A NEW APPROACH FOR SELF-CALIBRATING CAMERAS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

CAHİT GÜREL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MODELING AND DESIGN OF ENGINEERING SYSTEMS (MODES)
(MAIN FIELD OF STUDY: MECHATRONICS ENGINEERING)

FEBRUARY 2019

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

Prof. Dr. Ali KARA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Doctor of Philosophy in Modeling and Design of Engineering Systems (MODES) (Main Field of Study: Mechatronics Engineering) Department, Atilim University**.

Assoc. Prof. Dr. Ender KESKİNKILIÇ
Head of Department

This is to certify that we have read the thesis A NEW APPROACH FOR SELF-CALIBRATING CAMERAS submitted by CAHİT GÜREL and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Dr. Ahmet GÜNEŞ
Co-Supervisor

Asst. Prof. Dr. Hakan TORA
Supervisor

Examining Committee Members:

Asst. Prof. Dr. Baran USLU
Electrical & Electronics Eng. Dept., Atılım University

Asst. Prof. Dr. Hakan TORA
Aircraft Electrical & Electronics Dept., Atılım University

Asst. Prof. Dr. Erhan GÖKÇAY
Software Eng. Dept., Atılım University

Asst. Prof. Dr. A. Buğra KOKU
Mechanical Eng. Dept., ODTÜ

Asst. Prof. Dr. Kutluk Bilge ARIKAN
Mechanical Eng. Dept., TEDÜ

Date: February 11, 2019

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : CAHİT GÜREL

Signature :

ABSTRACT

A NEW APPROACH FOR SELF-CALIBRATING CAMERAS

Gürel, Cahit

Ph.D., Department of Modeling and Design of Engineering Systems (MODES)

(Main Field of Study: Mechatronics Engineering)

Supervisor : Asst. Prof. Dr. Hakan TORA

Co-Supervisor : Dr. Ahmet GÜNEŞ

February 2019, 51 pages

Camera is one of the most important sensors in robotic applications. Calibrated cameras provide more information than the uncalibrated ones. Intrinsic parameters of a camera can deteriorate due to mechanical and thermal changes in environment. Therefore self-calibration is required for robotic operations. Since self-calibration does not require any known template objects in the process, it is more flexible and extracting a few fixed points between calibration images is enough for self-calibration. We propose a new method for simpler and more accurate self-calibration method by incorporating some of the extrinsic parameters of camera along with some assumptions which are true for present day cameras. Moreover, we have included a basic point detection, tracking and association approach for the task. Proposed method is tested and compared with another self calibration method using synthetic data, a mobile robot with a camera in V-REP simulation environment and physical implementation with articulated robot arm. The results indicate the effectiveness of the new approach with respect to other self-calibration approaches for planer motion of the camera.

Keywords: self-calibration, tracking, epipolar geometry

ÖZ

KAMERALARIN ÖZDEVİMLİ KALİBRASYONU İÇİN YENİ BİR YAKLAŞIM

Gürel, Cahit

Doktora, Mühendislik Sistemlerinin Modellenmesi ve Tasarımı Ana Bilim Dalı

Tez Yöneticisi : Dr. Öğr. Üyesi Hakan TORA

Ortak Tez Yöneticisi : Dr. Ahmet GÜNEŞ

Şubat 2019, 51 sayfa

Kamera, robotik uygulamalardaki en önemli sensörlerden birisidir. Kalibre edilmiş bir kamera, kalibre edilmemişine göre daha fazla bilgi sunabilir. Ayrıca, kalibre edilmiş kameranın kalibrasyon değerleri mekanik ve termal değişikler nedeni ile zamanla bozulabilir. Dolayısıyla mobil robotik platformalar için özdevimli kamera kalibrasyonu ihtiyaç oluşturmaktadır. Özdevimli kamera kalibrasyonu için ölçüleri bilinen bir objeye ihtiyaç duyulmaması nedeni ile hem daha esnektir hem de görüntü üzerindeki az sayıda nokta eşleştirmesi kalibrasyon hessaplamaları için yeterlidir. Bu kapsamda, kameranın pozisyonu ve yönelim açısı bilgilerini katarak daha basit ve daha yüksek doğruluk değerine sahip yeni bir kalibrasyon metodu öneriyoruz. Önerilen metot sentetik veriler üzerinde, simülasyon ortamında ve gerçek bir donanım üzerinde test edilmiştir. Sonuçlar diğer özdevimli kamera kalibrasyonu metoduyla karşılaştırılmıştır. Düzlemsel hareket eden kameralar için sonuçlarımız diğer özdevimli kamera kalibrasyonu uygulamaları karşısında etkili olduğunu göstermektedir.

Anahtar Kelimeler: özdevimli kalibrasyon, takip etme, epipolar geometri

I dedicate this thesis to whom never leaves their great supports along the journey...

ACKNOWLEDGMENTS

First, I would like to express my enormous gratitude to my two thesis supervisors, Dr. Ahmet GÜNEŞ and Asst. Prof. Dr. Hakan TORA, for their encouragement, guidance and advice continuous during my research study. Their wisdom and expert guidance have been indispensable in the elaboration of this thesis and without their efforts, I would not have had the support to finish this work. It was a pleasure to work under the supervision of these two true gentlemen.

I take this opportunity to express my gratitude, sincere, and appreciation to Asst. Prof. Dr. Erhan GÖKÇAY and Asst. Prof. Dr. A. Buğra KOKU for their scientific advice and knowledge and many insightful discussions and suggestions.

I shall also thank to Asst. Prof. Dr. Kutluk Bilge ARIKAN and Asst. Prof. Dr. Baran USLU for oaring me their comments and suggestions to accomplish my work in appropriate way.

Furthermore, the special thanks go to my parents, Zübeyde and Celaletdin; my brother Ahmet; my biggest supporter Meral ADAY and my friend Çağatay ÖZTÜRK, for their continuous encouragement and support.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

$\mathbf{\Omega}$	:	Absolute conic
$\mathbf{Q}_{\infty}^{*}$	:	Absolute quadric
$\mathbf{P}$	:	Camera matrix
$\mathbf{\Omega}_{\infty}^{*}$	:	Dual of absolute conic
ω^{*}	:	Dual image of absolute conic
$\mathbf{e}$	:	Epipole
f	:	Focal length
f_x	:	Focal length in x direction
f_y	:	Focal length in y direction
$\mathbf{F}$	:	Fundamental matrix
$\mathcal{N}(\mu, \sigma)$	:	Gaussian distribution with mean μ and variance σ^2
ω	:	Image of absolute conic
π	:	Image plane
$\mathbf{K}$	:	Intrinsic parameters of camera, calibration matrix
$\mathbf{X}$	:	Matrix Representation
$\mathbf{H}$	:	Measurement model of Kalman filter
o_x	:	Principal point in x direction
o_y	:	Principal point in y direction
$\mathbf{R}$	:	Rotation matrix
α	:	Rotation angle about x axis
β	:	Rotation angle about y axis
γ	:	Rotation angle about z axis

X	:	Scalar Representation
s	:	Skew
$\mathbf{A}$	:	State transition matrix of Kalman filter
$\mathbf{x}_k$	:	State vector of Kalman filter
$\mathbf{T}$	:	Translation vector
T_X	:	Translation in x axis
T_Y	:	Translation in y axis
T_Z	:	Translation in z axis
$\mathcal{U}(a, b)$	:	Uniform distribution between a and b
$\hat{\mathbf{x}}$	:	Unit vector in x direction
$\hat{\mathbf{y}}$	:	Unit vector in y direction

CHAPTER 1

INTRODUCTION

Since camera can provide more information and the information from camera can be exploited in various ways, it is one of the most important sensors in robotic implementations. Even with a single frame, objects in the scene can be detected and classified. Cameras are used for surveillance, object tracking [2], object detection, object inspection [3], 3D reconstruction, navigation [4], depth estimation [5], Simultaneous Localization and Mapping (SLAM) [6], structure from motion [7] and recognition applications [8]. Nowadays metric information is becoming important for robotic implementations, i.e. autonomous car [9], visual servoing [10], object grasping [11], visual odometry [12], augmented reality [13], navigation and mapping [14]. Since metric information is only available with a calibrated camera, camera calibration is compulsory. With an uncalibrated camera, color detection, object or shape detection, pattern recognition, motion estimation in 2D are still possible but without the metric information [15].

Calibration procedures are divided into two main categories. Calibration with object and without object. When calibration is done with special objects whose dimensions are accurately known, the calibration method is called manual calibration [1]. The objects utilized for this kind of calibration can be 1D, 2D or 3D, such as a rod, a checker board or a bended checker board. The alternative approach is to extract information from the scene. Since no special objects or *a priori* information about the scene is required, this approach is called self-calibration.

Parameters of a camera can change under the mechanical effects or thermal changes. Thus, it would cause problems in the estimation of metric information of objects in

the scene and thus wrong decision making of robotic system. Therefore calibration parameters should be checked under these changes. If manual calibration is going to be performed, process of robotic system should be stopped and calibration has to be performed by users. However, self-calibration can be performed on the fly, any time anywhere and does not depend on the user operations [1].

In self-calibration process, a set of uncalibrated images in the static scene are used [16]. Since correlation information between the points in scenes are enough, self-calibration is also known as 0D approach and presents a more general framework. However, it also means higher theoretical and practical challenges, such as the extraction of points to be matched and accurate localization of points in images.

There are several self-calibration methods proposed in the literature. First known self-calibration method is performed by Faugeras in the 1992 [17]. Then, lots of different approaches and optimization methods are proposed and camera self-calibration is still an active research area. These methods are classified as direct or stratified methods [16]. The most commonly used method among the direct methods are the Absolute Dual Quadric (ADQ) method and Kruppa equations. ADQ uses absolute dual quadric instead of the absolute conic and requires minimum three images for calibration [18]. On the other hand, solving the calibration problem using Kruppa equations require same number of images but less computation than ADQ [17]. If the only unknown of parameter is the focal length of the camera, then two images are enough to calculate focal length using Kruppa Equations [16].

Since there is no calibration object in self-calibration, the calculations rely on the point correspondence. The problem of matching points between calibration images is a practical problem in the context of self-calibration. Most papers assume already matched points which is not realistic in real applications. However, the images may be difficult to relate and point matching in different images is not an easy task. Mismatches cause high errors in the estimation process. Matching the points between two different images can cause mismatches or false localization of points. Therefore, a more robust point matching process must be used. Since the points in two consecutive frames of a video do not change much, point tracking in video frames is a more reliable option. Several iterative algorithms are proposed that utilized the consecutive

frames of a video [19, 20, 21, 22]. The sequential approaches require knowledge on good prior distributions or manual initialization. As long as true associations between the points in images are provided, one step calculation of the calibration matrix using Kruppa equations is a better candidate however equations may fail in real case scenario due to the critical motions [23].

To extract points from an image, a general framework, which can provide points in almost every image, must be used. In this work, we have utilized Speed-Up Robust Features (SURF) [24] and simple corner detectors. Thus, multiple points are guaranteed to be found in each scene ensuring the fulfillment of minimal operational requirements for calibration. SURF features can be matched between two different images directly. The direct matching can be wrong in complicated environments and/or in case of duplicate objects or regions. Thus, tracking feature points decrease the wrong matches [25, 26]. Therefore, we detect and track feature points in each frame using a modified Kalman filter and Hungarian algorithm for point association which can overcome the problem of missed detections and false alarms in the frames.

Today's vision systems are generally digital, and production quality is increased, focal lengths along the horizontal and the vertical axes can be assumed equal. Moreover, the principal point of a camera is also located in the center of the image. Thus, unknown camera parameters can be reduced to a single parameter: the focal length [16]. If all camera parameters except focal length is known then camera is said to be semi-calibrated. Additionally, the camera may be a part of a mobile robotic system. Information of position and orientation of the camera give extra constraints on the calibration procedure. Since the relative motion between first and final position, and orientation of camera is known, then focal length estimation can lead to simple geometric problem for semi-calibrated camera case. The novelty of this thesis is tracking the feature points and performing the self-calibration of semi-calibrated camera on a mobile robot only using relative position and orientation of camera and without supervision of a user.

The thesis is organized as follows. In Chapter 2, base information about image formation, multiple view geometry, and both manual and self calibration method is given. In Chapter 3, an overview of the problem and the proposed algorithm is given. The

results of the proposed algorithm and comparison with other self-calibration method are presented in Chapter 4. Lastly, Concluding remarks are made in Chapter 5.

CHAPTER 2

MULTIPLE VIEW GEOMETRY AND SELF-CALIBRATION

2.1 Projective Geometry

Before moving on to the self-calibration methods and tracking methods, it is essential to introduce the projective geometry of a camera, epipolar geometry, and their relation to self-calibration.

Camera calibration algorithms are mostly based on the pinhole camera model. Pinhole camera model helps to project a point in the world coordinates to the pixel coordinates of the pinhole image plane. Simple form of pinhole camera model is given in Fig.2.1.

A point **X** is located at (X, Y, Z) position with respect to camera origin and the image plane π is located at a distance f in positive $z - direction$. z-direction is also called as *optical axis*. The intersection point of the image plane π and $z - axis$ is called principal point of π which is the origin of the image plane π. A ray that connects point **X** with camera origin is passed through the point **x** on π. The point **x** has a location (x, y, f) and this point is said to be projection of **X** on π. Then the position x and y can be expresses as;

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z} \tag{2.1}$$

In Eq.(2.1), world coordinate frame is coincide with camera coordinate frame. More general form of projecting on image plane is given in Fig.2.2. To project a point on to π, perspective projection is needed. This projection defines how a point that defined in world coordinates to is projected in pixel coordinates on the π. The perspective

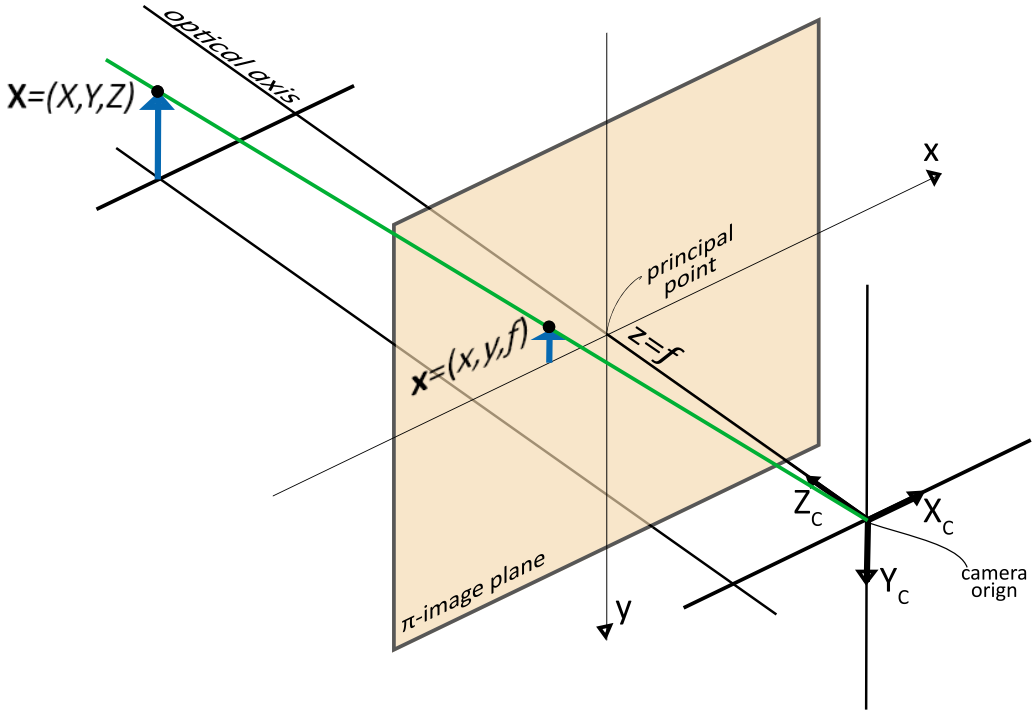


Figure 2.1: Pin Hole Model.

projection is defined by a matrix that is composed of focal length f in x and y direction, skew angle between x and y axis of π, and principle point of π. The matrix is 3×3 upper triangular matrix and called as **K** [27, 15].

$$\mathbf{K} = \begin{bmatrix} f_x & s & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

where f_x and f_y are the focal length in x and y axes, respectively, s is skew angle between x and y axis of π and o_x and o_y are principle points in x and y axes respectively. The **K** is also called as calibration matrix as well as intrinsic parameters. Each camera has specific parameters and they are unique.

The point **X** in Fig.2.2 that is defined in world coordinates $\mathbf{X_C}$ can be represented with respect to camera coordinate as $\mathbf{X_C} = \mathbf{R}(\mathbf{X_W} - \mathbf{T})$ where **R** and **T** are rotation and translation of camera coordinate system relative to world coordinate system. **R** and **T** are defined as extrinsic parameters of camera [28, 27, 15]. The rotation matrix **R**

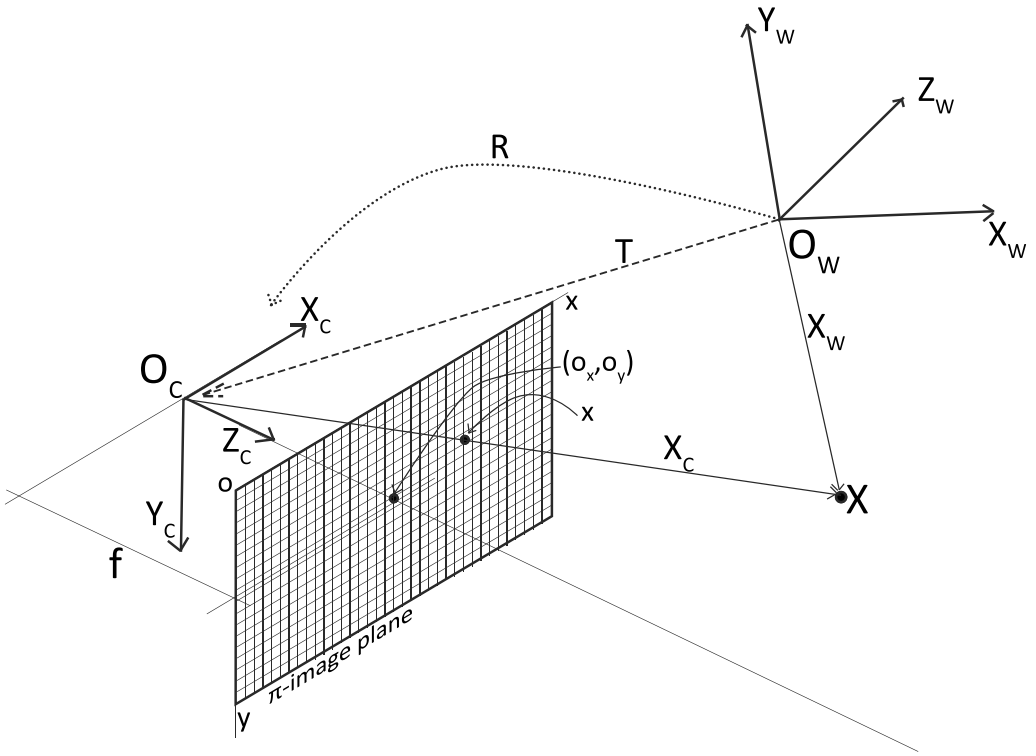


Figure 2.2: Projective geometry used in pinhole camera model. $\mathbf{O}_c$ is the optical center or origin of the coordinate system of camera, $\mathbf{O}_w$ is the origin of world coordinates. Points in real world are projected on π-image plane.

is defined by multiplication of rotation around X, Y, and Z axis with respect to world frame. The representation is given as

$$\mathbf{R} = \mathbf{R_x}\mathbf{R_y}\mathbf{R_z} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}_{3\times3} = \begin{bmatrix} \mathbf{R_1} \\ \mathbf{R_2} \\ \mathbf{R_3} \end{bmatrix}_{3\times3} \tag{2.3}$$

$$\mathbf{R_x}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \tag{2.4}$$

$$\mathbf{R_y}(\beta) = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \tag{2.5}$$

$$\mathbf{R_z}(\gamma) = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

where α, β and γ represents rotation around X, Y, and Z axis respectively. The translation vector $\mathbf{T}$ defines the translation of camera frame relative to world frame. Then it can be shown as

$$\mathbf{T} = \mathbf{O_C} - \mathbf{O_W} = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}_{3\times1} \tag{2.7}$$

Combination of extrinsic parameters can be represented as

$$[\mathbf{R} \mid \mathbf{t}] = \begin{bmatrix} \mathbf{R_1} & -\mathbf{R_1T} \\ \mathbf{R_2} & -\mathbf{R_2T} \\ \mathbf{R_3} & -\mathbf{R_3T} \end{bmatrix}_{3\times4} \tag{2.8}$$

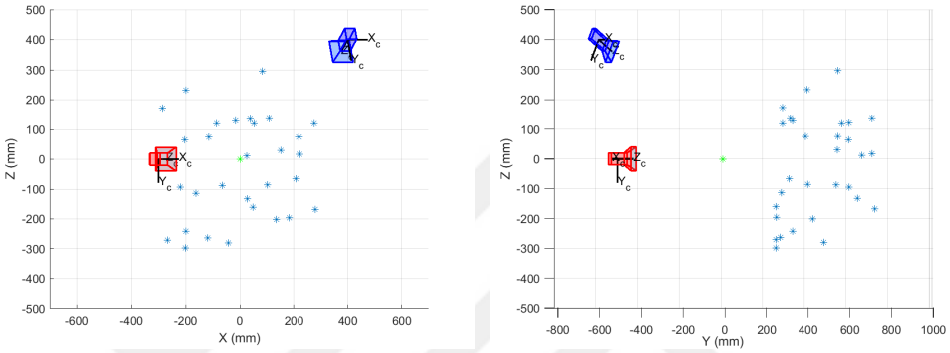
where $\mathbf{R_1}$, $\mathbf{R_2}$ and $\mathbf{R_3}$ represents the row vectors of $\mathbf{R}$. Thus perspective projection of $\mathbf{X}$ on to the π in Fig.2.2 can be written as

$$\hat{s}\mathbf{x} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{X} = \mathbf{PX} \tag{2.9}$$

where $\hat{s}$ is a unknown scale factor. The matrix $\mathbf{P}$ is a called as camera matrix and its size is 3×4. Therefore $\mathbf{X}$ has to be written in homogeneous coordinates as 4×1 vector and $\mathbf{x}$ has also be written in homogeneous coordinates as 3×1 vector [28, 15]. So rewriting Eq.2.9

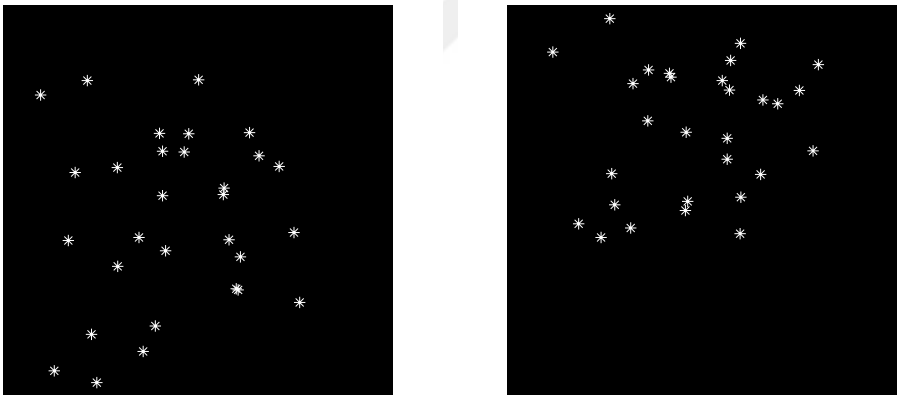
$$\hat{s}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.10}$$

Camera Calibration is a procedure that is performed for estimating 11 parameters of **P** that are f_x, f_y, s, o_x, o_y, α, β, γ, T_X, T_Y and T_Z. An example of perspective projection of randomly distributed points from two differently located camera image is given in Fig.2.3. In Fig.2.3, Camera 1's and Camera 2's position and orientation are shown as red and blue in 3D space in Fig. 2.3 (a) and Fig. 2.3 (b), respectively. In Fig. 2.3 (c), the projection of 3D points on to the image plane of Camera 1 is given as image. In Fig. 2.3 (d), the projection of 3D points on to the image plane of Camera 2 is given as image.



(a) View From XZ Plane

(b) View From YZ



(c) Image of Camera 1

(d) Image of Camera 2

Figure 2.3: Perspective Projection of Point on Cameras.

2.2 Epipolar Geometry

Camera calibration is performed on multiple images that are taken from different views. Therefore, understanding multiple view geometry is essential in calibration

procedure. Multiple view geometry is based on two view geometry. Schematic representation of two view geometry is given in Fig.2.4. A point $\mathbf{X}$ in the 3D space is projected as $\mathbf{x}$ and $\mathbf{x}'$ on π and π' respectively. As a result, $\mathbf{x}$ and $\mathbf{x}'$ are found to be corresponding points. The point O is optical center of π and similarly the point O' is the optical center of π'. The line that that connects two optical centers OO' is called as the *base line*. The intersection of π and π' with baseline is two image points $\mathbf{e}$ and $\mathbf{e}'$ are called as *epipole*. The line l_1 in π that pass from $\mathbf{e}$ and $\mathbf{x}$ is called as *epipolar line* corresponding to $\mathbf{x}'$. Similarly, The line l'_1 in π' that pass from $\mathbf{e}'$ and $\mathbf{x}'$ is called as *epipolar line* corresponding to $\mathbf{x}$. Then, a plane that passes from points O, P, and O' is called as *epipolar plane*. All of the geometry is called as *epipolar geometry* [15].

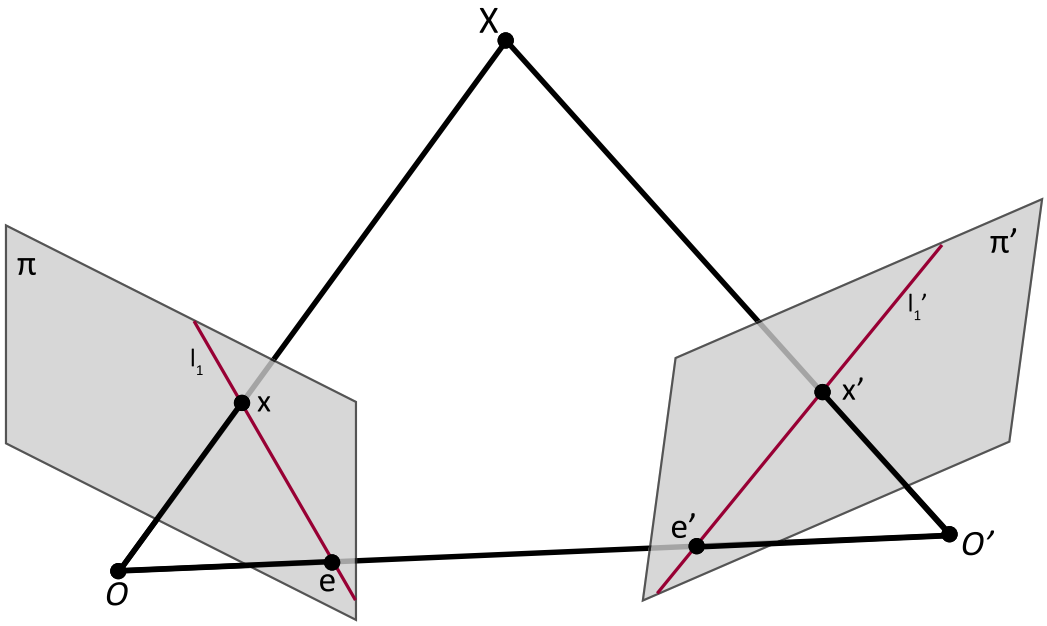


Figure 2.4: Epipolar geometry.

If $\mathbf{x}$ and $\mathbf{x}'$ are corresponding points, then $\mathbf{x}'$ must lie on l'_1 and conversely $\mathbf{x}$ must lie on l_1. This constraint is called as the *epipolar constraint*. Epipolar constraint is useful for stereo matching algorithm, because this constraint reduce the search area [16]. The algebraic representation of epipolar constraint can be shown as,

$$\mathbf{x}'^{T}\mathbf{F}\mathbf{x} = 0 \tag{2.11}$$

In Eq.(2.11), $\mathbf{F}$ is named as *fundamental matrix*. Some properties of *fundamental matrix* are;

- $\mathbf{F}$ is a rank 2 matrix with 7 degrees of freedom
- $l'_1 = \mathbf{Fx}$ is the epipolar line corresponding to $\mathbf{x}$
- $l_1 = \mathbf{F^T x'}$ is the epipolar line corresponding to $\mathbf{x'}$
- $\mathbf{Fe} = 0$
- $\mathbf{F}^T\mathbf{e'} = 0$
- $\mathbf{F} = \mathbf{K'}^{-T}[\mathbf{t}]_x\mathbf{R}\mathbf{K}^{-1}$ where $[\mathbf{t}]_x = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$
- $det(\mathbf{F}) = 0$
- $\mathbf{F} \simeq [\mathbf{e'}]_x\mathbf{H}_\infty$ where $\mathbf{H}_\infty$ is the infinity homography at plane at infinity

Due to the constraints on the $\mathbf{F}$, matching a minimum of 7 points is enough for calculations. As seen in properties, $\mathbf{F}$ contains the intrinsic parameters of cameras and extrinsic parameters relative to each π to π'. Let a point $\mathbf{x_i} = [u_i, v_i]^T$ in the first image be matched to a point $\mathbf{x'_i} = [u'_i, v'_i]^T$ then they must satisfy epipolar constraint $\mathbf{x'_i}^T\mathbf{F}\mathbf{x_i} = 0$. This constraint equation rewritten as a linear and homogeneous system in terms of 9 unknown coefficient matrix $\mathbf{F}$ [29]:

$$\mathbf{u_i}^T\mathbf{f} = 0 \tag{2.12}$$

where

$\mathbf{u_i} = [u_iu'_i, v_iu'_i, u'_i, u_iv'_i, v_iv'_i, u_i, v_i, 1]^T$

$\mathbf{f} = [F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}]$

Using Eq.(2.12), many solution method exist in literature to estimate $\mathbf{F}$. Direct Methods, Linear Least-Squares Techniques, Eigen Analysis, Nonlinear Minimization, and

Robust Methods are the implemented techniques for estimating fundamental matrix. More general, direct, and simplest method is *the normalized 8 point algorithm* [30, 16]. This methods requires a minimum of 8 matched points. The point sets x_i's and x_i''s are normalized according to normalizing transformation such as $\bar{x}_i = T x_i$ and $\bar{x}_i' = T' x_i'$. Then, linear homogeneous solution is performed to find $\bar{F}$ using Eq.(2.12) for normalized point set $\bar{x}_i$ and $\bar{x}_i'$. Then, rank 2 constraint is applied on $\bar{F}$. Finally, $F = T'^T \bar{F} T$ is the fundamental matrix that correspond the original data set x_i and x_i'.

On specific motion of image plane, **F** is shaped special and specific form. Specific camera motions are performed as, *x-axis pure translation,y-axis pure translation, z-axis pure translation, pure retinal translation, pure translation, retinal displacement*. Pure retinal translation is translating the image plane parallel to first image plane and retinal displacement is translating the image plane parallel to first image plane and rotating only around *z-axis*. Pure translation is moving the image plane in both *x*- and *y*-axis. The performed motions are relative to the first camera frame [31]. The formation of **F** under the specific motion is given in Table.2.1.

Epipoles can be directly evaluated from **F** [17, 16]. The unit norm eigenvector of the matrix $\mathbf{F}^T\mathbf{F}$ with the smallest eigenvalue is the epipole **e**. Conversely, the unit norm eigenvector of the matrix $\mathbf{F}\mathbf{F}^T$ with the smallest eigenvalue is the epipole $\mathbf{e}'$ [17]. Therefore, epipoles can be extracted from the singular value decomposition (SVD) of matrix **F** [16]. The **F** has rank 2, so SVD expansion is;

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{U}\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix}\mathbf{V}^T \tag{2.13}$$

where $\mathbf{F}^T\mathbf{u_3} = 0$ and $\mathbf{F}\mathbf{v_3} = 0$ are the null-vectors. Thus, the first epipole is $\mathbf{e} = \mathbf{v_3}$ and second epipole $\mathbf{e}' = \mathbf{u_3}$ where $\mathbf{u_3}$ and $\mathbf{v_3}$ are the third column vector of **U** and **V** respectively.

As described in Sec.2.1, **P** is the combination of intrinsic and extrinsic parameters. Without any knowledge about those parameters, it is difficult to directly estimate **P**. However, **P** is not unique thus it is possible to estimate the camera matrices of two cameras that build an epipolar geometry. Once the fundamental matrix is estimated

Table 2.1: Fundamental matrices arising from special motions

x-axis pure translation	$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$
y-axis pure translation	$\mathbf{F} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
z-axis pure translation	$\mathbf{F} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
pure retinal translation	$\mathbf{F} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ -a & -b & 0 \end{bmatrix}$
pure translation	$\mathbf{F} = \begin{bmatrix} 0 & c & a \\ -c & 0 & b \\ -a & -b & 0 \end{bmatrix}$
retinal displacement	$\mathbf{F} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{bmatrix}$

and using the SVD of $\mathbf{F}$, camera matrices could be written. First camera matrix may be assumed to be in the form of $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$. Then, second camera matrix can be estimated as [32];

Define matrices $\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$; $\mathbf{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $\bar{\mathbf{D}} = diag(\sigma_1, \sigma_2, \sigma_3)$ where σ_3 is arbitrary value. Then $\mathbf{F}$ can be rewritten as,

$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T = (\mathbf{U}\mathbf{Z}\mathbf{U}^T)(\mathbf{U}\mathbf{W}^T\bar{\mathbf{D}}\mathbf{V}^T) = \mathbf{S}\mathbf{M} \tag{2.14}$$

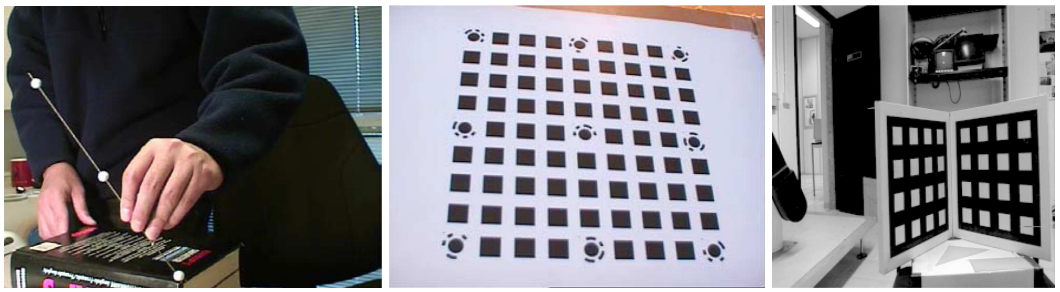
First and Second camera matrix may be defined respectively from $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ and Eq.(2.14) as;

$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}] \quad ; \quad \mathbf{P}' = [\mathbf{M} \mid \mathbf{u_3}] \tag{2.15}$$

2.3 Manual Calibration

A calibrated camera can give more information than an uncalibrated one. If camera will be used for surveillance, object tracking, object detection, object inspection, 3D reconstruction, navigation, depth estimation, augmented reality and recognition applications then calibration of camera is compulsory [15]. A calibrated camera can give metric information about the environment.

Calibration procedures are divided into two main categories; manual calibration and self-calibration. Manual calibration requires an object whose shape, dimensions are known. This object can be linear, planar or more complex, thus the names of calibration methods are known as 1D, 2D and 3D. Calibration objects are specifically designed and examples of calibration objects are given in Fig.2.5. On the other hand, self-calibration does not require any specific object or any measurement in the environment. Its processes is based on point extraction, matching and calculations using epipolar geometry [1].



(a) 1D Calibration Object (b) 2D Calibration Object (c) 3D Calibration Object

Figure 2.5: Three type of example calibration objects [1]

The process of camera calibration is the extraction or estimation of intrinsic and extrinsic camera parameters. There are 11 calibration parameters: f_x, f_y, s, o_x, o_y, α, β, γ, T_X, T_Y and T_Z. Given 2D-3D correspondence of $\mathbf{x_i} = (x_i, y_i)^T \leftrightarrow \mathbf{X_i} = (X_i, Y_i, Z_i)$ [33] and from Eq.(2.10),

$$x_i = \frac{X_i P_{11} + Y_i P_{12} + Z_i P_{13} + P_{14}}{X_i P_{31} + Y_i P_{32} + Z_i P_{33} + P_{14}} \tag{2.16}$$

$$y_i = \frac{X_i P_{21} + Y_i P_{22} + Z_i P_{23} + P_{24}}{X_i P_{31} + Y_i P_{32} + Z_i P_{33} + P_{14}} \tag{2.17}$$

Then,

$$X_i P_{11} + Y_i P_{12} + Z_i P_{13} + P_{14} - x_i X_i P_{31} - x_i Y_i P_{32} - x_i Z_i P_{33} - x_i P_{14} = 0 \tag{2.18}$$

$$X_i P_{21} + Y_i P_{22} + Z_i P_{23} + P_{24} - y_i X_i P_{31} - y_i Y_i P_{32} - y_i Z_i P_{33} - y_i P_{14} = 0 \tag{2.19}$$

Assume $\mathbf{x_i}$ and $\mathbf{X_i}$ are to be known. Then rearranging the homogeneous equation in terms of the elements of $\mathbf{P}$,

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & x_i X_i & x_i Y_i & x_i Z_i & x_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & y_i X_i & y_i Y_i & y_i Z_i & y_i \end{bmatrix} \mathbf{p} = \mathbf{0} \tag{2.20}$$

where $\mathbf{p} = [P_{11}, P_{12}, ..., P_{34}]^T$ and $\mathbf{0} = [0, 0]^T$. Call coefficient matrix of Eq.(2.20) as $\mathbf{G}_i$. For n point matches, concatenating all $\mathbf{G}_i$'s and get $\mathbf{Gp} = \mathbf{0}$. The matrix $\mathbf{G}$ has a size of $2n \times 12$. Thus, minimum 6 point matches are enough to determine $\mathbf{P}$.

After $\mathbf{P}$ is estimated than Intrinsic and Extrinsic Parameters can be recovered. Taking the first 3×3 sub matrix of $\mathbf{P}$ as $\mathbf{B}$ and 4th column as $\mathbf{b}$, and combining those two parts as $\mathbf{P} = [\mathbf{B} \;\; \mathbf{b}]$. Then,

$$\mathbf{B} = \mathbf{KR} \tag{2.21}$$

$$\mathbf{b} = \mathbf{Kt} \tag{2.22}$$

$$\mathbf{BB}^T = \mathbf{KK}^T \tag{2.23}$$

So,

$$\mathbf{R} = \mathbf{K}^{-1}\mathbf{B} \tag{2.24}$$

$$\mathbf{t} \quad = \mathbf{K}^{-1}\mathbf{b} \tag{2.25}$$

2.4 Self-Calibration

Self-calibration techniques do not require or use any defined calibration objects. No additional information is given to the system. The algorithms only rely on the point correspondences in a set of images. Calibration is performed by moving a camera in static scene and rigidity of scene constructs information about the internal camera parameters. To calculate all intrinsic parameters, a minimum of three images are enough. However, it is important that the right points are matched. This is satisfied by tracking the points through out the video sequence.

Self-calibration methods can be grouped into three categories [34]. The first one is camera intrinsic constraints which can be constant or varying intrinsic. The second one is camera intrinsic and motion constraint with constant or varying intrinsic. Lastly, Camera intrinsic and scene constraint with constant or varying intrinsic. Three of these categories rely on main constraint which is the camera has fixed or varying intrinsic. It means that, images can be taken by one camera with fixed intrinsic that is for constant intrinsic or using one camera with varying intrinsic (i.e. zooming) or using different cameras that is for varying intrinsic. The first category does not make any constraint on motion or scene. Camera can make any specific motion, i.e. general motion. In the second category, camera motion is restricted for defined combination of motions. For example, pure translation then pure rotation or retinal displacement than rotation in y-axis of camera, etc. In the last category, calibration is performed with some specific scene element without regarding the dimensions. For example, scene constraint can be horizontal or vertical lines that creates pattern, or any specific geometric shapes.

First known implementation of self-calibration is done by Faugeras in 1992 [17]. Later, many methods have been developed and implemented [18, 35, 19, 36]. Without the presence of a calibration object, an imaginary shape must be defined for calibration. This imaginary object is the *absolute conic* $\mathbf{\Omega}$ that lies on the plane at infinity π_∞ (see Fig. 2.6). Point $\mathbf{X} = [x_1, x_2, x_3, x_4]^T$ is defined in homogeneous coordinates and

is on $\mathbf{\Omega}$ at π_∞ where $x_4 = 0$ if it satisfies [17];

$$x_1^2 + x_2^2 + x_3^2 = 0 \tag{2.26}$$

Projection of Ω on π and π' are ω and ω'. ω and ω' are called as *image of absolute conic* (IAC). Under rigid motion and uniform changes in the scale, $\mathbf{\Omega}$ is invariant and have constant position relative to moving camera. Thus ω is constant [34]. Most of the self-calibration algorithms use epipolar transformation between ω and ω'. *Dual of the absolute conic* $\mathbf{\Omega}_\infty^*$ is the set of tangent lines to the $\mathbf{\Omega}$. It can be found as $\mathbf{\Omega}_\infty^* = det(\mathbf{\Omega})\mathbf{\Omega}^{-1}$ [37]. Thus, the *dual of image of absolute conic* (DIAC) ω^* is also invariant under rigid motion. DIAC encodes $\mathbf{K}$ such as $\omega^* \simeq \mathbf{K}\mathbf{K}^T$ [18]. Formation of DIAC is given in Eq.(2.27)

$$\omega^* = \mathbf{K}\mathbf{K}^T = \begin{bmatrix} f_x^2 + s^2 + o_x^2 & sf_y + o_x o_y & o_x \\ sf_y + o_x o_y & f_y^2 + o_y^2 & o_y \\ o_x & o_y & 1 \end{bmatrix} \tag{2.27}$$

Kruppa equation is based on the epipolar constraint of tangent epipolar lines l_1 and l_2 on ω^*, and l_1' and l_2' on $\omega^{*\prime}$ in π and π' respectively [17]. The constraint of $l_1 = (\mathbf{e}, \mathbf{x})$ is tangent to ω^* if it satisfies

$$(\mathbf{e} \times \mathbf{x})^T \omega^* (\mathbf{e} \times \mathbf{x}) = 0 \tag{2.28}$$

Eq.(2.28) is called as Kruppa Equation. This equation cannot be solved directly and has high computational burden. The simpler form of the Kruppa Equation is given in [31]. This approach has low computational load and is easily applicable. Using Eq.(2.28) for $l_1' = (\mathbf{e}', \mathbf{x}')$, we can write;

$$(\mathbf{e}' \times \mathbf{x}')^T \omega^{*'} (\mathbf{e}' \times \mathbf{x}') = \mathbf{x}'^T [\mathbf{e}']_x \omega^{*'} [\mathbf{e}']_x \mathbf{x}' = 0 \tag{2.29}$$

Similarly, l_1 can be written in terms of fundamental matrix property as $\mathbf{F}^T\mathbf{x}'$ and if it is tangential to ω than;

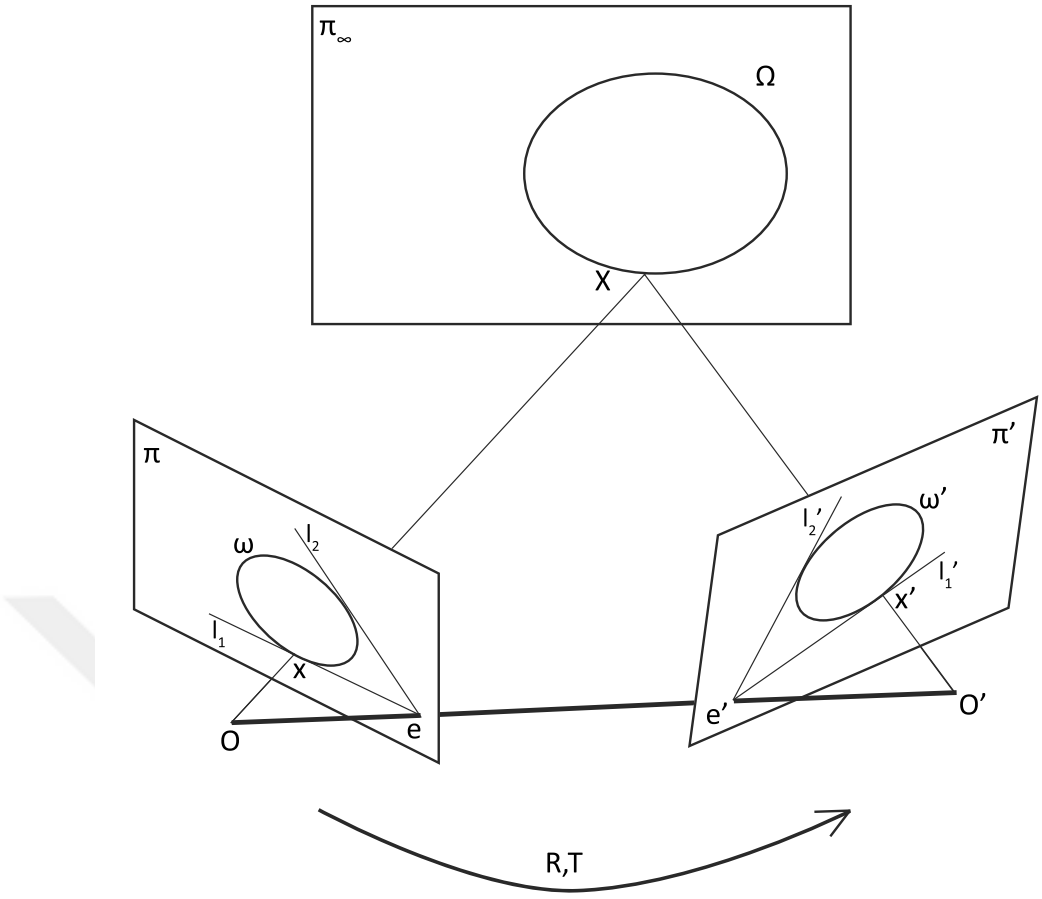


Figure 2.6: Epipolar geometry.

$$\mathbf{x'}^T \mathbf{F} \omega^* \mathbf{F}^T \mathbf{x'} = 0 \tag{2.30}$$

Then, writing Eq.(2.29) and Eq.(2.30) as equal, it can be concluded as Kruppa Equation. So,

$$[\mathbf{e'}]_x \omega^{*'} [\mathbf{e'}]_x = \mathbf{F} \omega^* \mathbf{F}^T \tag{2.31}$$

The Eq.(2.31) can be rewritten by using SVD components of fundamental matrix;

$$[\mathbf{u_3}]_x \omega^{*'} [\mathbf{u_3}]_x = \mathbf{U D V^T} \omega^* \mathbf{V D U^T} \tag{2.32}$$

$$\mathbf{U^T} [\mathbf{u_3}]_x \omega^{*'} [\mathbf{u_3}]_x \mathbf{U} = \mathbf{D V^T} \omega^* \mathbf{V D} \tag{2.33}$$

$$\begin{bmatrix}\mathbf{u_2} & \mathbf{u_1} & \mathbf{0}\end{bmatrix}^T \omega^{*'} \begin{bmatrix}\mathbf{u_2} & \mathbf{u_1} & \mathbf{0}\end{bmatrix} = \begin{bmatrix}\sigma_1 & & \\ & \sigma_2 & \\ & & 0\end{bmatrix} \mathbf{V}^T \omega^* \mathbf{V} \begin{bmatrix}\sigma_1 & & \\ & \sigma_2 & \\ & & 0\end{bmatrix} \tag{2.34}$$

$$\begin{bmatrix} \mathbf{u_2}^T\omega^{*'}\mathbf{u_2} & -\mathbf{u_2}^T\omega^{*'}\mathbf{u_1} & 0 \\ -\mathbf{u_1}^T\omega^{*'}\mathbf{u_2} & \mathbf{u_1}^T\omega^{*'}\mathbf{u_1} & 0 \\ 0 & & 0 \end{bmatrix} = \begin{bmatrix} \sigma_1^2\mathbf{v_1}^T\omega^*\mathbf{v_1} & \sigma_1\sigma_2\mathbf{v_1}^T\omega^*\mathbf{v_2} & 0 \\ \sigma_1\sigma_2\mathbf{v_1}^T\omega^*\mathbf{v_2} & \sigma_2^2\mathbf{v_2}^T\omega^*\mathbf{v_2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.35}$$

Eq.(2.35) could be rewritten in the form of homogeneous vector form.

$$\begin{bmatrix} \mathbf{u_2}^T\omega^{*'}\mathbf{u_2} \\ -\mathbf{u_1}^T\omega^{*'}\mathbf{u_2} \\ \mathbf{u_1}^T\omega^{*'}\mathbf{u_1} \end{bmatrix} = \begin{bmatrix} \sigma_1^2\mathbf{v_1}^T\omega^*\mathbf{v_1} \\ \sigma_1\sigma_2\mathbf{v_1}^T\omega^*\mathbf{v_2} \\ \sigma_2^2\mathbf{v_2}^T\omega^*\mathbf{v_2} \end{bmatrix} \tag{2.36}$$

The relation is equal up to a scale factor. To eliminate scaling problem vector cross product can be applied. So simplified version of Kruppa Equation becomes

$$\frac{\mathbf{u_2}^T\omega^{*'}\mathbf{u_2}}{\sigma_1^2\mathbf{v_1}^T\omega^*\mathbf{v_1}} = \frac{\mathbf{u_1}^T\omega^{*'}\mathbf{u_2}}{\sigma_1\sigma_2\mathbf{v_1}^T\omega^*\mathbf{v_2}} = \frac{\mathbf{u_1}^T\omega^{*'}\mathbf{u_1}}{\sigma_2^2\mathbf{v_2}^T\omega^*\mathbf{v_2}} \tag{2.37}$$

Eq.(2.37) has 2 independent and 1 dependent equations. Thus, the two constraints can be extracted from two views. If 5 intrinsic parameters are going to be estimated than 3 images are required to establish 5 different equations.

Three fundamental matrices can be estimated from pairwise image matching. If images are taken with the same camera and the camera is semi-calibrated which means principle point is known, skew is known and focal length in both axis is equal, then two image and one epipolar constraint is enough to solve one of the ratio equivalence of Eq.(2.37). The assumption is valid because principle point can be assumed being the mid point of digital image and manufacturing quality of sensor is well for zero skew value. Then DIAC becomes $\omega^* = diag(f^2, f^2, 1)$. First and second ratio equivalency or first and third ratio equivalency creates biquadric equation with one variable as focal length. So, it can be easily solved.

If the only unknown parameter is the focal length, then the solution of Kruppa equation is simple. But in some camera configurations, it results in a degenerate solution,

in other words an invalid focal length or multiple focal length results. The critical motion sequences for self-calibration methods were studied in detail [23, 38]. Generally two cases result in critical motions. These are when

- the optical axes are parallel to each other
- the optical axes are intersect at a finite point.

This is the case when camera makes pure translation, pure rotation, orbital motion and planar motion. In real images or video tests it is difficult to control if the camera makes critical motion or not. Thus, on Kruppa Equation tests the focal length estimation results are unstable, i.e. solution can reach to valid or invalid results. The results are directly affected by evaluation of fundamental matrix. If the matched points are distributed as a pure translation, the results usually become invalid. Therefore, matched points should not be concentrated in a region that generates camera's motion as a pure translation.

Another method is implementation of the *stratified solution* for self-calibration. The idea is to establish homography $\mathbf{H}_\infty$ on plane at infinity. The $\mathbf{H}_\infty$ is invariant under rigid motion as well as ω^*. If the plane at infinity is defined by $\pi_\infty = (\mathbf{p}^T, 1)^T$ and camera matrices are defined as $[\mathbf{M}^i \mid \mathbf{u_3}^i]$. Then, a linear relationship can be written between ω^* and $\mathbf{H}_\infty$ [39].

$$\mathbf{H}^i_\infty = \mathbf{M}^i - \mathbf{u_3}^i \mathbf{p}^T \tag{2.38}$$

$$\omega^{*'} = \mathbf{H}^i_\infty \omega^* {\mathbf{H}^i_\infty}^T \tag{2.39}$$

Evaluation of location of π_∞ is difficult. Therefore, $\mathbf{H}_\infty$ is also difficult with respect to the solution of Kruppa Equation. Stratified solution is more computationally heavy.

Another self-calibration model is constructed on the dual of absolute quadric, camera matrices $\mathbf{P}$ and ω^*. This method is called as *Absolute Dual Quadric* (ADQ) and first use of method is by [18]. A quadric Q is a surface in projective 3-space and it is defined by

$$\mathbf{X}^T\mathbf{Q}\mathbf{X} = \mathbf{0} \tag{2.40}$$

where $\mathbf{Q}$ is a 4×4 symmetric matrix and $\mathbf{X}$ is a point in homogeneous coordinates. A quadric is called degenerate if its matrix is singular. Dual of a quadric is a quadric. It represents the tangential planes to a point on $\mathbf{Q}$. The $\mathbf{\Omega}_\infty{}^*$ is a degenerate dual quadric that is absolute dual quadric $\mathbf{Q}_\infty{}^*$. $\mathbf{Q}_\infty{}^*$ is a 4×4 symmetric matrix with rank 3 and zero determinant [16]. It has a canonical form

$$\mathbf{Q}_\infty{}^* = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \tag{2.41}$$

The projection of $\mathbf{Q}_\infty^*$ using $\mathbf{P}$ is ω^* [18]. Algebraically;

$$\omega^* = \mathbf{P}\mathbf{Q}_\infty^*\mathbf{P}^T \tag{2.42}$$

If Eq.(2.42) is expanded,

$$\begin{bmatrix} f_x^2 + s^2 + o_x^2 & sf_y + o_x o_y & o_x \\ sf_y + o_x o_y & f_y^2 + o_y^2 & o_y \\ o_x & o_y & 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} Q^*_{\infty 11} & Q^*_{\infty 12} & Q^*_{\infty 13} & Q^*_{\infty 14} \\ Q^*_{\infty 12} & Q^*_{\infty 22} & Q^*_{\infty 23} & Q^*_{\infty 24} \\ Q^*_{\infty 13} & Q^*_{\infty 23} & Q^*_{\infty 33} & Q^*_{\infty 34} \\ Q^*_{\infty 14} & Q^*_{\infty 24} & Q^*_{\infty 34} & Q^*_{\infty 44} \end{bmatrix} \mathbf{P}^T \tag{2.43}$$

ω^* has 5 unknowns and $\mathbf{Q}_\infty^*$ has 10 unknowns. Due to the symmetries in ω^* only 6 equations can be extracted from equalities. Therefore, minimum three images are needed to generate 15 equations for 15 unknowns. Two types of solution can be performed. Linear and nonlinear solutions. For linear relation,

$$\omega^{*i}_{11} = \left(\mathbf{P}^i\mathbf{Q}_\infty^*\mathbf{P}^{iT}\right)_{11} \quad , \quad \omega^{*i}_{12} = \left(\mathbf{P}^i\mathbf{Q}_\infty^*\mathbf{P}^{iT}\right)_{12} \quad , \quad \omega^{*i}_{13} = \left(\mathbf{P}^i\mathbf{Q}_\infty^*\mathbf{P}^{iT}\right)_{13} \tag{2.44}$$

equations can be written where i represents the i^{th} camera matrix equation set. If some constraints can be given for ω^* than solution becomes easier and also decreases required number of equations. Linear equalities can be solved by least square methods

or iterative methods. However, linear solutions can lead to scaled values of parameters. Nonlinear method can overcome this consequence. Nonlinear equations can be written by taking the ratios of matrix components of ω^{*i} and $\mathbf{P}^i\mathbf{Q}_\infty^*\mathbf{P}^{iT}$, then cross multiplying them. This operation can be written symbolically as

$$\omega^* \wedge \left(\mathbf{P}^i\mathbf{Q}_\infty^*\mathbf{P}^{iT}\right) = 0 \tag{2.45}$$

Eq.(2.45) is equal to

$$\omega^{*i}{}_{AB}\left(\mathbf{P}^i\mathbf{Q}_\infty^*\mathbf{P}^{iT}\right)_{CD} - \omega^{*i}{}_{CD}\left(\mathbf{P}^i\mathbf{Q}_\infty^*\mathbf{P}^{iT}\right)_{AB} = 0 \tag{2.46}$$

where $A \leq B$, $C \leq D = 1...3$ are the indices of matrices ω^{*i} and $\mathbf{P}^i\mathbf{Q}_\infty^*\mathbf{P}^{iT}$ [40]. Nonlinear equation set can be solved using nonlinear least squares method, sequential quadratic programming, particle swarm optimization or other nonlinear optimization methods. When evaluating the Eq.(2.42), nonuniqueness of $\mathbf{P}^i$'s and unknown scale factor makes getting reliable results difficult. This problem is encountered in our tests as well as in other works [41].

CHAPTER 3

PROPOSED SELF-CALIBRATION METHOD

In practice, the points between images are difficult to associate, however most self-calibration approaches assume the points are already matched. Any mismatches degrade the performance of the self-calibration. There are different approaches to match the points. One of them is the direct approach which uses the properties of the extracted points [19, 20, 41, 42, 43, 44]. However, matching the points in different images is prone to errors. A more sensible approach is to track the locations of points through out the movement of the camera, thus tracking of points in image sequences.

Assuming, the camera is mounted on a mobile robot where the camera position and orientation can be acquired. Using epipolar constraints and geometric relations can provide a simple solution to focal length for a semi-calibrated camera where the only unknown is the focal length. In Fig.3.1, general framework of the proposed self-calibration method is given. Frames of the camera are acquired as image sequences with fix frame rate through the self-calibration procedure. When an image is acquired, the position and orientation of the robot as well as the camera are stored and feature points are extracted. The extracted feature point locations are used for tracking and assignment procedure.

After motion of the robot is completed for self-calibration procedure, the visible points are extracted from the stored tracked points that are visible through the all image sequences. The visible points are used as correspond point for estimating epipolar geometry. Fundamental matrix and epipoles are estimated. Next, using the geometry of motion of the camera and epipolar geometry, the focal length is estimated.

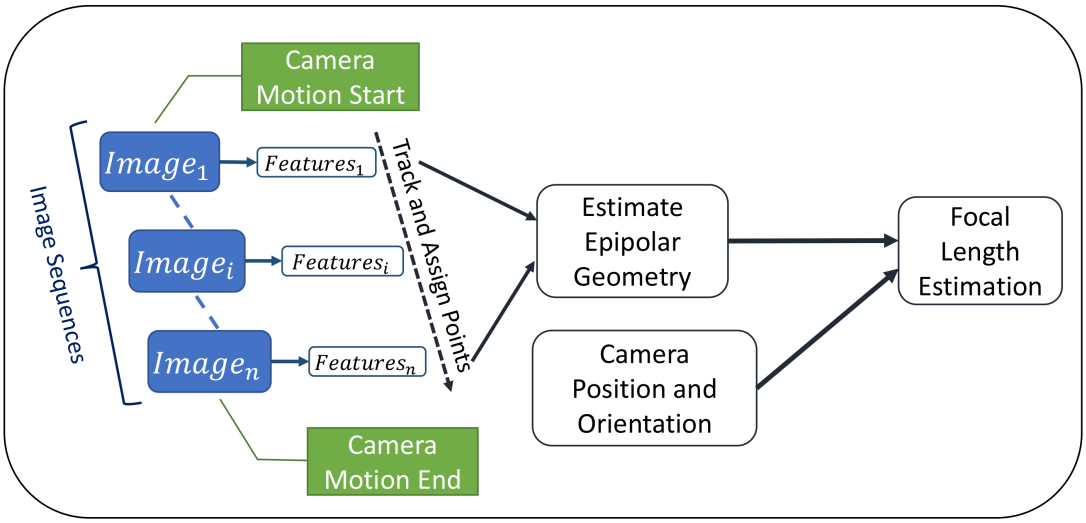


Figure 3.1: Framework of the proposed method.

3.1 Point Tracking and Association

Point tracking and association could increase the accuracy of the point matches in the complex or crowded environments. Therefore, feature points have to be extracted from each acquired image frame. First step of feature point extraction is detecting the interest regions in the image such as corner points[46]. Various detector methods in the literature include

- Moravec's Detector
- Harris Detector
- SUSAN Detector
- FAST Detector
- Hessian Detector
- Laplacian of Gaussian
- Difference of Gaussian
- Harris-Laplace
- Hessian-Laplace

- Gabor-Wavelet detector.

Then, around the detected point location, invariant feature vector is established. The feature vector, that may include statistics, parametric model, scale, orientation or combination of them, is the descriptor of the feature point [46]. The methods for feature descriptor include

- Scale Invariant Feature Transform (SIFT)
- Gradient Location-Orientation Histogram (GLOH)
- Speeded-Up Robust Features Descriptor (SURF)
- Local Binary Pattern (LBP)
- Binary Robust Independent Elementary Features (BRIEF).

Different feature extractor algorithms are used in our framework. The chosen method depends on the scene type. Both SURF and Harris Detector are implemented in different tests and simulations. The x and y coordinates of feature point are used for point tracking.

Various methods exist on tracking applications in image sequences but Kalman filter is a robust and established alternative [15, 45]. However, in image sequences false alarms and missed detections may occur. Moreover, the points from an image frame must be associated with the points in the next frame. Association is required to construct the epipolar geometry. The standard Kalman filter does not consider the missed detections and multiple object associations. These lacking features of Kalman filter must be incorporated to the standard Kalman filter. Assignment problem is overcome by generating a cost matrix based on log-likelihood function and using the Hungarian method for the minimization of association costs.

However, to distinguish between the missed detections and false alarms, the state vector is augmented with Confidence Level (CL). CL is incremented by 1 with each detection associated with the point in the previous frame and reduced by 1 if no associated detections occur. CL values are upper limited to 5 and if they decrease to 0,

the associated track is deleted. In Fig.3.2, the block diagram for point tracking and association is given for the k^{th} step.

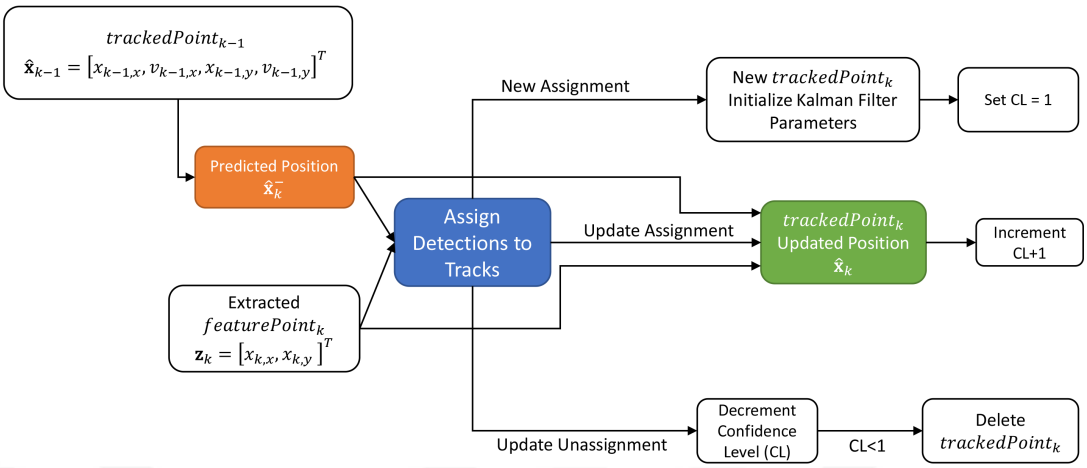


Figure 3.2: Block diagram of point tracking and association.

Kalman filter has two phases that are *Prediction Phase* and *Update Phase*. In Fig.3.2, Orange and Green boxes are Prediction and Update Phases of Kalman filter, respectively. Blue box is the *Association Phase* that assigns a point to a track or not based on the Euclidean distance measurement cost between points and tracks using Hungarian method.

The state vector at the k^{th} step for Kalman filter defined by $\mathbf{x}_k = [x_{k,x}, v_{k,x}, x_{k,y}, v_{k,y}]^T$ where $x_{k,x}$ and $x_{k,y}$ indicate the locations of a feature point in x and y direction, and $v_{k,x}$ and $v_{k,y}$ indicate the velocities of a feature point in x and y direction. The state transition matrix is created assuming the update period as 1, thus

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

and measurements are the pixels along x and y positions. The measurement model matrix is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.2}$$

If the k^{th} step of the process is considered, a tracked point in the $(k-1)^{st}$ step has a state vector $\hat{\mathbf{x}}_{k-1} = [x_{k-1,x}, v_{k-1,x}, x_{k-1,y}, v_{k-1,y}]^T$. The position of $\hat{\mathbf{x}}_{k-1}$ at k^{th} step is estimated as $\hat{\mathbf{x}}_k^-$ in the *Prediction Phase*. The feature point locations that are extracted in the k^{th} image frame are $\mathbf{z}_k$ measurement value. Then, Euclidean distance between set of $\mathbf{z}_k = [x_{k,x}, x_{k,y}]^T$'s and set of $\hat{\mathbf{x}}_k^-$'s are calculated. Using these distance results, a cost matrix is built. Then, Hungarian method is utilized to determine the detection and track associations.

There are three cases for *Association Phase*. The first case is when $\mathbf{z}_k$ is not assigned to any of the existing tracks. Then, $\mathbf{z}_k$ is generated as a *new track*. Then, a Kalman filter is initialized using the location of $\mathbf{z}_k$ and CL is set to 1. The second case is when a detection, $\mathbf{z}_k$, is assigned to a track, $\hat{\mathbf{x}}_k^-$. Then, the position of the assigned track is updated in the *Update Phase* and $\hat{\mathbf{x}}_k$ is obtained. In addition, CL is incremented with 1. The final case is when a track, $\hat{\mathbf{x}}_{k-1}$, is not assigned to any detection, $\mathbf{z}_k$, than the track's CL is decreased by 1. When the track $\hat{\mathbf{x}}_k$'s CL reaches to 0 then the track is deleted.

3.2 Focal Length Estimation

When the motion of the system ends, the orientation, initial and final positions of the camera are stored. Next step is to extract the visible points that are tracked throughout the motion. It is assumed that the camera is semi-calibrated where the only unknown parameter is the focal length. Since the resolution of the acquired image is known, than origin of the image, which is top left corner, is shifted to the center of image. Thus, location of extracted points are shifted by a number of pixels equal to the half of resolution. Then, the epipolar geometry is estimated from the extracted points. The initial and final positions and orientations of camera are shown in Fig.3.3. The black stars in this figure represents the visible tracked points throughout the motion.

The origin of the global world frame is at O_G. The first image is taken when the camera

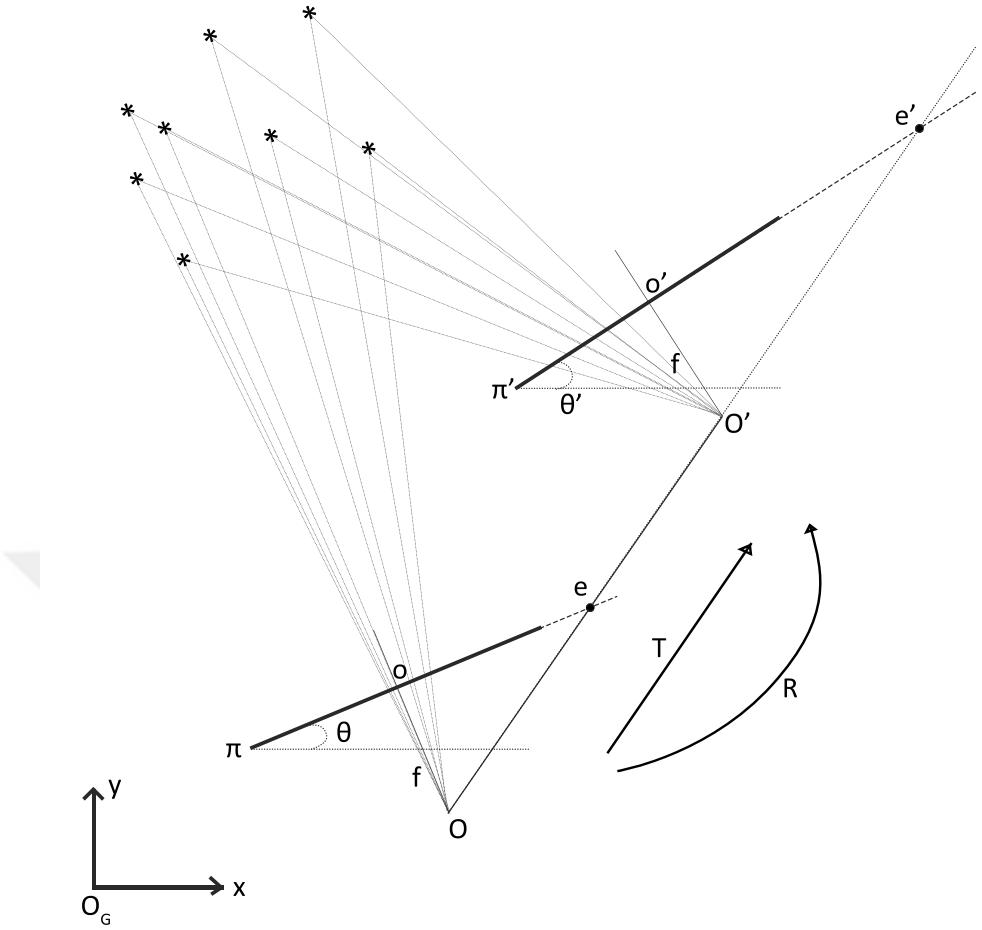


Figure 3.3: Two view geometry of Camera located on mobile robot.

is located initially at **O**. The second image is taken when the camera is located finally at **O**′. From **O** to **O**′ camera makes translation as **T** and rotation as **R**.

π is the image plane of the first image. The optical center of the camera for the first image is at **O** and its principle point is at **o**. The origin of the camera frame for the first image is at **o**. x and y axes of this frame coincides with π to its right and optical axis of π, respectively. Its heading angle is θ. The vector from **o** to **O** is

$$|\overrightarrow{oO}| = f\sin(\theta)\hat{\mathbf{x}} - f\cos(\theta)\hat{\mathbf{y}} \tag{3.3}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are the unit vectors of the global frame in the x and y directions, respectively. $|\overrightarrow{oO}|$ has magnitude of focal length f.

π' is the image plane of the second image. $\mathbf{O}'$ is the optical center, and $\mathbf{o}'$ is the principle point of the second image. The origin of the camera frame for the second image is at $\mathbf{o}'$ whose positive x-axis coincides with π' to its right and positive y-axis coincides with optical axis of π'. The vector from $\mathbf{o}'$ to $\mathbf{O}'$ is

$$|\overrightarrow{o'O'}| = f\sin(\theta')\hat{\mathbf{x}} - f\cos(\theta')\hat{\mathbf{y}} \tag{3.4}$$

has magnitude of focal length f and is in pixel unit.

Points in the 3D space are projected on π and π' which are found in *image1* and *image2*. Point matches can be established and epipolar geometry can be constructed. Connecting the optical center $\mathbf{O}$ to $\mathbf{O}'$ is $|\overrightarrow{OO'}|$ gives a baseline.

$$|\overrightarrow{OO'}| = T_x\hat{\mathbf{x}} + T_y\hat{\mathbf{y}} \tag{3.5}$$

where T_x and T_y are the x and y components of $\mathbf{T}$, respectively. $|\overrightarrow{OO'}|$ is defined in metric units. Intersections of $|\overrightarrow{OO'}|$ with π and π' give epipoles e and e' of *image1* and *image2*, respectively. $|\overrightarrow{oe}|$ and $|\overrightarrow{o'e'}|$ in pixel units are

$$|\overrightarrow{oe}| = e\cos(\theta)\hat{\mathbf{x}} + e\sin(\theta)\hat{\mathbf{y}} \;\;;\;\; |\overrightarrow{o'e'}| = e'\cos(\theta')\hat{\mathbf{x}} + e'\sin(\theta')\hat{\mathbf{y}} \tag{3.6}$$

where magnitude of $|\overrightarrow{oe}|$ and $|\overrightarrow{o'e'}|$ is e and e'. The vectors from $\mathbf{O}$ to e as $|\overrightarrow{Oe}|$ and from $\mathbf{O}'$ to e' as $|\overrightarrow{O'e'}|$ in pixel units. So,

$$|\overrightarrow{Oe}| = |\overrightarrow{oe}| - |\overrightarrow{oO}| = [e\cos(\theta) - f\sin(\theta)]\hat{\mathbf{x}} + [e\sin(\theta) + f\cos(\theta)]\hat{\mathbf{y}} \tag{3.7}$$

$$|\overrightarrow{O'e'}| = |\overrightarrow{o'e'}| - |\overrightarrow{o'O'}| = [e'\cos(\theta') - f\sin(\theta')]\hat{\mathbf{x}} + [e'\sin(\theta') + f\cos(\theta')]\hat{\mathbf{y}} \tag{3.8}$$

Since $|\overrightarrow{OO'}|$, $|\overrightarrow{Oe}|$ and $|\overrightarrow{O'e'}|$ are parallel to each other, then the ratio of $\hat{\mathbf{y}}$ and $\hat{\mathbf{x}}$ components must be equal.

$$\frac{|\overrightarrow{OO'}|_{\hat{\mathbf{y}}}}{|\overrightarrow{OO'}|_{\hat{\mathbf{x}}}} = \frac{|\overrightarrow{Oe}|_{\hat{\mathbf{y}}}}{|\overrightarrow{Oe}|_{\hat{\mathbf{x}}}} = \frac{|\overrightarrow{O'e'}|_{\hat{\mathbf{y}}}}{|\overrightarrow{O'e'}|_{\hat{\mathbf{x}}}} \tag{3.9}$$

Expanding Eq.(3.9)

$$\frac{T_y}{T_x} = \frac{e\sin(\theta) + f\cos(\theta)}{e\cos(\theta) - f\sin(\theta)} = \frac{e'\sin(\theta') + f\cos(\theta')}{e'\cos(\theta') - f\sin(\theta')} \tag{3.10}$$

Then, deriving f from first and second ratios from Eq.(3.10)

$$f_{1,2} = e\frac{T_y\cos(\theta) - T_x\sin(\theta)}{T_x\cos(\theta) + T_y\sin(\theta)} \tag{3.11}$$

and deriving f from first and third ratios from Eq.(3.10)

$$f_{1,3} = e'\frac{T_y\cos(\theta') - T_x\sin(\theta')}{T_x\cos(\theta') + T_y\sin(\theta')} \tag{3.12}$$

Thus, the focal length can be estimated if positions and orientations of the camera for two images are known.

CHAPTER 4

SIMULATIONS AND RESULTS

Simulations are performed to evaluate the proposed methods. Camera is assumed to be on a mobile robot. The position and the orientation of the camera is known under the random motion of mobile robot. We have done two different simulations and a hardware test. The first simulation is based on the synthetically generated 3D points in the scene. Points are projected on the image plane using predefined camera matrix. Then projected points are tracked over the random motions of the camera.

The focal length estimation is tested with and without errors point matching process. Additionally, error is introduced to the location of the robot. Second simulation is done in V-REP environment. A camera is attached on the differential drive robot and the robot makes curvilinear motion. While robot moves, camera captures the scene and the feature points are tracked over the motion.

In the hardware test, the proposed method is tested under real world situations. A webcam is attached to the 6 axes articulated robot arm and robot makes random motions on the horizontal plane. The extracted points are tracked and performance evaluation of focal length estimation is done.

4.1 Simulations with Synthetic Data

Simulations are performed on synthetically generated feature point locations that are randomly distributed and random movement trajectory of camera on 2D plane like it is assumed to be on the mobile robot. Two scenarios are created. In the first scenario, the feature points are distributed randomly in 3D space. The second scenario is based

on the feature points which randomly lie on two perpendicular walls. The resolution of the images are taken as 1024×1024, and the focal length is set to 1200 pixels. The objective is to estimate the focal length of the camera using the proposed method. The corresponding **K** is then,

$$\mathbf{K} = \begin{bmatrix} 1200 & 0 & 512 \\ 0 & 1200 & 512 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

Simulations are executed to evaluate and compare the proposed method and Kruppa equations with error free and with different errors on camera position and orientations and feature point position estimations. The first and the second ratios of Eq.(2.37), are used to estimate the focal length where only focal length is not known. Then, rearranging the polynomial in terms of focal length f_k.

$$Af_k^4 + Bf_k^2 + C = 0 \tag{4.2}$$

where

$$A = (u_{11}u_{12} + u_{21}u_{22})(v_{11}^2 + v_{21}^2)\sigma_1^2 + (v_{11}v_{12} + v_{21}v_{22})(u_{12}^2 + u_{22}^2)\sigma_1\sigma_2 \tag{4.3}$$

$$\begin{aligned} B = {} & (v_{11}v_{12} + v_{21}v_{22})u_{32}^2\sigma_1\sigma_2 + (v_{11}^2 + v_{21}^2)u_{31}u_{32}\sigma_1^2 \\ & +(u_{11}u_{12} + u_{21}u_{22})v_{31}^2\sigma_1^2 + (u_{12}^2 + u_{22}^2)v_{31}v_{32}\sigma_1\sigma_2 \end{aligned} \tag{4.4}$$

$$C = (u_{31}v_{31}\sigma_1 + u_{32}v_{32}\sigma_2)u_{32}v_{31}\sigma_1 \tag{4.5}$$

where u_{ij}, v_{ij} and $\sigma_{1,2}$ are the elements of **U**, **V** and **D** matrices respectively.

To test the performance of the proposed method and Kruppa equations, errors are introduced to the simulations. The noise added to the point locations is a Gaussian distribution. Several different noise variances are used in the simulations as 0, 1 and 25 pixels. The generated errors are added to the point locations. A similar approach is followed for camera positions. The variances of the Gaussian distributions for errors in camera positions are 0, 1 and 25 mm. Error is also introduced for heading

with variances 0°, 1° and 25°. 100 Monte Carlo runs are performed per scenario. At each run, motion of the robot and positions of the points are randomized. A sample view from one of the runs is given in Fig.4.1. In figure, blue markers represent 3D points in the space. Mobile robot (camera) moves for a duration of 50 frames. 90% of the points are visible through the all motion sequences and tracked successfully. The remaining of the points are tracked when they are becoming visible. Red marker represents positions of camera under the motion.

In the first scenario, the points are generated using a uniformly random probability distribution in a mixed fashion in 3D. The coordinates are given as

$$x_p \sim \mathcal{U}(-375, 375), \quad y_p \sim \mathcal{U}(500, 1000), \quad z_p \sim \mathcal{U}(-300, 300) \tag{4.6}$$

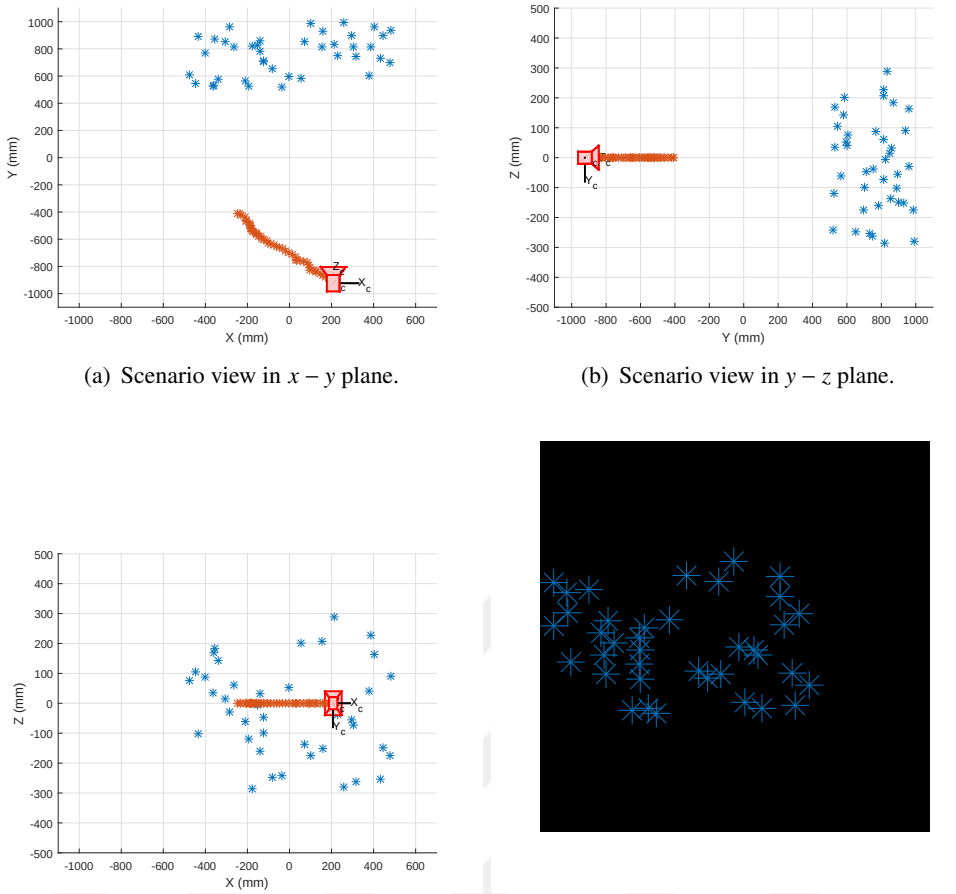
and the camera position and heading are changed with $\mathcal{U}(0, 20)$ and $\mathcal{N}(0, 0.1)$ at each step, respectively.

Results of the focal length estimation $f_{1,2}$, $f_{1,3}$ and f_k are given in Table 4.1, 4.2 and 4.3 using of Eq.(3.11), (3.12) and (4.2) respectively. In the results, 9 cases are evaluated for each possible case of point position detection error against Camera position and orientation errors. Where 0, 1, and 25 values are the variance values of the Gaussian distributions that are explained above. The results are given as mean and standard deviation of 100 simulation runs. In evaluation of Kruppa equation, half of the results are converged to complex numbers yielding invalid focal lengths. While evaluating these complex values to obtain the simulation results, they are replaced by the highest real values obtained in those simulations. In the Table. 4.3, number of invalid results are given in parenthesis.

Table 4.1: Simulation results for point cloud using $f_{1,2}$

		Camera Position and Orientation Errors		
		0	1	25
Point Position Detection Errors	0	1201.9 /41.9	1205.5 /61.9	1216.8 /247.3
	1	1209.7 /97.3	1212.6 /98.7	1224.7 /265.3
	25	1256.6 /605.1	1261.7 /629.6	1260.7 /580.2

In the second scenario, the points are generated using a uniformly random probability



(a) Scenario view in $x-y$ plane.

(b) Scenario view in $y-z$ plane.

(c) Scenario view in $x-z$ plane.

(d) Projection of points on the camera plane.

Figure 4.1: Scenario setup. Point locations are given with blue, the camera positions are with red.

Table 4.2: Simulation results for point cloud using $f_{1,3}$

		Camera Position and Orientation Errors		
		0	1	25
Point Position Detection Errors	0	1203.9 /31.2	1200.3 /46.3	1238.9 /194.1
	1	1208.5 /59.3	1205.0 /70.2	1244.0 /203.6
	25	1254.1 /406.7	1249.6 /402.7	1299.2 /495.8

Table 4.3: Simulation results for point cloud using f_k

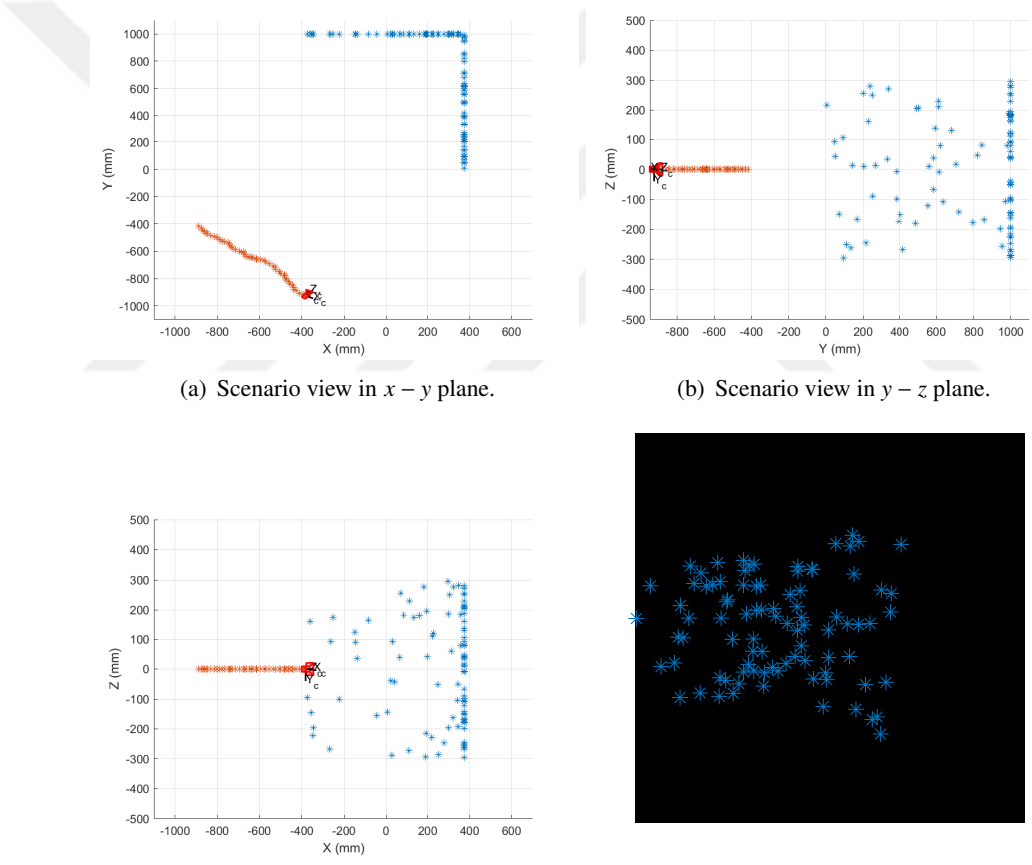
		Camera Position and Orientation Errors		
		0	1	25
Point Position Detection Errors	0	1413.8 /901.2(50)	1413.8 /901.2(50)	1413.8 /901.2(50)
	1	2841.7 /2412.3(46)	2841.7 /2412.3(46)	2841.7 /2412.3(46)
	25	2635.4 /1869.1(50)	2635.4 /1869.1(50)	2635.4 /1869.1(50)

distribution to create two perpendicular walls. The coordinates of the points are given as

$$x_p \sim \mathcal{U}(-375, 375), \quad y_p = 1000, \quad z_p \sim \mathcal{U}(-300, 300) \tag{4.7}$$

$$x_p = 375, \quad y_p \sim \mathcal{U}(0, 1000), \quad z_p \sim \mathcal{U}(-300, 300) \tag{4.8}$$

and the camera position and heading are changed with $\mathcal{U}(0, 20)$ and $\mathcal{N}(0, 0.1)$ at each step, respectively. One of sample view from the runs is given in Fig.4.2. Blue markers represent 3D points in the space. Red markers represent positions of camera under the motion.



(a) Scenario view in $x - y$ plane.

(b) Scenario view in $y - z$ plane.

(c) Scenario view in $x - z$ plane.

(d) Projection of points on the camera plane.

Figure 4.2: Scenario setup. Point locations are given with blue, the camera positions are with red.

Results of the f estimation are given in Table 4.4, 4.5 and 4.6. Evaluation procedure of results and representations are same with previous scenario results representation.

Table 4.4: Simulation results for two walled using $f_{1,2}$

		Camera Position and Orientation Errors		
		0	1	25
Point Position Detection Errors	0	1135.4 /665.3	1145.6 /620.9	1194.5 /685.0
	1	1138.8 /1641.5	1152.2 /1609.6	1196.0 /1673.4
	25	-359.9 /7398.4	-400.7 /7540.2	-326.9 /7628.1

Table 4.5: Simulation results for two walled using $f_{1,3}$

		Camera Position and Orientation Errors		
		0	1	25
Point Position Detection Errors	0	1071.5 /1314.1	1072.7 /1324.7	1138.3 /1068.8
	1	283.1 /7861.2	252.5 /8128.6	329.8 /8266.5
	25	-2053.3 /19475.5	-2083.3 /19649.4	-1618.9 /16559.5

Table 4.6: Simulation results for two walled using f_k

		Camera Position and Orientation Errors		
		0	1	25
Point Position Detection Errors	0	8778.5 /7406.9(50)	8778.5 /7406.9(50)	8778.5 /7406.9(50)
	1	1945.0 /1337.4(42)	1945.0 /1337.4(42)	1945.0 /1337.4(42)
	25	2708.9 /1758.0(50)	2708.9 /1758.0(50)	2708.9 /1758.0(50)

Results shows that f_k does not depend on the knowledge of position and orientation knowledge of camera, only requires point matches to estimate focal length. However as in seen results, f_k equation yields complex results in the half of the simulations. This is the due to the critical motion of the camera where Kruppa equation fails. In synthetic data simulation, the camera makes pure translation as a results it is natural to expect for failing Kruppa equations. On the other hand, proposed method has not failed under the pure translation. The standard deviation of $f_{1,2}$ and $f_{1,3}$ results is less than f_k results overall. Proposed method is more sensitive in point position detection error than camera position and orientation. Thus, it is necessary to track well and match precisely to reach good result. Position and orientation information of camera has less effect on estimation of focal length and this information error is expected for real robotic implementation.

4.2 Simulation on V-REP

Next step is the evaluation of the performance of V-REP simulations. In these simulations, a mobile robot with a camera is used. In V-REP, a 20×20 meter white floor is created. At each side 10 meter long white walls are put. The white color is selected to be segment features more easily. Sample view of simulation environment is given in Fig.4.3. Then, two white wall panels are put and fixed on to the floor . Size of the wall panels are 2×2 meter. In each wall panel, 4 different sizes of black rectangular shapes are attached. The detected corners of rectangles is used as feature points. One of the wall panel's center point is located at $(3657, -1750)$ mm in the x and y direction respectively and makes 15° with the x axis. Other's center point is located at $(3225, 25)$ mm in the x and y direction respectively and makes -35° with the x axis. Then from the library of V-REP, Pioneer P3-DX mobile robot is used without any specific reasons. Pioneer P3-DX is a differential drive robot and has two motor that drives two wheels. By controlling the velocities of motors, robot can move any desired path. Robot can be seen in the figure as red colored one. Then, a camera is mounted on the robot at 560 mm above the top of robot base. Camera sees forward direction of the robot. Camera's internal parameters set with same with synthetic experiment where focal length is 1200 pixel, and image size is 1024×1024 pixel.

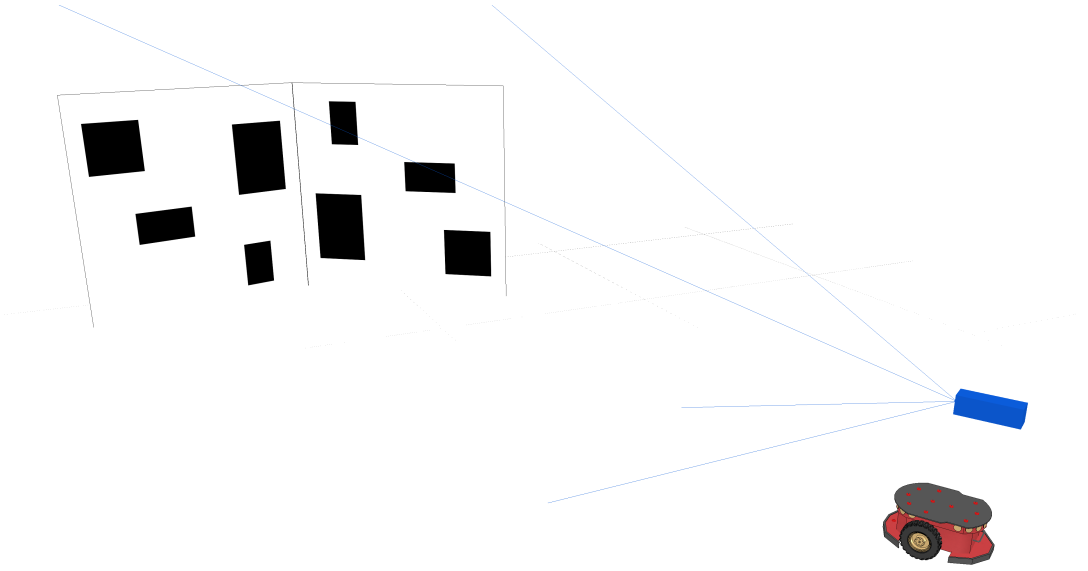


Figure 4.3: Implemented Test Scene on V-REP Simulator.

Robot should make curvilinear motion to perform successful focal length estimation. Result of a forward motion is the calculation of focal length as infinity. This happens due to degenerate configuration, since the epipoles are close to center of the image. Therefore, it is necessary to control the motion of robot to ensure that it sees the two wall panels. Before starting simulation, robot is placed where panels are visible by camera. Then, simulation is started from V-REP side for physical environment simulation. After that, the robot motion control, image processing, tracking and focal length estimation part is started. Robot's traveled distance and direction is determined by velocity changes of motor which are controlled throughout the simulation steps. The simulations are run 70 steps that results to 50 sec of physical motion in V-REP environment. Five types of paths are generated. Each has different starting points, end points, and directions. The paths are represented in the Fig.4.4. Star shaped points represents the acquired positions of the robot at each process steps in MATLAB. At the right part of figure, blue and black solid lines represent the locations of the wall panels. In all simulations, robot moves towards the walls and all paths are chosen as camera had to view wall panels during movement.

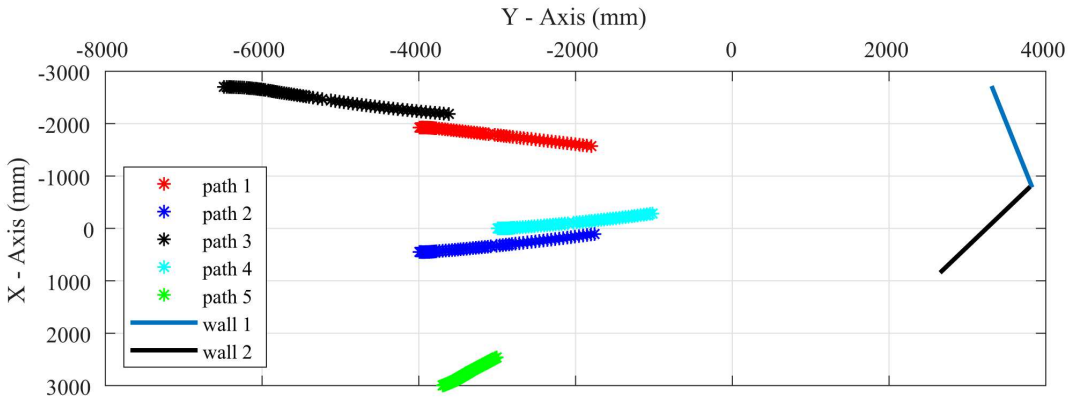


Figure 4.4: Path of camera that performed on V-REP Simulator.

In each path, 100 runs are performed. The starting and the destination points are the same in each path. However, due to numerical errors in the simulation and asynchronous communication between V-REP and MATLAB deviations occur. 100 different motion that robot makes during the movement along path 1 is given in Fig.4.4

Results of path 1, path 2, path 3, path 4 and path 5 for $f_{1,2}$, $f_{1,3}$ and f_k estimation are given in Table 4.7. The results represented as mean and standard deviation of

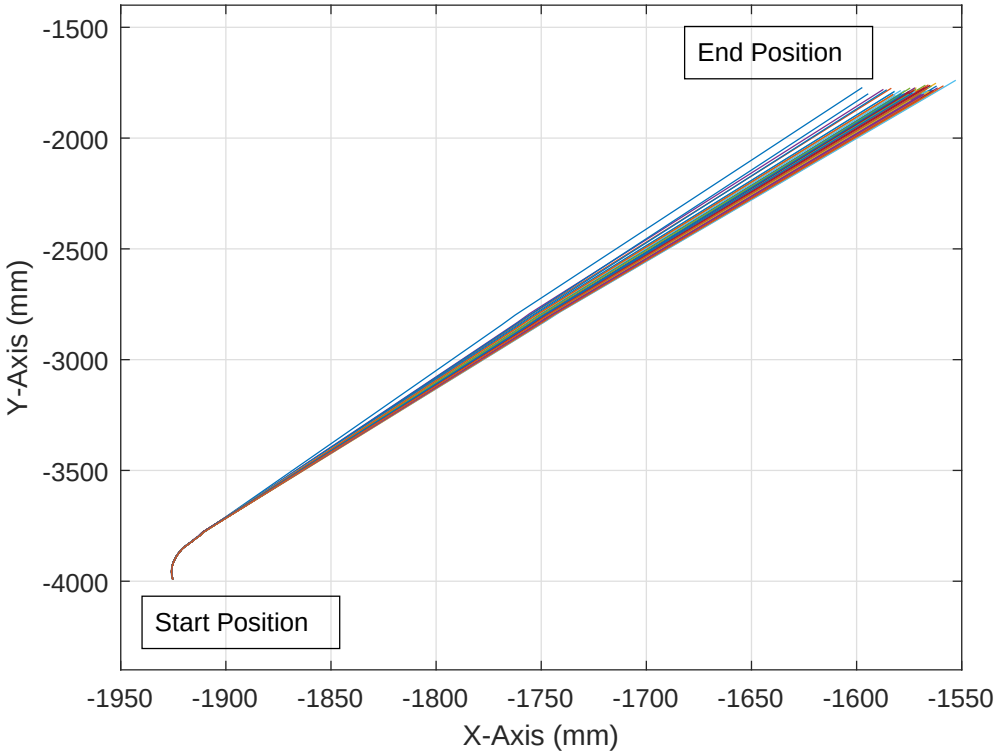


Figure 4.5: 100 motion of camera that performed on Path1 in the V-REP Simulator.

100 simulation runs. In evaluation of Kruppa equation, small number of the results are converged to complex number which means invalid focal length. In evaluation of Kruppa equation, half of the results are converged to complex numbers yielding invalid focal lengths. While evaluating these complex values to obtain the simulation results, they are replaced by the highest real values obtained in those simulations. In the Table.4.7, number of invalid results are given in parenthesis.

Table 4.7: Results of simulation

	$f_{1,2}$	$f_{1,3}$	f_k
path 1	1186.1 /90.9	1413.1 /1355.2	1151.1 /138.4(1)
path 2	1162.7 /44.0	1354.2 /192.8	1862.8 /957.6(28)
path 3	1144.9 /289.4	1010.0 /995.9	1494.0 /868.5(12)
path 4	1146.1 /89.7	1398.2 /372.3	1708.2 /1488.5(17)
path 5	1091.7 /310.6	3495.7 /6792.1	1181.3 /722.2(4)

Results indicate that Kruppa Equation reaches more real valued results in V-REP than

synthetic data simulations. Since V-REP provides a simulation based on realistic forces and reactions, the vehicle has some small motion along z axis as well. This helps the equations to recover from degeneracy problem. Thus, V-REP based simulation results are more accurate. The proposed method is much more effective than Kruppa equation results. In addition, in V-REP environment, objects are rendered to become visible in camera. Due to the rendering process some virtual movement may occur and the detected feature points, such as corners, from rendered image can yield only an approximate locations. These factors affect the results of both proposed method and Kruppa equation.

4.3 Hardware Test Results

After testing the performance of the proposed method in V-REP environment and obtaining satisfactory results, then physical testing is performed to evaluate the proposed method under real world conditions. It is necessary to acquire the position and orientation of camera with respect to fixed reference frame. Therefore, 6-axis articulated robot arm is used instead of the mobile robot platform. Since the location information acquisition is simpler and more accurate than mobile robot, it is a sensible approach to run the algorithm on a robot arm. In tests, KUKA KR210-L180 model 6-axis articulated robot arm is used which has a payload capacity of 210kg and 3300 mm radius of work envelope. At the tip point of 6th axis, Microsoft Lifecam Studio Webcam with 1920×1080 pixel resolution camera is used. In Fig.4.6, Robot platform and test scene are given.

Similar test environment with V-REP is setup in the workshop. The brown walls with bonded white papers are put in the scene. Than, Harris corner detector is applied to the acquired images to detect the corners of the white papers (see Fig.4.7). These corners are the feature points. Before the tests, webcam is manually calibrated with 2D calibration object using MATLAB Camera Calibration Tool. The manual calibration result for the focal length is found as 1437.84 pixels. This value is taken as reference f_r for error measurements.

At each step, the robot arm is moved randomly. At the end of each movement, actual

(a) 6 axis Robot arm (b) 6 axis Robot arm with webcam

Figure 4.6: Physical Test Setup

position of the camera is sent to the PC and then, an image is acquired from the webcam. The features are tracked for 60 steps. The method used for the generation of random motion for robot arm is given below. The robot makes diagonal movement towards the brown walls in the XY plane.

$X_{p0} \leftarrow \Delta X_p$

$Y_{p0} \leftarrow \Delta Y_p$

$\theta_0 \leftarrow \Delta\theta$

for $i = 1$ to 60 **do**

$X_{p_i} \leftarrow \Delta X_{p_{i-1}} + \mathcal{N}(-10, 1)$

$Y_{p_i} \leftarrow \Delta Y_{p_{i-1}} + \mathcal{N}(5, 1)$

$\theta_i \leftarrow \Delta\theta_{i-1} + \mathcal{N}(-0.1, 0.0025)$

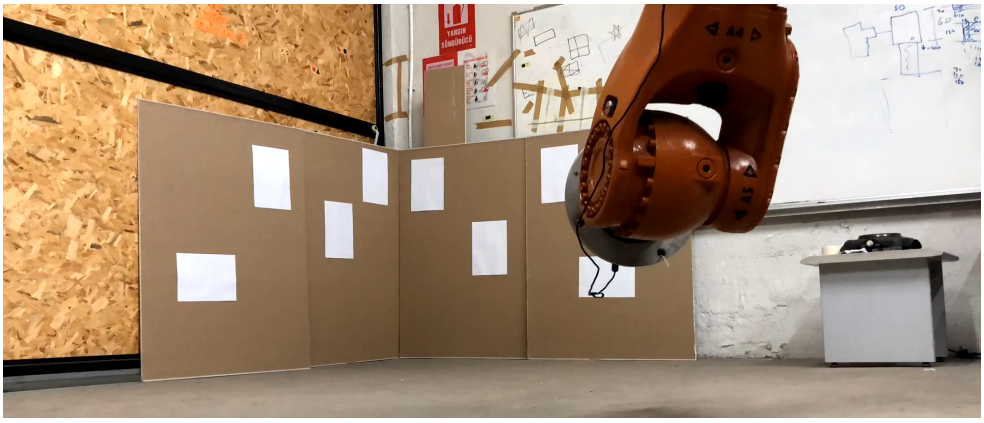
end for

$\Delta X_p \leftarrow -(X_{p_i} - X_{p_0}) + \mathcal{N}(0, 4)$

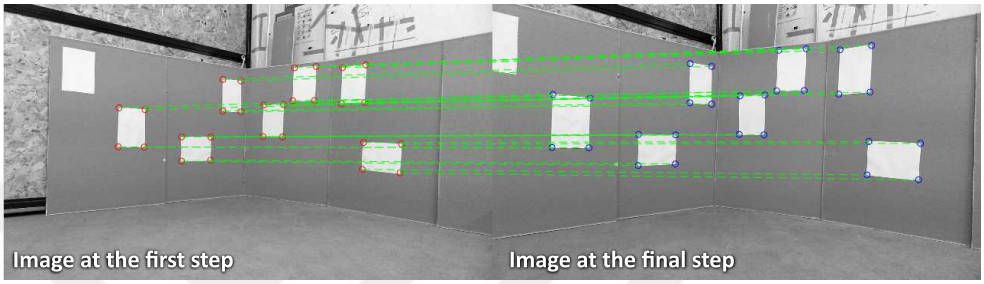
$\Delta Y_p \leftarrow -(Y_{p_i} - Y_{p_0}) + \mathcal{N}(0, 4)$

$\Delta\theta \leftarrow -(\theta_i - \theta_0) + \mathcal{U}(0, 1)$

Totally, 43 runs are performed with randomized motion of the camera. Positions of the



(a) Test scenario view



(b) Matched first and final image sequence

Figure 4.7: Test environment view and Matched points

tracks, position and orientation of robot/camera are collected over each steps. Then, the focal length is estimated at the end of each step. The information from the first and the last steps are used for the estimations. Median of focal length estimation at each step is calculated. In Kruppa equation calculation, almost half of the calculations resulted in complex numbers. To perform median operation, complex numbered results are omitted. Results are given in Fig.4.8. The reference value is equivalent to f_r.

Error value for each step of calculation is evaluated (see Fig.4.9). $f_{1,2}$ is more stable and reaches the steady state earlier than $f_{1,3}$. The estimations approach to a final value exponentially. Therefore, error model, f_e, is fit using $f_{1,2}$ as seen in Fig.4.9. And the result for the approximated function is

$$f_e(i) = 475.7 \exp(i)^{(-0.194)} + 170.8 \tag{4.9}$$

where i is the step number of the motion.

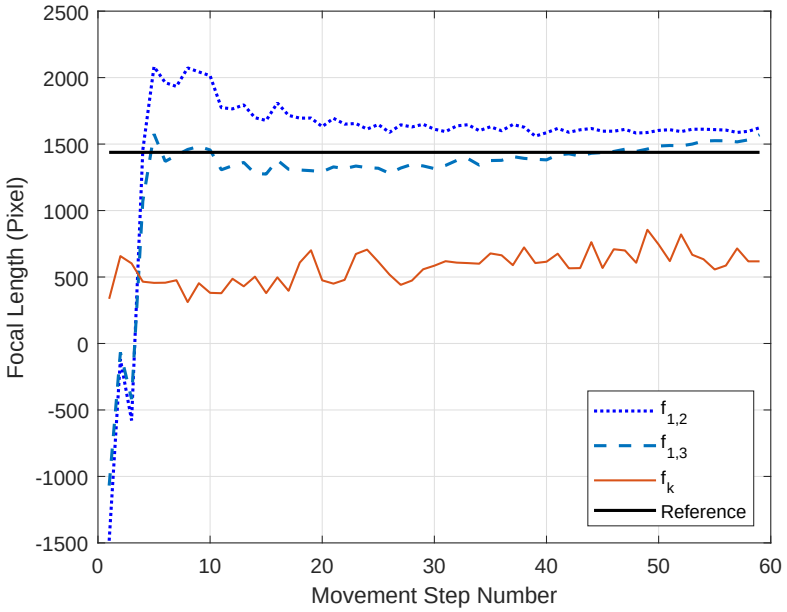


Figure 4.8: Focal estimation at each step of movement

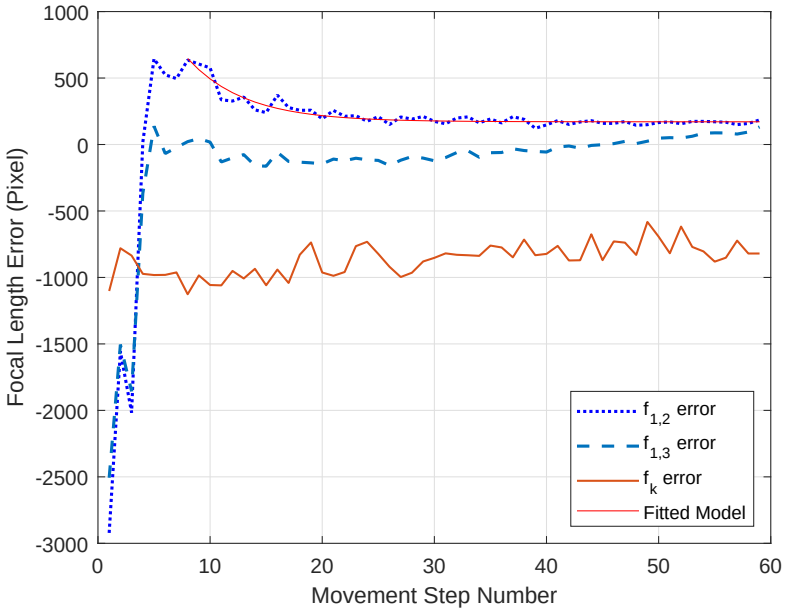


Figure 4.9: Focal estimation error at each step of movement and fitted error model

To evaluate the accuracy of the focal length estimation, reprojection is performed. Synthetic 3D points are projected on image plane using f_r in the calculations. Next, the image points are reprojected using the estimated focal length. The method for

evaluation is based on Fig.4.10. Image plane position and optical center location of cameras are exaggerated for visibility in the figure. For small sized image sensors, the distance between image plane and optical centers are close. Thus, the distance between three optical centers (O_s, O_r, O_b) may be omitted. As seen in the figure, three different points in space can projected onto the same pixel coordinates (green dot) for different cameras, i.e., black, red and blue dots are projected by f_b, f_r and f_s, respectively.

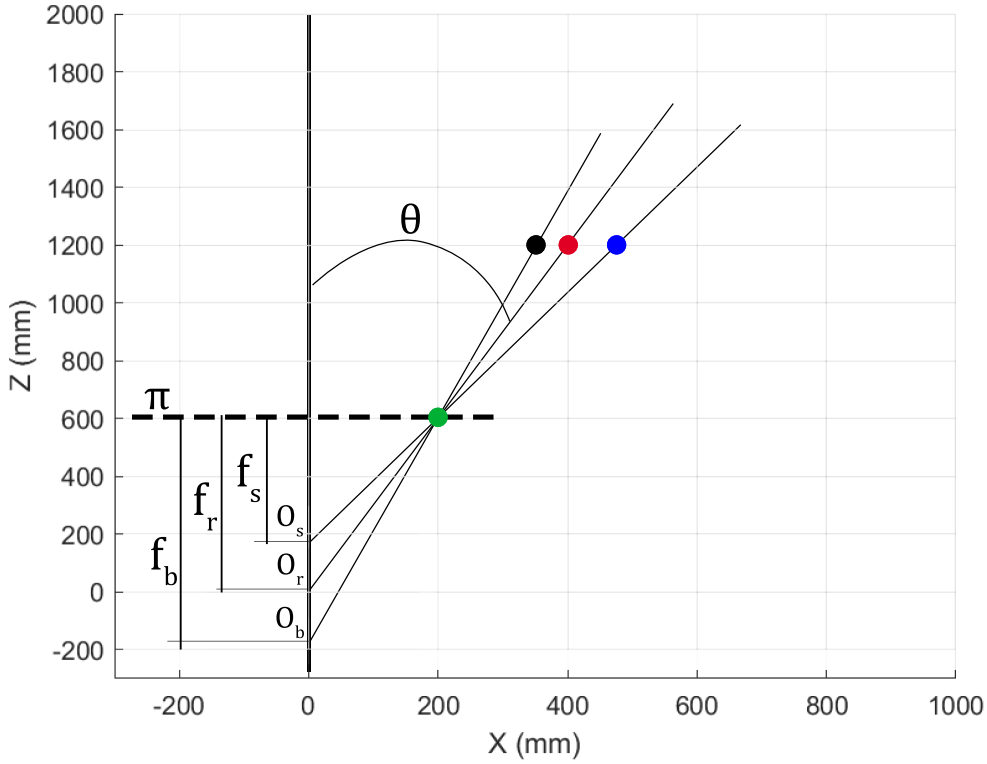


Figure 4.10: Reprojection of point based on different focal lengths

The red point is projected by the real focal length f_r. $\hat{f}$ is the estimated value for focal length. If $\hat{f} > f_r$, then the focal length corresponds to f_b and the point is reprojected to the black dot shown in Fig.4.10. In case, $\hat{f} < f_r$, then the estimated focal length becomes f_s in Fig.4.10. The reprojected point is then the blue dot. This reprojection error increases with the increasing θ angle and z distance.

To evaluate the performance of the proposed method, points are positioned at distances between 400 mm to 1600 mm in z direction. The results are shown in Table

4.8. Their angle θ to the optical axis also changes between 5° to 30°. The x displacement of 3D points are evaluated by $z\tan(\theta)$. In total, 42 points are evaluated. Using $\mathbf{K}_r$, pixel coordinates of the 3D points are acquired.

$$\mathbf{K}_r = \begin{bmatrix} 1437.84 & 0 & 960 \\ 0 & 1437.84 & 540 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

Then, pixel points are reprojected to the metric space using $\hat{\mathbf{K}}^{-1}$ where the entries of $\hat{\mathbf{K}}$ is calculated using the final value of $f_{1,2}$.

$$\hat{\mathbf{K}} = \begin{bmatrix} 1619.13 & 0 & 960 \\ 0 & 1619.13 & 540 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

The reprojection error of each 3D point is given in Table.4.8.

Table 4.8: Reprojection Error of Estimated Focal Length (in mm)

z \ θ	5°	10°	15°	20°	25°	30°
400 mm	-3.9	-7.8	-12.1	-16.4	-21.0	-25.9
600 mm	-5.8	-11.7	-18.1	-24.6	-31.5	-38.8
800 mm	-7.7	-15.6	-24.1	-32.8	-42.0	-51.8
1000 mm	-9.7	-19.5	-30.2	-41.0	-52.5	-64.7
1200 mm	-11.6	-23.3	-36.2	-49.1	-63.0	-77.7
1400 mm	-13.5	-27.2	-42.2	-57.3	-73.5	-90.6
1600 mm	-15.5	-31.1	-48.3	-65.5	-84.0	-103.6

As expected, real world test errors are slightly larger than the synthetic data and V-REP simulation results. This is due to the errors in feature point detections. The point locations in real world tests are not located perfectly. Naturally, errors increase. The proposed method outperforms the Kruppa equation results. Almost half of the Kruppa equation results are degenerate solutions. Thus, the proposed method can be utilized in a real world application.

CHAPTER 5

CONCLUSION

A new approach for self-calibration is presented in the thesis. This approach is applicable when the camera moves in $x - y$ plane and the camera is semi-calibrated. To increase the accuracy of calibration, the procedure also includes tracking and matching of points in consecutive video frames. Despite some constraints in motion of the camera, the likelihood of a situation causing degeneracy is less than when Kruppa equation is applied. In addition, the proposed method does not suffer from scaling problem like ADQ method does. Therefore, motion of the camera becomes important to get reliable results. The simulations are also done in V-REP environment. This helped to test the proposed method in a realistic environment. It is seen that the method works even better in V-REP simulations. In order to see the applicability of the method in SLAM problem, noise is introduced both to the vehicle location and to camera information. The results indicate that the method is robust against noise. Later, the method is tested in physical environment using a webcam that mounted on the articulated robot arm. Results show that the estimation results are adequate to implement in physical environment. Estimation results indicate that proposed method is better than Kruppa equation under the planer motion of the camera.

5.1 Discussion and Future Work

Although manual calibration gives more accurate results than self-calibration methods, it is impossible to implement as an online camera calibration procedure for mobile robotic implementations. The proposed method is developed and tested to work planar motion of a mobile robot with a camera for online camera calibration. The

proposed method derived for 2D motion of camera, therefore it could be derived for 3D motion of camera.

Displacement of the mobile robot determine the location of the epipoles. The epipoles have impact on the focal length estimation. The critical motion sequence for proposed method should be examined well and mobile robot should avoid critical motions. We plan to derive a cost function relating camera motion to self-calibration quality. Hence, it will be possible to use sensor control approach for path planning of the mobile robot as well as the camera.

All associated points are stored in our simulations even if the tracks die out. In the case of large number of feature points or new detections, tracking and association algorithms may fail due to large number of feature points. Parallel processing of tracks may help to handle the large number of points. Since closely placed points generate loose epipolar geometry and lead to invalid estimation, feature points must not be densely populated in a few regions in the image, but rather spread across the scene.

The formulation of the proposed method yields two focal length estimations instead of one, $f_{1,2}$ and $f_{1,3}$. However, the estimations for $f_{1,2}$ and $f_{1,3}$ differ in terms of means and standard deviations. To improve the accuracy of focal length estimation, optimization methods can be applied to the equations. In addition, it is possible to divide paths into the segments to obtain multiple equations of $f_{1,2}$ and $f_{1,3}$. Then, optimization methods could be implemented on each segment.

The real world implementation shows that the proposed method works adequetly. In addition, the trajectory of camera should be examined well in order to get more reliable results. After estimating the focal length, the tracked points can be used for 3D reconstruction of the scene or mapping problems.

REFERENCES

[1] Z. Zhang, *Emerging Topics in Computer Vision*, G. Medioni and S. B. Kang, Eds. Prentice Hall, 2004.

[2] Z. Chen, W. Liao, B. Xu, H. Liu, Q. Li, H. Li, C. Xiao, H. Zhang, Y. Li, W. Bao, and D. Yang, "Object tracking over a multiple-camera network," in *2015 IEEE International Conference on Multimedia Big Data*, April 2015, pp. 276–279.

[3] P. Tumas and A. Serackis, "Effective background subtraction algorithm for food inspection using a low-cost near infrared camera," in *2017 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, April 2017, pp. 1–4.

[4] X. Zhang, M. Chen, and X. Zhan, "A combined approach to single-camera-based lane detection in driverless navigation," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, April 2018, pp. 1042–1046.

[5] E. R. Davies, *Computer and Machine Vision, Fourth Edition: Theory, Algorithms, Practicalities*, 4th ed. Orlando, FL, USA: Academic Press, Inc., 2012.

[6] G. Bourmaud and R. Mégret, "Robust large scale monocular visual slam," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1638–1647.

[7] N. Nath, D. Braganza, and D. M. Dawson, "Position based structure from motion using a moving calibrated camera," in *2008 American Control Conference*, June 2008, pp. 1764–1769.

[8] T. Kijima, N. Sekiguchi, and H. Lim, "Study on object recognition by active stereo camera for clean-up robot," in *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, Oct 2017, pp. 27–31.

[9] T. Langner, D. Seifert, B. Fischer, D. Goehring, T. Ganjineh, and R. Rojas, "Traffic awareness driver assistance based on stereovision, eye-tracking, and head-up display," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3167–3173.

[10] G. Chesi and A. Vicino, "Visual servoing for large camera displacements," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 724–735, Aug 2004.

[11] C. Gurel, M. H. G. Zadeh, and A. Erden, "Development and implementation of rose stem tracing using a stereo vision camera system for rose harvesting robot," in *8th International Conference on Image Processing, Wavelet and Applications (IWW 2016)*, 2016.

[12] B. Silva, A. Burlamaqui, and L. Goncalves, "On monocular visual odometry for indoor ground vehicles," in *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, Oct 2012, pp. 220–225.

[13] B. Besbes, S. N. Collette, M. Tamaazousti, S. Bourgeois, and V. Gay-Bellile, "An interactive augmented reality system: A prototype for industrial maintenance training applications," in *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Nov 2012, pp. 269–270.

[14] L. W. Ming, L. Xue, C. X. Hui, and L. Q. Jun, "Development situation and trend of domestic and international aerial mapping camera," in *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, Aug 2011, pp. 527–530.

[15] P. Corke, *Robotics, Vision and Control*. Springer International Publishing, 2017.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.

[17] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *Proceedings of the Second European Conference on Computer Vision*, ser. ECCV '92. Berlin, Heidelberg: Springer-Verlag, 1992, pp. 321–334.

[18] B. Triggs, "Autocalibration and the absolute quadric," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1997, pp. 609–614.

[19] J. Civera, D. R. Bueno, A. J. Davison, and J. M. M. Montiel, "Camera self-calibration for sequential bayesian structure from motion," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ser. ICRA'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 3411–3416.

[20] R. I. Hartley, "Self-calibration of stationary cameras," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 5–23, Feb. 1997.

[21] T. Kohler, "Comparison of self-calibration algorithms for moving cameras," Master's thesis, University of Applied Sciences Munich, 2010.

[22] J. M. Frahm, "Camera self-calibration with known camera orientation," Ph.D. dissertation, der Technischen Fakultat der Christian-Albrechts-Universitat zu Kiel, 2005.

[23] P. Sturm, "Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1997, pp. 1100–1105.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speed-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2007.

[25] H. Shuo, W. Na, and S. Huajun, "Object tracking method based on surf," *AASRI Procedia*, vol. 3, pp. 351 – 356, 2012, conference on Modelling, Identification and Control.

[26] A. Perera, A. Pemasiri, S. Wijayarathna, C. Wijebandara, and C. Gamage, "Feature point tracking algorithm evaluation for augmented reality in handheld devices," in *Computer Vision - ACCV 2014 Workshops*, C. Jawahar and S. Shan, Eds. Cham: Springer International Publishing, 2015, pp. 275–288.

[27] B. Cyganek, *An Introduction to 3D Computer Vision Techniques and Algorithms*. USA: John Wiley & Sons, Inc., 2007.

[28] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[29] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *Int. J. Comput. Vision*, vol. 27, no. 2, pp. 161–195, Apr. 1998.

[30] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, June 1997.

[31] T. Vieville and D. Lingrand, "Using singular displacements for uncalibrated monocular visual systems," INRIA, Tech. Rep. RR-2678, Oct. 1995.

[32] R. Hartley, *Projective Reconstruction*. Boston, MA: Springer US, 2014, pp. 640–651.

[33] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*. Norwell, MA, USA: Kluwer Academic Publishers, 1996.

[34] E. E. Hemayed, "A survey of camera self-calibration," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, July 2003, pp. 351–357.

[35] G. Qian and R. Chellappa, "Bayesian self-calibration of a moving camera," *Computer Vision and Image Understanding*, vol. 95, no. 3, pp. 287–316, Sep. 2004.

[36] P. Sturm, "On focal length calibration from two views," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2, Dec 2001, pp. II–II.

[37] O. Faugeras and Q. Luong, *The geometry of multiple images : the laws that govern the formation of multiple images of a scene and some of their applications*. Cambridge (Mass.) : MIT press, 2001.

[38] P. Sturm, Z. L. Cheng, P. C. Y. Chen, and A. N. Poo, "Focal length calibration from two views: Method and analysis of singular cases," *Comput. Vis. Image Underst.*, vol. 99, no. 1, pp. 58–95, Jul. 2005.

[39] O. Faugeras, "Stratification of 3-d vision: projective, affine, and metric representations," *Journal of the Optical Society of America A*, vol. 12, pp. 46 548–4, 1995.

[40] B. Triggs, "Autocalibration from planar scenes," in *Proceedings of the 5th European Conference on Computer Vision-Volume I - Volume I*, ser. ECCV '98. London, UK, UK: Springer-Verlag, 1998, pp. 89–105.

[41] M. Ozuysal, “Manual and auto calibration of stereo camera systems,” Master’s thesis, METU, 2004.

[42] J. Ze-Tao, W. Wenhuan, and W. Min, “Camera autocalibration from kruppa’s equations using particle swarm optimization,” in *2008 International Conference on Computer Science and Software Engineering*, vol. 1, Dec 2008, pp. 1032–1034.

[43] S. L. L. Geng, S. Su, D. Cao, Y. Lei, and R. Ji, “A new camera self-calibration method based on csa,” in *2013 Visual Communications and Image Processing (VCIP)*, Nov 2013, pp. 1–6.

[44] H. Rastgar, “Robust self-calibration and fundamental matrix estimation in 3d computer vision,” Ph.D. dissertation, University of Ottawa, 2013.

[45] L. Fan, Z. Wang, B. Cail, C. Tao, Z. Zhang, Y. Wang, S. Li, F. Huang, S. Fu, and F. Zhang, “A survey on multiple object tracking algorithm,” in *2016 IEEE International Conference on Information and Automation (ICIA)*, Aug 2016, pp. 1855–1862.

[46] M. Hassaballah, A. A. Abdelmgeid, and H. A. Alshazly, “Image features detection , description and matching,” 2017.