STRATIFIED CALIBRATION AND GROUP SYNCHRONIZED FOCAL LENGTH ESTIMATION FOR STRUCTURE FROM MOTION ALGORITHMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AKIN ÇALIŞKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2017

Approval of the thesis:

STRATIFIED CALIBRATION AND GROUP SYNCHRONIZED FOCAL LENGTH ESTIMATION FOR STRUCTURE FROM MOTION ALGORITHMS

submitted by **AKIN ÇALIŞKAN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. A. Aydın Alatan
Supervisor, **Electrical and Electronics Engineering** _____

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Department, METU _____

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Erol Şahin
Computer Engineering Department, METU _____

Prof. Dr. Uğur Güdükbay
Computer Engineering Department, Bilkent University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: AKIN ÇALIŞKAN

Signature :

ABSTRACT

STRATIFIED CALIBRATION AND GROUP SYNCHRONIZED FOCAL LENGTH ESTIMATION FOR STRUCTURE FROM MOTION ALGORITHMS

Çalışkan, Akın
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. A. Aydın Alatan

July 2017, 86 pages

The estimation of unknown calibration parameters of the cameras without using any calibration pattern is critical for the performance of the 3D computer vision applications such as structure from motion, pose estimation, visual odometry, and it is still an open problem for the researchers. In this thesis, our contribution is two folded. First of all, we propose a novel stratified approach for estimating both the focal length and the radial distortion of a camera from given 2D point correspondences without knowing any calibration information, such as the focal length of a camera. We assume that the images share the same intrinsic parameters and we further assume that the optical image centers are known. Our method progresses first by showing that the distortion on the point coordinates can be removed without the knowledge of the true focal length by enforcing the epipolar geometry constraint. Next, by using the distortion free correspondences, we estimate the true focal length of the camera by enforcing the trace constraint. Secondly, we utilize the idea of estimation of calibration parameters from two cameras, and propose method for estimation of focal lengths of all cameras, which can be all different, used in structure from motion pipeline. The relation of focal lengths between two adjacent cameras are defined from a trace constraint, and this is followed by using the group synchronization method for all cameras in a dataset to estimate unknown focal lengths. In this step a novel energy function is proposed together with global optimum solution. The optimal solution

for this function gives the resultant focal lengths with an error which can be handled by bundle adjustment stage of structure from motion algorithms, even if very limited number of focal lengths are available. In both contributions, our methods are quite easy to implement compared to other methods in the literature and we demonstrate their accuracy and robustness against noise on synthetically generated data sets. Furthermore, we perform experiments for the accuracy of our first method on real image pairs by comparing our results against a method that uses a calibration pattern, and for the accuracy and complexity of the second method on real data sets used in structure from motion pipelines.

Keywords: Self-Calibration, Focal Length Estimation, Radial Distortion Removal, Structure From Motion, Multiple View Geometry, Non-Linear Optimization

ÖZ

HAREKETTEN YAPI ÇIKARIM ALGORİTMALARI İÇİN KATMANLI KALİBRASYON VE GRUP SENKRONİZE ODAK UZAKLIĞI KESTİRİMİ

Çalışkan, Akın
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. A. Aydın Alatan

Temmuz 2017 , 86 sayfa

Kalibrasyon için özelleşmiş levhaların kullanılmadığı ve bir kameranın kendine özgü parametrelerinin kestirimi 3 boyutlu bilgisayarlı görü uygulamalarının performansı için çok kritik bir önem taşımaktadır ve halen açık bir problemdir. Bu uygulamalara hareketten yapı çıkartma, poz kestirimi, görsel poz takibi örnek olarak verilebilir. Bu tezde yapılan katkı iki aşamalıdır. İlk olarak özgün bir yaklaşım ile odak derinliği ve radyal bozulmanın sadece iki tane görüntü kullanılarak kestirildiği katmanlı bir algoritma önerilmiştir. Bu yöntemde sadece iki görüntü arasındaki eşleşen noktalar kullanılmıştır ve herhangi bir kalibrasyon levhasından yararlanılmamıştır. Tezde önerilen yöntemde iki görüntünün de aynı kamera parametrelerine sahip olduğu ve optik görüntü merkezlerinin önceden bilindiği varsayılmıştır. Önerilen yöntemde öncelikle radyal bozulma, yerel gerçeklik odak derinliği bilgisi olmadan, epipolar geometri kısıtları kullanılarak kaldırılmaktadır. Sonrasında doğru odak derinliği, ağ diyagramı kısıtı ve radyal bozulmaya uğramamış nokta eşlemeleri kullanılarak hesaplanmaktadır. Tezdeki ikinci katkı olarak, iki görüntüden kamera parametrelerinin kestirim fikri kullanılarak, hepsi birbirinden farklı olabilecek kameralardan oluşan, hareketten yapı çıkartma algoritmasına girdi olarak verilen bir veri kümesinin hepsinin odak derinliğini bulmak önerilmektedir. Ardışık gelen iki kameranın odak derinliği arasındaki ilişki ağ diyagramı kısıtına dayanarak çıkartılmaktadır. Sonrasında, grup senkronizasyon metodu kullanılarak bilinmeyen bütün kameraların odak derinlikleri bulunmak-

tadır. Bu aşamada özgün bir maliyet fonksiyonu önerilmektedir ve bu fonksiyonun en optimal çözümü hesaplanmaktadır. Sonuç olarak hesaplanan odak derinliği değerlerindeki hata oranı, bilinen odak derinliği değerleri çok az bir oranda olsa bile, hareketten yapı çıkartma algoritmalarının demet duzenleme adımında düzeltilebilecek seviyededir. Yapılan iki katkıda da, önerilen yöntemler diğer metodlara kıyasla kolayca uygulanabilir yöntemlerdir ve sentetik veri deneylerinde gürültüye karşı gürbüzlükleri gösterilmiştir. Ayrıca, ilk önerilen yöntem gerçek veri setlerinde de denenmiş ve kalibrasyon levhası ile elde edilen sonuçlarla, bu yöntemden elde edilenler karşılaştırılmıştır. Yine ikinci önerilen yöntemin gerçek veri setleri kulanılarak gürbüzlüğü gözlemlenmiştir.

Anahtar Kelimeler: Öz Kalibrasyon, Odak Derinliği Kestirimi, Radyal Bozulma Düzeltimi, Hareketten Yapı Çıkarımı, Çoklu Görünüm Geometrisi, Doğrusal Olmayan Optimizasyon

To my wife and my parents

ACKNOWLEDGMENTS

First of all, I would like to thank my adviser Prof.Dr. Aydin Alatan for all of his contributions in order to develop my knowledge and abilities in the field of computer vision. He has not been only my adviser but also guidance for me about the life for last four years. If he didn't give me a chance to work at MMRG when I was a junior year student, I wouldn't be able to work this field with all unforgettable coworkers and supervisors.

I also would like to thank my co-adviser Engin Tola for all of his precious effort on me during this period. It is not possible to write all things he has done for me for last two years here; however, there are few things that I want to mention: First of all, he believed in me and guide me to find my way while I was looking for my research question with all his attention and respect to all my ideas. Then he taught me about how to be a researcher in the 3D computer vision while developing my abilities in mathematics, programming. which I will remember in all of my life.

Being a partf of MMRG is one of the best thing in my life, and it gave me chance to meet my friends that I care most, namely Yagiz, Ozan, Yeti, Beril, Emrecan, Emin, Selin. I want to thank Yagiz for his precious friendship and being my brother in this life. I always miss the moments we have shared together once there was a MMRG and also when I was an intern at DRZ. I also want to thank Emin Zerman for having great time with him discussing about the life and computer vision. I would like to thank Beril for great protectiveness that I have always needed, and the moments we share doing a great GIYBET stuff.

It has been precious to talk with Yeti about my thesis, sports, and all our common hobbies, I would like to thank him for his unique friendship. I also want to thank Emrecan (Bati), he has been always with me whenever I need him like a superhero, he has the definition of friendship in all manner. I couldn't forget all the moments we shared with Yeti and Emrecan about the food, especially the one named as KOKO. I also want to thank Selin for being my little sister, making the LAB enjoyful.

As I said before, I have had great time with my friends at MMRG, and I would like to thank all my colleagues, especially Alper Abi. He is a great advisor and teacher, I learnt a lot from him about how to be scientist.

I am also grateful to all my colleagues, and supervisors for their support and contributions on me, especially Zafer Arican, Serdar Gedik, Ahmet Saracoglu, Paul Beardsley, I have learned a lot from them, and I am always grateful for them all of their effort.

I would like to express my gratitude to my mother, Suna Caliskan, and father, Ali Caliskan. They spent all their life for their children's happiness and future life. They have supported all of my dreams, I have never felt alone because of their powerful protection in all my life.

I also want to express my gratitude to my lovely friends, Burcu, Baran, Umutcan, and Alperen. I have had great moments with them since the beginning of my university life. They have always supported me in all my actions, and I experienced the greatest moments with them. Especially, the unexpected RAKI nights with Umutcan and Baran made me feel great during the thesis period, and I would like to continue organizing them more often.

The biggest part of the acknowledgement goes to my wife, Ozumcan. For 6 great years that I spent with her, she have always believed my dreams and stayed with me. I am deeply grateful to her for all her unconditional support and unlimited patience during the thesis preparation period. I would like to thank Ozum for the moments that we spent hours with the discussion about academia and the future plans. I am happy that we will continue our life in London for the new adventures. I feel that I am the luckiest person in the world because I can look into her beautiful eyes.

In conclusion, I would like to thank my family, my wife, my friends and my colleagues for all things that I learnt from them the all moments we shared together.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

SfM	Structure from Motion
FOV	Field of View
COTS	Commercial off-the-shelf
QEP	Quadratic Eigenvalue Problem
EXIF	Exchangeable Image File
MSE	Mean Square Error
MST	Maximum Spanning Tree

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

There is no shortage of debate about how reconstructing the 3D models change the way human or robots visualize in many applications, including virtual reality [46], augmented reality [26, 3], mixed reality [27, 25], medical and scientific visualization [18, 29], defense, flight simulation and training [28]. With this technology, people can feel the atmosphere of being in a stadium while watching a football game [64], or architects can work on the real model of their work in a detailed 3D, Figure 1.1, or tourists can visit a city in full scale without going there exactly [1, 56], or self-driving cars can navigate themselves in a crowded city [65, 48].

That is not a dream to being integrated with these technologies in our daily life. However, although there are so much work so far, there will be much more effort needed to make this technology as being perfect as it will be. For this purpose, the research studies on 3D computer vision falls into two main groups that have been studied for more than 20 years. First of all, the visualization of 3D models for the natural human perception is one of the most significant challenges in this area [54, 43]. As an ultimate goal, the reconstructed models should be as real world object in terms of scale, color and interaction with others. Secondly, recovering the 3D structure, i.e. 3D models, of a scene is one of the most fundamental goals in 3D computer vision, which is strictly connected to the former group of research, and effect its success or failure. For this reason, researchers put their efforts to generate more realistic 3D models. The main direction on this research is mainly determined by the type of the

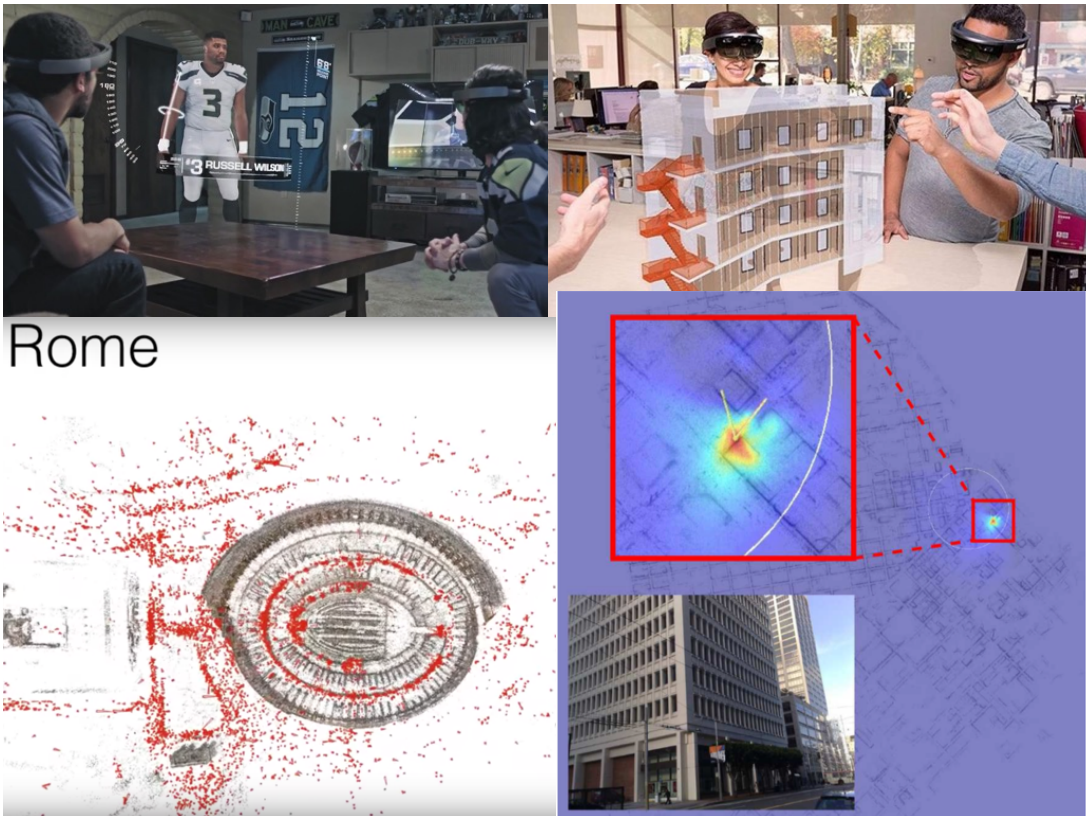


Figure 1.1: 3D Experience of watching a game in a stadium [64] , modeling the buildings, visiting a city [1, 56], navigating an autonomous car [65, 48].

capturing device used in the application. One group of research is conducted with the depth cameras [32, 17], such as products named as Kinect [14], intel real sense [13]. A sample 3D model generated with volumetric 3D reconstruction pipeline proposed in [32] is illustrated in Figure 1.2.

This approach reconstructs the 3D model of a scene from captured depth data [32] by merging them in an order. In details, this approach gets realistic models in real time with low resolution for indoor environments [57], while they could not be successful for outdoor scenes due to the depth sensor failure. Furthermore, there is another group of research for 3D model generation of a scene, which is recovering the 3D structure of a scene from set of images [1, 58, 56, 52, 51], and known as "multiview structure from motion (SfM)". The early research on SfM started with the estimation of structure from two views [21]. This is followed by utilizing more than two images, which are captured by the same camera, for generating the 3D structure. This approach is called *sequential*, or *incremental*, structure from motion [51] and illustrated

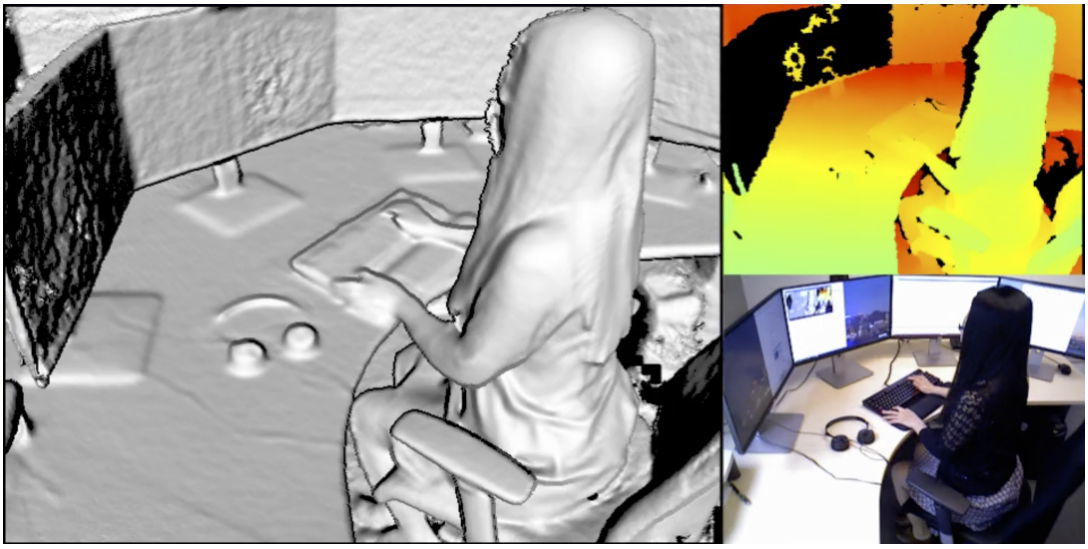


Figure 1.2: A Sample 3D model of a scene, which is generated by volumetric 3D reconstruction algorithm [32]

in Figure 1.3. The motivation of using more images for recovering the scene structure is to increase the resolution and the information inside the model.

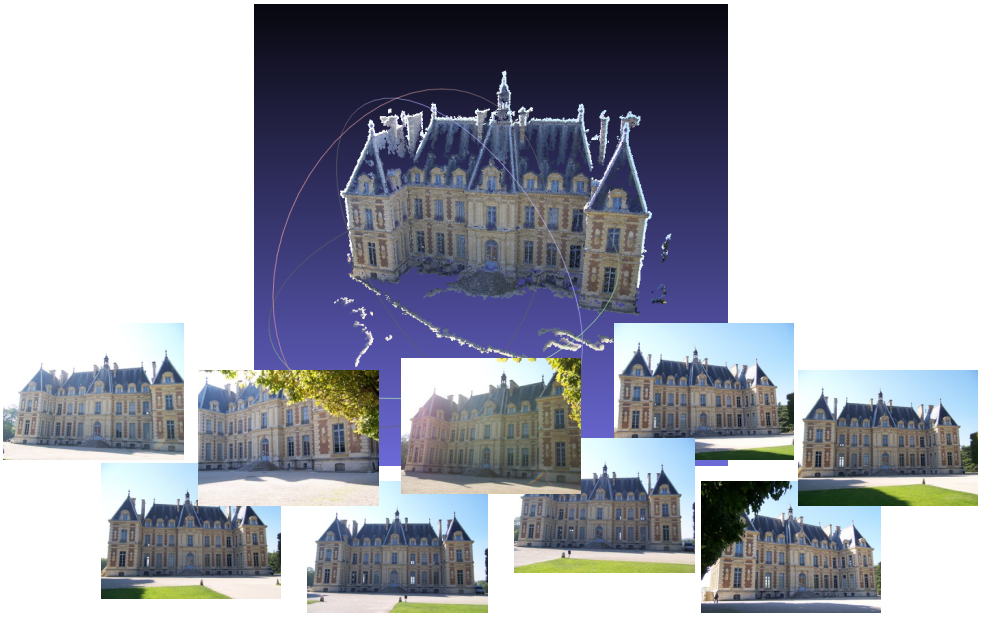


Figure 1.3: Incremental, or sequential, Structure from Motion: Set of images and resultant 3D model of a scene [51].

Furthermore, there is one more SfM approach [1, 58, 56] that generates 3D model from extremely large collection of photographs, such as those that can be download

from internet after a search for a keyword, for example searching "***ROME***" keyword in a Flicker [15] web site [1]. In Figure 1.4, sample photographs for Notre Dame cathedral and the resulting 3D model are illustrated [58]. In this case, it is possible to reconstruct a 3D model of a scene without going there, and to use a large set of photographs to increase the resolution of the model; however, this approach introduces researcher with new challenges due to processing of very large collection of data. For instance, as it is stated in [58], the downloaded photographs are from different cameras, such as point and shot cameras, action cameras,or smart phone cameras, and they are captured in various time intervals. This makes the feature-matching step more difficult and forces those algorithms to be more robust against different type changes [55].

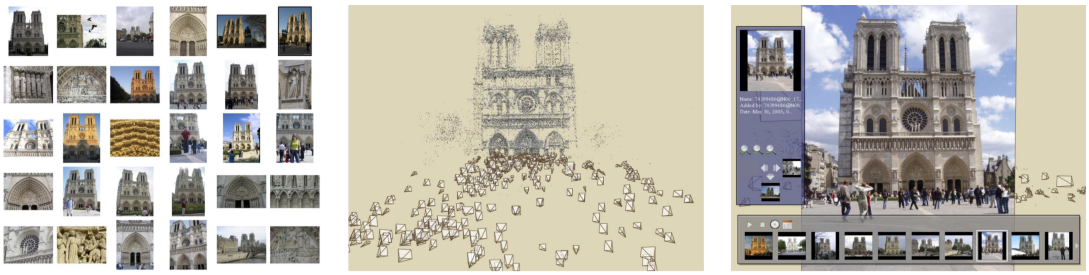


Figure 1.4: Global Structure from Motion: Large set of images and resultant 3D model of a scene [58].

Furthermore, the processing periods become longer, and they tend to increase exponentially as the number of photographs increases. It is also computationally complex to estimate global poses for each camera [61, 62] in global SfM pipelines.

As another complexity problem, global SfM approach uses all available Exchangeable Image File (EXIF) data, where information on image, such as camera intrinsic parameters, are stored, of the input set of photographs, and estimates the missing calibration parameter during the structure recovery stage [1, 53]. Although it seems that there is a solution to calibrate all the cameras using reprojection error minimization [21], i.e. bundle adjustment, the complexity of this stage become obstructive for large datasets. The time consumed during this stage could decrease by providing initial focal length values, i.e. the optimization algorithm in bundle adjustment stage could converge faster if the initial values of unknown focal lengths are close the real values [62]. Hence, alternative solutions are developed in order to increase performance of

this stage, which are called as focal length initialization algorithms.

In [31], focal length of cameras are estimated during the camera pose estimation phase and the these initial values of focal lengths are used in the bundle adjustment stage, which increases the performance of this step. Another work [62] proposes a solution for estimating the calibration parameters from two views and propagates this approach for each camera pair in the dataset. However, these methods solve focal length initialization up to some extend. In other words, the capability of these methods on solving initialization problem has limitations prescribed upon the different focal lengths [62]; they can only estimate focal length of a dataset if there is small variation in focal lengths. Hence, this is a real challenge and a fundamental problem for the researchers working on 3D computer vision. Apart from the accurate focal length initialization, considering that it is critical for 3D generation algorithms to be fast, initialization stage should be really fast not to be the bottleneck for the complexity of the full pipeline [53]. So, this is another constraint to be handled.

In conclusion, it is known that the goal of global SfM algorithms is to use all available photographs of a place, or scene, for recovering 3D structure, and it is critical for these algorithms to initialize the focal length of each camera fast and accurately. The goal of this thesis is fully matched with this purpose, and the contributions in this scope are explained in details during the next chapters.

1.2 Contribution of the Thesis

This thesis mainly focuses on the estimation of the calibration parameters for unknown cameras from a set of images, possibly captured with independent cameras or a video sequence whose calibration is unknown. The first goal of this thesis is to remove lens distortion, which is an inevitable effect for popular action cameras. After removal of lens distortion, the subsequent aim is to compute the focal length of a camera from two views without using any calibration pattern. A novel and easy way to implement approach is proposed and its results are more accurate than the relevant algorithms in the literature.

As a second goal of this thesis, a novel approach for estimating the focal lengths of

all cameras in a Structure from Motion (SfM) input dataset consisting of independently captured images is proposed. In this novel approach, the relation between the unknown focal lengths is derived from the geometric constraint between any pair of views, and extension of this idea to all of the cameras in a dataset is undertaken by minimization of a novel cost function. This technique also proposes an approach for estimating the focal lengths in parallel for all cameras.

1.3 Scope and Organization of Thesis

In this thesis, the intrinsic camera calibration from two views and the focal length estimation for all cameras in a set of pictures, which is a crucial input for any 3D modelling and reconstruction algorithms from 2D images, are examined. In Chapter 3, the focal length estimation algorithm from two views is described in details, and the result of the experiments are presented. The group synchronized focal length estimation algorithm, which is the second contribution in this thesis, is explored in Chapter 4. As a comprehensive guide for multiview-geometry and the literature for the proposed methods, Chapter 2 presents all necessary background information about camera calibration and geometric constraints. The thesis is concluded based on the observations from the simulations in Chapter 5.

CHAPTER 2

CAMERA CALIBRATION AND MULTI-VIEW GEOMETRY

The general purpose cameras are produced with a certain specifications, such as focal length, resolution of the sensor and lens model. However, due to possible errors during the production, these specifications are not reliable in any application that requires some precision. Hence, there are various approaches that are based on visual data for estimating the camera parameters, which is called as *camera calibration process*. Utilization of images from unknown cameras is another reason for the requirement of such calibration algorithms. This also requires estimating the camera parameters before using these images.

Understanding the lens model of the camera, and estimating the camera parameters are quite important in 3D computer vision due to the fact that accuracy of the geometric modeling algorithms strictly relies on the correctness of the calibration parameters [21]. Therefore, it is essential to understand camera models, which are described in Chapter 2.1. The intrinsic calibration process is explored in Chapter 2.3. Camera models are strictly related to the lens distortion, and this is investigated in Chapter 2.1. This is followed by the removal of lens distortion in Chapter 2.4 in order to utilize geometric constraints accurately, which is explored in Chapter 2.2.

2.1 Camera Models and Lens Distortion

A camera captures a scene after projecting an object point in 3D world to the 2D image plane. There are two main types of projection, namely central projection (e.g. perspective projection) and parallel projection (e.g. orthographic projection) [21]. In

this thesis, central projection is considered that is known to be more accurate with respect to its parallel counterpart. The group of cameras modeling with the central projection falls into two groups in terms of the location of the center of the cameras [21]. The center of the first group cameras at a finite location, whereas that of other group of cameras is at infinity. The name of the former one is finite cameras, and the latter is named as infinite cameras. In the scope of this thesis, finite cameras are utilized, which are mostly preferred in the applications.

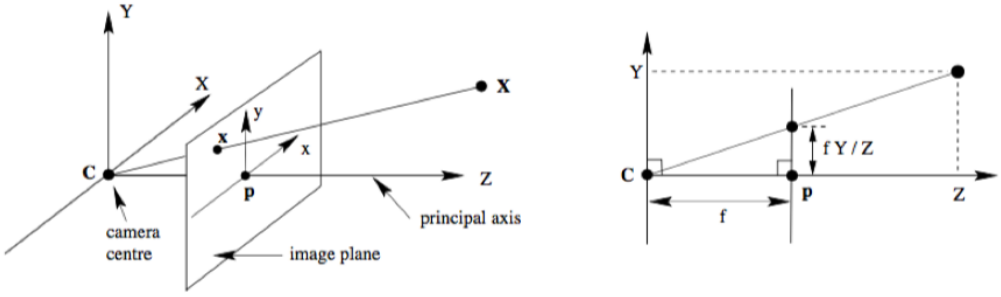


Figure 2.1: Pinhole Camera Model [21]

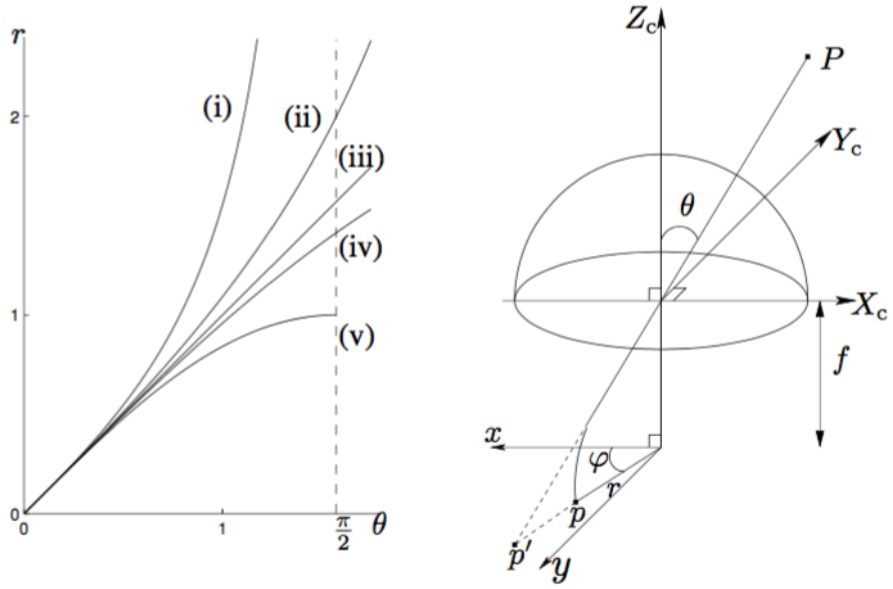


Figure 2.2: Fisheye Projection Models. (i) perspective projection, (ii) stereographic projection, (iii) equdistance projection, (iv) equisolid angle projection, (v) orthogonal projection [37]

Among the finite cameras, the most simplest and basic camera model is the pin hole camera model [21], which is illustrated in Figure 2.1. In this model, central projection

of the 3D point is a point on image plane, and the center of projection is the origin of the Euclidean coordinate system in this figure. The image plane is a planar surface at $z = f$, and f stands for the camera parameter about that optical component, namely focal length. In this model, C is the camera center, which is also a center of projection, and p is the principal point. A 3D point X projects to x, i.e. the ray aiming the center of projection from X intersects the image plane at x. In this model, the line starting from camera center and intersecting the image plane at principal point, which also perpendicular to the image plane at this point is called principal axis.

As a subgroup of the central projection, there is a type of projection model considered in the scope of this thesis, which is called Fisheye projection [37]. This projection type models the radial distortion on the images; in other words, the straight lines are not projected as linear curves to an image plane, and this causes geometric constraints fail. The main motivation behind the usage of Fisheye cameras is to enable capturing more scene regions in the image, i.e. field of view (FOV) of a camera is large. It should be noted that recently popularized action cameras are typically utilize such lenses that yield Fisheye distortion.

The fish eye projection falls into four groups [37], namely *stereographic projection*, *equdistance projection*, *equisolid projection*, and *orthogonal projection*. All of these projection is illustrated in Figure 2.2. Among these projection models, the equidistance model is the most commonly used model to calibrate Fisheye lens, so this model will be used in the following section where the radial distortion is removed.

2.2 Fundamentals of Multiview Geometry

In the case of one pair of cameras as illustrated in Figure 2.3, the 3D point X is projected as 2D point u to the image planes of cameras, and this projection is written as:

$$\lambda \mathbf{u} = P\mathbf{X} \tag{2.1}$$

where P is 3x4 matrix, λ is unknown scalar value, points are in homogeneous coordinates, u = $(u, v, 1)^T$, and X = $(x, y, z, 1)^T$. The projection matrix, P, can be rep-

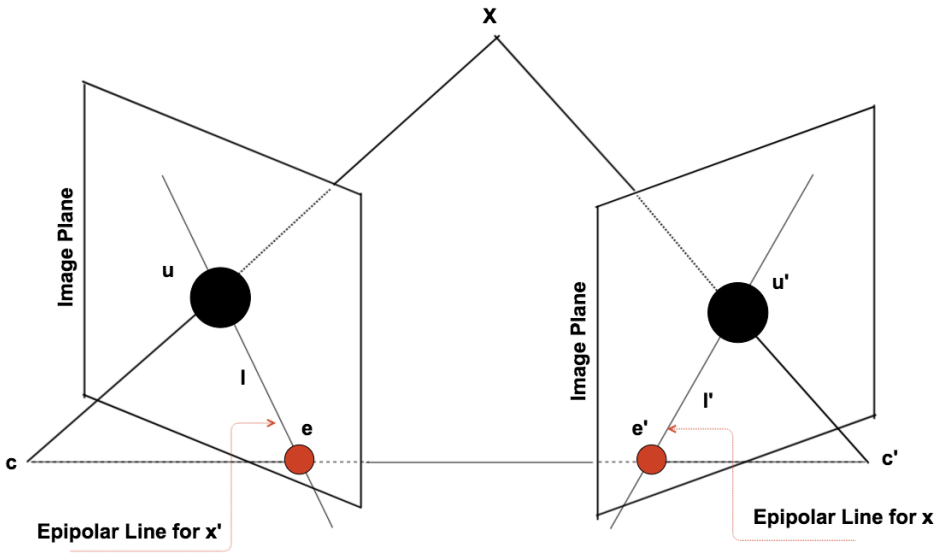


Figure 2.3: Epipolar Geometry Between Two Cameras

resented by 3x3 camera matrix, K, 3x3 rotation matrix, and 3x1 translation vector,t, which is given in Equation 2.2.

$$P = K\Big[R|t\Big] \tag{2.2}$$

Here K matrix contains camera intrinsic parameters, R matrix shows the pointing direction of the camera, and t states the camera position in 3D coordinate system. The general camera matrix is written as:

$$K = \begin{bmatrix} f & s & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

where f is focal length, c_x, c_y is the principal point coordinates, α is the aspect ratio of the pixels, s is the skew parameter. In the scope of this thesis, also as in the most of the digital cameras, the pixels of the cameras are square, $\alpha = 1$, and the skew is set to zero, $s = 0$. The principal point of the camera is very close to the image center, so it is taken as an initial value for principal point coordinates. The four (out of five) calibration parameters are set to some values which are quite close to their exact values; however, the focal length should be estimated correctly. Reducing the number

of unknown by setting these parameters to initial values is helpful for the estimation of focal length value, this is followed by estimating the exact values afterwards. In the scope of this thesis, the new camera matrix which is up to focal length is defined and it has the form:

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

The epipolar geometry is defined as intrinsic projective geometry between two views [21]. It only depends on the intrinsic parameters of the camera and the relative pose between two views. As it can be observed in Figure 2.3, the projection of X is u and u' on image planes, respectively. In this epipolar relation, 3D point X, and two camera centers constructs an epipolar plane. The epipole, $\mathbf{e}$ is the intersection point of the line connecting camera centers and the image plane. Epipolar line, $\mathbf{l}$, connect the projected point on image plane and the epipole, $\mathbf{e}$. Epipolar line reduces the search space of the corresponding point of u in the second image plane, which indicates that this point should be on the line, $\mathbf{l}'$.

For a camera pair in Figure 2.3, one can define an epipolar geometry relation, which is first stated by fundamental matrix, F. Fundamental matrix represents the intrinsic geometry. It is 3x3 matrix of rank 2. If point u is the projection of X on one image and the u' is on the other image, then there is a relation between u and u':

$$u'^T F u = 0 \tag{2.5}$$

The epipolar lines can be derived from fundamental matrix, F:

$$l' = Fu \tag{2.6}$$

$$l = Fu' \tag{2.7}$$

Secondly, in the case of fully calibrated cameras, i.e. calibration parameters are completely known, there is one more geometric relation defined by using Essential matrix,

E. The normalized image coordinates, $x = K^{-1}u$, and $x' = K^{-1}u'$ are related with essential matrix, E:

$$x'^T E x = 0 \tag{2.8}$$

where E is a 3x3 and rank-2 with two equal singular values. These specifications turns to the constraints on essential matrix:

$$det(E) = 0 \tag{2.9}$$

$$2EE^T E - tr(EE^T)E = 0 \tag{2.10}$$

Equation 2.9 is called the *rank constraint* [21] and the latter one is the most significant constraint used in the scope this thesis, which is called as *trace constraint* [50]. The detailed proof is given in the Appendix A. Nevertheless, the noise robustness of this constraint against correspondence errors, and the calibration algorithms constructed on this constraint show that this constraint is very crucial.

2.3 Estimation of Intrinsic Calibration Parameters

The estimation of intrinsic camera calibration parameters is a critical step for 3D computer vision algorithms. Among previous studies on this topic, there are two main approaches, which are explored in this chapter. The first group of approaches mainly relies on dedicated calibration images. For this scenario, the images of a predefined (calibration) pattern in the scene is utilized, and the parameters are estimated by using the point correspondences between these calibration images [6]. For the cases, for which it is not possible to capture calibration pattern by the camera (but arbitrary images are available for the camera), another group of approaches is available where they estimate the calibration parameters of the camera matrix in 2.4 by using point correspondences. The method in the latter group will be explored in Chapter 2.3.1.

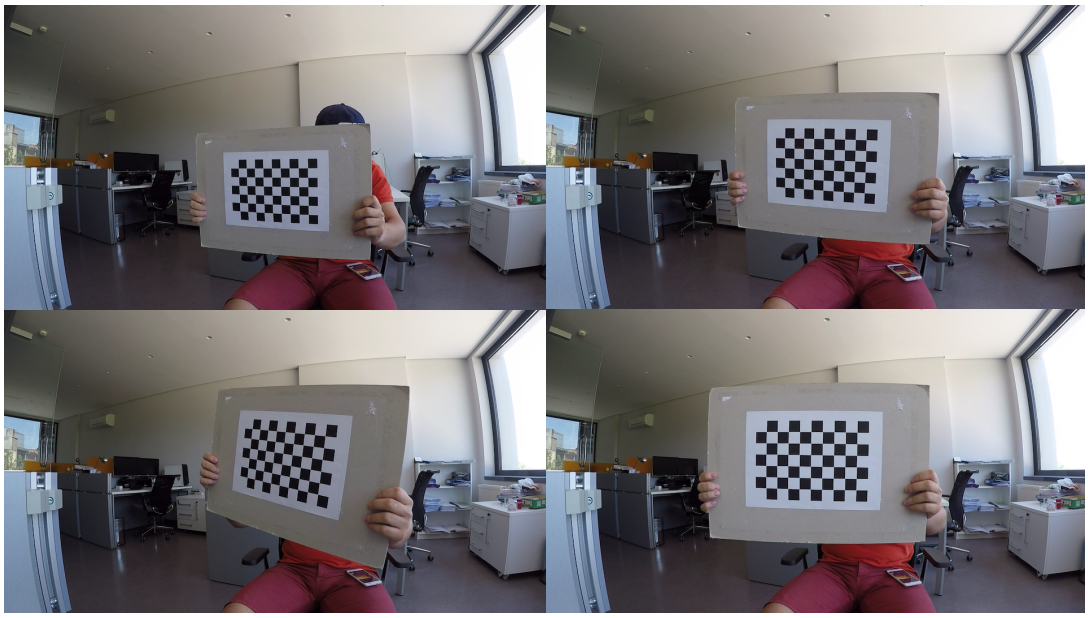


Figure 2.4: Sample Calibration Images of a Planar Calibration Pattern.

The former group of methods starts by capturing the calibration images. As it is illustrated in Figure 2.4, the images of calibration pattern are captured by the camera in a variety of different poses. Next, the corresponding points are detected as the corners of the calibration pattern and the plane structure and distances between corners of the calibration pattern are known. With these input data and the camera matrix in 2.3, this problem is modelled as a maximum likelihood estimation problem by using the cost function defined for corresponding points [6].

In the second group of estimation methods, the geometric relation between at least two views is utilized. Typically, the camera matrix for which the focal length is the only unknown is considered. The problem turns to a focal length estimation from multiple view, which is explored in details in the next chapter.

2.3.1 Focal Length Estimation From Two Views

For focal length estimation from two views, Sturm [60] proposes a method that links the fundamental matrix to the focal length by using Kruppa equations [23]. In Li [44], a system of polynomial equations are formed in terms of the trace constraint for the essential matrix [50] and the singularity constraint [21] on the fundamental

matrix in terms of focal length. Another polynomial equation solution is proposed by Stewenius [59] and Kukelova et al. [39] that again exploit the trace constraint for essential matrix and rank constraint for fundamental matrix in order to define those equations but the solution is obtained by using a Gröbner solver [39] and these methods differentiate from each other by their use of different basis selections. The most relevant work to the focal length estimation part of the proposed approach in this thesis is Kanatani et al. [34], which proposes a solution for focal length estimation from two views. In this approach [34], the previously derived constraint on essential matrix [33] is utilized to estimate focal length with the only fundamental matrix input.

Although utilization of the same constraint on the essential matrix for in the given solutions as in [34] is proposed, this constraint is tried to be solved in a quite different way so that the solution becomes more robust and operates under many different camera configurations compared to the algorithm presented in [34]. Kanatani et al. augment their previous work in [35] by using some heuristic metrics to improve the robustness of their earlier work. A detailed comparison of these algorithms with the proposed approaches is presented in Chapter 3 to highlight how the proposed way of solving the trace constraint compares to Kanatani in terms of performance. Finally, in a later study [36], Kanatani et al. proceed to show that their former approach is not feasible for two view problems and formulate their method for the three view case.

In distinction to the focal length estimation from two views, estimating the relative pose with both unknown focal length and radial distortion without using any calibration pattern is still an open issue. To the best of our knowledge, the algorithm proposed by Jiang et al. [30] is the only method for this purpose. In this method, a system of polynomial equations is generated by using the trace constraint and the singularity constraint for the essential matrix together with the epipolar constraint on the fundamental matrix that includes the distortion formulation via the one parameter division model [30]. Then this polynomial system is solved by using the Gröbner Basis method. Although this method solves the focal length problem together with radial distortion, it is computationally quite expensive and not feasible for practical applications.

2.4 Removal of Lens Distortion

The observed radial distortion should be removed in order to be able to use the epipolar geometry constraints for image pairs. Hence, there are some proposed methods in the literature for this purpose [6, 24, 38, 5, 19, 9, 42, 24, 40, 45, 10, 41, 39], which fall into two main groups: First of all, if the calibration images for the camera are available, the radial distortion model is utilized for point correspondences and lines on the calibration pattern. However, in the case of lack of calibration images, the previously proposed methods model this problem as the estimation of distortion parameters in order to satisfy epipolar constraints, which is explained in details in the next chapter.

2.4.1 Radial Distortion Estimation From Two Views

Radial distortion modeling for the estimation of epipolar geometry is quite a difficult task but still there are many algorithms already proposed for this purpose in the literature, [38, 5, 19, 9, 42, 24, 40, 45, 10, 41, 39]. Fitzgibbon [19] proposes a solution for the cameras that have the same radial distortion. In this method, epipolar constraint is utilized with one parameter division model and at least nine corresponding points are required. Since only epipolar constraint is utilized, the problem is formulated by a quadratic eigenvalue problem. Li et al. [45] also solve the radial distortion problem similar to [19] but they use kernel voting instead of median of the distribution in their estimation decision. Another method different than [19] is proposed by Barreto et al. [5], which utilizes the epipolar constraint for radial fundamental matrix. This method estimates the radial distortions of the cameras even if they are not identical. In distinction to the methods above, a minimal solution to the estimation of the epipolar geometry under radial distortion is proposed by [42, 10, 41, 38, 9, 39]. In these methods, epipolar constraint and singularity of the fundamental matrix is utilized with one parameter division distortion model. Moreover, in all of these techniques, a system of polynomial equations is solved by the Gröbner Basis method [16]. By using slightly different constraints or problem requirements, these algorithms differentiate from each other in terms of the number of minimal correspondences they require. Most recently, the difference between the minimal solvers are analyzed in a study by Kukelova et al. [40]. As a result of this study, authors propose a more robust method

which needs of minimum ten point correspondences. As a last group in the scope of radial distortion estimation, Brito et al. [24] propose a solution based on the fact that epipolar lines going through the distortion center should remain straight in the case of radial distortion. By using this observation, the proposed method estimate the constant radial distortion parameter from two views by using the one parameter division model.

2.5 3D modelling and the Effect of Calibration

In general, 3D model extraction relies on the epipolar geometry between image pairs, so it is crucial to estimate camera calibration parameters, and to remove radial distortion before step on the 3D model generation. That is, accurate triangulation requires to known exact camera calibration parameters [11, 12]. In case of any mistake on those parameters, it is inevitable to have erroneous 3D models and the error is expected to propagate as the number of input image increases. Hence, true focal length values are estimated during recovering of 3D model by minimizing the re-projection error, which is named as bundle adjusment stage in SfM algorithms [21].

2.6 Focal Length Estimation for Unknown Cameras in Structure from Motion Algorithms

Among the large set of images which is given as input to SfM algorithms, it is possible to face with the unknown focal lengths. For instance, it is easily observed in the global SfM algorithms, which utilize the large set of images downloaded from internet and many of them has no EXIF data [1]. There are two approaches in order to estimate unknown focal lengths in these datasets. First of all, bundle adjustment step can estimate the true focal length values using the known ones as recovering the 3D structure [51, 52]. This is considered as a solution for sequential SfM algorithms, where the projection matrix is computed in an order, and calibration parameters are extracted from them; however, this approach is not feasible for the global SfM algorithms due to the computational complexity. Hence, the second group of approach is emerged for this purpose. In this approach, the unknown focal lengths are estimated

by using the two view relations in the dataset, where unknown focal length is estimated by using its connected pair with known focal length [62, 8]. In general, these methods are limited to the variance of focal lengths in the dataset and it is required to know at least one of the focal lengths in image pair. In details, this causes the error propagation, and increase in the computational complexity of bundle adjustment step. In [8], the unknown focal lengths are estimated by using the connected views, in which one pair has known focal length. This method utilizes Gröbner basis approach to solve polynomial equations, which gives the estimated focal lengths. Another work [62] proposes an approach using two views relations in a connectivity graph of SfM pipeline, which again use the known focal lengths to estimate unknown ones in a two view relation. However, these approaches are able to solve focal length estimation problem in large datasets up to some extend. That is, the performance of this approach depends on the ratio of unknown focal lengths in dataset and the variation in the focal length distribution. Also, these methods are open to error propagation as the number of image in dataset increases [11, 12].

In this thesis, we propose another approach for estimating the unknown focal length using geometric constraint and without necessity to use known focal lengths for each pair. That is, it is not necessary to known one of the focal length in the adjacent relation in order to estimate unknown focal length, only their relation, which is computed using geometric constraint, is critical. Then, each relation is utilized in a novel cost function defined to estimate all unknown focal lengths at one step. The proposed method is able to estimate the unknown focal lengths for both sequential and global SfM algorithms as an initialization step. Hence, this makes bundle adjustment step converge fast with the accurate initialization.

CHAPTER 3

STRATIFIED ESTIMATION OF FOCAL LENGTH AND RADIAL DISTORTION

In this thesis, we study the problem of estimating radial distortion parameters and fixed focal length from two views. We have two main goals in this section: a computationally simple algorithm and robustness against image pixel noise. The proposed method operates in two successive stages: first, we use one parameter division distortion model of [19] and estimate a radial distortion parameter via transforming the problem to a quadratic eigenvalue one. After this step, we undistort the image correspondences and estimate the focal length by enforcing the trace constraint over the computed fundamental matrix by using the relation between the fundamental matrix and the essential matrix.

The contributions of our method is twofold: firstly, we show that the scaling parameter used in [19] for normalizing image coordinates can be theoretically up to an arbitrary scale. Depending on the utilized scale, the estimated radial distortion coefficient should change. However, using the input scale parameter and the estimated distortion coefficient corresponding to this scale produces the *same* undistorted image coordinates whether the scale is set to *including the correct parameter* or not. Therefore, we show that it is possible to remove the effects of the radial distortion problem from the image correspondences *without* the requirement of finding the correct radial distortion parameter. The true radial distortion parameter for the cameras can be obtained, only after the true focal length is utilized as the scaling parameter.

Our second contribution is the introduction of a new algorithm for estimating the focal length from a fundamental matrix between two images that share exactly the same

focal length. We extend the method of [7] for situations, where both focal lengths are not known for both of the images, as opposed to only one of them being unknown by developing the trace constraint equation (Appendix A) for the essential matrix in terms of the unknown focal length into a set of polynomial constraint equations. We introduce different ways of solving these equations and investigate their robustness under noise.

It should be noted that the methods in [34, 35] employ the same constraint for the estimation of the focal length. The unknown focal length is obtained by utilizing the first and second derivatives of the constraint equation. However, as reported by the authors themselves, the developed solution is prone to severe 'imaginary focal length' degeneracy and is not able to operate for fixated camera cases, i.e. when the epipolar line passes through the principal point. The proposed method, on the other hand, does not suffer from any of these degeneracy. Non-fixated camera assumption is observed to be a limiting condition for practical uses; even though the epipolar line might not pass through the image center, if it passes close to it, it causes the solution to be badly conditioned. Hence, these methods are not practical.

In this section, we explain our proposed solution for estimating both the focal length and the radial distortion parameters of a camera using two images of a general scene. We assume both of the images are captured with the same camera without any change in their parameters and from two different viewpoints. We further assume that the camera principal point is known and radial distortion has a first order characteristic with a single unknown parameter.

Our method operates in a *stratified* manner: we first remove the distortion effects without any knowledge about the camera parameters and without estimating the true distortion parameter and then estimate the focal length of the camera by using these undistorted image coordinates after imposing the so-called *trace constraint* over the estimated fundamental matrix. Once the true focal length is estimated, the true radial distortion parameter is recovered.

3.1 Radial Distortion Removal using Two-View Geometry

Let's define the intrinsic matrix and our distortion model more specifically. The camera is defined as an extended pinhole camera model with a one parameter division radial lens distortion proposed in [19]. The intrinsic matrix is defined as

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

where c_x and c_y are assumed to be at the center of the image, f is the unknown focal length and the distortion model is

$$p_u \sim \begin{bmatrix} x_d \\ y_d \\ 1 + \lambda {r_d}^2 \end{bmatrix}. \tag{3.2}$$

where $p_u = [x_u, y_u, 1]$ is the undistorted image coordinates, $p_d = [x_d, y_d, 1]$ is the distorted image coordinates, $r_d = \sqrt{x_d^2 + y_d^2}$ is the distance of the distorted coordinates from the image center and λ is the unknown radial distortion parameter. From this point on, we assume that the image coordinates (i.e. $(x_., y_.)$) are given with respect to a coordinate system center at the image origin.

In the next subsections, we will first show how to recover undistorted image correspondences from the distorted ones using the division model without any knowledge about the camera and then present our method for recovering the true focal length and the distortion parameter.

In this paper, as explained above, we use the extended pinhole camera model with a one-parameter division radial distortion model proposed in [19] (which is also used in [30], [41], [38] and [31]). According to the division model, an undistorted image point, $p_u = [x_u, y_u, 1]$ is computed from a corresponding distorted image point, $p_d = [x_d, y_d, 1]$ by using the formula:

$$p_u \sim \ p_d/(1 + \lambda r_d^2). \tag{3.3}$$

Developing this equation projectively, we have:

$$\begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \\ 1 + \lambda(x_d^2 + y_d^2) \end{bmatrix} \tag{3.4}$$

$$= \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} 0 \\ 0 \\ x_d^2 + y_d^2 \end{bmatrix} \tag{3.5}$$

$$\mathbf{x_u} = \mathbf{x_d} + \lambda \mathbf{r_d} \tag{3.6}$$

In [19], a quadratic eigenvalue problem (QEP) [63, 20] is formed by inserting this new definition for an undistorted point to the epipolar constraint:

$$\begin{aligned} (\mathbf{x'_d} + \lambda \mathbf{r'_d})^T F (\mathbf{x_d} + \lambda \mathbf{r_d}) &= 0 \\ \mathbf{x_d'^T} F \mathbf{x_d} + \lambda(\mathbf{r_d'^T} F \mathbf{x_d} + \mathbf{x_d'^T} F \mathbf{r_d}) + \lambda^2 \mathbf{r_d'^T} F \mathbf{r_d} &= 0 \end{aligned}$$

which is quadratic in the unknown λ and linear in F. Expanding and reorganizing the terms, the following equation is found:

$$(D_1 + \lambda D_2 + \lambda^2 D_3)\mathbf{f} = 0 \tag{3.7}$$

where **f** represents the entries of the fundamental matrix in column form and the three design matrices are equal to:

$$\begin{aligned} D_1 &= [x'_d x_d \;\; x'_d y_d \;\; x'_d \;\; y'_d x_d \;\; y'_d y_d \;\; y'_d \;\; x_d \;\; y_d \;\; 1] \\ D_2 &= \begin{bmatrix} 0 & 0 & x'_d r_d^2 & 0 & 0 & y'_d r_d^2 & x_d r_d'^2 & y_d r_d'^2 & r_d^2 + r_d'^2 \end{bmatrix} \\ D_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_d'^2 r_d^2 \end{bmatrix} \end{aligned} \tag{3.8}$$

Equation 3.7 is a quadratic eigenvalue problem (QEP) [63, 20], in which the aim is to find λ values assuming that f is a nonzero vector. The solution for this problem falls into two groups [63], namely the direct QEP solvers and the ones that linearize this problem and solve it as generalized eigenvalue problem. The latter group of solution requires D_1, D_2, D_3 matrices to be in a specific forms whereas the former group

approach propose a general solution. Hence, in the scope of this thesis, the second group of solution is utilized, which is available as MATLAB function , "polyeig". This solver is also implemented in our library[1] written in C++.

With the proposed solver, the QEP above has at most 10 solutions for which in practice no more than 6 are real [19]. The straightforward solution of this equation, however, is problematic due to numerical factors and therefore it is recommended in [19] to scale the image coordinates with the diagonal length of the image. Subtracting the image center and scaling the image coordinates is equivalent to normalizing the image coordinates using the inverse of the intrinsic matrix, i.e. dividing image coordinates by the focal length is equivalent to scaling. However, since focal length is not known, the image diagonal is proposed as a way to normalize image coordinates.

In theory, any scaling will not change the solution to the Equation 3.7 for the fundamental matrix. However, by scaling the coordinates, the estimated λ will be scaled with $\frac{1}{scale^2}$. This can be seen easily by going back to Equation 3.6 and making the scaling of x and y coordinates explicit. Additionally, while undistorting the image coordinates, the same scaling coefficient needs to be used while using the estimated λ. In Figure 3.1, we give an example of the recovered undistorted coordinates when using different scale parameters. As can be seen, results are equal to the order of -10. This means that, so long as a scaling parameter is used that prevents numerical errors in the solution of Equation 3.7, one can use the estimated λ value to recover the undistorted image coordinates.

In brief, in order to find out the undistorted image coordinates, we propose the following method: we first extract the image center from the correspondences and then scale them with 1000. Afterwards, within a RANSAC loop, we select 9 correspondences randomly and estimate the radial distortion parameter using Equation 3.7. As previously explained, this results in 4-6 solutions. As suggested by [19], we collect these possible solutions and select the median value of these as the λ-estimate for the 1000-scale solution. It is also possible to use kernel voting [45] for λ estimation. However, based on our synthetic data experiments, median based method has better accuracy under noisy observations.

[1] This is publicly available in `https://github.com/akcalakcal/Open_SfM_METU`

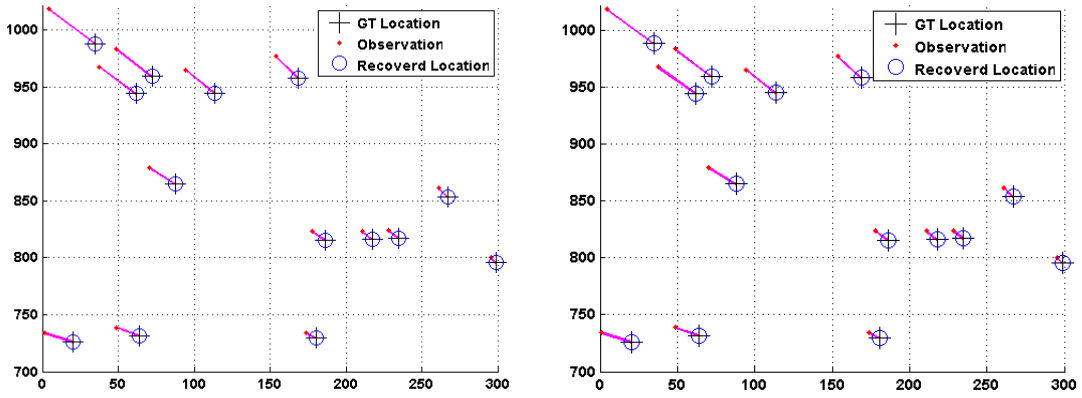


Figure 3.1: Recovering undistorted coordinates. Experiments are conducted for different scale parameters, i.e. $s = 1000$ (*left*) and $s = 4000$ (*right*). Mean square error values between the groundtruth location and the recovered location are $8.55x10^{-10}$, $6.75x10^{-10}$ for $s = 1000$ and $s = 4000$, respectively.

Once the radial distortion parameter is found with the selected scale parameter (i.e. 1000), we undistort the image coordinates according to Equation 3.6 using estimated lambda and the used scale. With these undistorted correspondences, we estimate the fundamental matrix using all the inliers of the previous stage via the standard normalized 8-point algorithm [21].

3.2 Trace Constraint Based Focal Length Estimation from Two Views

In the previous section, radial distortion is eliminated by using the epipolar constraint over the correspondences between the two views. In this section, we explain how to estimate the unknown focal length from the computed fundamental matrix. To reiterate, we assume that the focal length is the same for both of the images.

Our algorithm is an extension of [7]. In [7], the focal length estimation problem is formulated for the case where one of the images is calibrated and the focal length is found for only the second image. We follow a similar line of formulation where the trace constraint equation over the essential matrix is developed to find a polynomial in terms of the unknown focal length parameter with the estimated fundamental matrix. However, since we assume both of the images are uncalibrated, our polynomial is of higher degree.

The essential matrix has the property that its two singular values are of equal magnitude and the third singular value is zero. The trace constraint given below enforces this property (detailed proof is given in [50] and Appendix A):

$$2EE^TE - tr(EE^T)E = 0 \tag{3.9}$$

The above equation can be rewritten in terms of the fundamental matrix by a change of variable via $E = K^TFK$. Here, we use a simplified intrinsic matrix definition $K = diag(f, f, 1)$ for notational simplicity (fundamental matrix is computed accordingly from zero-centered coordinates):

$$G(f) \overset{\text{def}}{=} 2K^TFKKF^TK^TK^TFK - tr(K^TFKKF^TK^T)K^TFK = 0_{3\times3} \tag{3.10}$$

The $G(f)$ matrix above is only a function of the focal length given the fundamental matrix. In the ideal case, all of its entries should be zero for the true focal length value, i.e. $G(f) = 0_{3x3}$. However, due to noise, this does not occur and therefore to find the focal length the norm of the matrix $||G(f)||^2$ is minimized which is a 12^{th} degree polynomial whose coefficients are calculated using the entries of F, which is written in details in Appendix B. The minimization is accomplished by finding the roots of $\frac{dG(f)^TG(f)}{df} = 0$, an 11^{th} degree polynomial. Rejecting the $f = 0$ trivial solution, the local minima of this polynomial yields the correct focal length.

The elements of $G(f)^TG(f)$ matrix with row-major indices (1,2,4,5), (3,6,7,8) and (9) are 12^{th}, 10^{th} and 8^{th} degree polynomials, respectively. Since all elements of the matrix are non-negative and their sum is equal to zero in the ideal case, the value of each of the elements should be equal to zero (or should be as close to zero as possible for the noisy data case). Therefore, one can utilize these polynomial equations (Appendix B) individually to estimate a focal length value as well. If the $f = 0$ trivial solution is rejected the degrees of all the polynomials reduce to 8^{th}.

During our experiments, we have observed that solving the individual elements of the $G(f)^TG(f)$ matrix also produces the correct focal length; i.e. the roots of the 2^{nd}, 4^{th}, 6^{th} and 8^{th} entries of the $G(f)^TG(f)$ matrix produce the same focal length

estimates. Additionally, we have also observed that using the remaining entries of the $G(f)^T G(f)$ matrix for focal length estimation is quite susceptible to be corrupted by the noise and baseline of the camera pair. Therefore, compared to using the complete norm solution, using the solutions of the individual elements produce a much more robust algorithm.

Typically, the elements of $G(f)$ are quite small and hence the coefficients of the norm polynomial is quite close to the computation limit of most numerical libraries. Therefore, we recommend scaling the value of f in the K matrix before computing the coefficients i.e. $f = \alpha \hat{f}$ with $\alpha = 1000$.

Once the focal length is found as explained above from the computed fundamental matrix, we correct the estimated radial distortion parameter found in the previous stage by the true focal length via:

$$\lambda_{true} = \lambda_{est} * \frac{scale^2}{f^2} \tag{3.11}$$

3.3 Experimental Results

In this section, we present experiments on synthetic and real data sets to validate our approach. First, we show on synthetically constructed data sets numerical stability, noise and robustness properties of our pipeline. For some of these data sets, we also compare the proposed approach against other methods present in the literature. As our method is composed of two stages, we investigate both of these stages separately in order to understand their strengths and weaknesses. Finally, we demonstrate the performance of our approach on real data sets captured with different cameras.

3.3.1 Synthetic Data Experiments

In order to measure the accuracy and robustness of our proposed solution quantitatively, we generated synthetic sequences with known camera parameters and motions. The procedure for data generation can be explained as follows: 200 3D points within a cube centered at the origin are initialized randomly and projected onto two cameras

placed d_{cam} units away looking in the direction of the points with θ_{cam} degree angle between their z-axes. The angle separation mimics a turn-table type motion and it is randomly selected from the interval $[20, 40]$ degrees. The cameras are then initialized to have a 1024x1024 pixel resolution with a specific focal length and radial distortion parameter. Depending on the experiment below, we add different amounts of noise to the distorted image coordinates and use these in our experiments to recover the true focal length and the true radial distortion parameters.

3.3.1.1 Radial Distortion Removal Results

In this part, we measure the accuracy and robustness of removing radial distortion from the available image correspondences without knowing the true camera focal lengths. Here, we compare the performance of three methods: the method that we are using in our pipeline [19], a kernel voting based method [45] and a Gröbner based one [10]. It should be noted that the second algorithm operates very similarly with [19], except that the radial distortion parameter selection at the end is accomplished via a kernel voting scheme. Therefore, we use our own implementations below for the first two algorithms. The third algorithm [10], however, is relatively difficult to implement but its code is available online. We use the original implementation by the authors during the experiments below.

Numerical Stability for Radial Distortion Removal: Numerical stability of each of three methods, [19, 45, 10], is evaluated by using noise-free data with relative error between the estimated λ and the ground truth, $(\lambda - \lambda_{gt})/\lambda_{gt}$. The experiment is run for 5000 times and the resulting error distribution is shown in log_{10}-scale in Figure 3.2. As it is observed from the figure, the Gröbner-based method has poor numerical stability compared to the other two.

Noise Sensitivity for Radial Distortion Removal: In this experiment, distorted image pixel coordinates are perturbed with Gaussian noise of varying standard deviations $\sigma = [0, 0.01, 0.1, 0.5, 1.0]$ and the solvers are executed 300 times for each σ value. Then, the error between the estimated λ value and ground truth is calculated in a log_{10}-scale. As it can be observed in Figure 3.3, Fitzgibbon's method [19] performs

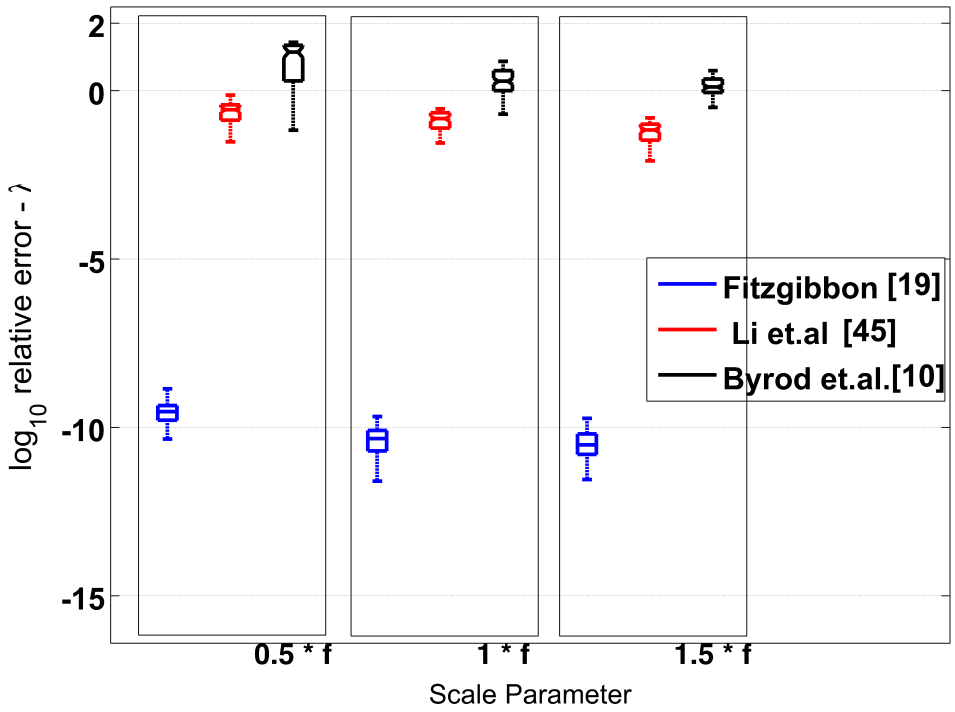


Figure 3.2: Comparison of λ estimation methods in terms of numerical stability of scale parameter

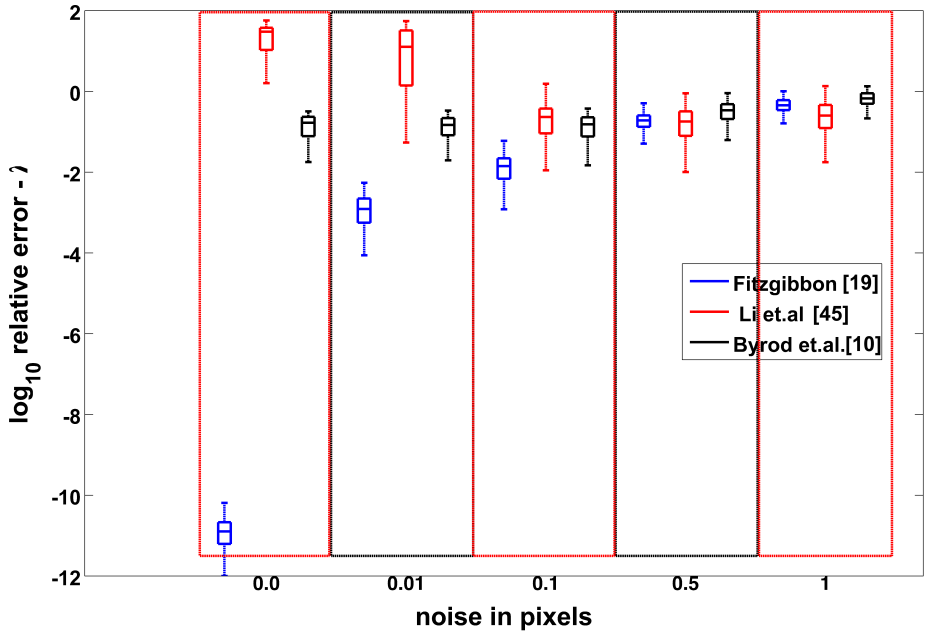


Figure 3.3: Noise sensitivity experiments are conducted for each method. Radial distortion is estimated 300 times and then distribution of relative error is plotted by using BOXPLOT function of MATLAB. Each box represents the values in between $\%15$ and $\%75$ of the distribution whereas horizontal line in each box stands for representing the median value in the distribution.

better than other the two methods in terms of noise sensitivity.

3.3.1.2 Focal Length Estimation Results

In this section, synthetically generated 3D points are projected onto the image planes without any radial distortion with the assumption that radial distortion is eliminated by algorithm that is explained in the previous step. The corresponding image pixel coordinates are utilized in order to estimate the focal length for the camera with the assumption that focal length values are identical.

Similarly, we compare the proposed algorithm for focal length estimation against other methods from the literature [60] [44]. All of these methods, including the proposed one, only require the fundamental matrix as input and produce an estimate for the focal length. As a result, the quality of the input fundamental matrix plays an important role in the estimated focal lengths for each of these algorithms. Here, we compare these methods in terms of their noise sensitivity and robustness to outliers with the same fundamental matrix estimated by using the normalized 8-point algorithm [22] within a RANSAC framework. We should also note that to avoid a degenerate turn-table type motion [60], we generate the poses of the camera in such a way that the camera translates along the x-axis and has small rotational perturbations, i.e. it is not strictly a pure translation motion. Before going forward with the comparisons, we first compare the robustness of two different constraint sets that have been introduced in Section 3.2, namely using the minimization of the norm of $G(f)$ or its individual elements.

Constraint Selection: In Figure 3.4, we compare the robustness of different constraint sets under noise with respect to small motions. In these experiments, we change the rotation angle between the two cameras and added $\sigma = 0.2$ Gaussian noise to the projected image coordinates and then tried to recover the focal length using the polynomial equations found by either using the norm of the $G(f)$ or using individual elements of the $G(f)^T G(f)$ matrix. We ran the experiment 300 times and then show the percentage of the truly recovered focal lengths in the y-axis. We assumed the focal length as true if it is within 5% of the actual value. As it can be observed from the results, as the camera axes start to align, i.e. rotation angle between

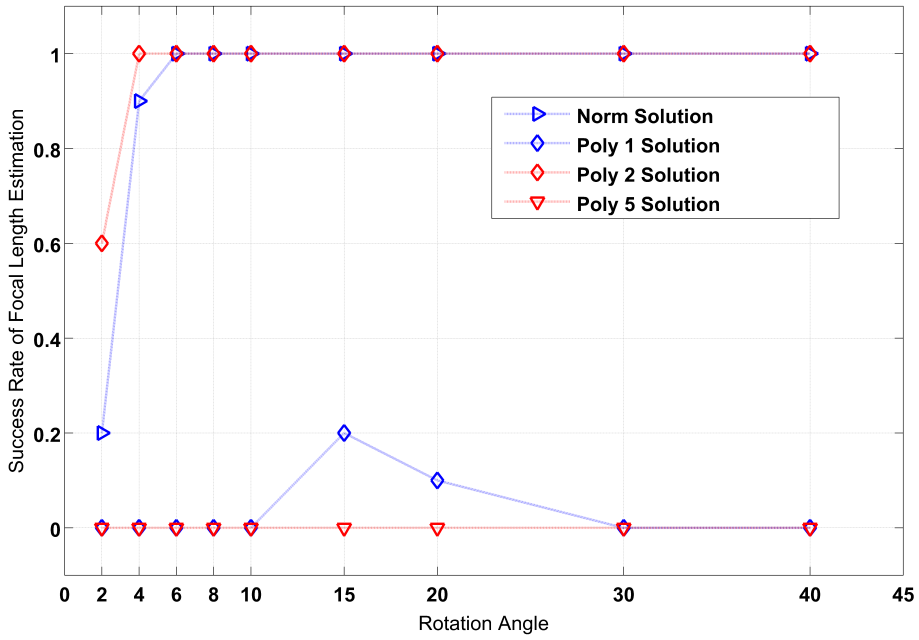


Figure 3.4: Success rate of focal length estimation is evaluated for different rotation angles. Gaussian noise with 0.2 standard deviation is added to the corresponding points. The focal length estimation is accepted as successful if the estimation error is less than $\%5$ of the true value. Success rate is observed for different solutions proposed for out method. There are three groups of solutions for which success rates are the same for all elements in a group. These three groups consists of the polynomials numbered as $\{1, 3\}$, $\{2, 4, 6, 8\}$ and $\{5, 7, 9\}$. Norm solution is compared with one of the polynomial solutions in each group for simplicity.

the axes gets smaller, the robustness of using the full 11^{th} degree polynomial suffers more compared to individual elements due to numerical problems. Hence, in practice, we use the focal length estimate from the $\{2^{nd}, 4^{th}, 6^{th}, 8^{th}\}$ polynomial entries of the $G(f)^T G(f)$ matrix. The remainder of the experiments will only be presented for this case.

Noise Sensitivity for Focal Length Estimation: In this experiment, undistorted image pixel coordinates are perturbed with Gaussian noise of varying standard deviations. This experiment is conducted with σ values of [0,0.01,0.1,0.5,1.0]. The proposed solver and the others are run 100 times for each σ value. Then, the relative error

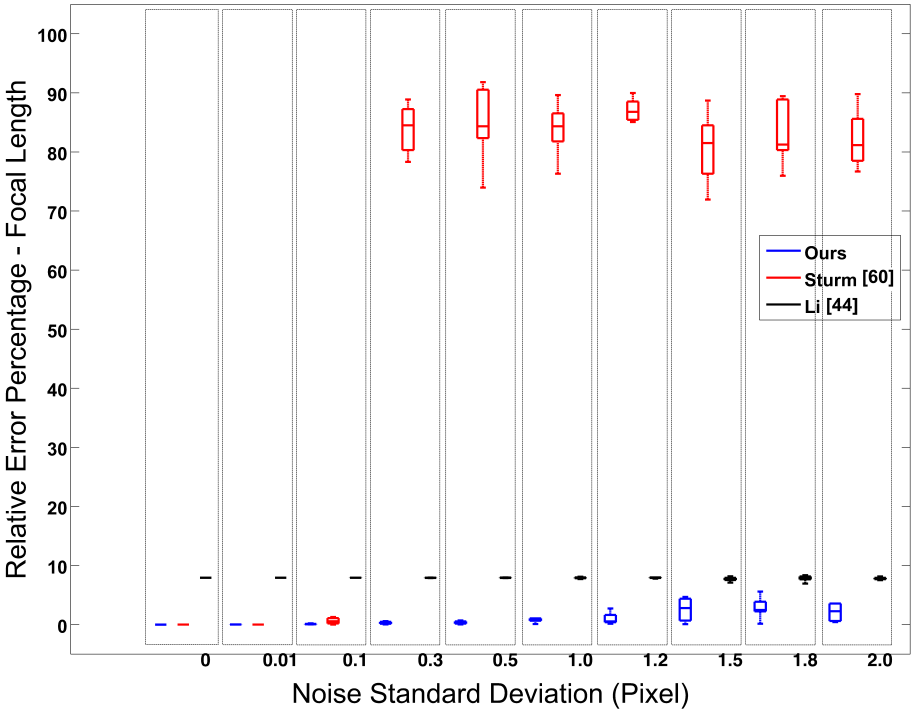


Figure 3.5: Radial distortion free corresponding points are perturbed with different Gaussian noise levels. Afterwards, focal length is estimated by using our proposed method, Sturm [60] and Li [44]. In this experiment camera motion is modeled so as to be a non-degenerate case for [60] in order to compare methods. For each noise level, focal length estimation algorithm runs for 100 times and the resultant percentage accuracy distribution is obtained. Distribution of relative error is plotted by using BOXPLOT function of MATLAB. Each algorithm is labeled with a different color. The line inside each box shows the median value of the distribution. Estimation results of Li [44] is strongly affected by noise, because the kernel voting is used with fixed parameters. Nonetheless, the proposed method by Sturm [60] is not numerically stable when compared with our solution, even though the noise sensitivity of [60] is comparable with our solver.

is computed between the true focal length and the estimated one in a log_{10}-scale. In Figure 3.5, the results are shown for comparison. The proposed method performs quite well compared to others. Additionally, while Sturm's method [60] works as good as ours for low noise situations, it quickly becomes unusable above $\sigma = 0.3$

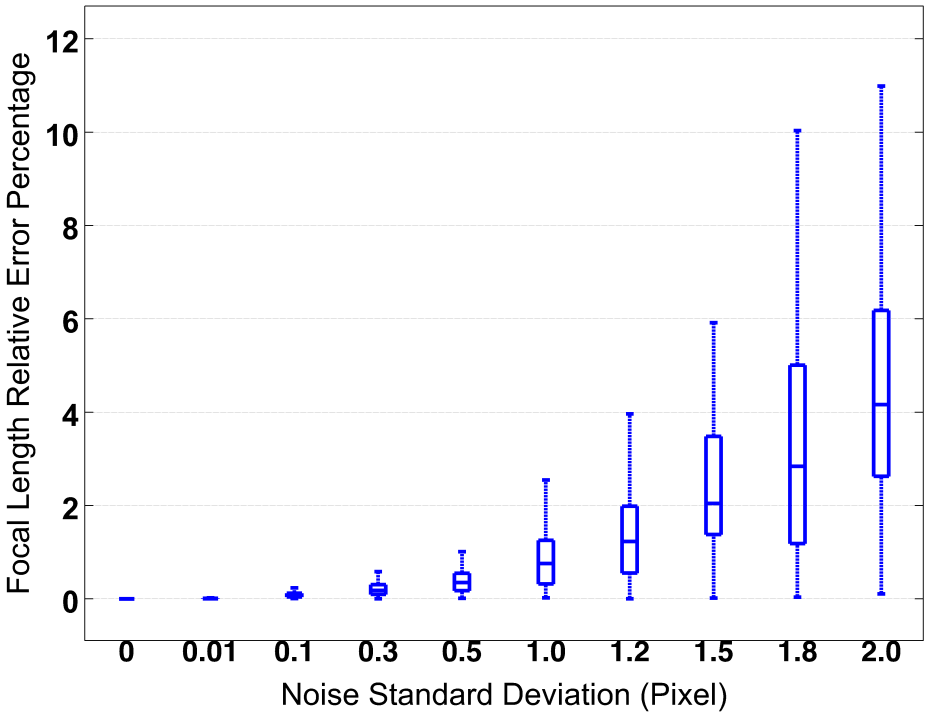


Figure 3.6: Radial distortion free corresponding points are perturbed with different Gaussian noise levels. Fundamental matrix is then calculated from those corresponding points. Afterwards, focal length is estimated by using our proposed method. In this experiment turntable motion is modeled with 20 degree rotation angle. For each noise level, focal length estimation algorithm runs for 100 times and the resultant percentage accuracy distribution is obtained. Distribution for each noise level is plotted by using BOXPLOT function of MATLAB. The blue box represents the values in the range of %15 and %75. Line inside the blue box shows the median value of the distribution.

noise levels. We also show the performance of our method separately for increased detail in Figure 3.6 with a turn-table type motion.

3.3.2 Real Data Experiments

In this section, we evaluate the overall pipeline for radial distortion removal and focal length estimation by using real data. We generate datasets by using a Commercial off-

the-shelf (COTS) camera (GoPro HERO 4 Black), which has a large Field-of-View (FOV). This camera has three different FOV settings, namely wide ($\sim 140°$), medium ($\sim 115°$) and narrow ($\sim 80°$). Different FOV values mean different amount of radial distortion. For medium and wide FOV modes, we capture 25 image pairs [2]. In Figure 3.7 (a-b) and (c-d) we show the example of radially distorted and undistorted image pairs captured by GoPro camera with medium and wide FOV settings.

Figure 3.7: In this experiment, stereo image pairs are captured by using GoPro Hero 4 Black camera in two different FOV mode, namely, Medium - $\sim 110°$ and Wide - $\sim 140°$ FOV modes. Images denoted by (a-b) and (c-d) are radially distorted and the undistorted images, respectively, in which removal of radial distortion is done by using our pipeline. Then focal lengths are estimated from these image pairs.

In these experiments, we use a resolution of 1920x1080 pixels for both cameras. First, SIFT features [47] are detected for both images and then tentative (putative) matches are obtained. The matches whose values are under certain threshold are selected as *best matches* and they are utilized for the estimation of the radial distortion parameter. The tentative matches are undistorted by using the estimated radial distortion parameter of the division model (3.6). Next, we run RANSAC algorithm with the configuration where the minimum number of iteration is 300 and the threshold is 3 pixels in order to compute a fundamental matrix, F, by utilizing these undistorted

[2] Datasets is publicly available at akincaliskan.net/focalDataset.zip

tentative matches. The computed F matrix is given as input to our proposed algorithm to estimate the focal length and the accuracy of out results is measured using a calibration pattern based method [3] . As it is stated before, the focal length estimation algorithm is run after the removal of radial distortion for 25 image pairs with different FOV values. In Figure 3.8, estimated focal length f_e and ground truth focal length f_{gt} values are plotted for datasets. We can see that the estimates for solvers are close to the ground truth value from the calibration. The median of relative error percentage distribution for $110°$ and $140°$ Field-of-View value are 7.18 and 8.11, respectively.

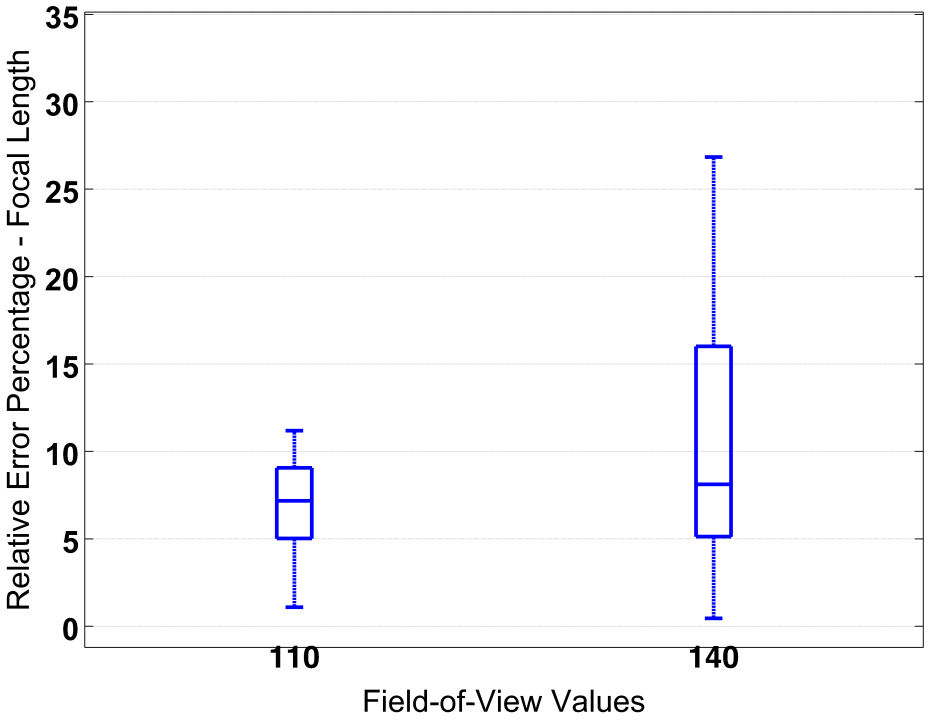


Figure 3.8: Proposed pipeline is evaluated by using randomly chosen 25 image pairs which are radially distorted and recorded by GoPro HERO 4 Black camera, in Medium and Wide FOV mode ($110°$ and $140°$ respectively). First radial distortion is removed, then focal length is estimated for each image pairs. As a result, estimated focal lengths are compared with the one that is computed by using calibration pattern, labeled as f_{gt}. That is, relative error percentage is computed by using ground truth focal lengths.

[3] C++ calibration code is publicly available at akincaliskan.net/thesisCodes.zip

CHAPTER 4

GROUP SYNCHRONIZED FOCAL LENGTH ESTIMATION FROM VIEWING GRAPH

Accuracy of the model generated as a result of SfM algorithms relies on the estimation accuracy of the intrinsic camera calibration parameters, especially the focal length estimate. Even if the focal length values can be estimated during the bundle adjustment step of any SfM pipeline, estimating the focal length correctly before the bundle adjustment step reduces the time consuming for any SfM pipeline. In the literature, as previously explained in Chapter 2, there are algorithms that is capable of estimating the focal length in a sequential order from the input dataset by using two view geometry. In this case, the accuracy depends on the variance of the values of focal lengths; mostly, it is preferred that the unknown focal lengths are equal to each other. Nevertheless, in this thesis, we propose a method for estimating focal lengths of each camera in SfM dataset. This method exploits the geometric relation between two views in order to derive a linear relation between focal lengths, which differentiate the proposed method from some other two view focal length estimation algorithms [39, 34, 35]. In other words, a novel constraint is proposed from the geometric constraint and group synchronization approach [4] is utilized with this constraint. Moreover, in this thesis, we test our method against different initialization scenarios, i.e. sequential focal length estimation using camera relations, and the one without any initialization. The proposed group synchronization based method outperforms the previously proposed focal length estimation algorithms for large SfM datasets.

It is easy to estimate the focal length of one camera, if the fundamental matrix de-

fined between two views and the focal length of the other camera are both known. This relation can be represented by a non-linear function, $P(\hat{F}, f2)$, Equation 4.2, where the details of this function is given in Appendix B. The value of this function turns out be equal to zero due to the trace constraint relation between two cameras, i.e. $P(\hat{F}, f2) = 0$. Using this relation, it is possible to solve for unknown focal length, if the other one is provided. The usage of trace constraint for estimation of unknown focal length is previously proposed in [7]. In Chapter 3, estimation of both of the unknown, but equal, focal lengths of two views is proposed. This solution is extended by the idea that there is one fundamental matrix available and none of the focal lengths of these two views are known. In this case, we only have nonlinear functions, $H(F, f1, f2)$, Equation 4.1 if cameras share the same focal lengths, and we require to derive a relation between these two focal length values if they are not the same. For this purpose, the possible focal length values are computed from the possible field of view hypothesis for the first camera. Then, the correspondent focal length values for the second camera are estimated for each of these hypotheses. As a result of this, the linear relation between focal lengths can be computed, which is called as constraint in this thesis. The selection of this function is important; i.e., if there is an approximate linear relation between focal lengths, it will be possible to extend our focal length estimation formulation as a group synchronization problem which is feasible for cost function modeling that can be solved by non-linear optimization methods.

4.1 Linear Modeling Between Focal Lengths

There is no linear relation defined between focal lengths of two adjacent cameras in case of only the fundamental matrix is available. If the two views share the same focal length, then this focal length is computed by using the geometric constraint between them (Chapter 3), or if one of the focal length is known, then the other camera's focal length is estimated by using the geometric relation between cameras [7]. This relation is based on trace constraint, which is explained in Appendix B. If the trace constraint is satisfied between two cameras, the Equation 3.9 holds for any Essential Matrix, E. Using this relation and replacing E matrix with the $E = {K_1}^T F K_2$, the following

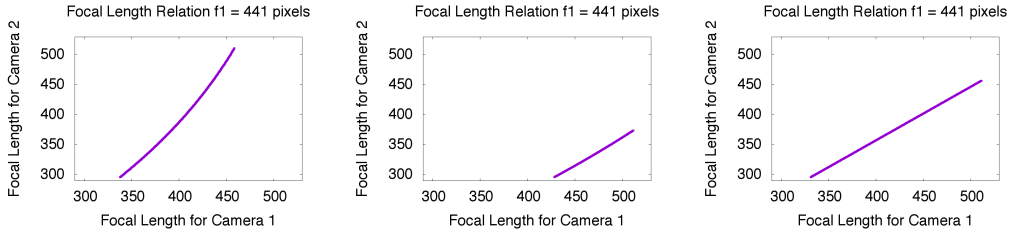
function is obtained whose value is equal to zero:

$$
\begin{aligned}
&H(F, focal1 = f, focal2 = f) = \\
&k_1 \times f^{12} + k_2 \times f^{10} + k_3 \times f^9 + k_4 \times f^8 + \\
&k_5 \times f^7 + k_6 \times f^8 + k_7 \times f^9 + k_8 \times f^8 + \\
&k_9 \times f^3 + k_{10} \times f^2 + k_{11} \times f^0 = 0
\end{aligned}
\tag{4.1}
$$

where coefficients, k_i, are defined in Appendix B. This relation can be replaced by another function, if one of the focal lengths is known. In this case, the known camera matrix, K_1 is merged with the fundamental matrix, F. Then, a new function, $P(\hat{F}, focal2)$, is obtained as below:

$$
\begin{aligned}
&P(\hat{F}, focal2 = f) = \\
&t_1 \times f^6 + t_2 \times f^4 + t_3 \times f^2 + t_4 = 0
\end{aligned}
\tag{4.2}
$$

where coefficients, t_i, are defined in Appendix B. This method for focal length estimation is very similar the one proposed in [7]. However, this utilized for derivation a linear relation between focal lengths, instead of finding an unknown focal length.



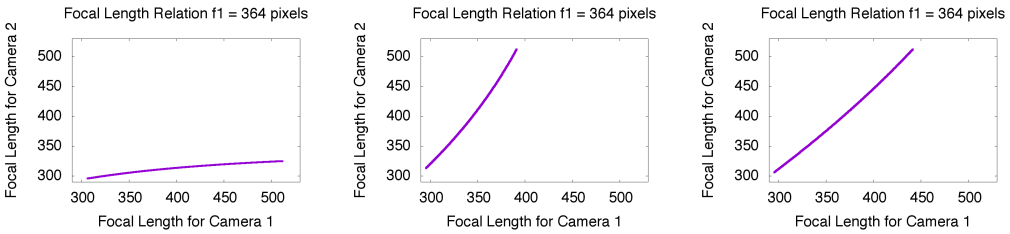
(a) Focal length, f_1, = 441 px, turntable angle,θ = 12 degree

(b) Focal length, f_1, = 441 px, turntable angle,θ = 18 degree

(c) Focal length, f_1, = 441 px, turntable angle,θ = 15 degree

Figure 4.1: Focal length relation under turntable motion. There are two cameras whose FOV value in the interval [90,120] and there is a turntable motion from first camera to second one in which the degree of motion is in the interval [10,20] degree. The Focal length of first camera is 441 pixels, and correspondent camera is chosen randomly from the adjacency matrix created during global SfM.

In this thesis, the linear relation is approximately estimated from the focal length pairs. These pairs are calculated by using Equation 4.2. For example, in Figure 4.1,

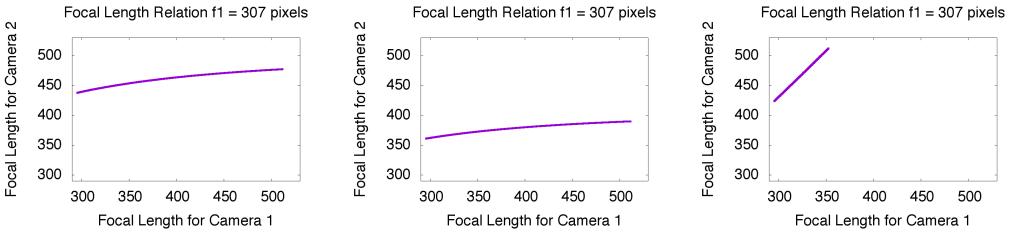


(a) Focal length, f_1, = 364 px, turntable angle,θ = 11 degree

(b) Focal length, f_1, = 364 px, turntable angle,θ = 16 degree

(c) Focal length, f_1, = 364 px, turntable angle,θ = 20 degree

Figure 4.2: Focal length relation under turntable motion. There are two cameras whose FOV value in the interval [90,120] and there is a turntable motion from first camera to second one in which the degree of motion is in the interval [10,20] degree. The Focal length of first camera is 364 pixels, and correspondent camera is chosen randomly from the adjacency matrix created during global SfM.



(a) Focal length, f_1, = 307 px, turntable angle,θ = 13 degree

(b) Focal length, f_1, = 307 px, turntable angle,θ = 11 degree

(c) Focal length, f_1, = 307 px, turntable angle,θ = 18 degree

Figure 4.3: Focal length relation under turntable motion. There are two cameras whose FOV value in the interval [90,120] and there is a turntable motion from first camera to second one in which the degree of motion is in the interval [10,20] degree. The Focal length of first camera is 307 pixels, and correspondent camera is chosen randomly from the adjacency matrix created during global SfM.

this relation is illustrated for different cases in which cameras are at different poses. These figures make it easy to visualize the linearization idea. In Figure 4.1, the focal length relations between a camera, which has focal length value of 441 pixels, and different cameras are plotted. The resulting plot shows that the focal length relation can be represented quite well by a linear function. This is followed by two other typical camera pairs, which are shown in Figure 4.2 and 4.3, for both of which a linear relation is observed again.

There is a linear function proposal in this thesis, which is called as $L1$, which maps one camera's focal length value to the other one's focal length after multiplying by a constant value. That is, two focal lengths are related by a *constraint*. Under this linear model, the estimation of the parameters of linear function can be simply obtained by least square fitting. As it is illustrated in Figure 4.1, 4.2, 4.3, most of the relations can be modeled by the $L1$ relation, and in the scope of this thesis, this relation will be assumed to be valid for focal length estimation.

4.2 Group Synchronization of a Graph

View graph [62] is a representation of a set of views that have pairwise geometrical relation, Figure 4.4. The aim of this study is to estimate focal length of each view, within the view graph assuming that some of the cameras have known calibration parameters. If all the relations within the graph is utilized for this purpose, this problem is then called as *synchronization* [4]. In other words, the goal of the synchronization problem is to infer the unknown states from the available measures in the network of nodes where the difference or the ratios of the connected nodes are known. The finite group is a set of limited number of elements together with group operation that satisfy the fundamental properties [4]. If the states of the nodes are represented by finite group elements, then the problem turns to *group synchronization*.

In order to define the focal length estimation from viewing graph problem in the scope a group synchronization problem, we need to represent it with the necessary mathematical relations. A finite digraph $\vec{G}$ is denoted by $\vec{G} = (V, E)$ with vertex set $V = \{1, 2, ..., N\}$ and edge set $E \subseteq \{1, 2, ..., N\}$ x $\{1, 2, ..., N\}$. In this thesis, $\vec{G}$ represent the viewing graph and it is assumed to be a *connected graph* unless otherwise stated. The viewing graph is also represented by an *adjacency matrix*, A, where $A_{ij} = 1$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise. Adjacency matrix is utilized in the optimization step of the group synchronization.

After fundamental definitions related to the graph theory, we need to define the *group synchronization* within the graph. Let $(\Sigma, *)$ be a *finite group* and the underlying set of this group is $\mathbb{R}$. In other words, the edge and vertex labeling of the graph,

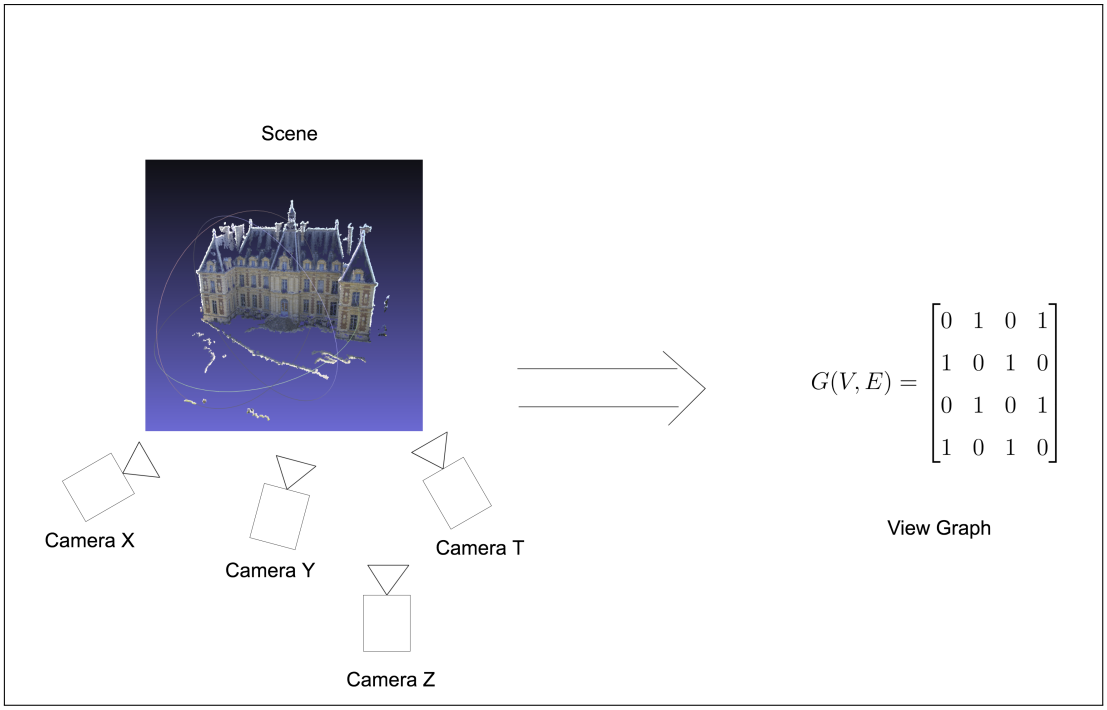


Figure 4.4: View Graph Generation. View graph is a representation of the relation between cameras in SfM algorithms [62]. "0" value stands for there is no matching between images on the camera position, where as "1" states that two images have enough points to be matched.

which will be mentioned later, are defined in $\mathbb{R}$. Since the aim is to find the group synchronization over the graph, we need to relate the finite group $(\Sigma, *)$ with the given graph using the proper labeling, which is called $\Sigma - labelled$ graph. If the labeling of the edge set of the graph is defined in Σ, then this is called $\Sigma - labelled$ graph. If the relation between the vertex label and the edge label defined by group operation $(*)$ hold for all the pairs in graph, then we can call it consistent labeling, i.e. *group synchronization of the graph*. Any estimation error which terminates this labeling results in a consistency error. Then, the group synchronization over the graph is not satisfied.

4.3 Trace Constraint Based Consistency Constraint

In our case, initially, each edge between any two nodes in the view graph is labeled by the Fundamental matrix, F, between two cameras viewing the same scene, as it is defined in any SfM algorithm; however, different from the case in [4], we should find a new labeling in order to use it during focal length estimation. The relation between two corresponding views is represented by F matrix, i.e. edge labeling, whereas the vertex labeling can be obtained by the corresponding focal length of each camera. Since F matrix and focal length, f, are defined in different underlying set of groups, namely $F \in \mathbb{R}^{3x3}$ and $f \in \mathbb{R}$, we should find edge labeling defined in the same set of f. In this study, we propose a method to extract edge labeling by using geometric relation between two views, i.e. fundamental matrix. As it is presented in Algorithm 1 below, in case of vertex labeling in which $x_i = f_i$, the edge labeling z_{ij} is computed from the fundamental matrix. The proposed method also satisfy the rule for being defined in the same set for both edge and vertex labeling.

4.4 Group Synchronized Focal Length Estimation Method

After defining the $\Sigma - labelled$ graph by the proposed labeling methods, now it is time to satisfy the group synchronization all over the graph by minimizing the *consistency error*. Consistency error is defined as the cumulative error within the graph between estimated edge labels and the edge labels computed from the ratio of estimated focal

Algorithm 1 Edge Labeling Algorithm

Input: Keypoint matches between connected nodes of the graph $\vec{G} = (V, E)$, $\forall P_i(x, y)$ and $\forall P_j(x, y)$. Possible focal length set, $S_f = \{..., f_k, .., f_l, ...\}$.

Output: Ratio of actual focal lengths of the connected nodes, x_i, x_j, i.e. cameras. Edge labeling result, i.e. z_{ij}

GENERATE the polynomial function, $P(\hat{F}, f_2) = 0$ from Equation 4.2.

INITIALIZE set P = { }, where focal length pairs are stored

for $f_1 \in S_f$ **do**

    COMPUTE $\hat{f}_2$ from $P(\hat{F}, f_2) = 0$

    ADDPAIR $(f_1, \hat{f}_2)$ to set P

end for

DRAW pairs in set P on two-dimensional plot. Estimate $f_2 = \alpha * f_1$

$A = B = 0$

for $(f_1, \hat{f}_2) \in P$ **do**

    A = A + ${f_1}^2$

    B = B + $f_1 * \hat{f}_2$

end for

$\alpha = A/B$

return $z_{ij} = \alpha$

lengths. In other words, our aim is to find the vertex values which yield the most consistent (estimated) edge labeling. This problem is modeled as an unconstrained optimization problem with an initial guess of vertex values, i.e. initial focal length values. After that, the rest of the problem is solved by a non-linear least square approach, namely Levenberg-Marquardt [49].

4.4.1 Initialization of Focal Length Values in Viewing Graph

Since unconstrained optimization problem is solved by non-linear least square method to obtain a solution, the initial vertex labeling values are crucial. In our formulation, a priori values of the vertex labeling values become the initial estimation of focal lengths of the cameras. Estimation of the focal lengths values can be obtained by selection of any path from the viewing graph. In this study, we also propose the robustness of the global synchronization methods to any path selected from viewing graph. However, path selection between two nodes (cameras) should be settled into a formulation. In order to make this path selection easier, we propose a method that find a maximum spanning tree (MST) starting from a vertex whose focal length is known a priori. The procedure is explained below in Algorithm 2: Only inputs to the algorithm is viewing graph with the related correspondences between connected nodes and the error threshold for the view graph's robustness to noise.

Although initialization is proposed as a crucial step for the next step, i.e. solution of the optimization problem, the experiments are conducted to show whether this step is necessary or not. For this purpose, in the experimental work, the initialization for unknown focal lengths are tested for both cases, namely initialization with MST, and initialization with arbitrary number. As a result of the experimental stages, arbitrary initialization is better than MST method, which is preferred in previous SfM studies [62]. Also, it is beneficial for the complexity of the overall solver to skip MST based initialization step, since the tree construction, and search in a tree are computationally costly for large set of datasets. Hence, arbitrary initialization is used in the proposed method. The result of experiments, that are explained in 4.5, shows that the proposed solver with given constraints is able to find the correct focal length values in accurate and fast way.

Algorithm 2 Focal Lengths Initialization In Viewing Graph

Input: Viewing graph, $\vec{G} = (V, E)$. Threshold value, T_{MSE} for the node elimination based on mean square error (MSE) of the edge labeling estimation.

Output: Focal length estimation path. $\{(node_i), (node_j), (node_k), ...\}$ starting from the node with known focal length

Change the edges of viewing graph with the number of correspondences between the nodes and name it as $\overrightarrow{G^u} = (V, E^u)$.

for Each edge $\in \overrightarrow{G^u}$ **do**

    Compute edge labeling z_{ij} from Algorithm 1

    Compute mean square error (MSE) of the estimation step: $err_{ij} = (\sum_j \hat{f_j}^2 - {f_j}^2)$

    if $err_{ij} > T_{MSE}$ **then**

        Remove the corresponding node from $\overrightarrow{G^u}$

    end if

end for

Estimate Maximum Spanning Tree [REF] from the $\overrightarrow{G^u}$.

Return the path, $\{(node_i), (node_j), (node_k), ...\}$, starting from the first node of the tree.

4.4.2 Non-Linear Optimization For Focal Length Estimation Problem

After the estimation of constraints of focal lengths of adjacent cameras, the set of focal length values which accounts for the minimum value of objective function are estimated. In this formulation, the objective function is non-linearly dependent to the focal length values; hence, this problem is solved by one of the common methods proposed by Levenberg-Marquat [49]. In order to satisfy the group synchronization, there should be a zero consistency error between the estimated edge labels and the computed ones from the vertex labeling. Let $Z_A = [z_{ij}]_{NxN}$ be a matrix consisting of the estimated edge labeling values from the fundamental matrix values according to the Algorithm 1. The value of the edge label for the same index in the Z_A is also computed from the vertex label, i.e. $z_{ij} = x_i/x_j$. By using the vector of the vertex labels, $X = \begin{bmatrix} x_1 & x_2 & ... & x_N \end{bmatrix}^T$ and $X_{inv} = \begin{bmatrix} {x_1}^{-1} & {x_2}^{-1} & ... & {x_N}^{-1} \end{bmatrix}$ the edge labeling matrix is also represented: $\hat{Z}_A = XX_{inv}$. The cost function for the group synchronization here defined as:

$$\min_{\begin{bmatrix} x_1 & x_2 & ... & x_N \end{bmatrix}^T} \sum_{k=1}^{NxN} \left(Z_A(k) - \hat{Z}_A(k)\right)^2 \tag{4.3}$$

where X_{init} is computed by using Algorithm 2 or initial values of unknown focal lengths are set to an arbitrary value. In order to solve the optimization problem, non-linear least square methods used with the choice of Cauchy Loss function. The implementation of the solver utilizes the Ceres library [2], which is the mostly preferred non-linear optimization library in SfM algorithms.

4.5 Experimental Results

The experiments are conducted with both synthetic and real data, which will be explored in details in the following part of the thesis. First of all, synthetic data experiments are designed to show how the focal length estimation method is robust to noise and how well is this proposed method solve the focal length estimation in case of minimum number of known focal lengths. Secondly, the real data experiments are set

up to justify the proposed method can be used in the SfM pipelines with the real data as an initial stage, and how successful is this method in the case of minimum number of known focal lengths.

4.5.1 Synthetic Data Experiments

The aim of the synthetic data experiments are three folded. First of all, the robustness of the method for estimation of focal lengths with given constraints is tested against noisy constants and the missing focal lengths values. Secondly, the robustness of the constraint in case of noisy pixel correspondences is investigated. Finally, this is followed by an experiment in which the full pipeline starting from point correspondences and ending up with the estimated focal length values is observed in the case of noisy pixel correspondences. In each group of experiments, the number of camera, that of missing focal lengths and the approximated linear functions are changing in order to observe robustness in a wide spectrum.

The synthetic experiments are conducted with randomly generated cameras whose field of view are selected from a realistic interval in practice. Poses between the cameras are computed from randomly generated rotation matrices and translation vectors. The pixel correspondences are computed by using these rotation matrices and translation vectors. In the first group of experiments, the constraints are computed from previously generated focal lengths and the focal length estimation accuracy is tested under different level of noise added to ground-truth constraint values. There are N number of cameras generated, and their FOV values are randomly selected from the interval $[90, 120]$ degrees. The size of images is selected 1024x1024 for convenience. The randomly generated focal lengths and the corresponding constraint are given as input to the focal length estimation algorithm which is proposed in Chapter 4.4.2, then the Gaussian noise with mean at zero and standard deviations at different values are added to the exact constraint values. These experiments are conducted for different number of cameras and various the number of broken focal lengths in each setup to investigate estimation accuracy relation between them.

In the first setup, an experiment is performed for the synthetically generated 10 cameras, 1 of them is broken, i.e. the broken focal length ratio is 10%. The robustness

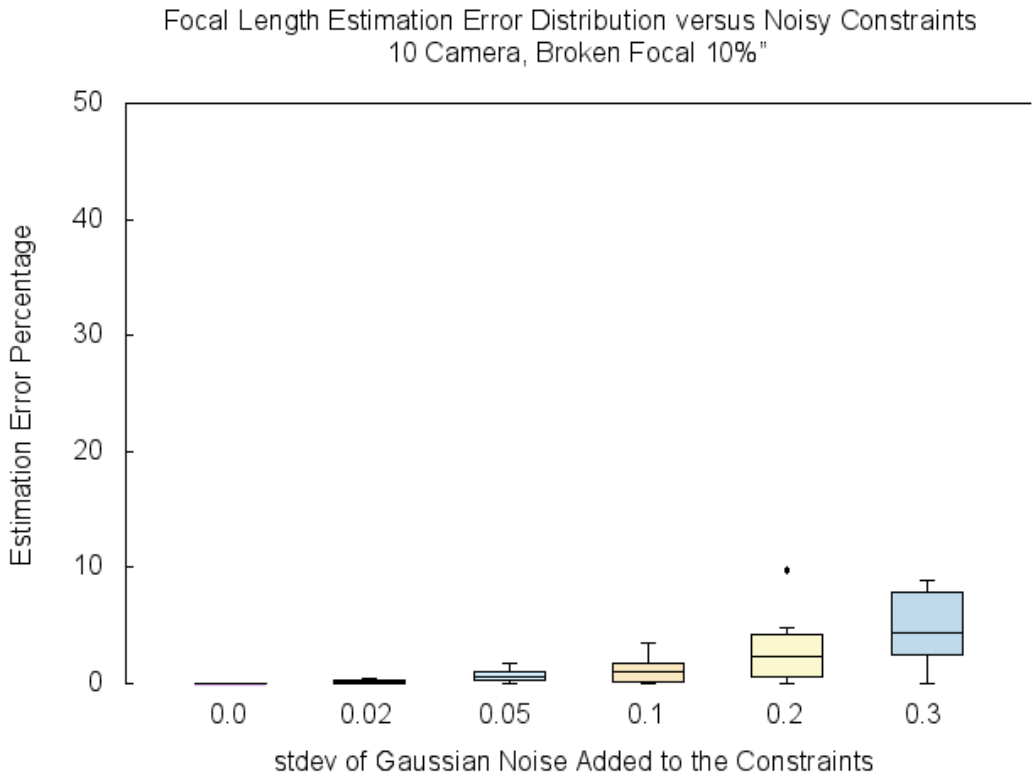


Figure 4.5: Robustness of Focal Length Estimation Algorithm Against Noise in Constraint Values. There are 10 cameras randomly generated and 1 of them have broken focal lengths. In this setup, the noise is added to the exact values of constraints which are calculaated using the $L1$ function defined in Chapter 4.1.

of the accuracy of estimation result is then plotted for different noise levels in Figure 4.5. The median of the focal length estimation error distribution increases while the standard deviation of the additive Gaussian noise gradually changing from 0 to 0.3. During the interval, the first three quantiles of the error distribution stays under 10%, which can be compensated by following steps in SfM pipeline. This result is due to the fact that the bundle adjustment algorithm handles the error under a certain level.

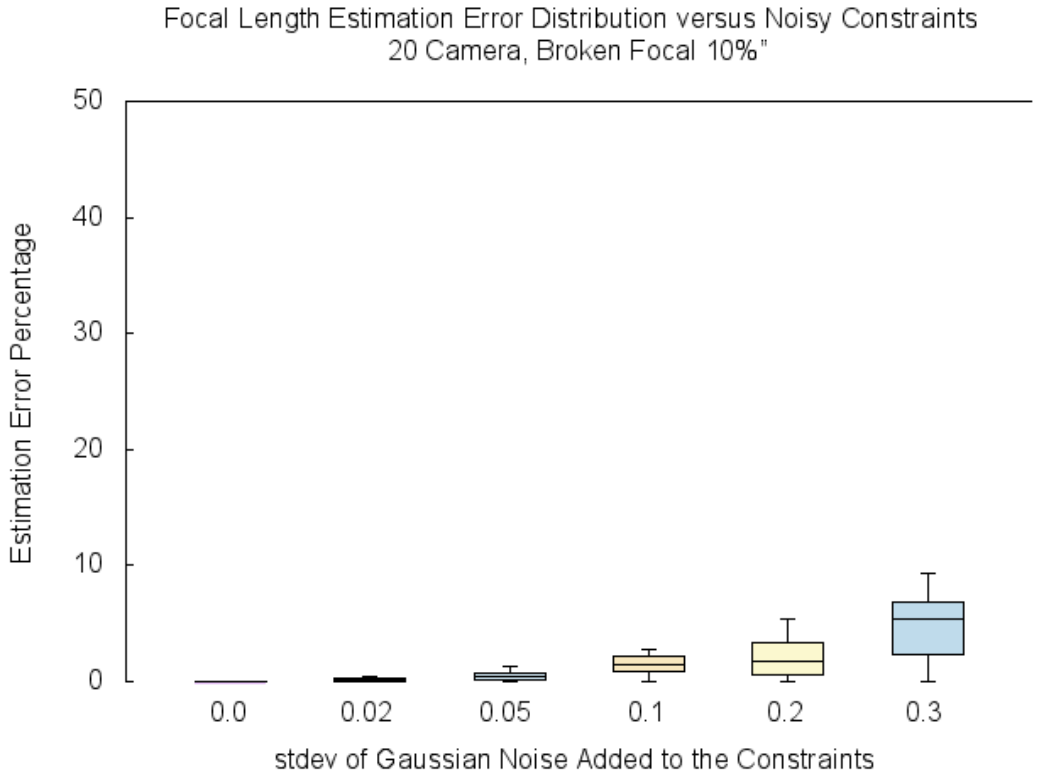


Figure 4.6: Robustness of Focal Length Estimation Algorithm Against Noise in Constraint Values. There are 20 cameras randomly generated and 2 of them have broken focal lengths. In this setup, the noise is added to the exact values of constraints which are calculaated using the $L1$ function defined in Chapter 4.1.

In the next phase of the experiments, the same setup is tested with different number of cameras but under same broken focal length ratio. In Figure 4.6, 20 cameras are randomly generated, and the the focal length estimation algorithm runs with constraints under different levels of noise. The trend of error distribution is slightly different than the one observed with 10 cameras, and the error distribution still stays in the safe region.

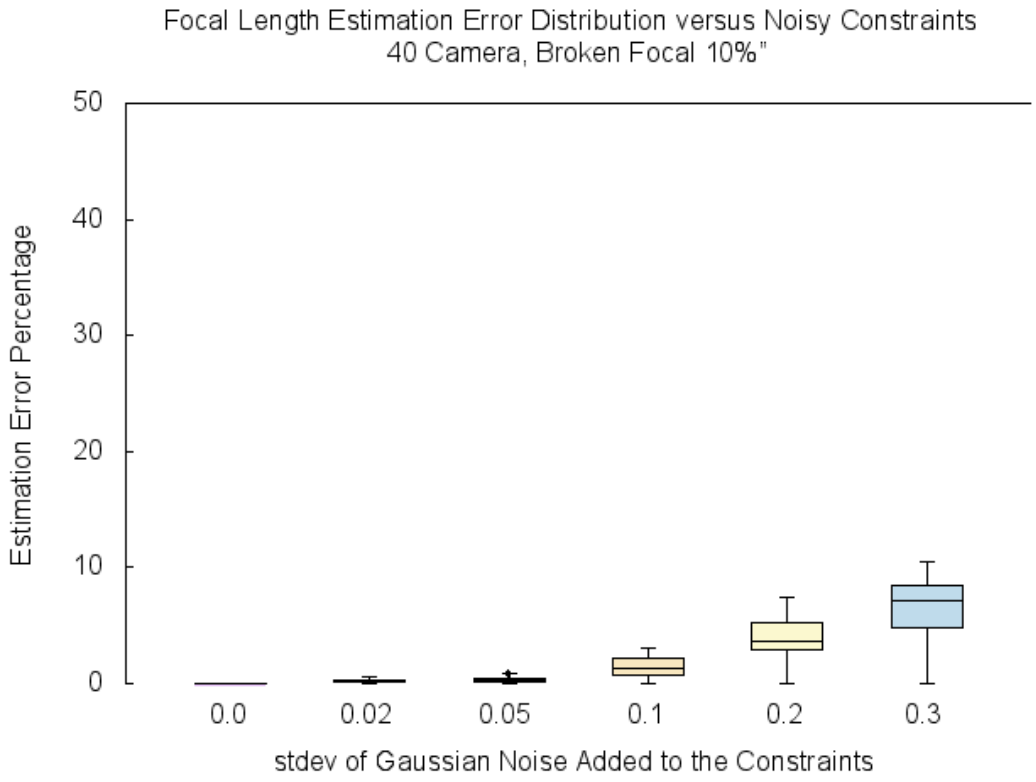


Figure 4.7: Robustness of Focal Length Estimation Algorithm Against Noise in Constraint Values. here are 40 cameras randomly generated and 4 of them have broken focal lengths. In this setup, the noise is added to the exact values of constraints which are calculated using the $L1$ function defined in Chapter 4.1.

This simulation is followed by other experiment conducted with 40 cameras with same broken focal length ratio at 10%. During the same noise interval, the error distributions are illustrated in Figure 4.7.

Different than the broken focal length ratio at 10%, the experiments for ratios at 20% and 50% are also conducted. The graphical results are not included in this thesis, but the important statistical measurements of estimation error distributions are tabulated in Table 4.1. In this table, the overall estimation error is presented; in some cases, the error is out of the secure region, and they are highlighted. Main trend shows that the ratio of broken focal length increases from 10% to 50%. Through this change, the accuracy of focal length estimation slightly declined. This is observed in all synthetic experiments with different cameras. According to the results of our synthetic data experiments, on the other hand, with a constant broken focal length ratio, it is observed that the accuracy of focal length estimation under noisy constraints slightly changes for the different number of cameras. However, the affect of noisy constraints is easily observed when the number of camera is at high level.

Table 4.1: Focal length estimation errors are shown in the table for different broken focal length ratios under noisy constraint values. The standard deviation of additive Gaussian noise are 0, 0.02, 0.05, 0.1, 0.2, 0.3. The ratio of broken focal lengths starts from 10% and gradually increases towards 50%.

		Broken f Ratio 10%					**Broken f Ratio** 20%					**Broken f Ratio** 50%				
		0.02	**0.05**	**0.1**	**0.2**	**0.3**	**0.02**	**0.05**	**0.1**	**0.2**	**0.3**	**0.02**	**0.05**	**0.1**	**0.2**	**0.3**
10 cams	**median**	0.161	0.559	1.083	2.558	4.873	0.378	0.760	0.910	1.774	5.934	0.619	0.390	1.831	3.541	8.128
	1^{st} **quan**	0.031	0.260	0.179	0.622	2.425	0.180	0.315	0.051	1.166	2.154	0.278	0.153	0.743	1.233	4.114
	3^{rd} **quan**	0.288	0.997	1.795	4.246	7.922	0.538	0.989	1.806	2.648	9.531	0.809	0.779	3.311	4.501	**10.99**
20 cams	**median**	0.378	0.760	0.910	1.774	5.934	0.178	0.656	0.925	2.317	5.569	0.157	0.760	0.516	3.534	5.991
	1^{st} **quan**	0.180	0.315	0.052	1.166	2.154	0.076	0.465	0.169	0.686	4.005	0.089	0.427	0.207	2.085	3.490
	3^{rd} **quan**	0.538	0.989	1.806	2.648	9.531	0.339	1.027	1.555	3.462	7.431	0.320	1.087	0.935	4.660	8.076
40 cams	**median**	0.148	0.256	1.383	3.706	7.202	0.079	0.285	0.940	4.245	6.675	0.346	0.305	0.633	5.398	**10.39**
	1^{st} **quan**	0.077	0.116	0.798	2.974	4.867	0.033	0.125	0.545	3.537	5.147	0.237	0.144	0.264	4.638	9.455
	3^{rd} **quan**	0.281	0.402	2.221	5.426	8.445	0.181	0.465	1.387	5.144	8.858	0.455	0.542	1.137	6.391	**11.97**
80 cams	**median**	0.111	0.600	1.726	6.061	**10.81**	0.089	0.640	1.458	8.388	**15.69**	0.096	0.313	1.946	6.121	**16.32**
	1^{st} **quan**	0.062	0.414	1.309	5.224	**10.12**	0.043	0.499	1.032	7.859	**14.87**	0.035	0.178	1.670	5.571	**15.43**
	3^{rd} **quan**	0.168	0.755	2.080	6.733	**11.91**	0.131	0.772	1.733	8.894	**16.46**	0.172	0.473	2.366	6.605	**17.32**

The result of this experiment is important for two main reasons which will be explored in the following part. First of all, this experiment shows the robustness of the constraint based focal length estimation method against the noise added to constraints. Secondly, the results state that constraint based focal length estimation method can

be utilized in an application if the constraint error under a certain level. This scenario is followed by the second group of experiments that are designed for observing the estimation error of constraint against the additive noise on pixel coordinates. It is still useful in SfM with this error ratios, however, the detailed experiments which prove this will be explored in the following chapter.

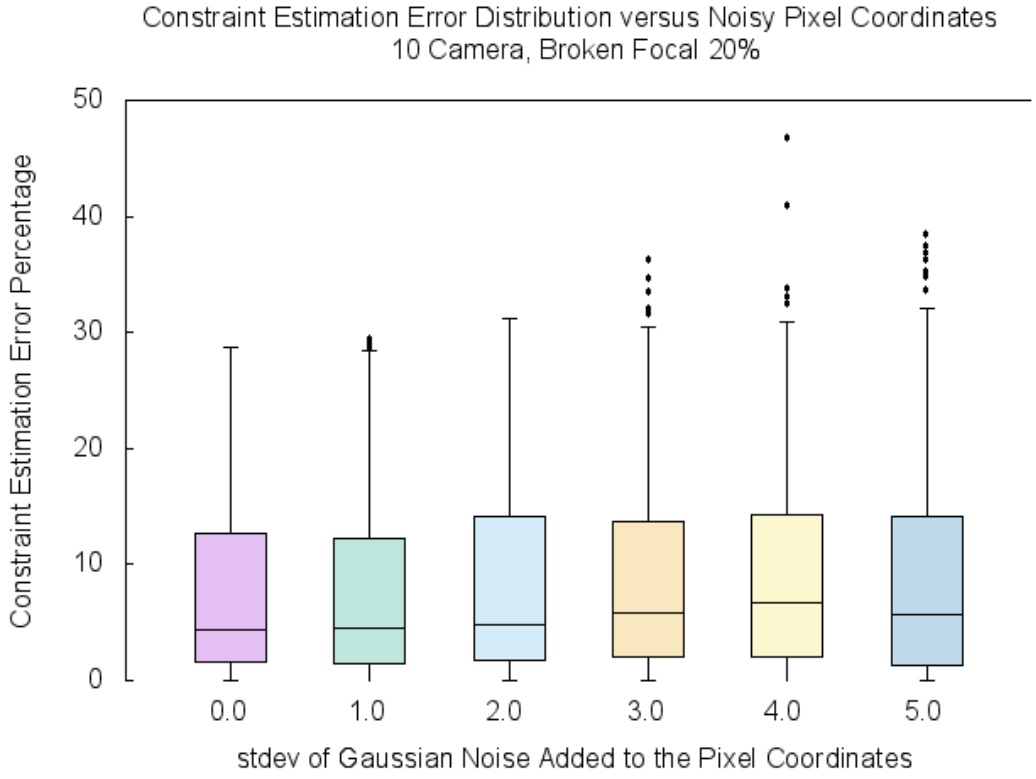


Figure 4.8: Robustness of Constraint Estimation Against Noise in Pixel Coordinates. here are 10 cameras randomly generated and 2 of them have broken focal lengths. In this setup, the noise is added to the pixel coordinates, and constraint is estimated using the $L1$ function defined in Chapter 4.1.

During the previous experiments, the robustness of the focal length estimation algorithm against the noisy constraint is demonstrated. This is followed by the experiment in which the robustness of the constraint estimation against noisy pixel correspondences is investigated. In these experiments, Gaussian noise is added to the pixel coordinates, and the constraints are estimated with these noisy correspondences. On the other hand, the ground truth constraints are computed from the ground truth focal lengths. The estimation accuracy is measured by the relative error on the estimation

of constraints. It is considered that the performance of constraint estimation should be investigated for different number of cameras and the ratio of the broken focal lengths. For this purpose, there are three group of experiments conducted. In the first group of experiments, the broken focal length is set to 20% and number of cameras generated is changing from 10 to 20 cameras. The relative estimation error of the constraints from noisy correspondences for 10 cameras is illustrated in Figure 4.8. The constraint error increases gradually with the increasing standard deviation of Gaussian noise. In details, although there is an increasing error in constraint estimation, the first three quantile of error distribution is far less than 20% , which is the limit of percentage error that can be handled by group synchronized focal length estimation method. This is followed by another experiment in which the broken focal length ratio is 40%. In Figure 4.9, it is illustrated that the constraint estimation is affected by the noisy pixel correspondences, and the increase in the standard deviation of Gaussian results in decreasing the accuracy of constraint estimation method. However, the first three quantile of the error distribution stays under 20%.

As a last experiment in this set, with the ratio of broken focal length of 60%, the constraint estimation error is observed and the results are plotted in Figure 4.10. The estimation error characteristics is not different from previous experiments, and the first three quartile is still under 20%. In details, the estimation error is its minimum value when the standard deviation is around 1 pixels, and it is followed by slight increase in the median of estimation error as standard deviation is incremented. Apart from the experiments with 10 cameras, there is one more group of experiments designed for 20 cameras and different ratio of broken focal lengths in order to observe estimation error in terms of number of cameras. The result of this experiment is illustrated in Table 4.2, which has the same values with the Figures 4.8, 4.9,4.10. There is a slight increase in constraint estimation error, if the number of camera increases, whereas the error values are close to each other for different noise level, and they are less than the critical ratios, 20%. As a second observation, ratio of broken focal length cannot cause big change of estimation error in the case of same pixel noise. In conclusion, for this group of experiments, the proposed constraint extraction method is robust to pixel noise and this robustness continue for different number of cameras, and this robustness stays for various ratios of broken focal lengths.

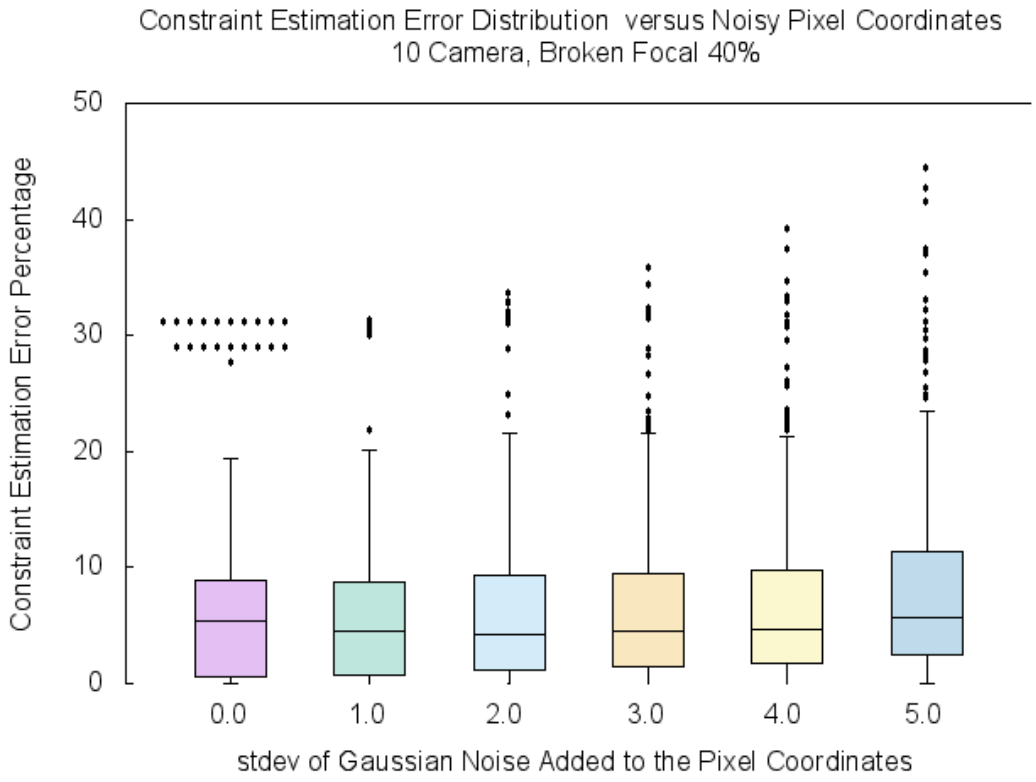


Figure 4.9: Robustness of Constraint Estimation Against Noise in Pixel Coordinates. here are 10 cameras randomly generated and 4 of them have broken focal lengths. In this setup, the noise is added to the pixel coordinates, and constraint is estimated using the $L1$ function defined in Chapter 4.1.

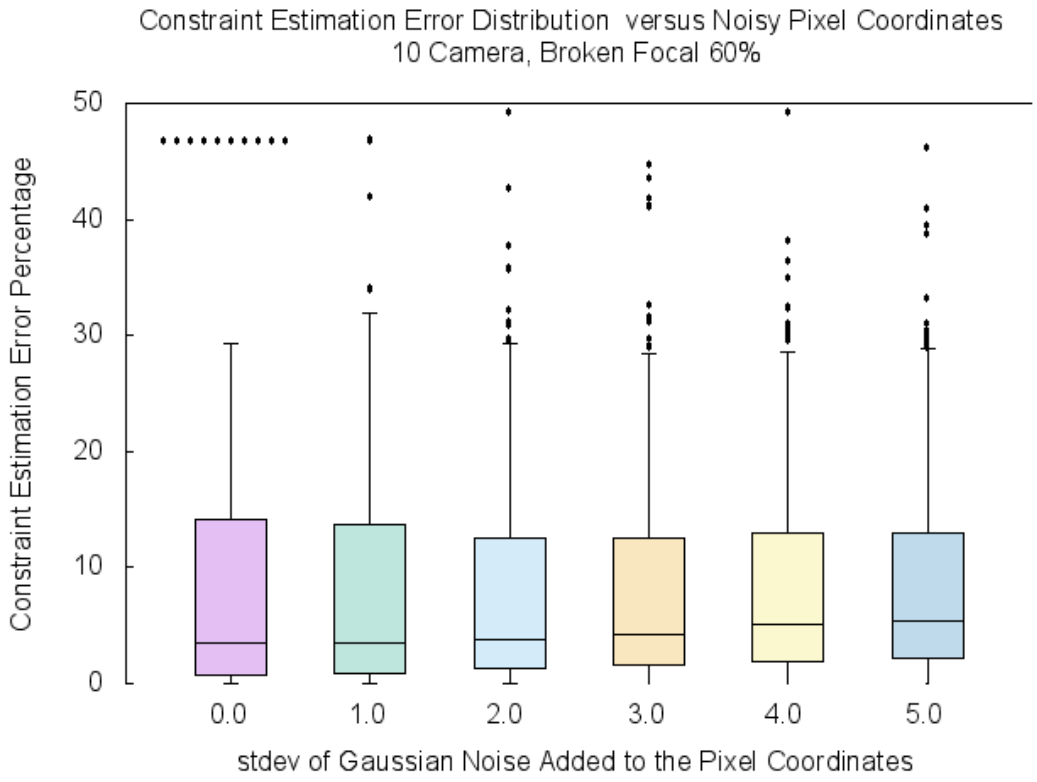


Figure 4.10: Robustness of Constraint Estimation Against Noise in Pixel Coordinates. There are 10 cameras randomly generated and 6 of them have broken focal lengths. In this setup, the noise is added to the pixel coordinates, and constraint is estimated using the $L1$ function defined in Chapter 4.1.

Table 4.2: Constraint estimation errors are shown in the table for different broken focal length ratios under noisy pixel coordinate values. The standard deviation of additive Gaussian noise are 0, 1.0, 2.0, 3.0, 4.0, 5.0. The ratio of broken focal lengths starts from 20% and gradually increases towards 60%.

		Broken f Ratio 20%					**Broken f Ratio** 40%					**Broken f Ratio** 60%				
		1.0	**2.0**	**3.0**	**4.0**	**5.0**	**1.0**	**2.0**	**3.0**	**4.0**	**5.0**	**1.0**	**2.0**	**3.0**	**4.0**	**5.0**
10 cams	**median**	4.540	5.006	6.020	7.098	6.453	4.915	4.611	5.003	4.885	5.692	3.393	3.713	4.539	5.440	5.686
	1^{st} **quan**	1.643	1.770	2.305	2.374	1.968	0.788	1.438	1.818	2.465	2.154	0.888	1.272	1.592	1.903	2.279
	3^{rd} **quan**	12.39	14.29	14.08	15.05	15.16	8.702	9.443	9.791	9.860	11.25	13.73	12.84	12.72	13.18	13.09
20 cams	**median**	6.946	6.335	7.454	7.328	8.109	5.337	5.431	4.987	5.934	6.544	6.554	7.584	7.002	9.189	8.927
	1^{st} **quan**	2.005	2.104	2.294	3.342	3.154	1.624	1.743	1.646	2.243	2.879	1.880	2.626	2.240	3.159	3.133
	3^{rd} **quan**	13.18	13.68	14.11	14.47	16.11	11.02	10.96	12.27	11.15	12.75	13.76	15.06	14.61	16.23	16.28

Based on the result of the previous step, i.e. the robustness of constraint estimation, we design a new set of experiments to test the full focal length estimation pipeline against pixel noise as it is planned in the beginning of experiment section. For this purpose, the first group of experiment is designed with randomly chosen 10 cameras and their connectivity graph. Then, the error of focal length estimation algorithm is tested in the case of that the Gaussian noise with various standard deviation values is added to the pixel values. The standard deviation of the noise are {0, 1.0, 2.0, 3.0, 4.0, 5.0}. For each broken focal length, the initial value is set to 100 which is very different the actual value.

In this experiment, first of all, the constraint is estimated from noisy correspondences and then the focal length estimation algorithm, which is based on the non-linear cost function, finds the unknown focal length values. The result of this experiments are conducted with that of previous experiment, i.e. the error distribution of constraint estimation is plotted in Figure 4.8 for ratio of broken focal 20% and the that of focal length estimation in Figure 4.11 and Figure 4.13 are the result of same experimental setup. This is due to the fact that it is essential to investigate the accuracy of focal length estimation with the error of constraint estimation.

There are two different figures for this experiment to show the performance of focal length estimation method. First of all, the relative estimation error is computed by using both known and unknown focal length values. It is important to observe how much does the known focal length change during the non-linear optimization stage. Intentionally, the known focal lengths are not set as constant variable in the solver,

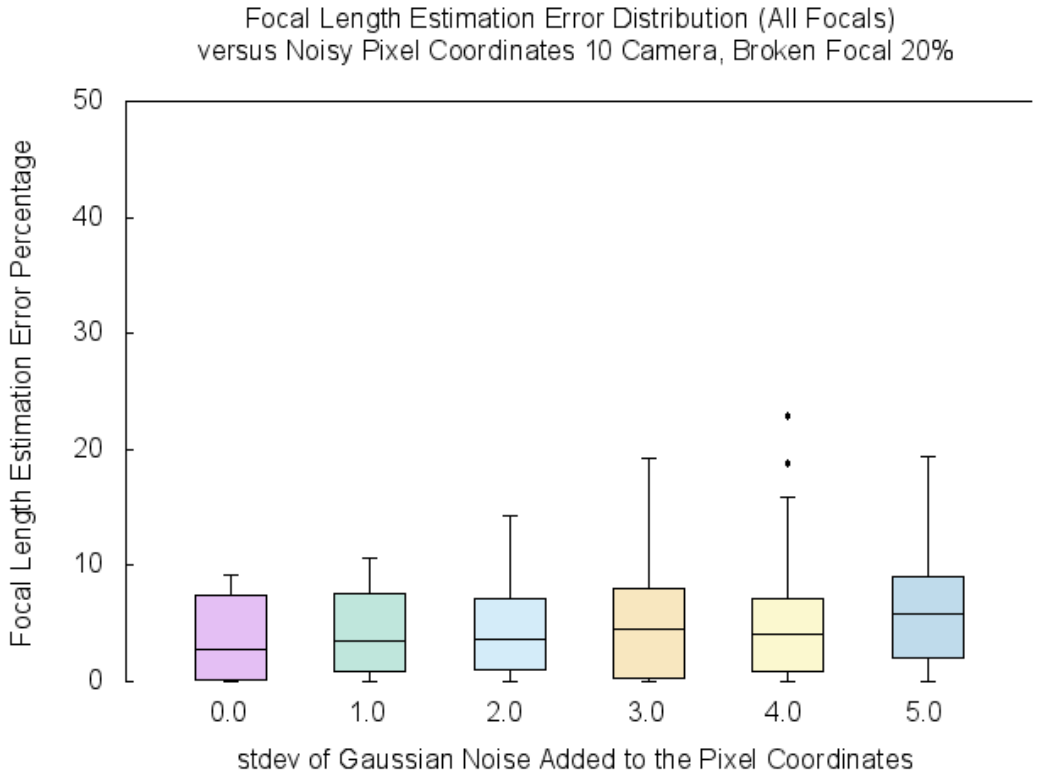


Figure 4.11: Robustness of Focal Length Estimation Against Noise in Pixel Coordinates. There are 10 cameras randomly generated and 2 of them have broken focal lengths. In this setup, the noise is added to the pixel coordinates, and constraint is estimated using the $L1$ function defined in Chapter 4.1, and group synchorized approach utilizes these constraints. This graph illustrates the focal lengh estimation error including all cameras in dataset

so it is valuable to compare the result for all focal length and that for only broken focal lengths. For this purpose, the error distribution of focal length estimation is illustrated in Figure 4.11 for all cameras. This is followed by the error distribution for only broken focal lengths in Figure 4.13. Although there is a slight change for the first three quartile of the distributions, these values increase with the incremental noise in pixels, but the median of these distributions are under 10% for all noise standard deviation values.

Apart from the experiment with 10 cameras and the ratio of broken focal length of 20%, there is also two group of experiments designed for different ratios, namely 40% and 60% with the 10 cameras. The estimation results for these ratios is illustrated in

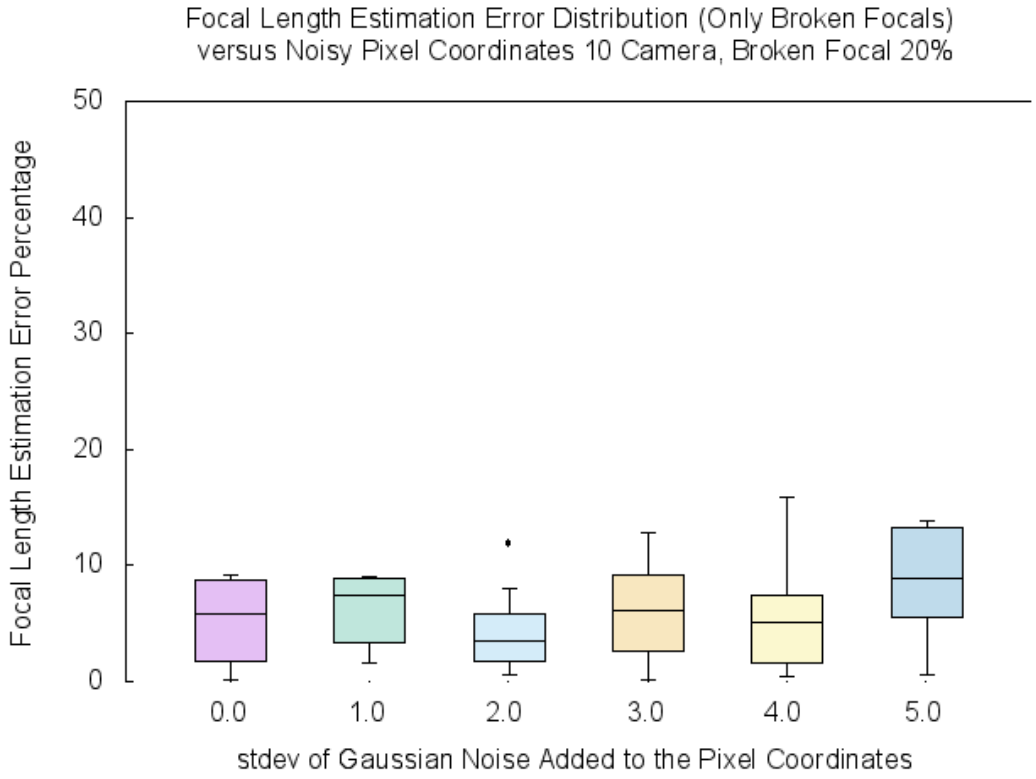
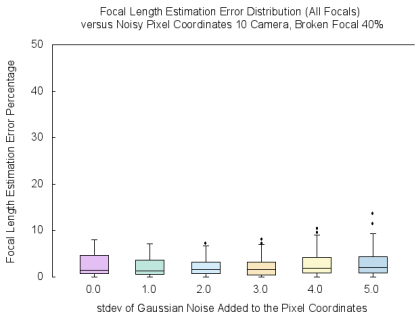


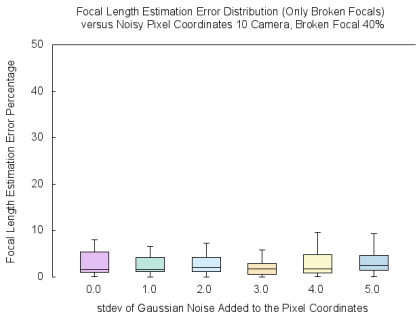
Figure 4.12: Robustness of Focal Length Estimation Against Noise in Pixel Coordinates. There are 10 cameras randomly generated and 2 of them have broken focal lengths. In this setup, the noise is added to the pixel coordinates, and constraint is estimated using the $L1$ function defined in Chapter 4.1, and group synchorized approach utilizes these constraints. This graph illustrates the focal lengh estimation error including all cameras in dataset

Figure 4.13 and Figure 4.14, respectively. The first three quartile of error distribution, in the case of ratio is equal to 40%, is under the 10% for all noise levels. During this interval, these values of the distribution starts at low levels, around 1%, and increase to the level around 5%. In the letter experiment, although the first quartile values of the error distribution more than that of the former experiment, they are still below the 10%. Also, these values gradually increase as the standard deviation changes from 1.0 px to 5.0 px.

Different than the experiments designed for 10 cameras and ratios at 20%, 40% and 60%, the experiments for different number of cameras are also conducted. The graphical results are not illustrated here, but the important statistical measurements of focal

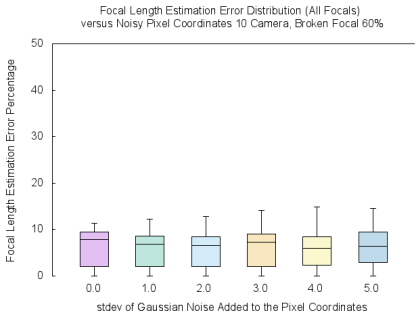


(a) Relative error is estimated for **all** focal lengths

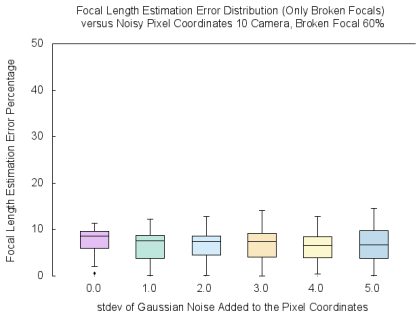


(b) Relative error is estimated for **broken** focal lengths

Figure 4.13: Robustness of Focal Length Estimation Against Noise in Pixel Coordinates. There are 10 cameras randomly generated and 4 of them have broken focal lengths. In this setup, the noise is added to the pixel coordinates, and constraint is estimated using the $L1$ function defined in Chapter 4.1, and group synchorized approach utilizes these constraints.



(a) Relative error is estimated for **all** focal lengths



(b) Relative error is estimated for **broken** focal lengths

Figure 4.14: Robustness of Focal Length Estimation Against Noise in Pixel Coordinates. There are 10 cameras randomly generated and 6 of them have broken focal lengths. In this setup, the noise is added to the pixel coordinates, and constraint is estimated using the $L1$ function defined in Chapter 4.1, and group synchorized approach utilizes these constraints.

length estimation error distributions are demonstrated in Table 4.3. In details, for the scenario with 10 cameras and 20% ratio, there is a slight change in the values of known focal lengths, so it causes a change in the statistical measurements, such as first/third quartile and median of distribution. This is different for the experimental

setup with 40% and 60% ratios, that is the solver doesn't make a considerable change on the values of known focals. Also, first three quartile of the error distribution of the focal length estimation are under 10% in these cases, though the quartile values gradually increase with the incremental noise on pixel coordinates.

Table 4.3: Focal length estimation errors, considering all cameras and only unknown focal length, are shown in the table for different broken focal length ratios under noisy pixel coordinate values. The standard deviation of additive Gaussian noise are 0, 1.0, 2.0, 3.0, 4.0, 5.0. The ratio of broken focal lengths starts from 20% and gradually increases towards 60%.

		Broken f Ratio 20%					**Broken f Ratio** 40%					**Broken f Ratio** 60%				
		1.0	**2.0**	**3.0**	**4.0**	**5.0**	**1.0**	**2.0**	**3.0**	**4.0**	**5.0**	**1.0**	**2.0**	**3.0**	**4.0**	**5.0**
10 cams, all focals	**median**	3.584	3.748	4.481	4.107	5.911	1.383	1.639	1.619	1.968	1.981	7.001	6.649	7.282	5.948	6.457
	1^{st} quan	1.129	1.064	0.359	0.882	2.235	0.631	0.783	0.484	0.829	0.971	2.355	2.145	2.058	2.403	3.147
	3^{rd} quan	7.597	7.338	8.082	7.324	9.084	3.616	3.173	3.183	4.335	4.448	8.667	8.404	9.116	8.458	9.570
10 cams, broken focals	**median**	7.517	3.748	6.264	5.147	8.970	1.639	2.223	1.816	1.743	2.512	7.619	7.470	7.375	6.627	6.864
	1^{st} quan	3.584	1.737	3.991	1.690	5.793	1.114	1.169	0.705	0.834	1.397	3.796	4.908	4.189	3.919	3.867
	3^{rd} quan	9.081	5.961	**10.57**	7.536	**13.78**	4.278	4.297	3.129	4.791	4.783	8.800	8.686	9.388	8.641	9.794
20 cams, all focals	**median**	8.006	5.384	5.384	4.551	8.021	3.636	9.002	7.846	9.677	8.931	6.114	4.448	3.563	8.558	4.984
	1^{st} quan	4.178	3.017	3.923	2.544	6.433	1.567	3.409	5.024	5.811	5.309	1.650	2.214	1.044	6.624	2.394
	3^{rd} quan	9.299	7.294	8.191	8.472	**11.30**	5.309	**10.58**	**10.73**	**11.13**	**11.37**	9.246	8.216	7.141	**11.25**	7.316
20 cams, broken focals	**median**	**11.31**	5.583	4.578	4.545	**12.57**	5.309	8.254	7.846	**10.04**	9.661	5.309	8.254	7.846	10.04	9.661
	1^{st} quan	9.299	0.6973	4.224	3.993	8.162	2.039	4.342	6.066	8.281	7.674	2.039	4.342	6.066	8.281	7.674
	3^{rd} quan	**13.19**	7.715	9.591	9.771	**14.68**	5.622	9.592	**10.73**	**11.26**	**11.37**	5.622	9.592	**10.73**	**11.26**	**11.37**

As a second group of experiment, there are randomly generated 20 cameras with different ratio of broken focal length. The error distributions of this experimental are written in Table 4.3. In details, for the ratios 20% and 60%, there is a slight change on known focal lengths during the optimization stage, whereas this change is far less than others in the experiment with 40% ratio. Apart from this observation, there is a slight increase in focal length estimation error if the number of camera changes from 10 to 20, i.e. the number of camera goes up. This is observed for the ratios of 20%, 40% and 60% during the experiment.

At the beginning of synthetic data experiemtns, there are three steps defined for the experiment stage, namely focal length estimation accuracy with noisy constraints, constraint estimation accuracy with noisy pixel coordinates and focal length estimation accuracy with again noisy pixel correspondences. As a result of this stage, it is observed that it is possible to estimate the unknown focal lengths from a geometric constraints by using the proposed method. The median of error distribution of focal length estimation stays under 10%, which is the safe region, if the constraint estima-

tion accuracy is above 80%. This slightly changes, but stays in safe region in general, if the number of cameras or the ratio of broken focal lengths increase.

Figure 4.15: The dataset is used for real data experiments. 10 images, at 3456 x 5184 resolution, from the cameras that have different focal length values are chosen. The focal length values are changing from 9000 to 19000 pixels.

4.5.2 Real Data Experiments

Apart from the synthetic data experiments, the proposed approach for focal length estimation is also tested with real data. The real data experiments are important for two main reasons, which will be explored in the following chapter. First of all, the relation between focal lengths of cameras in a real data set is observed, and it is critical to utilize proposed focal length estimation method in the thesis. Secondly, the performance of group synchronized focal length estimation is observed with the real data, and the result will show whether this method is useful for SfM algorithms or not. For this purpose, one of the known SfM data set [1] is utilized during the experiments. This datasets contains number of photographs of a scene , and the focal length value of each camera is changing from around 9000 pixels to 19000 pixels. The images are at 3456 x 5184 resolution. For the real data experiments, 10 cameras from the data

[1] This dataset is publicly available in `https://github.com/pyp22/datasets`

set are randomly chosen, which are illustrated in Figure 4.15.

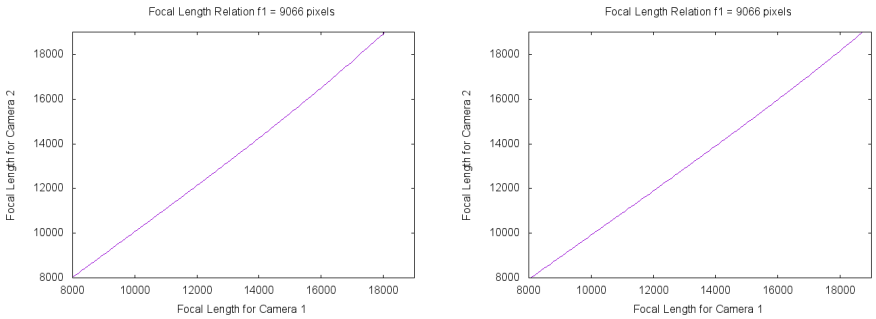
$$G(V,E) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Figure 4.16: The view graph for the dataset is generated after keypoint extraction and image matching stages within the input image set [62]

This is followed by extracting and describing the feature points for images, and then find matched image pairs in the dataset. As a result of the matching phase, the relation between images are represented in a view graph, which is given in Figure 4.16. Each row and column index stands for a camera in a dataset. The library[2] developed for the thesis is utilized for all of these stages. With the view graph input, as a first stage in the experiments, relation between focal lengths are computed by using Algorithm 1. As an observation, the relation between a set of focal length pairs are illustrated in Figure 4.17. The focal length from a logical interval is chosen for camera 0 and the correspondent focal lengths are computed by using Equation 4.2 for cameras 1 and 5. It is observed that this relation is modeled by linear function, $L1$, which is defined in Chapter 4.1. This is followed by estimating all constraints between connected image pairs for this dataset. With the estimated constratins and the EXIF data for input images, the relative error of constraint estimation is computed. It is observed that the median error of constraint estimation is 21%, which is very close to safe region defined in synthetic data experiments. With this constraint estimation accuracy, this is followed by the the performance analysis of focal length estimation algorithm, which

[2] The library developed in c++ is publicly available in `https://github.com/akcalakcal/Open_SfM_METU`

is proposed in this thesis.



(a) Focal Length, f_1, = 9066 px (b) Focal Length, f_1, = 9066 px

Figure 4.17: Focal length relation between two cameras in a dataset, which is utilized for real data experiments. The relation between camera 0 and two other cameras, namely camera 1 and 5, are plotted. The focal length of the camera 0 is given as 9066 pixels in the EXIF data.

As a second group of experiments, error distribution of the focal length estimation algorithm with different ratios of broken focal length, and the various initial values for unknown focal lengths are tested. First of all, the ratio of broken focal length is set to 20% and the focal length of unknown cameras are estimated. This is repeated by 100 times for randomly chosen broken cameras each time with the same ratio, and the initial focal length value for the broken cameras is set to 100. The result of of this experiments is illustrated in Table 4.4. It is observed that the first three quartile of error distribution is around 0 value. This is slightly changed if the ratio of unknown focal length is increased to 40%. During this experiment, the third quartile of the error distribution rise around 13% values, whereas the median of the distribution remains around 0 value. This is followed by another experiment, which is designed for 60% ratio of broken focal length. This is the starting point to think about the limits of the proposed focal length estimation algorithm in this thesis. During the real data experiments, it is experienced to test proposed algorithm for higher ratio of broken focal lengths than the ones that is tested for the synthetic data experiments. This is because, the aim of this experiment is to explore the limit of this algorithm, so the experiments with the 60% ratio of broken focal length is critical step. As a result of this experiment, the median of error distribution gets closer to the arbitrary error boundary defined during the synthetic data experiments, which is 10%, but remains

in the safe region.

With the same number of camera, which is 10, and initial value for focal lengths value, which is 100, there is another group of experiments designed for 70% and 80% ratio of broken focal lengths. During these experiments, the the median of error distribution increased together with the third quartile, which means that some of the unknown focal length values are set to correct ones whereas others are slightly different than the ground truth values. This is followed by the increased of median value up to 37.05% in the case of 80% ratio of broken focal lengths, which shows that the focal length estimation method with the $L1$ linearization starts to fail at this level. It may be considered by the reader that this is also because of the initial value for the unknown focal length; hence, the second group of experiments are conducted for this purpose. In details, this group of experiments is constructed with the same ratio of broken focal lengths but with different initial values for the non-linear optimization steps. Each column in Table 4.4 stands for this experiment for different ratios. It is observed that the estimation results slightly change for different percentage of ratios, i.e. initial value is not the main factor to be considered for the focal length estimation algorithm, and solver can start with any initial value.

In conclusion, the performance of the focal length estimation value is limited with the ratio of broken focal length, and the median of the distribution reach to higher values if the ratio is more than 70%. So, this method is useful up to ratio of unknown focal length in a SfM pipeline.

Table 4.4: The proposed focal length estimation algorithm is tested with real data, which is used in SfM algorithms. The ratio of broken focal length in the dataset is changing from 20% to 80%, and the estimation accuracy is observed with these ratios.

		Broken f Ratio				
		20%	**40%**	**60%**	**70%**	**80%**
10 cams, f_{init} = 100	**median**	0.0	0.0	9.42	18.62	37.05
	1^{st} **quan**	0.0	0.0	0.0	0.0	8.54
	3^{rd} **quan**	0.0	14.33	33.86	46.98	73.50
10 cams, f_{init} = 1000	**median**	0.0	0.0	10.69	18.53	36.56
	1^{st} **quan**	0.0	0.0	0.0	0.0	7.98
	3^{rd} **quan**	0.0	14.89	34.96	49.88	73.49
10 cams, f_{init} = 4000	**median**	0.0	0.0	10.86	19.14	36.55
	1^{st} **quan**	0.0	0.0	0.0	0.0	8.55
	3^{rd} **quan**	0.0	13.14	34.07	47.50	73.47
10 cams, f_{init} = 8000	**median**	0.0	0.0	9.72	17.73	35.94
	1^{st} **quan**	0.0	0.0	0.0	0.0	7.99
	3^{rd} **quan**	0.0	13.62	35.17	49.76	73.43

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this study, we have proposed a novel stratified calibration algorithm for estimating both the focal length and the radial distortion parameter of a camera from given 2d point correspondences. We show here that by using the one parameter division model of [19], it is possible to recover the undistorted image correspondences. Moreover, we have introduced a completely new method for recovering the camera focal length by enforcing the trace constraint over the essential matrix via the estimated fundamental matrix. Finally, we have shown with synthetic and real experiments that our method compares favorably against other approaches presented in the literature.

As a second contribution in this thesis, novel method for estimating the unknown focal lengths in SfM dataset using the geometric constraints between image pairs and group synchronization based on derived linear relation is proposed. In conclusion, it is stated that the proposed approach is useful as an initialization stage in SfM algorithms, after that the performance of the method is shown in the experimental stages. As a result of synthetic data experiments, the performance of the proposed methods is tested for the noise sensitivity. The results show that this method is robust to the pixel noise. Another group of experiments are conducted with the real data. The result of real data experiments illustrated that this method is useful for the SfM pipelines as an initial stage. If we know the focal lengths of the at least 30% of all cameras in a dataset, it is possible to recover unknown focal lengths with the median of the estimation error under 10%.

As a future work of this study, it is a research problem to increase the performance of the group synchronized focal length estimation method by developing the model

between unknown focal lengths.

REFERENCES

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[3] Oytun Akman. *Robust augmented reality*. PhD thesis, TU Delft, Delft University of Technology, 2012.

[4] Federica Arrigoni, Andrea Fusiello, and Beatrice Rossi. Camera motion from group synchronization. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 546–555. IEEE, 2016.

[5] Joao P Barreto and Kostas Daniilidis. Fundamental matrix for cameras with radial distortion. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 625–632. IEEE, 2005.

[6] Jean-Yves Bouguet. Camera calibration toolbox for matlab. 2004.

[7] José Henrique Brito, Christopher Zach, Kevin Köser, Manuel Ferreira, and Marc Pollefeys. One-sided radial-fundamental matrix estimation. In *BMVC*, pages 1–12, 2012.

[8] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. 3d reconstruction from image collections with a single known focal length. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1803–1810. IEEE, 2009.

[9] Martin Byröd, Matthew Brown, and Karl Åström. Minimal solutions for panoramic stitching with radial distortion. In *The 20th British Machine Vision Conference*. British Machine Vision Association (BMVA), 2009.

[10] Martin Byröd, Zuzana Kukelova, Klas Josephson, Tomas Pajdla, and Kalle Åström. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[11] LF Cheong and X Xiang. Error characteristics of sfm with unknown focal length. In *Proceedings of Asian Conference on Computer Vision*. Springer, 2006.

[12] Loong-Fah Cheong and Xu Xiang. Behaviour of sfm algorithms with erroneous calibration. *Computer Vision and Image Understanding*, 115(1):16–30, 2011.

[13] Intel Corp. Intel real sense depth camera, 2017.

[14] Microsoft Corp. Microsoft kinect depth camera, 2017.

[15] Yahoo Corp. Flickr photo sharing web site, 2017.

[16] David A Cox, John Little, and Donal O'shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.

[17] Sundara Tejaswi Digumarti, Gaurav Chaurasia, Aparna Taneja, Roland Siegwart, Amber Thomas, and Paul Beardsley. Underwater 3d capture using a low-cost commercial depth camera. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[18] 3D doctor. Vector based 3d imaging, modelling and measurement software, 2017.

[19] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–125. IEEE, 2001.

[20] Sven Hammarling, Christopher J Munro, and Francoise Tisseur. An algorithm for the complete solution of quadratic eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)*, 39(3):18, 2013.

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

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

[23] Richard I Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Transactions on pattern analysis and machine intelligence*, 19(2):133–135, 1997.

[24] Jose Henrique Brito, Roland Angst, Kevin Koser, and Marc Pollefeys. Radial distortion self-calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1368–1375, 2013.

[25] Microsoft Hololens. Microsoft hololens, 2017.

[26] Apple Inc. Apple arkit, 2017.

[27] DoubleMe Inc. Doubleme mixed reality, 2017.

[28] FlyInside Inc. Flyinside virtual realtiy based flight simulation solutions, 2017.

[29] Open Inventor. High-performance 3d software development tools, 2017.

[30] Fangyuan Jiang, Yubin Kuang, Jan Erik Solem, and Kalle Åström. A minimal solution to relative pose with unknown focal length and radial distortion. In *Computer Vision–ACCV 2014*, pages 443–456. Springer, 2014.

[31] Klas Josephson and Martin Byrod. Pose estimation with radial distortion and unknown focal length. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2419–2426. IEEE, 2009.

[32] O. Kahler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S Torr, and D. W. Murray. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Transactions on Visualization and Computer Graphics (Proceedings International Symposium on Mixed and Augmented Reality 2015*, 22(11), 2015.

[33] Kenichi Kanatani. *Geometric computation for machine vision*. Oxford University Press, Inc., 1993.

[34] Kenichi Kanatani and Chikara Matsunaga. Closed-form expression for focal lengths from the fundamental matrix. In *Proc. 4th Asian Conf. Comput. Vision*, volume 1, pages 128–133. Citeseer, 2000.

[35] Kenichi Kanatani, Atsutada Nakatsuji, and Yasuyuki Sugaya. Stabilizing the focal length computation for 3-d reconstruction from two uncalibrated views. *International Journal of Computer Vision*, 66(2):109–122, 2006.

[36] Yasushi Kanazawa, Yasuyuki Sugaya, and Kenichi Kanatani. Decomposing three fundamental matrices for initializing 3-d reconstruction from three views. *IPSJ Transactions on Computer Vision and Applications*, 6(0):120–131, 2014.

[37] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

[38] Yubin Kuang, Jan Erik Solem, Fredrik Kahl, and Kalle Åström. Minimal solvers for relative pose with a single unknown radial distortion. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 33–40. IEEE, 2014.

[39] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *Computer Vision–ECCV 2008*, pages 302–315. Springer, 2008.

[40] Zuzana Kukelova, Jan Heller, Martin Bujnak, Andrew Fitzgibbon, and Tomas Pajdla. Efficient solution to the epipolar geometry for radially distorted cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2309–2317, 2015.

[41] Zuzana Kukelova and Tomas Pajdla. A minimal solution to the autocalibration of radial distortion. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[42] Zuzana Kukelova and Tomas Pajdla. Two minimal problems for cameras with radial distortion. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[43] Oliver Kutter, André Aichert, Christoph Bichlmeier, Jörg Traub, SM Heining, B Ockert, Ekkehard Euler, and Nassir Navab. Real-time volume rendering for high quality visualization in augmented reality. In *International Workshop on Augmented environments for Medical Imaging including Augmented Reality in Computer-aided Surgery (AMI-ARCS 2008), New York, USA*, 2008.

[44] Hongdong Li. A simple solution to the six-point two-view focal-length problem. In *European Conference on Computer Vision*, pages 200–213. Springer, 2006.

[45] Hongdong Li and Richard Hartley. A non-iterative method for correcting lens distortion from nine point correspondences. *OMNIVIS 2005*, 2:7, 2005.

[46] Oculus LLC. Oculus virtual reality, 2017.

[47] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[48] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, 2015.

[49] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[50] Stephen Maybank. *Theory of reconstruction from image motion*, volume 28. Springer Science & Business Media, 2012.

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

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

[53] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 26:305–364, 2017.

[54] Nitish Padmanaban, Robert Konrad, Tal Stramer, Emily A Cooper, and Gordon Wetzstein. Optimizing virtual reality for all users through gaze-contingent and adaptive focus displays. *Proceedings of the National Academy of Sciences*, page 201617251, 2017.

[55] Filip Radenovic, Johannes L Schonberger, Dinghuang Ji, Jan-Michael Frahm, Ondrej Chum, and Jiri Matas. From dusk till dawn: Modeling in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5488–5496, 2016.

[56] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[57] Thomas Schöps, Torsten Sattler, Christian Häne, and Marc Pollefeys. 3d modeling on the go: Interactive 3d reconstruction of large-scale scenes on mobile devices. In *3D Vision (3DV), 2015 International Conference on*, pages 291–299. IEEE, 2015.

[58] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[59] Henrik Stewénius, David Nistér, Fredrik Kahl, and Frederik Schaffalitzky. A minimal solution for relative pose with unknown focal length. *Image and Vision Computing*, 26(7):871–877, 2008.

[60] Peter Sturm. On focal length calibration from two views. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–145. IEEE, 2001.

[61] Chris Sweeney, John Flynn, and Matthew Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 483–490. IEEE, 2014.

[62] Chris Sweeney, Torsten Sattler, Tobias Hollerer, Matthew Turk, and Marc Pollefeys. Optimizing the viewing graph for structure-from-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 801–809, 2015.

[63] Françoise Tisseur and Karl Meerbergen. The quadratic eigenvalue problem. *SIAM review*, 43(2):235–286, 2001.

[64] Gadget View. Microsoft hololens : The future of nfl, 2017.

[65] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2704–2712, 2015.

APPENDIX A

DERIVATION OF TRACE CONSTRAINT

Essential matrix, E, has important properties; by definition, E matrix is rank-2 matrix, and two of three singular values are same and the third one is equal to 0. With these properties, E matrix is written as a multiplication of a skew-symmetric matrix and the rotation matrix [21]. Hence, an essential matrix is written as:

$$E = T_a R \tag{A.1}$$

where R is a $3x3$ rotation matrix, and T_a is a $3x3$ skew symmetric matrix, which is written for any vector $a \in \mathbb{R}^3$, and $a = [a_1, a_2, a_3]^T$:

$$T_a = \begin{bmatrix} 0 & -a_3 & -a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{A.2}$$

This is followed by rewriting the *trace constraint* in Equation 2.10, of E matrix:

$$EE^T E = (0.5) * tr(EE^T)E \tag{A.3}$$

First of all, the left side of this equation is represented by T_a and R matrices:

$$EE^T E = T_a R R^T {T_a}^T T_a R \tag{A.4}$$

$$EE^T E = T_a {T_a}^T T_a R \tag{A.5}$$

Secondly, the same replacement is applied for the right side:

$$tr(EE^T)E = tr(T_a {T_a}^T)T_a R \tag{A.6}$$

With this representations, the problem turns to satisfaction of the following equation:

$$T_a {T_a}^T T_a = \frac{1}{2} * tr(T_a {T_a}^T)T_a \tag{A.7}$$

Let x be an arbitrary vector in $\mathbb{R}^3$. The multiplication of x with $T_a {T_a}^T$ yields the following equation:

$$\begin{aligned} T_a {T_a}^T x &= T_a(a \times x) \\ T_a {T_a}^T x &= (a \times x) \times a \\ T_a {T_a}^T x &= (a.a)x - (a.x)a \end{aligned} \tag{A.8}$$

This is followed by:

$$\begin{aligned} T_a {T_a}^T &= (a.a)I - (a \otimes a) \\ T_a {T_a}^T T_a &= (a.a)T_a \end{aligned} \tag{A.9}$$

Apart from the left side, the right side of the equation is written as following:

$$\begin{aligned} &\frac{1}{2} * tr(T_a {T_a}^T) = \frac{1}{2} * tr((a.a)I - (a \otimes a)) = a.a \\ &\frac{1}{2} * tr(T_a {T_a}^T)T_a = (a.a)T_a \end{aligned} \tag{A.10}$$

In conclusion, it is shown that left and right side of the Equation A.7 is equal to each other. So, proof is done.

APPENDIX B

DERIVATION OF NON-LINEAR EQUATION FOR FOCAL LENGTH ESTIMATION BASED ON TRACE CONSTRAINT

In the scope of this thesis, there are two different approach proposed for focal length estimation, which utilize the trace constraint [50]. First of all, in Chapter 3.2, focal length can be estimated if two camera share the same focal length and the fundamental matrix which defines the epipolar geometry between these views is known. Secondly, if the fundamental matrix between two view is estimated and one of the camera's focal length is known, then the focal length of other camera can be estimated. In the following, the functions used for the focal length estimation are derived in details form the trace constraints and geometric constraints.

First of all, trace constraint of an essential matrix is written as it is stated in [50]:

$$2EE^TE - tr(EE^T)E = 0 \tag{B.1}$$

The essential matrix, E, is also writin in terms of fundamental matrix ,F, and the camera matrices, K, of cameras, which are same in this problem:

$$E = K^TFK \tag{B.2}$$

As it is explained in the Chapter 2.1, the camera matrix in the form of 2.4 is preferred in order to model this solution for focal length estimation. This is followed by compute essential matrix E in terms of elements of fundamental matrix F, and the the shared focal length of cameras.

$$E = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.3}$$

$$E = \begin{bmatrix} f^2 \times f_{1,1} & f^2 \times f_{1,2} & f \times f_{1,3} \\ f^2 \times f_{2,1} & f^2 \times f_{2,2} & f \times f_{2,3} \\ f \times f_{3,1} & f \times f_{3,2} & f \times f_{3,3} \end{bmatrix} \tag{B.4}$$

With this form of essential matrix, E, the first part of the *trace constraint*, which is EE^T is constructed:

$$EE^T = \begin{bmatrix} (f^4{f_{1,1}}^2 + f^4{f_{1,2}}^2 + f^2{f_{1,3}}^2) & (f^4 f_{1,1}f_{2,1} + f^4 f_{1,2}f_{2,2} + ff_{1,3}f_{2,3}) & (f^3 f_{1,1}f_{3,1} + f^3 f_{1,2}f_{3,2} + ff_{1,3}f_{3,3}) \\ (f^4 f_{2,1}f_{1,1} + f^4 f_{2,2}f_{1,2} + f^2 f_{2,3}f_{1,3}) & (f^4{f_{2,1}}^2 + f^4{f_{2,2}}^2 + f^2{f_{2,3}}^2) & (f^3 f_{2,1}f_{3,1} + f^3 f_{2,2}f_{3,2} + ff_{2,3}f_{3,3}) \\ (f^3 f_{3,1}f_{1,1} + f^3 f_{3,2}f_{1,2} + ff_{3,3}f_{1,3}) & (f^3 f_{3,1}f_{2,1} + f^3 f_{3,2}f_{2,2} + ff_{3,3}f_{2,3}) & (f^2{f_{3,1}}^2 + f^2{f_{3,2}}^2 + {f_{3,3}}^2) \end{bmatrix} \tag{B.5}$$

This is followed by the construction of EE^TE, which is demonstrated in a vector form for inconvenience:

$$vec(EE^TE) = \begin{bmatrix} ((f^6{f_{1,1}}^3 + f^6 f_{1,1}{f_{1,2}}^2 + f^4 f_{1,1}{f_{1,3}}^2) + (f^6 f_{1,1}{f_{2,1}}^2 + f^6 f_{1,2}f_{2,2}f_{2,1} + f^3 f_{1,3}f_{2,3}f_{2,1}) + (f^4 f_{1,1}{f_{3,1}}^2 + f^4 f_{1,2}f_{3,2}f_{3,1} + f^2 f_{1,3}f_{3,3}f_{3,1})) \\ ((f^6{f_{1,1}}^2 f_{1,2} + f^6{f_{1,2}}^3 + f^4{f_{1,3}}^2 f_{1,2}) + (f^6 f_{1,1}f_{2,1}f_{2,2} + f^6 f_{1,2}{f_{2,2}}^2 + f^3 f_{1,3}f_{2,3}f_{2,2}) + (f^4 f_{1,1}f_{3,1}f_{3,2} + f^4 f_{1,2}{f_{3,2}}^2 + f^2 f_{1,3}f_{3,3}f_{3,2})) \\ ((f^5{f_{1,1}}^2 f_{1,3} + f^5{f_{1,2}}^2 f_{1,3} + f^3{f_{1,3}}^3) + (f^5 f_{1,1}f_{2,1}f_{2,3} + f^5 f_{1,2}f_{2,2}f_{2,3} + f^2 f_{1,3}f_{2,3}f_{2,3}) + (f^3 f_{1,1}f_{3,1}f_{3,3} + f^3 f_{1,2}f_{3,2}f_{3,3} + ff_{1,3}{f_{3,3}}^2)) \\ ((f^6 f_{2,1}{f_{1,1}}^2 + f^6 f_{2,2}f_{1,2}f_{1,1} + f^4 f_{2,3}f_{1,3}f_{1,1}) + (f^6{f_{2,1}}^3 + f^6{f_{2,2}}^2 f_{2,1} + f^4{f_{2,3}}^2 f_{2,1}) + (f^4 f_{2,1}{f_{3,1}}^2 + f^4 f_{2,2}f_{3,2}f_{3,1} + f^2 f_{2,3}f_{3,3}f_{3,1})) \\ ((f^6 f_{2,1}f_{1,1}f_{1,2} + f^6 f_{2,2}f_{1,2}{f_{1,2}}^2 + f^4 f_{2,3}f_{1,3}f_{1,2}) + (f^6{f_{2,1}}^2 f_{2,2} + f^6{f_{2,2}}^3 + f^4{f_{2,3}}^2 f_{2,2}) + (f^4 f_{2,1}f_{3,1}f_{3,2} + f^4 f_{2,2}{f_{3,2}}^2 + f^2 f_{2,3}f_{3,3}f_{3,2})) \\ ((f^5 f_{2,1}f_{1,1}f_{1,3} + f^5 f_{2,2}f_{1,2}f_{1,3} + f^3 f_{2,3}{f_{1,3}}^2) + (f^5{f_{2,1}}^2 f_{2,3} + f^5{f_{2,2}}^2 f_{2,3} + f^3{f_{2,3}}^3) + (f^3 f_{2,1}f_{3,1}f_{3,3} + f^3 f_{2,2}f_{3,2}f_{3,3} + ff_{2,3}{f_{3,3}}^2)) \\ ((f^5 f_{3,1}{f_{1,1}}^2 + f^5 f_{3,2}f_{1,2}f_{1,1} + f^3 f_{3,3}f_{1,3}f_{1,1}) + (f^5 f_{3,1}{f_{2,1}}^2 + f^5 f_{3,2}f_{2,2}f_{2,1} + f^3 f_{3,3}f_{2,3}f_{2,1}) + (f^3{f_{3,1}}^3 + f^3{f_{3,2}}^2 f_{3,1} + ff_{3,3}{}^2 f_{3,1})) \\ ((f^5 f_{3,1}f_{1,1}f_{1,2} + f^5 f_{3,2}f_{1,2}{f_{1,2}}^2 + f^3 f_{3,3}f_{1,3}f_{1,2}) + (f^5 f_{3,1}f_{2,1}f_{2,2} + f^5 f_{3,2}{f_{2,2}}^2 + f^3 f_{3,3}f_{2,3}f_{2,2}) + (f^3{f_{3,1}}^2 f_{3,2} + f^3{f_{3,2}}^3 + ff_{3,3}{}^2 f_{3,2})) \\ ((f^4 f_{3,1}f_{1,1}f_{1,3} + f^4 f_{3,2}f_{1,2}f_{1,3} + f^2 f_{3,3}{f_{1,3}}^2) + (f^4 f_{3,1}f_{2,1}f_{2,3} + f^4 f_{3,2}f_{2,2}f_{2,3} + f^2 f_{3,3}{f_{2,3}}^2) + f^2{f_{3,1}}^2 f_{3,3} + f^2{f_{3,2}}^2 f_{3,3} + {f_{3,3}}^3)) \end{bmatrix} \tag{B.6}$$

As a second part of the *trace constraint*, the following equation is written:

$$
\begin{aligned}
&vec(tr(EE^T)E) = \\
&\begin{bmatrix}
(f^6({f_{1,1}}^3 + f_{1,2}{}^2 f_{1,1} + {f_{2,1}}^2 f_{1,1} + f_{2,2}{}^2 f_{1,1}) + f^4(f_{1,3}{}^2 f_{1,1} + f_{2,3}{}^2 f_{1,1} + {f_{3,1}}^2 f_{1,1} + f_{3,2}{}^2 f_{1,1}) + f^2 f_{3,3}{}^2 f_{1,1}) \\
(f^6({f_{1,1}}^2 f_{1,2} + {f_{1,2}}^3 + {f_{2,1}}^2 f_{1,2} + f_{2,2}{}^2 f_{1,2}) + f^4 f_{1,3}{}^2 f_{1,2} + f_{2,3}{}^2 f_{1,2} + {f_{3,1}}^2 f_{1,2} + f_{3,2}{}^2 f_{1,2}) + f^2 f_{3,3}{}^2 f_{1,2}) \\
(f^5({f_{1,1}}^2 f_{1,3} + f_{1,2}{}^2 f_{1,3} + {f_{2,1}}^2 f_{1,3} + f_{2,2}{}^2 f_{1,3}) + f^3({f_{1,3}}^3 + f_{2,3}{}^2 f_{1,3} + {f_{3,1}}^2 f_{1,3} + f_{3,2}{}^2 f_{1,3}) + f f_{3,3}{}^2 f_{1,3}) \\
(f^6({f_{1,1}}^2 f_{2,1} + f_{1,2}{}^2 f_{2,1} + {f_{2,1}}^3 + f_{2,2}{}^2 f_{2,1}) + f^4(f_{1,3}{}^2 f_{2,1} + f_{2,3}{}^2 f_{2,1} + {f_{3,1}}^2 f_{2,1} + f_{3,2}{}^2 f_{2,1}) + f^2 f_{3,3}{}^2 f_{2,1}) \\
(f^6({f_{1,1}}^2 f_{2,2} + f_{1,2}{}^2 f_{2,2} + {f_{2,1}}^2 f_{2,2} + {f_{2,2}}^3) + f^4(f_{1,3}{}^2 f_{2,2} + f_{2,3}{}^2 f_{2,2} + {f_{3,1}}^2 f_{2,2} + f_{3,2}{}^2 f_{2,2}) + f^2 f_{3,3}{}^2 f_{2,2}) \\
(f^5({f_{1,1}}^2 f_{2,3} + f_{1,2}{}^2 f_{2,3} + {f_{2,1}}^2 f_{2,3} + f_{2,2}{}^2 f_{2,3}) + f^3(f_{1,3}{}^2 f_{2,3} + {f_{2,3}}^3 + {f_{3,1}}^2 f_{2,3} + f_{3,2}{}^2 f_{2,3}) + f f_{3,3}{}^2 f_{2,3}) \\
(f^5({f_{1,1}}^2 f_{3,1} + f_{1,2}{}^2 f_{3,1} + {f_{2,1}}^2 f_{3,1} + f_{2,2}{}^2 f_{3,1}) + f^3(f_{1,3}{}^2 f_{3,1} + f_{2,3}{}^2 f_{3,1} + {f_{3,1}}^3 + f_{3,2}{}^2 f_{3,1}) + f f_{3,3}{}^2 f_{3,1} \\
(f^5({f_{1,1}}^2 f_{3,2} + f_{1,2}{}^2 f_{3,2} + {f_{2,1}}^2 f_{3,2} + f_{2,2}{}^2 f_{3,2}) + f^3(f_{1,3}{}^2 f_{3,2} + f_{2,3}{}^2 f_{3,2} + {f_{3,1}}^2 f_{3,2} + {f_{3,2}}^3) + f f_{3,3}{}^2 f_{3,2}) \\
(f^4({f_{1,1}}^2 f_{3,3} + f_{1,2}{}^2 f_{3,3} + {f_{2,1}}^2 f_{3,3} + f_{2,2}{}^2 f_{3,3}) + f^2(f_{1,3}{}^2 f_{3,3} + f_{2,3}{}^2 f_{3,3} + {f_{3,1}}^2 f_{3,3} + f_{3,2}{}^2 f_{3,3}) + {f_{3,3}}^3)
\end{bmatrix}
\end{aligned}
\tag{B.7}
$$

Finally, the previously defined function $G(f)$, Equation 3.10 is stated in terms of *focal length*, f, and the elements of fundamental matrix, F. In fact,

$$
G(f) = \begin{bmatrix}
g(f)_{1,1} & g(f)_{1,2} & g(f)_{1,3} \\
g(f)_{2,1} & g(f)_{2,2} & g(f)_{2,3} \\
g(f)_{3,1} & g(f)_{3,2} & g(f)_{3,3}
\end{bmatrix}
\tag{B.8}
$$

and each elements of $G(f)$ is stated in following:

$$
\begin{aligned}
&g(f)_{1,1} = \\
&(f^6(2{f_{1,1}}^3 + 2f_{1,1}f_{1,2}{}^2 + 2f_{1,1}{f_{2,1}}^2 + 2f_{1,2}f_{2,2}f_{2,1} - {f_{1,1}}^3 - f_{1,2}{}^2 f_{1,1} - {f_{2,1}}^2 f_{1,1} - f_{2,2}{}^2 f_{1,1}) \\
&+ f^4(2f_{1,1}f_{1,3}{}^2 + 2f_{1,1}{f_{3,1}}^2 + 2f_{1,2}f_{3,2}f_{3,1} - f_{1,3}{}^2 f_{1,1} - f_{2,3}{}^2 f_{1,1} - {f_{3,1}}^2 f_{1,1} - f_{3,2}{}^2 f_{1,1}) \\
&+ f^3(2f_{1,3}f_{2,3}f_{2,1}) \\
&+ f^2(2f_{1,3}f_{3,3}f_{3,1} - f_{3,3}{}^2 f_{1,1}))
\end{aligned}
\tag{B.9}
$$

$$
\begin{aligned}
&g(f)_{1,2} = \\
&(f^6(2f_{1,1}f_{1,2} + 2{f_{1,2}}^3 + 2f_{1,1}f_{2,1}f_{2,2} + 2f_{1,2}{f_{2,2}}^2 - {f_{1,1}}^2f_{1,2} - {f_{1,2}}^3 - {f_{2,1}}^2f_{1,2} - {f_{2,2}}^2f_{1,2}) \\
&+ f^4(2{f_{1,3}}^2f_{1,2} + 2f_{1,1}f_{3,1}f_{3,2} + 2f_{1,2}{f_{3,2}}^2 - {f_{1,3}}^2f_{1,2} - {f_{2,3}}^2f_{1,2} - {f_{3,1}}^2f_{1,2} - {f_{3,2}}^2f_{1,2}) \\
&+ f^3(2f_{1,3}f_{2,3}f_{2,2}) + f^2(2f_{1,3}f_{3,3}f_{3,2} - {f_{3,3}}^2f_{1,2}))
\end{aligned}
\tag{B.10}
$$

$$
\begin{aligned}
&g(f)_{1,3} = \\
&(f^5(2{f_{1,1}}^2f_{1,3} + 2{f_{1,2}}^2f_{1,3} + 2f_{1,1}f_{2,1}f_{2,3} + 2f_{1,2}f_{2,2}f_{2,3} - {f_{1,1}}^2f_{1,3} - {f_{1,2}}^2f_{1,3} - {f_{2,1}}^2f_{1,3} - {f_{2,2}}^2f_{1,3}) \\
&+ f^3(2{f_{1,3}}^3 + 2f_{1,1}f_{3,1}f_{3,3} + 2f_{1,2}f_{3,2}f_{3,3} - {f_{1,3}}^3 - {f_{2,3}}^2f_{1,3} - {f_{3,1}}^2f_{1,3} - {f_{3,2}}^2f_{1,3}) \\
&+ f^2(2f_{1,3}{f_{2,3}}^2) + f(f_{1,3}{f_{3,3}}^2 - {f_{3,3}}^2f_{1,3}))
\end{aligned}
\tag{B.11}
$$

$$
\begin{aligned}
&g(f)_{2,1} = \\
&(f^6(2f_{2,1}{f_{1,1}}^2 + 2f_{2,2}f_{1,2}f_{1,1} + 2{f_{2,1}}^3 + 2{f_{2,2}}^2f_{2,1} - {f_{1,1}}^2f_{2,1} - {f_{1,2}}^2f_{2,1} - {f_{2,1}}^3 - {f_{2,2}}^2f_{2,1}) \\
&+ f^4(2f_{2,3}f_{1,3}f_{1,1} + 2{f_{2,3}}^2f_{2,1} + 2f_{2,1}{f_{3,1}}^2 + 2f_{2,2}f_{3,2}f_{3,1} - {f_{1,3}}^2f_{2,1} - {f_{2,3}}^2f_{2,1} - {f_{3,1}}^2f_{2,1} - {f_{3,2}}^2f_{2,1}) \\
&+ f^2(2f_{2,3}f_{3,3}f_{3,1} - {f_{3,3}}^2f_{2,1}))
\end{aligned}
\tag{B.12}
$$

$$
\begin{aligned}
&g(f)_{2,2} = \\
&(f^6(2f_{2,1}f_{1,1}f_{1,2} + 2f_{2,2}{f_{1,2}}^2 + 2{f_{2,1}}^2f_{2,2} + 2{f_{2,2}}^3 - {f_{1,1}}^2f_{2,2} - {f_{1,2}}^2f_{2,2} - {f_{2,1}}^2f_{2,2} - {f_{2,2}}^3) \\
&+ f^4(2f_{2,3}f_{1,3}f_{1,2} + 2{f_{2,3}}^2f_{2,2} + 2f_{2,1}f_{3,1}f_{3,2} + 2f_{2,2}{f_{3,2}}^2 - {f_{1,3}}^2f_{2,2} - {f_{2,3}}^2f_{2,2} - {f_{3,1}}^2f_{2,2} - {f_{3,2}}^2f_{2,2}) \\
&+ f^2(2f_{2,3}f_{3,3}f_{3,2} - {f_{3,3}}^2f_{2,2}))
\end{aligned}
\tag{B.13}
$$

$$
\begin{aligned}
&g(f)_{2,3} = \\
&(f^5(2f_{2,1}f_{1,1}f_{1,3} + 2f_{2,2}f_{1,2}f_{1,3} + 2{f_{2,1}}^2f_{2,3} + 2{f_{2,2}}^2f_{2,3} - {f_{1,1}}^2f_{2,3} - {f_{1,2}}^2f_{2,3} - {f_{2,1}}^2f_{2,3} - {f_{2,2}}^2f_{2,3}) \\
&+ f^3(2f_{2,3}{f_{1,3}}^2 + 2{f_{2,3}}^3 + 2f_{2,1}f_{3,1}f_{3,3} + 2f_{2,2}f_{3,2}f_{3,3} - {f_{1,3}}^2f_{2,3} - {f_{2,3}}^3 - {f_{3,1}}^2f_{2,3} - {f_{3,2}}^2f_{2,3}) \\
&+ f(2f_{2,3}{f_{3,3}}^2 - {f_{3,3}}^2f_{2,3}))
\end{aligned}
\tag{B.14}
$$

$$
\begin{aligned}
&g(f)_{3,1} = \\
&(f^5(2f_{3,1}{f_{1,1}}^2 + 2f_{3,2}f_{1,2}f_{1,1} + 2f_{3,1}{f_{2,1}}^2 + 2f_{3,2}f_{2,2}f_{2,1} - {f_{1,1}}^2f_{3,1} - {f_{1,2}}^2f_{3,1} - {f_{2,1}}^2f_{3,1} - {f_{2,2}}^2f_{3,1}) \\
&+ f^3(2f_{3,3}f_{1,3}f_{1,1} + 2f_{3,3}f_{2,3}f_{2,1} + 2{f_{3,1}}^3 + 2{f_{3,2}}^2f_{3,1} - {f_{1,3}}^2f_{3,1} - {f_{2,3}}^2f_{3,1} - {f_{3,1}}^3 - {f_{3,2}}^2f_{3,1}) \\
&+ f(2{f_{3,3}}^2f_{3,1} - {f_{3,3}}^2f_{3,1}))
\end{aligned}
\tag{B.15}
$$

$$
\begin{aligned}
&g(f)_{3,2} = \\
&(f^5(2f_{3,1}f_{1,1}f_{1,2} + 2f_{3,2}{f_{1,2}}^2 + 2f_{3,1}f_{2,1}f_{2,2} + 2f_{3,2}{f_{2,2}}^2 - {f_{1,1}}^2f_{3,2} - {f_{1,2}}^2f_{3,2} - {f_{2,1}}^2f_{3,2} - {f_{2,2}}^2f_{3,2}) \\
&+ f^3(2f_{3,3}f_{1,3}f_{1,2} + 2f_{3,3}f_{2,3}f_{2,2} + 2{f_{3,1}}^2f_{3,2} + 2{f_{3,2}}^3 - {f_{1,3}}^2f_{3,2} - {f_{2,3}}^2f_{3,2} - {f_{3,1}}^2f_{3,2} - {f_{3,2}}^3) \\
&+ f(2{f_{3,3}}^2f_{3,2} - {f_{3,3}}^2f_{3,2}))
\end{aligned}
\tag{B.16}
$$

$$
\begin{aligned}
&g(f)_{3,3} = \\
&(f^4(2f_{3,1}f_{1,1}f_{1,3} + 2f_{3,2}f_{1,2}f_{1,3} + 2f_{3,1}f_{2,1}f_{2,3} + 2f_{3,2}f_{2,2}f_{2,3} - {f_{1,1}}^2f_{3,3} - {f_{1,2}}^2f_{3,3} - {f_{2,1}}^2f_{3,3} - {f_{2,2}}^2f_{3,3}) \\
&+ f^2(2f_{3,3}{f_{1,3}}^2 + 2f_{3,3}{f_{2,3}}^2 + 2{f_{3,1}}^2f_{3,3} + 2{f_{3,2}}^2f_{3,3} - {f_{1,3}}^2f_{3,3} - {f_{2,3}}^2f_{3,3} - {f_{3,1}}^2f_{3,3} - {f_{3,2}}^2f_{3,3}) \\
&+ (2{f_{3,3}}^2 - {f_{3,3}}^2))
\end{aligned}
\tag{B.17}
$$

For the simplicity, $G(f)$ matrix is denoted by focal length, f, and $a_{i,j}$ coefficients, which will be beneficial for the next step:

$$
\begin{aligned}
&G(f) = \\
&\begin{bmatrix}
(f^6(a_{1,1}) + f^4(a_{1,2}) + f^3(a_{1,3}) + f^2(a_{1,4})) & (f^6(a_{2,1}) + f^4(a_{2,2}) + f^3(a_{2,3}) + f^2(a_{2,4})) & (f^5(a_{3,1}) + f^3(a_{3,2}) + f^2(a_{3,3}) + f(a_{3,4})) \\
(f^6(a_{4,1}) + f^4(a_{4,2}) + f^2(a_{4,3})) & (f^6(a_{5,1}) + f^4(a_{5,2}) + f^2(a_{5,3})) & (f^5(a_{6,1}) + f^3(a_{6,2}) + f(a_{6,3})) \\
(f^5(a_{7,1}) + f^3(a_{7,2}) + f(a_{7,3})) & (f^5(a_{8,1} + f^3(a_{8,2}) + f(a_{8,3})) & (f^4(a_{9,1}) + f^2(a_{9,2}) + (a_{9,3}))
\end{bmatrix}
\end{aligned}
\tag{B.18}
$$

In the ideal case, the $L2$ norm of $G(f)$ should be equal to 0 with the true focal length.

So, the norm of the matrix, $\|G(f)\|^2$, is calculated as 12^{th} degree polynomial:

$$\begin{aligned}
&\|G(f)\|^2 = \\
&k_1 \times f^{12} + k_2 \times f^{10} + k_3 \times f^9 + k_4 \times f^8 + \\
&k_5 \times f^7 + k_6 \times f^8 + k_7 \times f^9 + k_8 \times f^8 + \\
&k_9 \times f^3 + k_{10} \times f^2 + k_{11} \times f^0 = 0
\end{aligned} \tag{B.19}$$

The correspondent k_i values are written in the following set of equations:

$$\begin{aligned}
k_1 &= a_{1,1}a_{1,1} + a_{2,1}a_{2,1} + a_{4,1}a_{4,1} + a_{5,1}a_{5,1} \\
k_2 &= 2a_{1,1}a_{1,2} + 2a_{2,1}a_{2,2} + a_{3,1}a_{3,1} + 2a_{4,1}a_{4,2} + 2a_{5,1}a_{5,2} + a_{6,1}a_{6,1} \\
&+ a_{7,1}a_{7,1} + a_{8,1}a_{8,1} \\
k_3 &= 2a_{1,1}a_{1,3} + 2a_{2,1}a_{2,3} \\
k_4 &= 2a_{1,1}a_{1,4} + a_{1,2}a_{1,2} + 2a_{2,1}a_{2,4} + 2a_{3,1}a_{3,2} + 2a_{4,1}a_{4,3} + a_{4,2}a_{4,2} + 2a_{5,1}a_{5,3} \\
&a_{5,2}a_{5,2} + 2a_{6,1}a_{6,2} + 2a_{7,1}a_{7,2} + 2a_{8,1}a_{8,2} + a_{9,1}a_{9,1} \\
k_5 &= 2a_{1,2}a_{1,3} + 2a_{2,2}a_{2,3} + 2a_{3,1}a_{3,3} \\
k_6 &= 2a_{1,2}a_{1,4} + a_{1,3}a_{1,3} + a_{1,4}a_{1,3} + 2a_{2,2}a_{2,4} + a_{2,3}a_{2,3} + 2a_{3,1}a_{3,4} \\
&+ a_{3,2}a_{3,2} + 2a_{4,2}a_{4,3} + 2a_{5,2}a_{5,3} + 2a_{6,1}a_{6,3} + a_{6,2}a_{6,2} + 2a_{7,1}a_{7,3} \\
&+ a_{7,2}a_{7,2} + a_{8,1}a_{8,3} + a_{8,2}a_{8,2} + a_{9,1}a_{9,2} \\
k_7 &= a_{1,3}a_{1,4} + 2a_{2,3}a_{2,4} + 2a_{3,2}a_{3,3} \\
k_8 &= a_{1,4}a_{1,4} + a_{2,4}a_{2,4} + 2a_{3,2}a_{3,4} + a_{3,3}a_{3,3} + a_{4,3}a_{4,3} + a_{5,3}a_{5,3} \\
&+ 2a_{6,2}a_{6,3} + 2a_{7,2}a_{7,3} + 2a_{8,2}a_{8,3} + 2a_{9,1}a_{9,3} + a_{9,2}a_{9,2} \\
k_9 &= 2a_{3,3}a_{3,4} \\
k_{10} &= a_{3,4}a_{3,4} + a_{6,3}a_{6,3} + a_{7,3}a_{7,3} + a_{8,3}a_{8,3} + 2a_{9,2}a_{9,3} \\
k_{11} &= a_{9,3}a_{9,3}
\end{aligned} \tag{B.20}$$

where the $a_{i,j}$ values are stated in Equation B.18.

The definition of function, $H(F, focal1 = f, focal2 = f) = 0$, which is given in

Chapter 4.1, is equal to norm of $G(f)$ with given fundamental matrix, F. In other words, the function defined in Equation 4.1 is stated as following:

$$\begin{aligned} &H(F, focal1 = f, focal2 = f) = \\ &k_1 \times f^{12} + k_2 \times f^{10} + k_3 \times f^9 + k_4 \times f^8 + \\ &k_5 \times f^7 + k_6 \times f^8 + k_7 \times f^9 + k_8 \times f^8 + \\ &k_9 \times f^3 + k_{10} \times f^2 + k_{11} \times f^0 = 0 \end{aligned} \tag{B.21}$$

Apart from estimation of equal focal length of two views using the geometric constraint, i.e. fundamental matrix, F, there is another approach for estimating the unknown focal length if the other focal length is known in addition to F matrix. In this case, camera matrix of known camera, K_1 is merged with the F matrix, which is called $\hat{F}$, and essential matrix is represented in terms of $\hat{F}$ and $focal2$. This is followed by rewriting *trace constraint* using this relation. Similar steps with the previous derivation is conducted, and the $\hat{G}(f)$ matrix is computed in terms of unknown focal length, f. The $L2$ norm of this matrix should be 0 with the true focal length. So, the norm of this matrix, $\|\hat{G}(f)\|^2$, is calculated as 6^{th} degree polynomial:

$$\begin{aligned} &\|\hat{G}(f)\|^2 = \\ &t_1 \times f^6 + t_2 \times f^4 + t_3 \times f^2 + t_4 = 0 \end{aligned} \tag{B.22}$$

The correspondent t_i values are written in the following set of equations utilizing the elements of $\hat{F}$ matrix:

$$\begin{aligned} t_1 &= \hat{a}_{1,1}^2 + \hat{b}_{1,1}^2 + \hat{c}_{1,1}^2 + \hat{d}_{1,1}^2 + \hat{e}_{1,1}^2 + \hat{g}_{1,1}^2 \\ t_2 &= 2\hat{a}_{1,1}\hat{a}_{1,2} + 2\hat{b}_{1,1}\hat{b}_{1,2} + 2\hat{c}_{1,1}\hat{c}_{1,2} \\ &+ 2\hat{d}_{1,1}\hat{d}_{1,2} + 2\hat{e}_{1,1}\hat{e}_{1,2} + 2\hat{g}_{1,1}\hat{g}_{1,2} \\ &+ \hat{h}_{1,1}^2 + \hat{j}_{1,1}^2 + \hat{k}_{1,1}^2 \\ t_3 &= \hat{a}_{1,2}^2 + \hat{b}_{1,2}^2 + \hat{c}_{1,2}^2 + \hat{d}_{1,2}^2 + \hat{d}_{1,2}^2 + \hat{g}_{1,2}^2 \\ &+ 2\hat{h}_{1,1}\hat{h}_{1,2} + 2\hat{j}_{1,1}\hat{j}_{1,2} + 2\hat{k}_{1,1}\hat{k}_{1,2} \\ t_4 &= \hat{h}_{1,2}^2 + \hat{j}_{1,2}^2 + \hat{k}_{1,2}^2 \end{aligned} \tag{B.23}$$

where the coefficients are related to the elements of $\hat{F}$ as it is stated in the following equations:

$$\hat{a}_{1,1} = ((\hat{f}_{1,1}\hat{f}_{1,1}\hat{f}_{1,1})) + (\hat{f}_{1,1}((\hat{f}_{1,2}\hat{f}_{1,2}))) + (\hat{f}_{1,1}((\hat{f}_{1,3}\hat{f}_{1,3}))) + (\hat{f}_{1,1}((\hat{f}_{2,1}\hat{f}_{2,1})))$$
$$+ (2(\hat{f}_{1,2}\hat{f}_{2,2}\hat{f}_{2,1})) + (2(\hat{f}_{1,3}\hat{f}_{3,3}\hat{a}_{2,1})) - (\hat{f}_{1,1}((\hat{a}_{2,2}\hat{f}_{2,2}))) - (\hat{f}_{1,1}((\hat{f}_{2,3}\hat{f}_{2,3})))$$

(B.24)

$$\hat{a}_{1,2} = (\hat{f}_{1,1}((\hat{f}_{3,1}\hat{f}_{3,1}))) + (2(\hat{f}_{1,2}\hat{f}_{3,2}\hat{f}_{3,1})) + (2(\hat{f}_{1,3}\hat{f}_{3,3}\hat{f}_{3,1})) - (\hat{f}_{1,1}((\hat{f}_{3,2}\hat{f}_{3,2})))$$
$$- (\hat{f}_{1,1}((\hat{f}_{3,3}\hat{f}_{3,3})))$$

(B.25)

$$\hat{b}_{1,1} = (\hat{f}_{1,2}((\hat{f}_{1,1}\hat{f}_{1,1}))) + ((\hat{f}_{1,2}\hat{f}_{1,2}\hat{f}_{1,2})) + (\hat{f}_{1,2}((\hat{f}_{1,3}\hat{f}_{1,3}))) + (2\hat{f}_{1,1}\hat{f}_{2,1}\hat{f}_{2,2}) +$$
$$(\hat{f}_{1,2}((\hat{f}_{2,2}\hat{f}_{2,2}))) + (2\hat{f}_{1,3}\hat{f}_{3,3}\hat{f}_{2,2}) - (\hat{f}_{1,2}((\hat{f}_{2,1}\hat{f}_{2,1}))) - (\hat{f}_{1,2}((\hat{f}_{2,3}\hat{f}_{2,3})))$$

(B.26)

$$\hat{b}_{1,2} = (2\hat{f}_{1,1}\hat{f}_{3,1}\hat{f}_{3,2}) + (\hat{f}_{1,2}((\hat{f}_{3,2}\hat{f}_{3,2}))) + (2\hat{f}_{1,3}\hat{f}_{3,3}\hat{f}_{3,2}) - (\hat{f}_{1,2}((\hat{f}_{3,1}\hat{f}_{3,1})))$$
$$- (\hat{f}_{1,2}((\hat{f}_{3,3}\hat{f}_{3,3})))$$

(B.27)

$$\hat{c}_{1,1} = (\hat{f}_{1,3}((\hat{f}_{1,1}\hat{f}_{1,1}))) + (\hat{f}_{1,3}((\hat{f}_{1,2}\hat{f}_{1,2}))) + ((\hat{f}_{1,3}\hat{f}_{1,3}\hat{f}_{1,3})) + (2\hat{f}_{1,1}\hat{f}_{2,1}\hat{f}_{2,3})$$
$$+ (2\hat{f}_{1,2}\hat{f}_{2,2}\hat{f}_{2,3}) + (\hat{f}_{1,3}((\hat{f}_{2,3}\hat{f}_{2,3}))) - (\hat{f}_{1,3}((\hat{f}_{2,1}\hat{f}_{2,1}))) - (\hat{f}_{1,3} \times ((\hat{f}_{2,2} \times \hat{f}_{2,2})))$$

(B.28)

$$\hat{c}_{1,2} = (2\hat{f}_{1,1}\hat{f}_{3,1}\hat{f}_{3,3}) + (2\hat{f}_{1,2}\hat{f}_{3,2}\hat{f}_{3,3}) + (\hat{f}_{1,3}((\hat{f}_{3,3}\hat{f}_{3,3}))) - (\hat{f}_{1,3}((\hat{f}_{3,1}\hat{f}_{3,1})))$$
$$- (\hat{f}_{1,3}((\hat{f}_{3,2}\hat{f}_{3,2})))$$

(B.29)

$$\hat{d}_{1,1} = (\hat{f}_{2,1}((\hat{f}_{1,1}\hat{f}_{1,1}))) + (2\hat{f}_{2,2}\hat{f}_{1,2}\hat{f}_{1,1}) + (2\hat{f}_{1,3}\hat{f}_{2,3}\hat{f}_{1,1}) + ((\hat{f}_{2,1}\hat{f}_{2,1}\hat{f}_{2,1})) + (\hat{f}_{2,1}((\hat{f}_{2,2}\hat{f}_{2,2}))) + (\hat{f}_{2,1}((\hat{f}_{2,3}\hat{f}_{2,3}))) - (\hat{f}_{2,1}((\hat{f}_{1,2}\hat{f}_{1,2}))) - (\hat{f}_{2,1} \times ((\hat{f}_{1,3} \times \hat{f}_{1,3})))$$

(B.30)

$$\hat{d}_{1,2} = (\hat{f}_{2,1}((\hat{f}_{3,1}\hat{f}_{3,1}))) + (2\hat{f}_{2,2}\hat{f}_{3,2}\hat{f}_{3,1}) + (2 \times \hat{f}_{2,3}\hat{f}_{3,3}\hat{f}_{3,1}) - (\hat{f}_{2,1}((\hat{f}_{3,2}\hat{f}_{3,2}))) - (\hat{f}_{2,1}((\hat{f}_{3,3}\hat{f}_{3,3})))$$

(B.31)

$$\hat{e}_{1,1} = (2\hat{f}_{1,1}\hat{f}_{2,1}\hat{f}_{1,2}) + (\hat{f}_{2,2}((\hat{f}_{1,2}\hat{f}_{1,2}))) + (2\hat{f}_{1,3}\hat{f}_{2,3}\hat{f}_{1,2}) + (\hat{f}_{2,2}((\hat{f}_{2,1}\hat{f}_{2,1}))) + ((\hat{f}_{2,2}\hat{f}_{2,2}\hat{f}_{2,2})) + (\hat{f}_{2,2}((\hat{f}_{2,3}\hat{f}_{2,3}))) - (\hat{f}_{2,2}((\hat{f}_{1,3}\hat{f}_{1,3}))) - (\hat{f}_{2,2}((\hat{f}_{1,1}\hat{f}_{1,1})))$$

(B.32)

$$\hat{e}_{1,2} = (2\hat{f}_{2,1}\hat{f}_{3,1}\hat{f}_{3,2}) + (\hat{f}_{2,2}((\hat{f}_{3,2}\hat{f}_{3,2}))) + (2\hat{f}_{2,3}\hat{f}_{3,3}\hat{f}_{3,2}) - (\hat{f}_{2,2}((\hat{f}_{3,1}\hat{f}_{3,1}))) - (\hat{f}_{2,2}((\hat{f}_{3,3}\hat{f}_{3,3})))$$

(B.33)

$$\hat{g}_{1,1} = (2\hat{f}_{1,1}\hat{f}_{2,1}\hat{f}_{1,3}) + (2\hat{f}_{2,2}\hat{f}_{1,2}\hat{f}_{1,3}) + (\hat{f}_{2,3}((\hat{f}_{1,3}\hat{f}_{1,3}))) + (\hat{f}_{2,3}((\hat{f}_{2,1}\hat{f}_{2,1}))) + (\hat{f}_{2,3}((\hat{f}_{2,2}\hat{f}_{2,2}))) + ((\hat{f}_{2,3}\hat{f}_{2,3}\hat{f}_{2,13})) - (\hat{f}_{2,3}((\hat{f}_{1,1}\hat{f}_{1,1}))) - (\hat{f}_{2,3}((\hat{f}_{1,2}\hat{f}_{1,2})))$$

(B.34)

$$\hat{g}_{1,2} = (2\hat{f}_{2,1}\hat{f}_{3,1}\hat{f}_{3,3}) + (2\hat{f}_{2,2}\hat{f}_{3,2}\hat{f}_{3,3}) + (\hat{f}_{2,3}((\hat{f}_{3,3}\hat{f}_{3,3}))) - (\hat{f}_{2,3}((\hat{f}_{3,1}\hat{f}_{3,1}))) - (\hat{f}_{2,3} \times ((\hat{f}_{3,2} \times \hat{f}_{3,2})))$$

(B.35)

$$\hat{h}_{1,1} = (\hat{f}_{3,1}((\hat{f}_{1,1}\hat{f}_{1,1}))) + (2\hat{f}_{1,2}\hat{f}_{1,1}\hat{f}_{3,2}) + (2\hat{f}_{1,3}\hat{f}_{1,1}\hat{f}_{3,3}) + (\hat{f}_{3,1}((\hat{f}_{2,1}\hat{f}_{2,1}))) + (2\hat{f}_{2,2}\hat{f}_{2,1}\hat{f}_{3,2}) + (2\hat{f}_{2,1}\hat{f}_{2,3}\hat{f}_{3,3}) - (\hat{f}_{3,1}((\hat{f}_{1,2}\hat{f}_{1,2}))) - (\hat{f}_{3,1}((\hat{f}_{1,3}\hat{f}_{1,3}))) - (\hat{f}_{3,1} \times ((\hat{f}_{2,2} \times \hat{f}_{2,2}))) - (\hat{f}_{3,1} \times ((\hat{f}_{2,3} \times \hat{f}_{2,3})))$$

(B.36)

$$\hat{h}_{1,2} = ((\hat{f}_{3,1}\hat{f}_{3,1}\hat{f}_{3,1})) + (\hat{f}_{3,1}((\hat{f}_{3,2}\hat{f}_{3,2}))) + (\hat{f}_{3,1}((\hat{f}_{3,3}\hat{f}_{3,3}))) \tag{B.37}$$

$$\begin{aligned}\hat{j}_{1,1} &= (2\hat{f}_{1,1}\hat{f}_{1,2}\hat{f}_{3,1}) + (\hat{f}_{3,2}((\hat{f}_{1,2}\hat{f}_{1,2}))) + (2\hat{f}_{1,3}\hat{f}_{1,2}\hat{f}_{3,3}) + (2\hat{f}_{2,1}\hat{f}_{2,2}\hat{f}_{3,1}) \\ &+ (\hat{f}_{3,2}((\hat{f}_{2,3}\hat{f}_{2,2}))) + (2\hat{f}_{2,3}\hat{f}_{2,2}\hat{f}_{3,3}) - (\hat{f}_{3,2}((\hat{f}_{1,1}\hat{f}_{1,1}))) - (\hat{f}_{3,2}((\hat{f}_{1,3}\hat{f}_{1,3}))) \\ &- (\hat{f}_{3,2}((\hat{f}_{2,1}\hat{f}_{2,1}))) - (\hat{f}_{3,2}((\hat{f}_{2,3}\hat{f}_{2,3})))\end{aligned} \tag{B.38}$$

$$\hat{j}_{1,2} = (\hat{f}_{3,2}((\hat{f}_{3,1}\hat{f}_{3,1}))) + ((\hat{f}_{3,2}\hat{f}_{3,2}\hat{f}_{3,2})) + (\hat{f}_{3,2}((\hat{f}_{3,3}\hat{f}_{3,3}))) \tag{B.39}$$

$$\begin{aligned}\hat{k}_{1,1} &= (2\hat{f}_{1,1}\hat{f}_{1,3}\hat{f}_{3,1}) + (2\hat{f}_{1,2}\hat{f}_{1,3}\hat{f}_{3,2}) + (\hat{f}_{3,3}((\hat{f}_{1,3}\hat{f}_{1,3}))) + (2\hat{f}_{2,1}\hat{f}_{3,1}\hat{f}_{2,3}) \\ &+ (2\hat{f}_{2,2}\hat{f}_{3,2}\hat{f}_{2,3}) + (\hat{f}_{3,3}((\hat{f}_{2,3}\hat{f}_{2,3}))) - (\hat{f}_{3,3}((\hat{f}_{1,1}\hat{f}_{1,1}))) - (\hat{f}_{3,3}((\hat{f}_{1,2}\hat{f}_{1,2}))) \\ &- (\hat{f}_{3,3}((\hat{f}_{2,1}\hat{f}_{2,1}))) - (\hat{f}_{3,3}((\hat{f}_{2,2}\hat{f}_{2,2})))\end{aligned} \tag{B.40}$$

$$\hat{k}_{1,2} = (\hat{k}_{3,3}((\hat{k}_{3,1}\hat{k}_{3,1}))) + (\hat{k}_{3,3}((\hat{k}_{3,2}\hat{k}_{3,2}))) + ((\hat{k}_{3,3} \times \hat{k}_{3,3} \times \hat{k}_{3,3})) \tag{B.41}$$

In conclusion, the definition of function, $P(\hat{F}, focal2 = f)$, which is given in Chapter 4.1, is equal to norm of $\hat{G}(f)$ with given fundamental matrix,F, and known focal length. That is, the function defined in Equation 4.2 is stated as following:

$$\begin{aligned}&P(F, focal2 = f) = \\ &t_1 \times f^6 + t_2 \times f^4 + t_3 \times f^2 + t_4 = 0\end{aligned} \tag{B.42}$$