.A.F. Seber and C.J. Wild. *Nonlinear Regression*. Wiley, 2003.

[98] J.L. Shanks, S. Treitel, and J.H. Justice. Stability and synthesis of two-dimensional recursive filters. *IEEE Transactions on Audio and Electroacoustics*, 20(2):115–128, 1972.

[99] H.-Y. Shum and S.B. Kang. A review of image-based rendering techniques. In *IEEE/SPIE Visual Communications and Image Processing Conference (VCIP)*, pages 2–13, 2000.

[100] C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, 4th edition, 1980.

[101] S.M. Smith. A new class of corner finder. In *British Machine Vision Conference (BMVC)*, 1992.

[102] N. Snavely. *Scene Reconstruction and Visualization from Internet Photo Collections*. PhD thesis, University of Washington, 2008.

[103] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80:189–210, 2008.

[104] C.V. Stewart. Bias in robust estimation caused by discontinuities and multiple structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:818–833, 1997.

[105] C.V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[106] G.W. Stewart. Perturbation theory for the singular value decomposition. *SVD and Signal Processing, II: Algorithms, Analysis and Applications*, pages 99–109, 1991.

[107] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, 2005.

[108] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE*

International Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2008.

[109] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1100–1105, 1997.

[110] P. Sturm. A case against Kruppa's equations for camera self-calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1199–1204, 2000.

[111] P. Sturm. Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. *Image and Vision Computing*, 20(5–6):415–426, 2002.

[112] P. Sturm, Z. Cheng, Ch. Chen, P. Yu, and A.N. Poo. Focal length calibration from two views: Method and analysis of singular cases. *Computer Vision and Image Understanding*, 99(1):58–95, 2005.

[113] T. Thormhlen, H. Broszio, and A. Weissenfeld. Keyframe selection for camera motion and structure estimation from multiple views. In *European Conference on Computer Vision (ECCV)*, pages 523–535, 2004.

[114] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[115] B.J. Tordoff and D.W. Murray. Guided-MLESAC: Faster image transform estimation by using matching priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1523–1535, 2005.

[116] P.H.S. Torr. *Outlier Detection and Motion Segmentation*. PhD thesis, University of Oxford, 1995.

[117] P.H.S. Torr, A. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, 1999.

[118] P.H.S. Torr and A.W. Fitzgibbon. Invariant fitting of two view geometry or "in defiance of the 8 point algorithm". In *British Machine Vision Conference (BMVC)*, 2002.

[119] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.

[120] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[121] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, 2008.

[122] B. Triggs. Autocalibration and the absolute quadric. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 609–614, 1997.

[123] B. Triggs. Autocalibration from planar scenes. In *European Conference on Computer Vision (ECCV)*, pages 89–105, 1998.

[124] B. Triggs, Ph. McLauchlan, R.I. Hartley, and A. Fitzgibbon. Bundle adjustment a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883, pages 298–372, 2000.

[125] E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision*. Prentice Hall, 1998.

[126] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[127] A. Wald. *Sequential Analysis*. Wiley, 1947.

[128] A. Whitehead and G. Roth. Estimating intrinsic camera parameters from the fundamental matrix using an evolutionary approach. *EURASIP Journal on Applied Signal Processing*, 8:1113–1124, 2004.

[129] L. Zhang, H. Rastgar, D. Wang, and A. Vincent. Maximum likelihood estimation sample consensus with validation of individual correspondences. In *International Symposium on Visual Computing (ISVC)*, pages 447–456, 2009.

[130] W. Zhang and J. Kosecka. Ensemble method for robust motion estimation. In *25 Years of RANSAC, (CVPR Workshop)*, 2006.

[131] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15(1):59–76, 1997.

[132] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.