KINEMATIC ANALYSIS OF A SLIDER CRANK MECHANISM VIA A PRE-CALIBRATED VISION SYSTEM DEVELOPED BY USING TWO COMMERCIAL CAMERAS

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

BY

M. HİLMİ ERALP

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

DECEMBER 2014
Approval of the thesis

KINEMATIC ANALYSIS OF A SLIDER CRANK MECHANISM VIA A PRE-CALIBRATED VISION SYSTEM DEVELOPED BY USING TWO COMMERCIAL CAMERAS

submitted by MEHMET HİLMİ ERALP in partial fulfillment of the requirements for the degree of Master Science in Mechanical Engineering Department, Middle East Technical University by,

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

Prof. Dr. Tuna Balkan
Head of the Department, Mechanical Engineering

Prof. Dr. Reşit Soylu
Supervisor, Mechanical Engineering Dept., METU

Examing Committee Members:

Prof. Dr. M. Kemal Ö zgören
Mechanical Engineering Dept., METU

Prof. Dr. Reşit Soylu
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Ergin Tönük
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Yiğit Yazıcıoğlu
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. İl k ay Ulusoy
Electrical and Electronics Engineering Dept., METU

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

Name, Last Name: M. Hilmi ERALP

Signature:
ABSTRACT

KINEMATIC ANALYSIS OF A SLIDER CRANK MECHANISM VIA A PRE-CALIBRATED VISION SYSTEM DEVELOPED BY USING TWO COMMERCIAL CAMERAS

Eralp, Mehmet Hilmi

M.S., Department of Mechanical Engineering

Supervisor: Prof. Dr. Reşit Soylu

December 2014, 225 Pages

There are two main objectives of this study. The first objective is to develop a vision system consisting of 2 inexpensive commercial cameras. In general, by self-calibration methods reconstruction of a scene by using uncalibrated images is performed up to a scale only. However, in this thesis reconstruction of a scene is to be performed such that one obtains the actual values of the distances in the scene. For this purpose, it is assumed that the extrinsic parameters of the cameras are known. Therefore, one needs to determine the intrinsic parameters of the cameras only. In order to calculate the intrinsic parameters, two methods, that take advantage of the simplified Kruppa equations and the equal eigenvalue theorem, are used. The results obtained via the two methods are compared with the results obtained by using a calibration pattern. A triangulation process is then performed to calculate several known distances in the scene by using the method that gives better results for the intrinsic parameters. The actual and estimated distances obtained via the vision system are then presented and compared.
The second objective of this study is to perform kinematic analysis of a slider crank mechanism by using the developed vision system. The position, velocity and acceleration analyses of the slider crank mechanism are realized by using several markers that are attached on the moving links of the mechanism. The positions of the markers are calculated by using the vision system. This data is then utilized to determine the joint variables, joint velocities and joint accelerations of the slider crank. The results thus obtained via an encoder attached to the input link of the mechanism are compared with the results obtained via the developed vision system. The effects of the locations of the markers and the effects of the number of markers used on the accuracy of the results are also investigated.

ÖZ

BİR KRANK BİYEL MEKANİZMASININ ÖN KALİBRELI, 2 TİCARİ KAMERA KULLANILARAK GELİŞTİRİLEN GÖRÜŞ SİSTEMİ VASİTASIYLA KİNEMATİK ANALİZİ

Eralp, Mehmet Hilmi
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Reşit Soylu
Aralık 2014, 225 Sayfa


To My Family

For Their Love and Support
Firstly, I would like to express my thanks to my supervisor Prof. Dr. Reşit Soylu for giving me the opportunity for doing this research and his endless support throughout this thesis. Furthermore, I am grateful to him since he provided all devices used in this study.

Secondly, I want to thank Hakan Mencek deeply because he provided the mechanism which has been built by him and helped a lot during experiments.

I would also like to thank all my friends for their help and support during this study.

Also, I would like to thank to all people of faculty who gave the support.

And finally, I wish to express my deepest thanks to my family for every kind of support and understanding.
# TABLE OF CONTENT

ABSTRACT..................................................................................................................v
ÖZ...............................................................................................................................vii
ACKNOWLEDGEMENTS..........................................................................................x
TABLE OF CONTENTS.............................................................................................xi
LIST OF FIGURES...................................................................................................xiv
LIST OF TABLES.....................................................................................................xix
LIST OF SYMBOLS..................................................................................................xx

**CHAPTERS**

1. INTRODUCTION.................................................................................................... 1
   1.1 Objective of the Thesis................................................................................... 1
   1.2 Scope of the Thesis....................................................................................... 2
   1.3 Literature Survey........................................................................................ 3
   1.4 Outline of the Thesis .................................................................................. 11

2. THE VISION SYSTEM AND CAMERA MODEL [1], [2] ................................. 13
   2.1 Human Vision and Computer Vision ......................................................... 13
   2.2 Pinhole Camera Model................................................................................ 15
   2.3 CCD Camera Model.................................................................................... 18
   2.4 Camera Lens and Distortion...................................................................... 20

3. IMAGE PROCESSING AND MARKER DETECTION ...................................... 25
   3.1 MARKER DETECTION............................................................................. 25
      3.1.1 Detecting the Red Color .................................................................. 25
      3.1.2 Median Filtering .............................................................................. 26
      3.1.3 Filtering via Threshold ..................................................................... 26
      3.1.4 Region Labelling ............................................................................. 27
      3.1.5 Filtering via Area ............................................................................. 28
**LIST OF FIGURES**

**FIGURES**

Figure 1: Stereopsis [32] .................................................................................................. 4
Figure 2: The Calibration Object used in the Tsai Approach [33] ................................. 5
Figure 3: Calibration Pattern for Zhang Technique [3] ................................................ 6
Figure 4: 1D Calibration Object [34] ........................................................................... 6
Figure 5: Epipolar Geometry and Absolute Conic [29] .............................................. 7
Figure 6: Pure Rotation Example [8] ......................................................................... 8
Figure 7: Experimental Setup of Self - Calibration Depth from Refraction [26] ........... 9
Figure 8: Absolute Quadric in Plane at Infinity and Its Image [37] ............................ 10
Figure 9: Sketch of An Eyeball [2] ........................................................................... 14
Figure 10: Pinhole Camera Model [2] ........................................................................ 15
Figure 11: Pinhole Camera Geometry and Similar Triangles [2] ............................... 16
Figure 12: Pixel Coordinate System ........................................................................ 18
Figure 13: Pixel Shape ................................................................................................ 19
Figure 14: Thin Lens Model [39] ............................................................................. 21
Figure 15: Salt and Pepper Noise ............................................................................. 26
Figure 16: 4 - connected and 8 - connected pixels .................................................... 27
Figure 17: Sample Stages of Marker Detection .......................................................... 29
Figure 18: Epipolar Planes ....................................................................................... 31
Figure 19: Projection of a 3D Point in Two Image Planes [1] ................................... 32
Figure 20: Epipolar Line [1] .................................................................................... 33
Figure 21: Point Transfer with Homography [1] ...................................................... 35
Figure 22: Calibration Pattern Viewed at Different Orientations ............................. 51
Figure 23: Calibration Pattern .................................................................................. 53
Figure 24: Rectangular Shape on the Image ............................................................. 54
Figure 25: Extracted Corner Points ........................................................................ 55
Figure 26: Example of Output for Bouguet Camera Calibration Toolbox ................. 59
Figure 27: Visualization of Estimated Extrinsic Parameters ..................................... 59
Figure 28: Example of Stereo Calibration Results .................................................. 60
Figure 29: Visualization of the Stereo Parameters ..................................................... 61
Figure 30: Absolute Conic and Its Image ................................................................. 65
Figure 31: Absolute Conic and Absolute Dual Quadric ............................................. 67
Figure 32: Epipolar Lines Tangent to Conic .............................................................. 72
Figure 33: Selected Multiple Images ....................................................................... 80
Figure 34: Error Graph of $\alpha_x$ .......................................................................... 85
Figure 35: Error Graph of $\alpha_y$ .......................................................................... 86
Figure 36: Error Graph of $x_0$ ................................................................................ 86
Figure 37: Error Graph of $y_0$ ................................................................................ 87
Figure 38: Error Graph of $\alpha_x$ .......................................................................... 87
Figure 39: Error Graph of $\alpha_y$ .......................................................................... 88
Figure 40: Error Graph of $x_0$ ................................................................................ 88
Figure 41: Error Graph of $y_0$ ................................................................................ 89
Figure 42: Left View of a Scene with Detected Corner Points .................................. 96
Figure 43: Right View of a Scene with Detected Points ............................................ 96
Figure 44: Left View with Selected Few Corner Points ............................................ 97
Figure 45: Right View with Selected Few Corner Points ........................................... 97
Figure 46: Left View with Selected Corner Points .................................................... 101
Figure 47: Right View with Selected Corner Points ................................................ 101
Figure 48: Original View of 3D scene .................................................................... 102
Figure 49: x- y View of 3D Scene ........................................................................... 102
Figure 50: x- z View of 3D Scene ............................................................................ 103
Figure 51: y- z View of 3D Scene ............................................................................ 103
Figure 52: The slider crank mechanism .................................................................. 105
Figure 53: Markers on the mechanism .................................................................... 108
Figure 54: Curve Fit for position of marker $M_{2,3}$ ............................................... 110
Figure 55: Curve Fit for position of marker $M_{3,1}$ ............................................... 111
Figure 56: Illustration of Slider Crank Mechanism ................................................. 113
Figure 57: Encoder Data for a specific time interval ............................................... 121
Figure 58: Fourier series curve fit in $\theta_3$ .............................................................. 122
Figure 59: $\theta_2$ vs time as an example ................................................................... 123
Figure 60: Joint variables of the slider crank mechanism ........................................ 124
Figure 61: Results of the kinematic analysis by using marker M2,1 ......................... 127
Figure 62: Results of the kinematic analysis by using marker M2,1 [continued] .... 128
Figure 63: Results of the kinematic analysis by using markers M2,1 and M2,2 ....... 131
Figure 64: Results of the kinematic analysis by using markers M2,1 and M2,2 [continued] ............................................................................................................... 132
Figure 65: Results of the kinematic analysis by using the marker triple M2,3, M3,4 and M4,1 ........................................................................................................................... 134
Figure 66: Results of the kinematic analysis by using the marker triple M2,3, M3,4 and M4,1 [continued] ........................................................................................................ 135
Figure 67: Results of the kinematic analysis by using markers M2,2, M2,3, M3,4 and M4,1 ........................................................................................................................... 137
Figure 68: Results of the kinematic analysis by using markers M2,2, M2,3, M3,4 and M4,1 [continued] ........................................................................................................ 138
Figure 69: Velocities and Accelerations obtained via the vision system and via the encoder .................................................................................................................................. 145
Figure 70: An Example of Digital Image .................................................................. 155
Figure 71: Matrix Representation of Digital Image [40] ........................................ 156
Figure 72: Binary Image .......................................................................................... 158
Figure 73: Representation of Grayscale Image .......................................................................................... 159
Figure 74: Full Color Image ..................................................................................... 160
Figure 75: Primary and Secondary Colors of the RGB Model .............................. 161
Figure 76: RGB Color Cube [40] ............................................................................. 162
Figure 77: CMY Color Model .................................................................................. 163
Figure 78: HSV Color Circle [40] ............................................................................. 164
Figure 79: HSV Color Cone ..................................................................................... 165
Figure 80: Harris Corner Detector ........................................................................... 168
Figure 81: Susan Corner Detector [41] .................................................................... 170
Figure 82: Results of the kinematic analysis by using marker M2,2 ......................... 182
Figure 83: Results of the kinematic analysis by using marker M2,2 [continued] .... 183
Figure 84: Results of the kinematic analysis by using marker M2,3 ......................... 184
Figure 85: Results of the kinematic analysis by using marker M2,3 [continued] .... 185
Figure 86: Results of the kinematic analysis by using marker M2,4 ......................... 186
xvi
Figure 87: Results of the kinematic analysis by using marker M2,4 [continued].... 187
Figure 88: Results of the kinematic analysis by using marker M3,1 ............... 188
Figure 89: Results of the kinematic analysis by using marker M3,1 [continued]...... 189
Figure 90: Results of the kinematic analysis by using marker M3,2 ............... 190
Figure 91: Results of the kinematic analysis by using marker M3,2 [continued]...... 191
Figure 92: Results of the kinematic analysis by using marker M3,3 ............... 192
Figure 93: Results of the kinematic analysis by using marker M3,3 [continued]..... 193
Figure 94: Results of the kinematic analysis by using marker M3,4 ............... 194
Figure 95: Results of the kinematic analysis by using marker M3,4 [continued]..... 195
Figure 96: Results of the kinematic analysis by using marker M4,1 ............... 196
Figure 97: Results of the kinematic analysis by using marker M4,1 [continued]..... 197
Figure 98: Results of the kinematic analysis by using marker M4,2 ............... 198
Figure 99: Results of the kinematic analysis by using marker M4,2 [continued]..... 199
Figure 100: Results of the kinematic analysis by using marker M4,3 .............. 200
Figure 101: Results of the kinematic analysis by using marker M4,3 [continued]... 201
Figure 102: Results of the kinematic analysis by using marker M4,4 .............. 202
Figure 103: Results of the kinematic analysis by using marker M4,4 [continued]... 203
Figure 104: Results of the kinematic analysis by using markers M2,1 and M2,3... 206
[continued] ............................................................................................................... 207
Figure 105: Results of the kinematic analysis by using markers M2,1 and M2,3
Figure 106: Results of the kinematic analysis by using markers M2,3 and M2,4... 208
[continued] ............................................................................................................... 209
Figure 107: Results of the kinematic analysis by using markers M2,3 and M2,4
Figure 108: Results of the kinematic analysis by using markers M3,2 and M3,3... 210
[continued] ............................................................................................................... 211
Figure 109: Results of the kinematic analysis by using markers M3,2 and M3,3
Figure 110: Results of the kinematic analysis by using markers M3,3 and M3,4... 212
[continued] ............................................................................................................... 213
Figure 111: Results of the kinematic analysis by using markers M3,3 and M3,4
Figure 112: Results of the kinematic analysis by using markers M4,1 and M4,2... 214
[continued] ............................................................................................................... 215
LIST OF TABLES

TABLES

Table 1: Results Obtained by using Equal Eigenvalues Method with Different Initial Guesses ................................................................................................................................. 81
Table 2: Results Obtained by using the Simplified Kruppa Equations with Different Initial Guesses ............................................................................................................ 81
Table 3: Results of the Simplified Kruppa Equations ................................................ 83
Table 4: Results of Equal Eigenvalue Theorem ......................................................... 84
Table 5: Image Coordinates and 3D Point Coordinates ............................................. 99
Table 6: Estimated Distance between Points ........................................................... 100
Table 7: Error Percentage between Estimated Distance and Real Distance Values 100
Table 8: Distances between Different Markers ........................................................ 112
Table 9: Polar Coordinates of the Markers .............................................................. 115
Table 10: Average Errors of Joint Variables ........................................................... 140
Table 11: RMS Values of Error in Joint Variables .................................................. 141
LIST OF SYMBOLS

M (x, y, z) : 3D point in the scene
m (x, y) : 2D image point of 3D point M
f : focal length
α : skew angle
px, py : pixel dimensions in horizontal and vertical directions, respectively
K : camera calibration matrix
αx, αy : focal lengths in horizontal and vertical directions, respectively
x0, y0 : principal points
s : skew value
(xn, yn) : normalized coordinates of image point
do : distance between object and center of lens
di : distance between image and center of lens
δu_i^r, δv_i^r : infinitesimal radial displacement of i\(^{th}\) point
δu_i^t, δv_i^t : infinitesimal tangential displacement of i\(^{th}\) point
(u_0, v_0) : image coordinates of principal point
(ũ_i, ũ_i) : image coordinates of i\(^{th}\) scene point in pinhole projection
k_1, k_2 : radial distortion coefficients
p_1, p_2 : tangential distortion coefficients
u_i, v_i : corrected image point of i\(^{th}\) point
$D_w, D_v$ : coefficients to convert metric into pixel

$s_u$ : scale factor

fps : frame per second

$F_p$ : matrix representing pixel values of an image

$L$ : number of the gray level

$k$ : number of bits

$R, G, B$ : primary colors which are red, green, blue

$C, M, Y$ : secondary colors which are cyan, magenta, yellow

$B(x, y)$ : value for binary image for the pixel $(x, y)$

$T$ : threshold

$I(x, y)$ : function showing the value of grayscale

$w(x, y)$ : window factor

$m(\Delta x, \Delta y)$ : gradient value in any direction

$I_x, I_y$ : intensity values

$\lambda_1, \lambda_2$ : eigenvalues

$X$ : 3D point in the scene

$x$ : image point of point $X$ in first image plane

$x'$ : image point of point $X$ in second image plane

$C, C'$ : camera centers of first camera and second camera, respectively

$\pi$ : epipolar plane

$l, l'$ : epipolar lines in first image plane and second image plane, respectively
e, e': epipoles in first image plane and second image plane, respectively

F: Fundamental Matrix

E: Essential Matrix

$H_\pi$: homography matrix with respect to plane $\pi$

P, P': camera projection matrices of first and second camera, respectively

$P^*, P'^*$: Pseudo – inverse of camera projection matrices

A: matrix involving homogeneous set of equations

U, D, V: singular value decomposition matrices

$m_0 (m_1, m_2)$: centroid plane in image

T, T': transformation matrices for first and second image planes, respectively

t: translation vector

R: rotation matrix

om: rotation vector

$r_1, r_2, r_3$: columns of rotation matrix R

H: homography matrix

$h_1, h_2, h_3$: columns of homography matrix H

u: image point of scene point at infinity

w: absolute conic

$Q_\infty^*$: absolute quadric

$X_j$: 3D point coordinates of $j^{th}$ point of scene
$x^i_j$ : image point of $X_j$ in $i^{th}$ camera

$P_{Mi}$ : camera projection matrix with projective ambiguity

$X_{Mj}$ : 3D point of $X_j$ with projective ambiguity

$w^*$ : dual of absolute conic

$u_1, u_2, u_3$ : columns of matrix $U$

$v_1, v_2, v_3$ : columns of matrix $V$

$r, s$ : eigenvalues of matrix $D$

$\rho_1, \rho_2, \rho_3$ : numerator terms of Kruppa Equations

$\theta_1, \theta_2, \theta_3$ : denominator terms of Kruppa Equations

$w_1, w_2, w_3$ : weight factors

$L_1, L_2, L_3$ : linearly dependent equations of Kruppa

$F$ : function describing triangulation

$K_{left}$ : camera calibration matrix of left camera

$K_{right}$ : camera calibration matrix of right camera

$D$ : distance between 3D points in the scene or between markers

$l$ : number of links in the mechanism

$j$ : number of joints in the mechanism

$NM$ : number of markers on the mechanism

$M_{i,j}$ : $j^{th}$ marker on link $i$

$(L_{i,j}, Y_{i,j})$ : are polar coordinates of marker $M_{i,j}$ with respect to a body fixed frame attached to link $i$

$O_mX_mY_mZ_m$ : mechanism coordinate frame

$O_cX_cY_cZ_c$ : camera reference frame

$\theta_2, \theta_3, s_4$ : joint variables of the slider crank mechanism
$\left[m R_c \right]$: rotation matrix which relates a vector expressed in camera and reference frames

$u_x, u_y, u_z$: unit vectors in x, y and z directions of the frame, respectively

$NF$: number of frame
CHAPTER 1

INTRODUCTION

1.1 Objective of the Thesis

Three dimensional reconstruction of a scene from uncalibrated images is one of the most challenging problems in computer vision. In order to perform reconstruction of the scene, the internal parameters and the external parameters of the cameras, with respect to each other, should be estimated. In literature, reconstruction is performed up to a scale, i.e., the actual dimensions, or positions, cannot be known by self-calibration. However, in this thesis, it is aimed to obtain the reconstruction of the scene with actual dimensions without using any calibration object. Therefore, the external parameters, which are the relative position or relative motion of cameras, are assumed to be fixed and known. In the light of that, a vision system consisting of two cameras is developed in this thesis. Today, digital cameras are quite cheap. Hence, in this thesis, digital cameras are used for the vision system and a CCD camera model is assumed for camera calibration. In order words, there are five internal parameters to be estimated during self-calibration. After determining the internal and external parameters, the reconstruction can be performed and position of any point in the scene with respect to the camera reference frame can be calculated.

After developing the vision system, kinematic analysis of a slider crank mechanism, which is readily available since it has been built for a TUBITAK project [50], is performed by using the vision system. Several markers are attached to the moving links of the mechanism. These markers are tracked by the developed vision system and by using the algorithm developed in this study, the joint variables of the mechanism are obtained as functions of time. The results obtained by the vision system are then compared with the results obtained by an encoder attached to the input link of the slider crank mechanism. Using the time history of the joint variables, velocity and the acceleration analysis of the mechanism are also performed.
1.2 Scope of the Thesis

As stated before, self–calibration, i.e., estimating the internal parameters from uncalibrated images, is the first objective of this thesis. In order to perform self–calibration, we need to set up a pre–calibrated multi–camera vision system. The system is labelled to be pre–calibrated since the external parameters are assumed to be known.

After the multi–camera vision system is set up, a scene is recorded with the cameras being in the video mode. These recorded movies have to be converted into individual image frames. In order to obtain these image frames, a developed MATLAB® code is used. If one does not want to use MATLAB® to obtain the image frames, he/she can use another readily available software.

In this thesis, self–calibration is performed off–line. In other words, calibration is performed after the recording rather than during the recording. After obtaining the images, these images can be processed by means of various ways. In summary, in order to calibrate the cameras, the following steps are applied.

- Synchronization of images from the two cameras
- Finding corresponding 2D corner points between the images
- Determination of the algebraic relations between the images
- Performing self–calibration and determining the internal parameters

The external parameters, on the other hand, are estimated by using the Bouguet camera calibration toolbox with a calibration pattern [31]. As stated before, in this study it is assumed that the extrinsic parameter of the cameras with respect to each other are fixed and known.
Once the cameras are calibrated, the vision system can be used to perform kinematic analysis of any mechanism, or any machine. In this thesis, a slider crank mechanism is analyzed. It should be noted that in order to perform the kinematic analysis, the procedure applied to the slider crank mechanism may be applied to any other mechanism as well. The following steps are followed to perform the kinematic analysis of the slider crank mechanism:

- Finding the positions of the markers in the camera reference frame for each frame
- Curve fitting process to smooth the position data of the markers
- Performing position analysis and obtaining the joint variables of the mechanism by utilizing the positions of the markers
- Performing velocity and acceleration analysis by taking time derivatives of the joint variables.

1.3 Literature Survey

Camera calibration has been a very challenging area for the last 20 years. There are various methods to achieve camera calibration. Some of these methods include the use of a calibration object while some of them do not require a calibration object, which is self-calibration.

Fusing the pictures, which are recorded by two eyes, and realizing the differences between them allow us to gain the sense of depth [2]. So, the authors state that in order to obtain the depth information at least two cameras which are recording the same scene are needed. This task is known as stereopsis. (Figure 1)
Calibration techniques are categorized as (i) calibration using 3D calibration object, (ii) calibration using 2D calibration object, (iii) calibration using 1D calibration object and (iv) self–calibration.

A popular and traditional method for calibration using a 3D object is the Tsai approach [33]. A 3D calibration object is used for this technique. In Tsai’s experiments, the calibration object is created by impressing a template of instant lettering graphics sheet containing 16 squares whose dimensions are 2 in x 1.5 in x 0.5 in. These squares are on the top of a block steel. (Figure 2). Calibration is done by using the corners of these squares.
For calibration using a 2D calibration object, Zhengyou Zhang suggests a technique in which the camera records a planar pattern at a number of (at least two) different orientations [3], [4]. Either the camera or the planar pattern can be moved to capture planar pattern at different orientations. In addition to the intrinsic and the extrinsic parameters, the radial lens distortion coefficients can be found. In the light of this technique, a MATLAB® toolbox has been released [31]. Recently, MATLAB® has developed its own module to calibrate a camera by using this technique. The planar pattern used in this technique is shown in Figure 3.
In addition to calibration using a 2D calibration object, Zhengyou Zhang also discusses a calibration technique using 1D calibration object [34]. In this technique, points aligned on a line are used as seen in Figure 4. One of the points should be fixed. Otherwise, calibration is not possible.

Self-calibration is introduced by S. Maybank, O.D. Faugeras and Q.T. Luong [29]. They have realized the calibration process by using a moving camera recording a static
scene. It is shown that calibration can be performed by using point correspondences and fundamental matrices between the image sequences. Constant intrinsic parameter, epipolar constraint and the absolute conic concept are the main parts of this theory. The epipolar constraint and the absolute conic concept are illustrated in Figure 5.

[15] suggests a method to retrieve Euclidean calibration. Projective calibration, affine calibration and Euclidean calibration are performed step by step to reach the Euclidean calibration. Modulus constraint is the novel part of the method. Later, [17] improved the method to obtain metric calibration from only three images. A. Zisserman et al. and R. Horaud et al. extended this method for a stereo rig [16]. P. Sturm examined the critical motion sequences from multiple image pairs [35].

[9] suggests a global optimization technique to obtain Euclidean reconstruction from several views of same camera which is moving. Projection reconstruction, quasi-affine
reconstruction and Euclidean reconstruction is obtained, respectively. Afterwards, Hartley extended this method for a camera which is recording the scene at the same point but with different orientations [8]. Also, he introduced the concept of transformation matrix between image planes. Since the images are taken using the same location, point correspondence and finding the transformation matrix is easier. Clearly, the method is based upon pure rotation. An example of pure rotation is given in Figure 6.

Qiang Ji and Songtao Dai improved the study of Hartley by stating that pure rotation assumption is unrealistic since the optical center of the camera is often unknown [10]. Hence, the rotation is performed about an unknown and fixed point, which is near the optical center. So, there should be a translational offset between the optical center and the point about which the rotation is performed about. In both Hartley’s case and Qiang Ji et al. ‘s case, the camera rotates with an unknown angle. However, in Qiang Ji et al.’s technique, the camera should rotate with same amount of angle.
David Nister et al. studied non-parametric self-calibration by observing motion in the distorted image [36]. They focused on the case of three infinitesimal rotations. In this method, reconstruction of points is possible up to projective ambiguity. [26] introduced a method which is self-calibration depth from refraction. In this method, a scene is recorded firstly by a fixed camera and then same scene is captured by placing a transparent medium between the scene and the camera. Correspondence points of images are used to find the orientation of the parallel planar faces of the medium and depths of scene points. The experimental setup can be seen in Figure 7.

![Experimental Setup of Self-Calibration Depth from Refraction](image)

Figure 7: Experimental Setup of Self-Calibration Depth from Refraction [26]

Triggs has suggested a new method for self-calibration in order to obtain Euclidean reconstruction of three or more views taken by a moving camera with fixed and unknown intrinsic parameters [37]. He uses the absolute quadric concept which is a degenerate quadric consisting of planes tangent to the absolute conic (Figure 8). This absolute quadric is obtained by using a constrained nonlinear minimization technique.
Anders Heyden and Kalle Astrom have shown that self-calibration is possible in the case of zooming cameras if skew is 0 and the aspect ratio is taken as 1 [38]. A zooming camera implies that the intrinsic parameters of the camera are changing continuously. At the end, they have obtained Euclidean reconstruction up to a similarity transformation. Thao Dang, Christian Hoffmann and Christoph Stiller have shown that continuous stereo self-calibration by camera parameter tracking, in case of varying intrinsic parameters, is possible [11]. It is assumed that initial guesses for the camera calibration parameters are readily available. They developed a setup consisting of three cameras which can be rotated about their vertical axes.

M.J. Brooks et al. have studied metric reconstruction with a dynamic stereo head [27]. In this technique, the scene is viewed by a moving stereo head. This stereo head consists of two cameras and each camera can vary its angle of vergence. In their study,
the cases such that knowing distance between the cameras of the stereo head or cameras of the stereo head have same intrinsic parameters have been examined.

Like the stereo head, there are several vision systems consisting of stereo cameras. Kinect and Bumblebee2 are two examples of them. Both of them can be used to obtain 3D coordinates of a point in the scene. [43] presents the comparison of them for indoor environment. In [43], it is shown that Kinect is not suitable in case of bright daylight and Bumblebee2 has a high cost and the scene must be well illustrated. Also, it is shown that resolution of Kinetic is very low which can cause higher value of error.

There are different kind of applications of the computer vision. For example, [44], [45] shows the application of the computer vision in determining strain and stress distribution. In these studies, an extensometer is used to examine deformation of the specimen. The calibration of extensometer is performed by using the calibration specimen. Another application is that analysis of the human walking (gait) by using the vision system shown in [46], [47]. In [48], a study of walking posture analysis based on vision system by extracting the body line, neck line, center of gravity and gait width is shown. However, the calibration of the vision system is not mentioned in these studies.

1.4 Outline of the Thesis

The scope and objective of the thesis, accompanied with a literature survey is given in Chapter 1.

Chapter 2 gives brief information about vision systems and camera models which exist in literature. Also, the concept of lens distortion effect and camera calibration matrix are presented in this chapter. In Chapter 3, digital image is discussed. Image types and image processes are presented. Several methods are given in order to find the corner points. Also, the markers detection are explained step by step in this chapter. Chapter
4 discusses algebraic relations between images. The fundamental matrix and the essential matrix are defined in this chapter. Some methods, which are used to find these matrices, are also presented. Chapter 5 gives brief information about the Bouguet camera calibration toolbox. The functions of the toolbox and some important issues about the toolbox are presented. Self-calibration methods are presented in Chapter 6. Also, experimental results by self-calibration methods are presented. In chapter 7, triangulation is discussed. Furthermore, reconstruction is explained by using triangulation and some experimental result are presented. Chapter 8 includes the kinematic analysis of the slider crank mechanism. The results obtained by using vision system and comparison them with the results obtained by encoder are presented and there is also a discussion part about the results.

Finally, Chapter 9 concludes the thesis with a brief summary and few suggestions for future work.
CHAPTER 2

THE VISION SYSTEM AND CAMERA MODEL [1], [2]

In this chapter, vision systems and camera models are introduced. Also, similarities between human vision systems and computer vision systems, consisting of cameras to reconstruct the environment, are investigated.

2.1 Human Vision and Computer Vision

Descartes removed the eye of an ox and scraped its black part to make it transparent, and then observed the inverted image of a scene from a darkened room on the eyes. (Prienne, 1967). Similar experiments have been performed firstly by Scheiner. He performed experiments firstly with the eyes of sheep and oxen. After that, he performed the same experiment with a human eye in 1625. However, the first person who claimed formation of a converted retinal image is Kepler in 1604. (Polyak, 1957).
Figure 9 illustrates the sections of an eyeball. The iris and the pupil control the amount of light penetrating the eyeball. The cornea and the crystalline lens refract the light to create the retinal image. Retina is the section where the image is formed [2].

In order to receive the depth information, there are some binocular cues (such as stereopsis, eye converge) and monocular cues (such as size and motion parallax). Here, the stereopsis is taken into account. Since the eyes are placed with a distance between them, the same scene view is obtained from slightly different positions and angles. So, the brain can produce a sensation of depth by using these two views (Poggio, 1984).

In the light of human or animal vision systems, computer vision tries to perceive the world in an artificial way. Like humans having two eyes, computer vision systems should view the scene at least with two different angles at the same time. Therefore,
the measurement of depth can be possible and it allows to reconstruct of a 3D scene. In computer vision systems, 2D projections, images, of a real world scene are obtained by using cameras. These images capture two kinds of information, which are geometric and photometric. Geometric information consists of positions, points, lines, curves etc., while photometric information consists of intensity and color. Many tasks, such as recognition and modeling, can be accomplished by using that information.

### 2.2 Pinhole Camera Model

The simplest model of the camera is the pinhole camera model. In this model, the center of projection is the origin of a world coordinate frame. Also, the image plane is placed at a position of focal length. There are two images planes in a pinhole camera model. They are the actual image plane and the virtual image plane. Virtual image plane, which is in front of the pinhole, show us the inverted image. (Figure 10)

![Image](image.png)

*Figure 10: Pinhole Camera Model [2]*
As seen in Figure 11, the principal axis is the line which is perpendicular to the image plane from the center of coordinate system. The principal point is the intersection of the principal axis and the image plane. A line joining a 3D point of a scene and the center of projection intersects the image plane and forms the image point on the 2D image plane. In Figure 11, the 3D point is designated by $\mathbf{M}$ and the image of it is designated by $\mathbf{m}$.
If $\mathbf{M} = [X, Y, Z]$ and $\mathbf{m} = [x, y]$, we can obtain the following relations by using similar triangles seen in Figure 11.

\[
\frac{x}{X} = \frac{f}{z} \\
\frac{y}{Y} = \frac{f}{z} \quad (2.1)
\]

The coordinates of the image point can be obtained as

\[
x = f \frac{x}{z} \\
y = f \frac{y}{z} \quad (2.2)
\]

So;

\[
\mathbf{m} = [f \frac{x}{z}, f \frac{y}{z}] \quad (2.3)
\]

The relation between the world and the images points can be written by homogeneous vectors as

\[
\begin{bmatrix}
  fX \\
  fY \\
  f
\end{bmatrix} =
\begin{bmatrix}
  f & 0 & 0 & 0 \\
  0 & f & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix} \quad (2.4)
\]
2.3 CCD Camera Model

A charge-coupled device (CCD) converts light into electrical charge. A charge-coupled device contains digital imaging elements which are pixels. We can assume that each pixel is a small part of the original image. In a CCD camera model, the coordinates of image points are represented by pixel numbers, while they are represented as coordinates with respect to a world coordinate system in a pinhole camera model. The pixel coordinate system is illustrated in Figure 12.

Figure 12: Pixel Coordinate System
The shape of a pixel can be square, rectangular, or even ellipse. The shape of pixel introduces a scale factor in x and y directions. The scale factor is 1 for square pixels. It is assumed that the cameras which are used in this thesis have square pixels. Hence, their scale factor will be taken to be 1.

Figure 13 shows us a pixel shape and some related parameters. When the pixel is square, $p_x$ and $p_y$ are equal to each other. $\alpha$ is the skew angle, or the skew shortly. It is seen that the skew angle is zero when the pixel shape is square, or rectangular.

Since the coordinates of the image points are represented by pixel dimensions, the focal length of the CCD cameras should also be represented in pixel dimensions. If $\alpha_x$ and $\alpha_y$ are the focal lengths in the x and y directions respectively, then:

\[ \alpha_x = f_{p_x} \quad \text{and} \quad \alpha_y = f_{p_y} \]  

(2.5)
Also, let us designate the principal points, in the x and y directions, as \( x_0 \) and \( y_0 \), respectively. It is noted that the principal points do not have to be on the center point of the image plane.

In the light of these parameters, we can define the camera calibration matrix as:

\[
K = \begin{bmatrix}
\alpha_x & s & x_0 \\
0 & \alpha_y & y_0 \\
0 & 0 & 1
\end{bmatrix}
\] (2.6)

Hence, the homogenous vectors of the normalized image coordinates and the measured image coordinates can be related as:

\[
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix} = \begin{bmatrix}
\alpha_x & s & x_0 \\
0 & \alpha_y & y_0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_n \\
y_n \\
1
\end{bmatrix}
\] (2.7)

where \( x_n \) and \( y_n \) are the normalized image coordinates in the x and y directions, respectively.

2.4 Camera Lens and Distortion

Lens is an important part of a camera. A lens basically blocks the most of the light and selects one ideal light ray coming from a point of the scene object. It is used for recreating the image more accurately on an imaging sensor. A lens can be fixed to the camera or it may be interchangeable depending on the purpose. Hence, lenses may have different focal lengths. Figure 14 shows a thin lens model.
The equation of a thin lens is:

\[
\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}
\]

(2.8)

where \(d_o\) is the distance between the object and the center of the lens, \(d_i\) is the distance between the image and the center of the lens and \(f\) is the focal length of the lens.

Lenses can cause aberrations and distortions because of manufacturing. Therefore, some corrections should be applied to the image coordinates. The most commonly
used correction is for the radial lens distortion which causes the actual image point to be displaced radially in the image plane [6]. The radial distortion can be modelled by using the following equations:

\[
\begin{bmatrix}
\delta u_i^r \\
\delta v_i^r
\end{bmatrix} = \begin{bmatrix}
\hat{u}_i(k_1r_i^2 + k_2r_i^4 + \ldots) \\
\hat{v}_i(k_1r_i^2 + k_2r_i^4 + \ldots)
\end{bmatrix}
\]

(2.9)

where,

\(\delta u_i^r, \delta v_i^r\): infinitesimal radial displacement of \(i^{th}\) point in the x and y directions, respectively

\(k_1, k_2\): the coefficients for radial distortion

\(r_i = (\hat{u}_i^2 + \hat{v}_i^2)^{1/2}\)

Tangential distortion is caused by imperfections in centering of the lenses and manufacturing deficiencies.

The expression for tangential distortion is as following:

\[
\begin{bmatrix}
\delta u_i^t \\
\delta v_i^t
\end{bmatrix} = \begin{bmatrix}
2p_1\hat{u}_i\hat{v}_i + p_2(r_i^2 + 2\hat{u}_i^2) \\
p_1(r_i^2 + 2\hat{v}_i^2) + 2p_2\hat{u}_i\hat{v}_i
\end{bmatrix}
\]

(2.10)

where,

\(\delta u_i^t, \delta v_i^t\): infinitesimal tangential displacement of \(i^{th}\) point in the x and y directions, respectively

\(p_1, p_2\): the coefficients for tangential distortion.

There exists other distortion effects in the literature. For instance, there is a correction term for the cases on which the image axes are not orthogonal (Melen, 1994). The
other one is the thin prism distortion. The reason of thin prism distortion is the
imperfect lens design and manufacturing in addition to assembly. However, this effect
can be included in the radial and tangential distortions. In most cases, distortion effect
is not significant [6]. Also, in the case of radial and tangential distortions, only two
coefficients are enough to compensate.

After mentioning distortion effects, a camera model which includes the pinhole
projection and CCD camera model with distortion effect can be given by the equation:

\[
\begin{bmatrix}
  u_i \\
  v_i
\end{bmatrix}
= \begin{bmatrix}
  D_u s_u (\tilde{u}_i + \delta u^r_i + \delta u^t_i) \\
  D_v (\tilde{v}_i + \delta v^r_i + \delta v^t_i)
\end{bmatrix} + \begin{bmatrix}
  u_0 \\
  v_0
\end{bmatrix}
\] (2.11)

Here,

(u_i, v_i): the corrected coordinates of (\tilde{u}_i,\tilde{v}_i)

(D_u, D_v): the coefficient to convert metric to pixel

s_u: the scale factor

(\delta u^r_i, \delta v^r_i): the radial component of distortion

(\delta u^t_i, \delta v^t_i): the tangential component of distortion

In this study, we will use a CCD camera model and the distortion effects are not taken
into account.

Properties of the cameras are used in this study are given below:

- Model of the camera: Casio® Ex – Z750
- Resolution at video mode: 640 x 480 pixels
- Memory card type: Kinston 2GB.
- Maximum duration of video recording: 1 hour for 2 GB card.
- Frame rate: 30 fps
CHAPTER 3

IMAGE PROCESSING AND MARKER DETECTION

Basic information about images and image processing is given in Appendix A.

3.1 MARKER DETECTION

In order to perform kinematic analysis of a readily available slider crank mechanism [50], various markers are attached on the moving links of the mechanism. The markers that are used are circular shape, with a diameter of 13 mm and red in color. The reason for using red markers is that it is easier to detect the red, green and blue colors in full a color (RGB) image than the other colors.

Since the markers are circular, centers of the circles are tracked. In order to detect the markers, some filtering operations are applied to the images.

3.1.1 Detecting the Red Color

Full color images are stored as an array of M x N x 3. For an RGB image, the red color is represented by array of M x N x 1. Hence, the elements of this array are used to detect the red color. Some colors, such as magenta, can have some red color value in the array of M x N x 1. It is necessary to eliminate these colors as much as possible. For this purpose, the grayscale values of the pixels are subtracted from the red color values of the pixels. At the end of this process, only red, or, reddish objects in the scene are detected. The red objects will be seen to be brighter than the other color objects.
3.1.2 Median Filtering

Median filtering is a nonlinear process. In this thesis, 2D median filtering is performed. Median filtering is mostly used to reduce the salt and pepper noise. An example of the salt and pepper noise is presented in Figure 15.

![Figure 15: Salt and Pepper Noise](image)

Although there is very little salt and pepper noise in the frames of the slider crank mechanism, this filtering process is still performed in order to obtain better results.

3.1.3 Filtering via Threshold

After obtaining the red objects in the scene, the images should be converted into black and white images, which are binary images, in order to detect the markers. We need to specify a threshold while converting the images into binary ones. This threshold is in
the range of $[0, 1]$. The threshold can be chosen around the upper limit for the frames where there is no blur. However, in the case of blurred images (because the motion is too fast), if the threshold is chosen close to the upper limit, data loss can occur. So, for the blurred images, a lower threshold should be preferred. The regions below this threshold will be black and the regions above the threshold will be white. One should also note that the threshold value may be different within the image sequence because of the light conditions of the scene. After applying the threshold, regions other than the ones corresponding to the markers may also appear in the binary image. Therefore, additional filtering operations are necessary.

3.1.4 Region Labelling

Labelling helps us to distinguish the white regions from each other. In this filtering, black regions remain as 0, while white regions are labelled as 1, 2 and so on. This process is performed by means of either 4 – connectivity or 8 – connectivity object. The difference between the 4 and 8 – connectivity objects can be seen in Figure 16.

![Figure 16: 4 - connected and 8 - connected pixels](image)

Figure 16: 4 - connected and 8 - connected pixels
3.1.5 Filtering via Area

This filtering process helps us to remove the regions which are out of the range where range is defined by the user. The area of markers (in terms of pixels) can be determined manually. Here, the user should define a range such that the area of the region is in the range \([\text{areamin}, \text{areamax}]\). At the end of the filtering, the white regions which are within the range will remain unchanged while other regions turn into black.

3.1.6 Detection of Marker and Center of Marker

The markers used are circular in shape. This feature helps us to find the markers and their centers. MATLAB® has a function which detects circular shapes. This function is “imfindcircles”. It requires a radius range in the form \([\text{radiimin}, \text{radiimax}]\). One should note that this function also gives the center points of the circular shapes. The circles which are out of this range are not detected. However, in the case of blurred images, it is very hard to detect circular shapes (corresponding to the markers). For this reason, instead of detecting the circular shapes, the centroid of the blurred area is determined to represent the center of the marker.

After all these processes are performed, there may still be some regions which resemble marker. Such regions should be eliminated manually.

One can summarize the processes as follows.

1) Detection of the red color objects
2) 2D Median filtering (“medfilt2” function of MATLAB® can be used.)
3) Threshold filtering to convert the image into a binary image.
4) Labeling the regions of the image (“bwlabel” function of MATLAB® can be used.)
5) Area filtering (“bwareaopen” function of MATLAB® can be used.)
6) Detection of the circular regions and their centers, or the centroids of the blurred areas (“imfindcircles” function, or “regionprops” function of MATLAB® can be used.)
These operations for one image are illustrated in Figure 17.

Figure 17: Sample Stages of Marker Detection
CHAPTER 4

FUNDAMENTAL MATRIX

In this part, a brief information about the fundamental matrix is given. The fundamental matrix is based on the epipolar geometry concept. If the intrinsic parameters of a camera are not known, epipolar geometry is represented by the fundamental matrix. Otherwise, epipolar geometry is represented by the essential matrix.

4.1 Epipolar Geometry

The epipolar geometry between two views is essentially the geometry of the intersection of the image planes with the pencil of planes having the baseline as an axis [1]. Here, the baseline is a line joining the center of the cameras. (See Figure 18).
Suppose that a 3D point is projected on the two image planes. Assume that the image point of this point, which is shown as $X$, is $x$ on the first image plane and $x'$ on the second image plane. In Figure 19, it is seen that the rays back-projected from $x$ and $x'$ intersect at the point $X$.

![Figure 19: Projection of a 3D Point in Two Image Planes](image)

In Figure 19, $C$ and $C'$ represent the camera centers, or the projection centers. Clearly, the rays back-projected from the image points $x$ and $x'$ and the baseline form the epipolar plane, $\pi$.

Suppose that only $x$ is known. So, the epipolar plane $\pi$ can be found by the ray back-projected from $x$ and the baseline. $x'$ lies in the plane $\pi$. Another feature that can help
one to find \( x' \) is the second image plane. As it can be seen in Figure 20, the epipolar plane \( \pi \) intersects with the second image plane on a line.

In Figure 20, it is seen that the intersection of the epipolar plane \( \pi \) and the second image plane is the line \( l' \). This line is called the epipolar line on the second image plane. Therefore, line \( l' \) helps one in the search for the corresponding point. Instead of searching the entire image plane, one can search only on the line \( l' \) (to find the corresponding point on the second image plane). So, the epipolar constraint can be described as follows:

For an image point in the first image plane, the corresponding point \( x' \) on the second image plane lies on the line \( l' \) and similarly for an image point \( x' \) in the second image plane, the corresponding point \( x \) on the first image plane lies on the line \( l \).
Considering Figure 20, point e represents the image point of the second camera center and point e’ represents the image point of the first camera center. These points are known as the epipolar points, or the epipoles. Epipoles are the intersection of all epipolar lines and there is only one epipole for each image. To sum it up, epipole is the point that shows image of other camera center and it is intersection point of baseline with image plane. Epipolar plane is a plane containing the baseline and the ray back-projected from the image point. Epipolar line is the intersection of the epipolar plane with the image plane.

### 4.2 Fundamental Matrix

The fundamental matrix is the algebraic representation of the epipolar geometry. It defines a map from a point in one image plane to its corresponding line in the other image plane. The definition of the fundamental matrix $F$ is given as:

$$l' = Fx$$

$$l = F^T x'$$

(4.1)

As it is known, the point $x$ lies on the epipolar line $l$ and the point $x'$ lies on the epipolar line $l'$, leading to the relations:

$$x'^T l' = 0$$

$$x^T l = 0$$

(4.2)

Using the equations (4.1) and (4.2), one obtains:

$$x'^T F x = 0$$
Consider, now, a plane $\pi$ which is not passing through either of the two camera centers as shown in Figure 21.

The ray back projected from $x$ in the first image plane intersects the plane $\pi$ at the point $X$. Then point $X$ is projected onto the second image plane yielding point $x'$. This procedure is known as the point transfer via plane. Thus, there is a 2D homography, $H_\pi$, mapping each $x$ to $x'$. This relation may be represented via the equation:

$$x^T F x' = 0 \quad (4.3)$$
\[ x' = H_\pi x \quad (4.4) \]

In addition to that, since all epipolar lines pass through epipoles, one can write down the equations of the epipolar lines as follows:

\[ l' = e'x x' = [e']_x x' \quad (4.5) \]

where \([e']_x\) is the skew symmetric matrix of epipole.

By using equations (4.4) and (4.5), one obtains:

\[ l' = [e']_x H_\pi x \quad (4.6) \]

Now, combining equations (4.6) and (4.1),

\[ l' = [e']_x H_\pi x = Fx \quad (4.7) \]

Thus, the fundamental matrix can be obtained as:

\[ F = [e']_x H_\pi \quad (4.8) \]

Since fundamental matrix represents a mapping from the 2 dimensional projective plane onto the 1 dimensional epipolar lines, the rank of it is 2.
Now, let the camera projection matrices of the two camera are $P$ and $P'$. Therefore, one can write down the following relations:

$$x = PX$$

$$x' = P'X$$  \hspace{1cm} (4.9)$$

leading to

$$X = P^+x = P'^+x'$$  \hspace{1cm} (4.10)$$

where $P^+$ and $P'^+$ designate the pseudo-inverse of $P$ and $P'$ matrices

The null vector of the camera projection matrix is the center of camera. Hence,

$$PC = 0$$  \hspace{1cm} (4.11)$$

The image of the camera center of the first image plane in the second plane is given by

$$e' = P'C$$  \hspace{1cm} (4.12)$$

And by eliminating 3D point $X$ in equation (4.10), one obtains

$$x' = P'P^+x$$  \hspace{1cm} (4.13)$$

By combining equations (4.5), (4.12) and (4.13), one obtains:
\[ l' = (P' C) x (P' P^+ x) = [e'] x (P' P^+ x) = F x \]

Therefore, F is obtained to be

\[ F = [e'] x P' P^+ \] (4.14)

The fundamental matrix F has seven degrees of freedom. It is a 3x3 matrix defined up to a scale. So, there are nine elements and the common scaling is insignificant which leads to eight independent ratios. In addition to that, det [F] = 0 which removes one more degree of freedom since the rank of it is 2 [1].

4.3 Computation of the Fundamental Matrix

In this thesis, 3 methods are given for the computation of the fundamental matrix. Referring to equation (4.3), the fundamental matrix is defined via the equation

\[ x'^T F x = 0 \]

This equation can be used to compute the fundamental matrix. Assume that x and x' are two corresponding points in the two images and \( x = (x, y, 1)^T \) and \( x' = (x', y', 1)^T \) are the homogeneous coordinates of the image points. Furthermore, let the fundamental matrix be represented as:

\[
F = \begin{bmatrix}
f_{11} & f_{12} & f_{13} \\
f_{21} & f_{22} & f_{23} \\
f_{31} & f_{32} & f_{33}
\end{bmatrix}
\] (4.15)
where $f_{ij}$ is the $j^{th}$ element of the $i^{th}$ row of the fundamental matrix.

Each given pair of matching points gives one linear equation in the unknown elements of $F$. One can obtain this linear equation as:

$$x'x f_{11} + x'y f_{12} + x'f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0$$

(4.16)

If one represents the elements of the $F$ matrix as a row vector, in row-major order, he/she can rewrite equation (4.16) as:

$$(x'x, x'y, x', y'x, y'y, x', y', x, y, 1)^T f = 0$$

(4.17)

where $f$ is the vector form of the fundamental matrix.

For a given set of $n$ matching points, equation (4.17) can be written in matrix form, yielding,

$$
\begin{bmatrix}
    x_1'x_1 & x_1'y_1 & x_1'y_1 & y_1'x_1 & y_1'y_1 & x_1 & y_1 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    x_n'x_n & x_n'y_n & x_n'y_n & y_n'x_n & y_n'y_n & x_n & y_n & 1 \\
\end{bmatrix} f = Af = 0
$$

(4.18)

Clearly, the equation (4.18) represents a homogenous set of linear equations. Hence, vector $f$ can be determined only up to a scale. In order to find a solution, the rank of
matrix A must be at most 8. In case the rank of A is exactly 8, the solution is unique up to a scale factor and can be found by using linear methods [1].

In practice, point coordinates involve noises and the rank of matrix A may be 9 since matrix A has 9 columns. In this case, the solution for the vector f can be found in the least-square sense. The singular value decomposition technique can be used to find least-square solution. The details of the singular value decomposition is given in Appendix B. The least square solution is the singular vector corresponding to the smallest singular value of A which is the last column of V where A = UDVᵀ, where U and V are real, or complex unitary matrices and D is the rectangular diagonal matrix with non-negative real numbers.

4.3.1 The Normalized 8 – Point Algorithm [1]

The 8 – point algorithm is the simplest method of computing the fundamental matrix, involving no more than the construction and solution of a set of linear equations [1]. The 8 – point algorithm is introduced by Longuet Higgins, in 1981. Hartley improved this technique by using the normalization method [1].

Suppose that the pair of corresponding points x and x’ are in the form of (100, 100, 1)ᵀ. It is seen that the x and y coordinates of point, which are 100, are much higher than 1. Therefore, the corresponding A matrix will be in the form of A = (10⁴, 10⁴, 10², 10⁴, 10⁴, 10², 10², 10², 1). Increasing the term 1 of matrix A by 100 means a huge change in image points whereas increasing the term 10⁴ of matrix A by 100 means only a slight change. That is why all the entries of matrix A should have similar magnitudes [2].

Briefly, normalization can be done by applying the following steps:

1. The points are translated so that their centroid is at the origin.
2. The points are scaled so that RMS distance from the origin is equal to $\sqrt{2}$.
3. This transformation process should be applied to the images independently.

As can be anticipated from the name of algorithm, 8 pair of matching points are necessary in order to implement the normalized 8–point algorithm. After normalizing the points, a linear solution can be found by using equation (4.18). After that, one needs to make sure that fundamental matrix satisfies the singularity constraint, because the rank of the fundamental matrix must be 2. This singularity constraint enforcement can be provided by using the singular value decomposition (SVD). Suppose that the linear solution for the fundamental matrix is $F'$.

Let $F' = UDV^T$ be SVD of $F'$.

If noise is presented, $D$ will be the diagonal matrix $D = \text{diag} (r, s, t)$ where $r \geq s \geq t$. In order to satisfy the singularity constraint, the last diagonal element, $t$, must be 0. So, $t$ is replaced by 0 in the diagonal matrix $D$ and the new $F'''$ can be found via the equation $F''' = U\text{diag}(r, s, 0)V^T$. After finding $F'''$, the correct fundamental matrix $F$ can be found by applying denormalization.

The steps of the normalized 8–point algorithm are summarized below.

1. Normalization: Transform the images coordinates to the normalized coordinates via the equation $x_n = Tx$ and $x_n' = T'x'$ where $x_n$ and $x_n'$ are normalized image point coordinates.

Centroid of points for $n$ set of image points:

$$m_n = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$m_n' = \frac{1}{n} \sum_{i=1}^{n} x_i'$$

Let $m_n = (m_1, m_2, 1)^T$ and $m_n' = (m'_1, m'_2, 1)^T$
Define \( s = \left( \frac{1}{2n} \sum_{i=1}^{n} (x_i - m_1)^2 + (y_i - m_2)^2 \right)^{1/2} \) and
\[
s' = \left( \frac{1}{2n} \sum_{i=1}^{n} (x'_i - m'_1)^2 + (y'_i - m'_2)^2 \right)^{1/2}
\]
for image points \( x_i = (x_i, y_i)^T \) and \( x'_i = (x'_i, y'_i)^T \) on the first and second image planes.

Therefore, \( T \) and \( T' \) can be written as:
\[
T = \begin{bmatrix}
s^{-1} & 0 & -s^{-1}m_1 \\
0 & s^{-1} & -s^{-1}m_2 \\
0 & 0 & 1
\end{bmatrix}
\]
\[
T' = \begin{bmatrix}
s'^{-1} & 0 & -s'^{-1}m'_1 \\
0 & s'^{-1} & -s'^{-1}m'_2 \\
0 & 0 & 1
\end{bmatrix}
\]

2. Linear Solution: Find a linear solution for \( F' \) by using equation (4.18)
3. Singularity Constraint: Find \( F'' \) to enforce \( \det(F') = 0 \) by using the SVD.
4. Denormalization: Find the correct \( F \) by denormalizing \( F'' \) such that \( F = T'TF''T \).

### 4.3.2 The Algebraic Minimization Algorithm

The advantage of the normalized 8-point algorithm is that it is simple and rapid. However, this method is not optimal numerically, because all the entries of fundamental matrix have different importance. So, an alternative method can be used to find the singular matrix \( F' \) which minimizes the \( \|Af'\| \) subject to \( \|f'\| = 1 \).

The fundamental matrix may be written as \( F = M[e]_x \), [1]. Here, \( M \) is a non–singular 3x3 matrix and \([e]_x \) is the skew–symmetric matrix of epipole in the first image. Suppose that epipole \( e \) is known and the fundamental matrix \( F \) is to be found. One can write the equation \( F = M[e]_x \) in term of the vectors of \( f \) and \( m \) (which are the vector form of \( F \) and \( M \) in row major order) as \( f = Em \). Here, \( E \) is a 9x9 matrix given by
\[
E = \begin{bmatrix}
[e]_x & 0 & 0 \\
0 & [e]_x & 0 \\
0 & 0 & [e]_x
\end{bmatrix}
\]

Therefore, the minimization problem becomes:

Minimize \(|AEml|\) subject to the condition \(|Em| = 1\).

This minimization problem can be solved by using the more constrained minimization algorithm [1]. The more constrained minimization algorithm is given in Appendix C. Epipole \(e\) will be varied to realize the minimization. The initial value of epipole \(e\) can be found as the null vector of the fundamental matrix which is found by using the normalized 8 – point algorithm. Hence, one obtains

\[Fe = 0 \quad (4.22)\]

Also, the Levenberg – Marquardt algorithm can be used to vary the epipole \(e\).

The algebraic minimization algorithm is summarized below.

1. Find an approximate fundamental matrix \(F_0\) by using the normalized 8 – point algorithm
2. Find the epipole in the first image as the right null vector of \(F_0\).
3. Staring with the initial value of epipole \(e\) and computing \(E\), vary \(e\) by using the Levenberg – Marquardt algorithm and minimize the algebraic error \(|Af|\) with \(f\) which is calculated by the more constrained minimization algorithm.
4. In minimization, converging \(f\) represent the fundamental matrix which is desired.
4.3.3 Geometric Distance Algorithm

In this method, a minimization of a geometric image distance is implemented. Here, the gold standard method, which is suggested by Hartley [1], is given as the geometric distance algorithm. Geometric distance algorithms are important when there is noise during experiments. According to Hartley [1], other algorithms yield accurate results and are easier to implement, however, they are not optimal when the image error is Gaussian. In order to implement the geometric distance minimization algorithm, an initial value of parameters for non–linear minimization and a parameterization of cost function are necessary. The initial value for the non–linear minimization can be obtained by using one of the linear methods.

4.3.3.1 The Gold Standard Method

In this method, it is assumed that there is noise involved in the image point according to the Gaussian distribution. So, one tries to minimize the objective function $g$

$$g = \sum_i {d(x_i, \hat{x}_i)^2 + d(x_i', \hat{x}_i')^2} \quad (4.23)$$

Here, $d(\cdot)$: the geometric distance between points

$X_i$ and $x_i'$: the measured coordinates

$\hat{x}_i$ and $\hat{x}_i'$: the estimated true coordinates which satisfy $\hat{x}_i^T F \hat{x}_i = 0$ where $F$ is the estimated fundamental matrix

While minimizing the error function, a pair of camera matrices can be defined such that $P = [I|0]$ and $P' = [e'|x_F|e']$ where $I$ is the 3x3 identity matrix and 0 is 3x1 zero matrix. So, we can write
\[ \dot{x}_i = PX_i \]
\[ \dot{x}_i' = P'X_i \]  \hspace{1cm} (4.24)

Therefore, one can vary \( P' \) and \( X_i \) to minimize the equation (4.23) by using the Levenberg–Marquardt algorithm. \( X_i \) can be obtained by using triangulation method which will be explained later.

In addition to that, Sampson distance minimization may be used. In Sampson distance, the cost function \( p \) can be written as:

\[
 p = \sum_i \frac{(x_i^T F x_i)^2}{(F x_i)_1^2 + (F x_i)_2^2 + (F^T x_i)_1^2 + (F^T x_i)_2^2} \]  \hspace{1cm} (4.25)

Here, \( (\ )_i \) is the \( i^{th} \) component of a 3–vector.

The gold standard algorithm is summarized below.

1. Compute the initial fundamental matrix by using linear method.
2. Assume the camera matrices to be \( P = [I|0] \) and \( P' = [\varepsilon' F | \varepsilon'] \)
3. Estimate \( X_i \) by using the triangulation method such that \( \dot{x}_i = PX_i \) and \( \dot{x}_i' = P'X_i \).
4. Minimize the cost function given by equation (4.23)

### 4.3.3.2 Parameterization of the Fundamental Matrix

In order to implement the non–linear minimization method, the fundamental matrix must be parameterized. While parameterizing of the fundamental matrix, the following relations will be helpful:
Parameterization can be done by using only one epipole, or by using two epipoles. In
the one epipole parameterization technique, one notes that \( f_3 = a f_1 + b f_2 \), where \( f_i \)
represents the \( i \)th column of the fundamental matrix and \( a, b \) are two multipliers. This
expression can be written because rank of the fundamental matrix is 2. So, the
fundamental matrix can be represented as:

\[
F = \begin{bmatrix}
    f_{11} & f_{12} & af_{11} + bf_{12} \\
    f_{21} & f_{22} & af_{21} + bf_{22} \\
    f_{31} & f_{32} & af_{31} + bf_{32}
\end{bmatrix}
\]  

(4.28)

Clearly, there are 8 parameters which are \( f_{11}, f_{12}, f_{21}, f_{22}, f_{31}, f_{32}, a, b \).

The main disadvantage of this parameterization method is that it cannot be used when
the first two columns of the fundamental matrix are linearly dependent. In this case,
the third column of the fundamental matrix cannot be written in terms of the first two
columns. Therefore, other column (other than third one) can be written in terms of the
remaining columns. For instance, the first column of the fundamental matrix can be
expressed in terms of the second and the third columns. In addition to that, the first
two columns of the fundamental matrix are linearly dependent when the epipole is at
infinity such that \( e = (e_1, e_2, 0)^T \). One has to define the multipliers \( a \) and \( b \) as well.
These multipliers can be defined such that \( e = (a, b, -1)^T \) which is the right epipole of
the fundamental matrix. For best results, the parameterization should be done such that
the largest entry in absolute value is 1, [1].
The second method uses two epipoles for the parameterization. Suppose that the two epipoles are given by \((a, b, -1)\)\(^T\) and \((a', b', -1)\)\(^T\). So, the fundamental matrix can be expressed as:

\[
F = \begin{bmatrix}
    f_{11} & f_{12} & af_{11} + bf_{12} \\
    f_{21} & f_{22} & af_{21} + bf_{22} \\
    a'f_{11} + b'f_{21} & a'f_{12} + b'f_{22} & a'a'f_{11} + a'b'f_{12} + b'a'f_{21} + b'b'f_{22}
\end{bmatrix}
\]

\[(4.29)\]

In addition, one may arbitrarily set one of the independent parameters to 1 in order to achieve a minimum set of parameters for both the one epipole and two epipoles parameterizations.

### 4.3.4 Computation of the Fundamental Matrix by the Using RANSAC Algorithm

While computing the fundamental matrix, a set of matching points is used. However, there can be wrong matching points which are outliers. In order to obtain best results for the fundamental matrix, the outliers should be removed from the set. RANSAC algorithm is suitable for this task. RANSAC is an algorithm for robust model fitting by selecting a minimum sample set required for the model. So, the correct fundamental matrix can be found by using a subset which does not contain any outliers. In this thesis, Matlab\(^\text{®}\) is used to find the fundamental matrix during this thesis since it has a function, “estimateFundamentalMatrix”, to calculate the fundamental matrix by using RANSAC.
4.4 Essential Matrix

Essential matrix is a form of the fundamental matrix. The essential matrix is valid when the cameras are calibrated, i.e., when the intrinsic parameters of the cameras are known. Essential matrix was firstly introduced by Longuet and Higgins. The essential matrix is defined via the equation:

\[ x_n^T E x_n = 0 \]  \hspace{1cm} (4.30)

where \( x_n \) and \( x_n' \) are the normalized coordinates of the image points and \( E \) denotes the essential matrix. The normalized coordinates can be found by using the calibration matrices of the cameras and the following expressions.

\[ x_n = K^{-1} x \]
\[ x_n' = K'^{-1} x' \]  \hspace{1cm} (4.31)

By combining equations (4.3), (4.30) and (4.31), one can express the essential matrix in terms of the fundamental matrix and calibration matrices, yielding

\[ E = K'^{-T} F K \]  \hspace{1cm} (4.32)

The essential matrix can also be expressed in terms of the rotation matrix \( R \) and the translation vector \( t \), yielding

\[ E = [t]_x R = R[R^T t]_x \]  \hspace{1cm} (4.33)

where \([ ]_x\) represent the skew symmetric matrix.
Hence, the essential matrix has five degrees of freedom. There exists three degrees of freedom associated with the rotation and three degrees of freedom associated with the translation yielding six degrees of freedom all together. However, there is a scale ambiguity since the essential matrix is a homogeneous quantity. In addition to that, the essential matrix has two nonzero and equal eigenvalues, whereas the third eigenvalue is zero.
In this chapter, Bouguet camera calibration toolbox of MATLAB® is discussed. The positions of two cameras are assumed to be fixed with respect to each other. Therefore, the extrinsic parameters of the two cameras are taken from the results of the Bouguet camera calibration toolbox. The calibration is performed via the technique specified in [3], [4]. Hence, calibration is performed by using a planar calibration pattern. The calibration pattern should be viewed by each camera at various, different orientations (Figure 22).

Figure 22: Calibration Pattern Viewed at Different Orientations
5.1 Functions of the Bouguet Camera Calibration Toolbox

5.1.1 Reading Images

After capturing the calibration pattern at different orientations, the first step is to store the selected images into the calibration toolbox. In this process, the type of the image (i.e., whether the image is full color, or binary, or grayscale) is not important. The toolbox will read the images and automatically convert them to grayscale images.

5.1.2 Extract Grid Corners

This function helps one to specify the corresponding points in different views. The corresponding points are chosen from the corner points of the calibration pattern. The Figure 23 shows an example of the calibration pattern that is used for the Bouguet camera calibration toolbox.
Only 4 points, corresponding to the corner points of the calibration pattern, are chosen. These 4 points should form a rectangular shape on the image. This rectangular shape is shown in Figure 24. The points which are shown by circles are the points chosen by the user.
After specifying the rectangular shape on each image, the toolbox finds the corresponding points in the different views automatically. These corresponding points are chosen such that they exist inside of the rectangular shape. One should specify that the first clicked point is chosen to be the origin. In Figure 24, the origin is specified with O. After finding the corresponding points, one must enter the length of an edge of the square on the calibration pattern. This length helps one to find a relation between the world points and the image points. The origin is taken as the point (0, 0) on the pattern and the coordinates of remaining points are found depending on the dimension of the square.
Figure 25 shows the last form of the image after extracting the corresponding points. At the end of the extracting corner point process, a set of detected 2D points and a set of detected 3D points are obtained. These two sets will be used later during the calibration procedure.


5.1.3 Calibration

In the calibration step, the intrinsic and the extrinsic parameters are obtained by using the sets of detected 2D and 3D points. Calibration is implemented in two steps. First one is the initialization. The initialization step gives a closed form solution for all parameters except for the lens distortion parameters. The projection equation for the Bouguet camera toolbox is given below.

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  1
\end{bmatrix} = \begin{bmatrix}
  r_1 & r_2 & r_3 & t
\end{bmatrix} \begin{bmatrix}
  X \\
  Y \\
  0 \\
  1
\end{bmatrix} = \begin{bmatrix}
  r_1 & r_2 & t
\end{bmatrix} \begin{bmatrix}
  X \\
  Y \\
  1
\end{bmatrix} = H \begin{bmatrix}
  X \\
  Y \\
  1
\end{bmatrix} \tag{5.1}
\]

Here,  

- \( s \) : the scalar parameter  
- \( x_1 \) and \( x_2 \) : the image coordinates in horizontal and vertical directions  
- \( K \) : the calibration matrix  
- \( r_1, r_2 \) and \( r_3 \) : the columns of rotation matrix  
- \( t \) : the translation vector  
- \( X, Y \) : the coordinates of 3D point  
- \( H \) : the homography matrix between 2D points and 3D points

Clearly, the coordinates of the 3D points in the Z direction are zero. Since the 3D coordinate frame is chosen such that each point on the calibration pattern has its z coordinate to be zero.

Since \( r_1 \) and \( r_2 \) are orthogonal, one can write the following two constraints:
\[ h_1^T K^{-T} K^{-1} h_2 = 0 \]
\[ h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2 \]  \quad (5.2)

As discussed before, \( K^{-T} K^{-1} \) describes the image of the absolute conic and it is a symmetric matrix. So, it can be represented by a 6 x 1 vector. Let us describe a matrix 
\( A = K^{-T} K^{-1} \) and matrix 
\( H = [h_1 \ h_2 \ h_3] \) where 
\( h_1 = (h_{11} \ h_{12} \ h_{13})^T \) versa. Therefore, the left hand side of equation (5.2) can be written as:

\[ h_1^T A h_2 = v_{12}^T a \]
\[ h_1^T A h_1 - h_2^T A h_2 = (v_{11} - v_{22})^T a \]  \quad (5.3)

where \( a \): the 6 x 1 vector consisting of the elements of matrix \( A \).

\[ v_{ij} = 
\begin{bmatrix}
  h_{i1} h_{j1} & h_{i1} h_{j2} + h_{i2} h_{j1} & h_{i2} h_{j2} & h_{i3} h_{j1} + h_{i1} h_{j3} & h_{i3} h_{j2} + h_{i2} h_{j3} & h_{i3} h_{j3}
\end{bmatrix} \]

Therefore, equation (5.3) can be expressed as:

\[
\begin{bmatrix}
  v_{12}^T \\
  (v_{11} - v_{22})^T
\end{bmatrix} b = Vb = 0
\]  \quad (5.4)

If we have \( N \) images, matrix \( V \) given in equation (5.4) is a 2N x 6 matrix. \( b \) is calculated by using the SVD of matrix \( V \) and calibration matrix \( K \) is found by using the symmetric matrix \( B \) as well as scalar value. In addition to that, the extrinsic parameters are found by using the camera calibration matrix and the columns of homography matrix \( H \) via the equations
\[ r_1 = sK^{-1}h_1 \]
\[ r_2 = sK^{-1}h_2 \]
\[ r_3 = r_1 \times r_2 \]
\[ t = sK^{-1}h_3 \]  \hspace{1cm} (5.5)

After finding the estimated values of the intrinsic and the extrinsic parameters, the radial distortion coefficients are solved by using the linear least–square techniques. Determination of the distortion coefficients will not be discussed here. If anyone is interested in the distortion coefficients, he/she can refer to references [3], [4]. The values found for the parameters are used as initialization for second step which is nonlinear optimization. This optimization is performed via the maximum likelihood inference. Assume that the image points are corrupted by distributed noise and we are given N number of images with m number of points on the calibration pattern. So, the maximum likelihood estimate can be obtained by minimizing the following expression:

\[ r = \min \sum_{n=1}^{N} \sum_{m=1}^{m} \|x_{ij} - \mathbf{x}(K, R_i, t_i, X_j)\|^2 \]  \hspace{1cm} (5.6)

Here, \( x_{ij} \) : the image point of the 3D point \( X_j \) in the image plane of the \( i^{th} \) camera
\( \mathbf{x} \) : the projection point found by the parameters which are calculated before

After optimization, the results are displayed as shown in Figure 26.
5.1.4 Show Extrinsic

This function provides visualization of the estimated extrinsic parameters. So, it shows the position of the cameras and the calibration pattern at different orientations (with

Figure 26: Example of Output for Bouguet Camera Calibration Toolbox

Figure 27: Visualization of Estimated Extrinsic Parameters
respect to the fixed camera or the position of the camera with respect to the fixed
 calibration pattern). Figure 27 shows an example of such a visualization.

5.1.5 Stereo Calibration

Bouguet camera calibration toolbox has a part that calculates the position of a camera,
with respect to the other, when two cameras are used. It should be noted that the
cameras should be synchronized. That means the views of two cameras should be taken
at the same time. In this thesis, the synchronization is done by using a flashlight. While
the two cameras are in the video mode, the flashlight is turned on and off. The two
frames on which the light is seen firstly are the corresponding frames of the two
cameras. In the same way, the two frames on which the light is off firstly are
corresponding frames of the two cameras. Then, the toolbox can calculate the position
of the cameras with respect to each other. At the end, the solution is displayed as seen
in Figure 28.

![Figure 28: Example of Stereo Calibration Results](image)

Intrinsic parameters of left camera:

- Focal length
- Principal point
- Imaging matrix

Intrinsic parameters of right camera:

- Focal length
- Principal point
- Imaging matrix

Extrinsic parameters (position of right camera wrt left camera):

- Rotation vector
- Translation vector

Note: The numerical errors are approximately three times the standard deviations (for reference).
In Figure 28, om and T vectors represent the rotation and translation vector of the right camera with respect to the left. Let the coordinates of a 3D point be $X_L$ in the left camera coordinate frame and $X_R$ in the right camera coordinate frame. Hence, one can write the following expression:

$$X_R = R \times X_L + T$$  \hspace{1cm} (5.7)

Here, $R$ is the rotation matrix. This rotation matrix $R$ can be obtained by using “Rodrigues” function which exists in the toolbox, so $R = \text{rodrigues}(om)$. The visualization of the estimated stereo extrinsic parameters can be seen in Figure 29.

Figure 29: Visualization of the Stereo Parameters
During stereo calibration, all intrinsic and extrinsic parameters are recomputed. One may observe that the uncertainties of the intrinsic parameters are smaller after stereo calibration. The reason is that stereo optimization is realized globally and it is implemented over a minimal set of unknown parameters.

As stated before, the position vectors of the cameras, with respect to each other, are determined by using the Bouguet camera calibration toolbox. Recall that we assume cameras are fixed with respect to each other and their positions are known.
CHAPTER 6

SELF – CALIBRATION

6.1 Kruppa Equations

The usage of the Kruppa equations in computer vision was first introduced by Faugeras, Luong and Maybank [29]. In order to implement the Kruppa equations, one needs to know the fundamental matrix and the two independent quadric equations of the dual image of absolute conic. Firstly, the absolute conic are discussed, followed by the Kruppa equations in this chapter.

6.1.1 Absolute Conic

One of the most important concepts for camera self – calibration is the absolute conic and image of the absolute conic. Basically, the absolute conic, $\Omega_\infty$, is a point conic in the plane at infinity. In a metric frame, if a point $X = [x_1, x_2, x_3, x_4]^T$, in homogeneous form, exists on the absolute conic, it must satisfy the following relation:

$$x_1^2 + x_2^2 + x_3^2 = x_4 = 0 \quad (6.1)$$

Recall that if a point is at infinity, this point must satisfy $x_4 = 0$ in homogeneous form.

The most important characteristic of the absolute conic is that the absolute conic is invariant under Euclidean transformations, i.e., its image is independent of camera pose. The moon can be given as an example for the absolute conic. When someone drives on a straight road, or walk on a straight road, he/she has an impression that
moon is following him. Absolute quadric is more general, in the sense that its image is invariant under translations and rotations.

It has been stated that the image of the absolute conic is invariant under Euclidean transformations. On the other hand, it has been shown in the previous chapters that an image does not depend on the camera pose only but also on the internal parameters of the camera, such as the focal length, or principal point. Therefore, the image of the absolute conic does not depend on the camera pose, i.e., it depends on the intrinsic parameters of the camera. Hence, if the intrinsic parameters of a camera are constant, then the image of the absolute conic is constant, too.

If a point $X$ is on the absolute quadric, the equation of it can be written as:

$$X^T X = 0 \quad (6.2)$$

Assume that there is a camera whose camera matrix is $P$ and whose internal matrix is $K$. Since the projection of the absolute conic does not depend on the camera pose, we can write the image of a point as:

$$u = KX \quad (6.3)$$

where $u$ is the image point. Solving $X$ from equation (6.3), one obtains

$$X = K^{-1}u \quad (6.4)$$

By inserting equation (6.4) into equation (6.2), one obtains
\[ u^T K^{-1} K^{-1} u = 0 \]

Hence, the image of the absolute conic is obtained to be

\[ w = K^{-1} K^{-1} \]

where, \( w \) represents the image of the absolute conic.

Figure 30: Absolute Conic and Its Image
6.1.2 Absolute Dual Quadric

The absolute dual quadric, $Q^*_\infty$, is the dual of the absolute conic and it is a degenerate dual quadric in 3 – space. The absolute dual quadric is formed by planes which are tangent to the absolute conic. Therefore, the absolute conic can be called as the rim of the absolute dual quadric, or simply rim quadric. Algebraically, the absolute dual quadric is a symmetric 4 x 4 homogeneous matrix which has a rank of 3. It can be shown as:

$$Q^*_\infty = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$$ (6.5)

Where, $I$ is the 3 x 3 identity matrix and $0$ is the 3 x 1 zero vector.

Like the absolute conic, the absolute dual quadric is also invariant under Euclidean transformations. The reason for preferring the absolute dual quadric is that it is much simpler to project it into the image plane compared to the absolute conic.

Assume that a plane is represented by $\pi = (v^T, k)^T$. If this plane is in the envelope of $Q^*_\infty$, it should satisfy that

$$\pi^T Q^*_\infty \pi = 0$$ (6.6)

From equations (6.5) and (6.6) with $\pi = (v^T, k)^T$, one obtains

$$v^Tv = 0$$ (6.7)

where $v$ is a 3 x 1 vector.
Hence, \( v \) represents the line which is the intersection of plane \( \pi \) and the plane at infinity. So, this line is tangent to the absolute conic if and only if equation (6.7) is satisfied. This implies that the absolute dual quadric is made up of the planes which are tangent to the absolute conic [1].

The image of the absolute dual conic is the inverse of the image of the absolute conic. Hence, one obtains

\[
w^* = \text{inverse}(w) = KK^T \tag{6.8}
\]

After finding the absolute dual quadric or absolute conic, the camera calibration matrix, \( K \), can be found by using Choleski factorization.

![Figure 31: Absolute Conic and Absolute Dual Quadric](image-url)
6.2 Self – Calibration with the Kruppa Equations

6.2.1 Problem Statement

Suppose that we have N numbers of cameras and each of these cameras has an associated camera projection matrix designated by $P_i$, where $i = 1, \ldots, N$. Furthermore, M number of 3D points with coordinates $X_j$, where $j = 1, \ldots, M$. Using the projection matrices, one may write down the equation

\[ x^i_j = P_i X_j \]  \hspace{1cm} (6.9)

where, $x^i_j$ is the image point of $X_j$ in the $i^{th}$ image plane.

The aim is to determine the metric reconstruction from the uncalibrated image points by determining the camera parameters and by using the triangulation method. Assume that the projective reconstruction is known. In order to upgrade the projective reconstruction to the metric reconstruction, a rectifying homography matrix $H$ should be determined. After determining the $H$ matrix, the projective reconstruction can be converted to the metric one via the equations:

\[ P_{M_i} = P_i H \]

\[ X_{M_j} = H^{-1} X_j \]  \hspace{1cm} (6.10)

The subscript $M$ indicates that the camera matrix $P$ is in the form of metric reconstruction. In equation (6.10), the $H$ matrix can be written as:

\[ H = \begin{bmatrix} A & t \\ k^T & 1 \end{bmatrix} \]  \hspace{1cm} (6.11)
where, A is the 3 x 3 matrix, t and v are the 3 x 1 vector and k is the scalar.

It should be noted that the obtained metric reconstruction is a scaled version of the real scene. Assume that the world coordinate frame coincides with the first camera frame. So, the rotation matrix $R_1$ for the first camera is a 3x3 identity matrix $I$ and the translation vector $t$ is a 3x1 zero vector. Therefore, the first camera matrix $P_{M1}$ can be written as:

$$P_{M1} = K_1[I|0]$$  \hspace{1cm} (6.12)

Since the projective camera matrix $P_1 = [I|0]$ and $P_{M1} = P_1H$, the equation (6.12) implies

$$[K_1|0] = [A \quad t]$$ \hspace{1cm} (6.13)

leading to $A = K_1$ and $t = 0$. Let, now, $k = 1$ to fix the scale of reconstruction. The matrix $H$ is obtained as:

$$H = \begin{bmatrix} K_1 & 0 \\ v^T & 1 \end{bmatrix}$$ \hspace{1cm} (6.14)

In metric reconstruction, the coordinates of the plane at infinity do not change and the elements $K_1$ and $v$ of matrix $H$ represents the plane at infinity in projective reconstruction. Hence, one obtains

$$\pi_{\infty} = H^{-T} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} K_1^{-T} & -K_1^{-T}v \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -K_1^{-T}v \\ 1 \end{bmatrix}$$ \hspace{1cm} (6.15)
If it is assumed that $\pi_\infty = (p^T, 1)^T$, one can show that $p = -K_1^{-T}v$.

Finally, the matrix H can be expressed as:

$$H = \begin{bmatrix} K_1 & 0 \\ -p^TK_1 & 1 \end{bmatrix} \quad (6.16)$$

If the calibration matrix of the first camera, $K_1$, and the coordinates of the plane at infinity are known, one can upgrade the projective reconstruction to metric reconstruction by calculating the rectifying homography matrix H. Conversely, if H matrix is known, one can obtain the calibration matrix of the first camera and the coordinates of the plane at infinity. Note that, 8 parameters, which are 3 for p and 5 for $K_1$, are sufficient to describe the H matrix.

Suppose that the projective camera matrices for the other views are given by $P_i = [A_i | a_i]$. So, one obtains by using equation (6.10):

$$K_iR_i = (A_i - a_ip^T)K_1 \quad \text{for } i = 2, \ldots, N \quad (6.17)$$

The rotation matrix can be obtained from equation (6.17) as:

$$R_i = K_i^{-1}(A_i - a_ip^T)K_1 \quad (6.18)$$

Since rotation matrix R is orthogonal, one must have $RR^T = 1$. By using this information, $R_i$ can be eliminated from equation (6.18) as follows.

$$K_iK_i^T = (A_i - a_ip^T)K_1K_1^T(A_i - a_ip^T)^T \quad (6.19)$$
The following equation can be obtained by inserting equation (6.8) into equation (6.19):

\[ w_i^* = (A_i - a_i p^T)w_i^*(A_i - a_i p^T)^T \]  \hspace{1cm} (6.20)

Or

\[ w_i = (A_i - a_i p^T)^{-T}w_1(A_i - a_i p^T)^{-1} \]  \hspace{1cm} (6.21)

Equations (6.20) and (6.21) are the basic equations for self-calibration. Equation (6.21) is simply the inverse of equation (6.20). The self-calibration methods are variations of solving the equations above two equations. Firstly, \( w_i \) or \( w_i^* \) are found by employing iterative methods. After that, \( K_i \) can be calculated by performing Cholesky factorization.

Suppose that the views are taken by the same camera and the internal parameters are constant for multiple views, leading to \( w_i = w_1 \) and \( w_i^* = w_1^* \). Therefore, equations (6.19), (6.20) and (6.21) can be written as follows:

\[ K_1 K_1^T = (A_i - a_i p^T)K_1 K_1^T(A_i - a_i p^T)^T \]

\[ w_1^* = (A_i - a_i p^T)w_1^*(A_i - a_i p^T)^T \]

\[ w_2 = (A_i - a_i p^T)^{-T}w_2(A_i - a_i p^T)^{-1} \]  \hspace{1cm} (6.22)

Since each side of the equations are 3 x 3 matrix, each view other than first one yields 5 constraints. Also, it has been stated before that we have 8 parameters (5 for K matrix and 3 for p) to decide. Therefore, in order to obtain a solution, one must have \( 5(m - 1) \geq 8 \). Clearly, we can obtain a solution when \( m \geq 3 \) where \( m \) is the number of the views.
6.3.2 The Kruppa Equations

The Kruppa equations are based on the epipolar constraint and the absolute conic. The Kruppa equations are the algebraic representations of the correspondence of epipolar lines tangent to a conic [1]. (See Figure 32).

In Figure 32, C and C' designate the images of a conic Cw in two different image planes. Suppose that C* and C*' are the duals of C and C'. l₁ and l₂ are the epipolar tangent lines in the first image plane the epipole of which is e. l₁' and l₂' are the epipolar tangent lines in the second image plane the epipole of which is e'. These two epipolar tangent lines in the first image plane can be combined into a single degenerate point conic given by the equation:

Figure 32: Epipolar Lines Tangent to Conic
\[ C_t = [e]_x C^*[e]_x \]  \hspace{1cm} (6.23)

where \( C_t \) represent the degenerate point conic and 3 x 3 matrix.

Similarly, the same equation can be written for the second image as:

\[ C_t' = [e']_x C'^*[e']_x \]  \hspace{1cm} (6.24)

The transformation equation relating the two point conics is given by

\[ C_t' = H^{-T} C_t H^{-1} \]  \hspace{1cm} (6.25)

where \( H \) is the transformation matrix between the two image planes.

By combining equations (6.23), (6.24) and (6.25), one obtains

\[ [e']_x C'^*[e']_x = H^{-T} [e]_x C^*[e]_x H^{-1} \]  \hspace{1cm} (6.26)

It has already been shown that \( F = H^{-T} [e]_x \) in Chapter 4. Therefore, equation (6.26) can be written as:

\[ [e']_x C'^*[e']_x = FC^*F^T \]  \hspace{1cm} (6.27)
The images C and C' are the images of the absolute conic and it can be shown that C* = w* and C*' = w*'. Also, the transformation matrix H can be written as H = H∞. Therefore, equation (6.27) yields

\[ [e']_x w^* [e']_x = F w^* F^T \quad (6.28) \]

The Kruppa equations given by equation (6.28) are difficult to apply. Hartley developed a simplified version of the Kruppa Equations [1]. He used the SVD of the fundamental matrix to express the simplified Kruppa Equations.

The fundamental matrix has a rank of 2, the SVD of the fundamental matrix F can be written as:

\[ F = U D V^T = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} r & s & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T \quad (6.29) \]

Here, u₃ and v₃ are the null vectors of F such that \( F^T u^3 = 0 \) and \( F v^3 = 0 \), (i.e., \( e = v_3 \) and \( e' = u_3 \)).

Therefore, equation (6.28) can be expressed as:

\[ [u_3]_x w^{*'} [u_3]_x = U D V^T w^* V D U^T \quad (6.30) \]

By pre–multiplying the left hand side of equation (6.30) by \( U^T \) and post–multiplying it by \( U \), one obtains
Applying the same procedure to the right hand side of equation (6.30), one obtains:

\[
DV^T w^V D = \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} V^T w^V \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} r^2 v_1^T w^* v_1 & rs v_1^T w^* v_2 & 0 \\ rs v_1^T w^* v_2 & s^2 v_2^T w^* v_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\]

(6.32)

Equation (6.32) is equivalent to the following expression:

\[
\left( \begin{array}{c} u_2^T w^* u_2 \\ -u_2^T w^* u_2 \\ u_2^T w^* u_1 \end{array} \right) \times \left( \begin{array}{c} r^2 v_1^T w^* v_1 \\ rs v_1^T w^* v_2 \\ s^2 v_2^T w^* v_2 \end{array} \right) = 0
\]

(6.33)

It is clear that equations (6.31) and (6.32) are equivalent to each other. Equation (6.33) implies that the two vectors on the left hand side of the equation must be parallel to each other. Hence, one obtains the following three equations:

\[
\frac{u_2^T w^* u_2}{r^2 v_1^T w^* v_1} = \frac{u_2^T w^* u_2}{rs v_1^T w^* v_2} = \frac{u_2^T w^* u_1}{s^2 v_2^T w^* v_2}
\]

(6.34)

Note that only two of these three equations are linearly independent.
6.3.3 Solving the Simplified Kruppa Equations

The simplified Kruppa equations can be utilized to solve self-calibration problems. Assume that the images are taken with the same camera, leading to $w^* = w^{*/'}$. For only one camera, equation (6.34) yields:

$$
\begin{align*}
\frac{u^*_1 w^* u_2}{r^2 v_1^* w^* v_1} &= - \frac{u^*_1 w^* u_2}{r s v_1^* w^* v_1} = \frac{u^*_1 w^* u_1}{s v_1^* w^* v_2} \\
&= \frac{\rho_1}{\delta_1} = \frac{\rho_2}{\delta_2} = \frac{\rho_3}{\delta_3}
\end{align*}
$$

(6.35)

where $w^* = KK^T$

The simplified Kruppa equations can be utilized in two different ways. In the first method, the terms of the Kruppa equations can be subtracted from each other. Therefore, 3 linearly dependent equations can be shown as:

$$
\begin{align*}
\frac{u^*_1 w^* u_2}{r^2 v_1^* w^* v_1} - \frac{u^*_1 w^* u_2}{r s v_1^* w^* v_1} &= \frac{u^*_1 w^* u_1}{s v_1^* w^* v_2} = \frac{\rho_1}{\delta_1} = \frac{\rho_2}{\delta_2} = \frac{\rho_3}{\delta_3} \\
&= 0
\end{align*}
$$

(6.36)

where, $\rho_1, \rho_2, \rho_3$ : numerator terms of the Kruppa equations

$\delta_1, \delta_2, \delta_3$ : denominator terms of the Kruppa equations

The left hand side of the above 3 equations can be used to define the cost function
\[ c = \sum_{i=1}^{N} w_1 \left( \frac{\rho_1}{\delta_1} - \frac{\rho_2}{\delta_2} \right)^2 + w_2 \left( \frac{\rho_2}{\delta_2} - \frac{\rho_3}{\delta_3} \right)^2 + w_3 \left( \frac{\rho_3}{\delta_3} - \frac{\rho_1}{\delta_1} \right)^2 \]  

(6.37)

which is to be minimized, where \( N \) is the number of view other than the first view and \( w_1, w_2, w_3 \) are weights.

The reason for using three Kruppa equations (although only two of them are linearly independent) is to improve the accuracy of the solution. It is assumed that the weights are uniform in order to decrease the difficulty of minimization process.

The second method of using the simplified Kruppa equations for self–calibration is to obtain three equations from equation (6.35) by cross multiplication. These three equations are given below.

\[
\begin{align*}
L_1 & = \rho_1 \delta_2 - \rho_2 \delta_1 \\
L_2 & = \rho_2 \delta_3 - \rho_3 \delta_2 \\
L_3 & = \rho_3 \delta_1 - \rho_1 \delta_3
\end{align*}
\]

(6.38)

Next, using \( L_1, L_2 \) and \( L_3 \), one can define the cost function

\[
c = \sum_{i=1}^{N} w_1 L_1^2 + w_2 L_2^2 + w_3 L_3^2
\]

(6.39)

which is to be minimized.

Here, \( N \) represent the number of views other than the first one and \( w_1, w_2, w_3 \) are the weighting coefficients which are assumed to be uniform. In this thesis, the second
method of utilizing the simplified Kruppa equations is employed. Using appropriate minimization techniques, $w^*$ is found. Then, the calibration matrix $K$ can be obtained by Cholesky factorization.

### 6.3 Self – Calibration with Equal Eigenvalues

A second method utilized for self – calibration is to use the essential matrix. Note that the essential matrix is a calibrated version of the fundamental matrix. If the images are taken with the same camera whose calibration matrix is $K$, the following equation can be written.

$$E = K^T FK$$  \hfill (6.40)

where, $F$ is the fundamental matrix.

The essential matrix has two nonzero and equal eigenvalues. This feature can be used for calibration purposes. The purpose is to find a calibration matrix $K$ that makes two eigenvalues of the essential matrix $E$ equal (or, as close to equal as possible). Suppose that two eigenvalues of the essential matrix $E$ are $\lambda_1$ and $\lambda_2$. The define the function $f$ via the equation

$$f = 1 - \frac{\lambda_2}{\lambda_1}$$  \hfill (6.41)

Clearly, when the two eigenvalue are equal to each other, the function $f$ will be zero. Hence, the goal is to minimize $f$. Therefore, the cost function $c$ for multiple views is defined to be:

$$c = \sum_{i=1}^{N} w_i (1 - \frac{\lambda_2}{\lambda_1})$$  \hfill (6.42)
Here, \( N \) is the number of views other than the first one and \( w_i \) is the weighting factor. Each weighting factor can be normalized to a range from zero to one. However, as stated before, the weighting factors are taken to be equal to each other for practical purposes.

6.4 Experimental Results of Two Methods

In this section, experimental results related to the simplified Kruppa equations and the eigenvalue technique are examined. The obtained results are then compared with the ones obtained by using Bouguet calibration toolbox. After obtaining the frames, we choose 24 of them in order to perform the calibration process. One should recall that these frames must be viewed at different positions. The eight selected frames are shown in Figure 33.
An important aspect common to both techniques is the initialization of minimization. By using 8 selected frames, one can show the effects of initialization. The only constraint imposed during minimization is that the focal lengths in horizontal and vertical directions are equal. Table 1 shows the results obtained by using different initial guesses in the equal eigenvalues technique.
### Table 1: Results Obtained by using Equal Eigenvalues Method with Different Initial Guesses

<table>
<thead>
<tr>
<th>Initial Values</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>αx</td>
<td>αy</td>
</tr>
<tr>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>650</td>
<td>650</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

As it can be seen easily from Table 1, the image centers are used as the initial guesses for principal points. If we apply the same procedure and initialization for the simplified kruppa equations, we obtain the results shown in Table 2.

### Table 2: Results Obtained by using the Simplified Kruppa Equations with Different Initial Guesses

<table>
<thead>
<tr>
<th>Initial Values</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>αx</td>
<td>αy</td>
</tr>
<tr>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>650</td>
<td>650</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
From the tables, it can be easily seen that initialization does not change the results significantly unless they are too absurd.

Now we will investigate the effect of image numbers and method on the results. To achieve this goal, we have applied the calibration procedure by using from 4 to 24 views. The results obtained by using the simplified Kruppa equations and equal eigenvalues techniques are shown in Table 3 and Table 4. Note that the following initial guesses were used for all cases:

\[ \alpha_x = \alpha_y = 700 \]
\[ x_0 = 320 \text{ and } y_0 = 240 \]
Table 3: Results of the Simplified Kruppa Equations

<table>
<thead>
<tr>
<th>Number of Images</th>
<th>$\alpha_x$ and $\alpha_y$</th>
<th>$x_0$</th>
<th>$y_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1076.9</td>
<td>490</td>
<td>31.9</td>
</tr>
<tr>
<td>5</td>
<td>395.2</td>
<td>188.5</td>
<td>246.1</td>
</tr>
<tr>
<td>6</td>
<td>364.6</td>
<td>181.6</td>
<td>248.8</td>
</tr>
<tr>
<td>7</td>
<td>541.7</td>
<td>231.2</td>
<td>224.4</td>
</tr>
<tr>
<td>8</td>
<td>755</td>
<td>289.8</td>
<td>222.9</td>
</tr>
<tr>
<td>9</td>
<td>780.5</td>
<td>327.7</td>
<td>168</td>
</tr>
<tr>
<td>10</td>
<td>791.6</td>
<td>333.1</td>
<td>162.2</td>
</tr>
<tr>
<td>11</td>
<td>736.2</td>
<td>387.6</td>
<td>46.4</td>
</tr>
<tr>
<td>12</td>
<td>752.8</td>
<td>385.6</td>
<td>71.4</td>
</tr>
<tr>
<td>13</td>
<td>765.4</td>
<td>380.9</td>
<td>86.2</td>
</tr>
<tr>
<td>14</td>
<td>790.3</td>
<td>367.6</td>
<td>122.2</td>
</tr>
<tr>
<td>15</td>
<td>797.6</td>
<td>362.8</td>
<td>134.6</td>
</tr>
<tr>
<td>16</td>
<td>801.3</td>
<td>360.2</td>
<td>140.8</td>
</tr>
<tr>
<td>17</td>
<td>801.3</td>
<td>360.2</td>
<td>140.8</td>
</tr>
<tr>
<td>18</td>
<td>801.3</td>
<td>360.2</td>
<td>140.8</td>
</tr>
<tr>
<td>19</td>
<td>801.3</td>
<td>360.2</td>
<td>140.8</td>
</tr>
<tr>
<td>20</td>
<td>802</td>
<td>359.8</td>
<td>142</td>
</tr>
<tr>
<td>21</td>
<td>660.1</td>
<td>40.5</td>
<td>612.2</td>
</tr>
<tr>
<td>22</td>
<td>682.5</td>
<td>34.8</td>
<td>597.9</td>
</tr>
<tr>
<td>23</td>
<td>682.6</td>
<td>34.8</td>
<td>597.8</td>
</tr>
<tr>
<td>24</td>
<td>687.5</td>
<td>36.1</td>
<td>595.4</td>
</tr>
</tbody>
</table>
### Table 4: Results of Equal Eigenvalue Theorem

<table>
<thead>
<tr>
<th>Number of Images</th>
<th>$\alpha_x$ and $\alpha_y$</th>
<th>$x_0$</th>
<th>$y_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>682.6716</td>
<td>231.5415</td>
<td>290.6958</td>
</tr>
<tr>
<td>5</td>
<td>710.3321</td>
<td>230.1543</td>
<td>250.1878</td>
</tr>
<tr>
<td>6</td>
<td>533.6353</td>
<td>170.1036</td>
<td>261.2115</td>
</tr>
<tr>
<td>7</td>
<td>625.4671</td>
<td>214.6122</td>
<td>267.7546</td>
</tr>
<tr>
<td>8</td>
<td>693.3487</td>
<td>249.6778</td>
<td>284.3153</td>
</tr>
<tr>
<td>9</td>
<td>694.8596</td>
<td>246.3057</td>
<td>277.3451</td>
</tr>
<tr>
<td>10</td>
<td>814.9259</td>
<td>285.9470</td>
<td>241.4271</td>
</tr>
<tr>
<td>11</td>
<td>885.6409</td>
<td>308.6337</td>
<td>211.6786</td>
</tr>
<tr>
<td>12</td>
<td>856.4041</td>
<td>300.6389</td>
<td>227.0759</td>
</tr>
<tr>
<td>13</td>
<td>844.0160</td>
<td>299.7114</td>
<td>230.5727</td>
</tr>
<tr>
<td>14</td>
<td>848.3849</td>
<td>295.4251</td>
<td>228.0971</td>
</tr>
<tr>
<td>15</td>
<td>854.8865</td>
<td>295.5138</td>
<td>225.9425</td>
</tr>
<tr>
<td>16</td>
<td>859.2849</td>
<td>283.3793</td>
<td>223.0458</td>
</tr>
<tr>
<td>17</td>
<td>864.8179</td>
<td>284.6376</td>
<td>215.5329</td>
</tr>
<tr>
<td>18</td>
<td>869.0143</td>
<td>286.4103</td>
<td>214.7112</td>
</tr>
<tr>
<td>19</td>
<td>872.1264</td>
<td>287.7295</td>
<td>214.0974</td>
</tr>
<tr>
<td>20</td>
<td>854.5340</td>
<td>278.3906</td>
<td>221.4971</td>
</tr>
<tr>
<td>21</td>
<td>855.0811</td>
<td>279.4402</td>
<td>217.8246</td>
</tr>
<tr>
<td>22</td>
<td>856.7569</td>
<td>266.8290</td>
<td>218.4219</td>
</tr>
<tr>
<td>23</td>
<td>851.4706</td>
<td>273.4265</td>
<td>222.5004</td>
</tr>
<tr>
<td>24</td>
<td>859.1806</td>
<td>260.7043</td>
<td>223.6296</td>
</tr>
</tbody>
</table>
For the same records, the results obtained by using the Bouguet camera calibration toolbox are as follows:

\[ \alpha_x = 749.9798, \quad \alpha_y = 745.4922 \]

\[ x_0 = 284.4588, \quad y_0 = 249.3364 \]

When the results are examined, one can say that equal eigenvalue theorem is more stable than the simplified Kruppa equations. Considering especially the principal points, there are some very “ill” results in the simplified Kruppa equations. Therefore, it can be concluded that the simplified Kruppa equations are more sensitive to noise.

Now, assume that the results which are obtained by the Bouguet camera calibration toolbox are exactly correct. The percent errors associated with the focal lengths and the location of principal points can then be computed. The error plots associated with the simplified Kruppa equations and the method of equal eigenvalue theorem are given by Figures 34, 35, 36, 37, 38, 39, 40, 41.

![Figure 34: Error Graph of $\alpha_x$](image-url)
Figure 35: Error Graph of $a_y$

Figure 36: Error Graph of $x_0$
Figure 37: Error Graph of $y_0$

Figure 38: Error Graph of $\alpha_x$
Figure 39: Error Graph of $\alpha_y$

Figure 40: Error Graph of $x_0$
From the error graphs, it is observed that the simplified Kruppa equations gives better results for the focal lengths. However, the errors associated with the principal points are quite large even if the number of views increases. Although the equal eigenvalue theorem gives worse results for the focal lengths than the simplified Kruppa equations do, it absolutely gives better results for the principal points (than the simplified Kruppa equations). Therefore, the equal eigenvalue theorem is preferred to use in this thesis.

Figure 41: Error Graph of $y_0$
CHAPTER 7

STEREO TRIANGULATION

7.1 Problem Statement

Suppose that the calibration matrices of the cameras, which are $K_1$ and $K_2$, are known. One should note that camera matrices actually are not known, but estimated. In addition to that, let the extrinsic parameters, which are the translation and rotation vectors, are known. Note that the extrinsic parameters are assumed to be known since the cameras are assumed to be fixed with respect to each other. Therefore, once the extrinsic parameters are estimated, they can be used for other experiments over and over, as long as the positions of the cameras, with respect to each other, remain the same. In this thesis, these extrinsic parameters are estimated by using the Bouguet camera calibration toolbox. The extrinsic parameters are estimated by using a calibration pattern since estimation via a calibration pattern removes the scale ambiguity. On the other hand, if we try to estimate them by using self-calibration methods and then try to estimate a structure, we can obtain only a scaled structure where the scale is not known. Other than knowing the intrinsic and extrinsic parameters, one also needs the image point coordinates of a 3D point in the two image planes so that we can obtain the 3D coordinates of a point which exists in the scene.

Let the measured image points for a 3D point $X$ be $x$ and $x'$. If there is no noise in the measured data, the rays back–projected from the image points will meet at a point which is $X$. However, in the case of noise, the rays back–projected from the image points are skew. In that case, the 3D point $X$ is estimated to be the closest point to both rays back–projected from the image points. Let the camera matrices including the intrinsic and extrinsic parameters be $P$ and $P'$. Then we can define a function $f$ to find 3D point $X$ such that
\( X = \varphi(x, x', P, P') \) (7.1)

### 7.2 Linear Triangulation

Let the camera projection matrices \( P \) and \( P' \) be given by \( P = K [I|0] \) and \( P' = K' [R|t] \).

It is assumed that the world coordinate frame is attached to the left camera coordinate frame. The relations between \( X \), \( x \) and \( x' \) are given by

\[
x = PX
\]
\[
x' = P'X
\] (7.2)

These relations can be combined into a single equation yielding

\[
AX = 0
\] (7.3)

where \( A \) is the matrix involving the elements of \( x, x', P \) and \( P' \).

Equation (7.3) is linear in terms of \( X \). On the other hand, in order to eliminate the scale factor equations (7.2) can be written in the form

\[
x \times (PX) = 0
\]
\[
x' \times (P'X) = 0
\] (7.4)

since \( x \) and \( PX \) are parallel, as well as \( x' \) and \( P'X \).
If we let \( x = [x_1, x_2, 1]^T \) and \( x' = [x'_1, x'_2, 1]^T \), from equation (7.4), one obtains

\[
\begin{align*}
&x_1(p_1^TX) - (p_1^TX) = 0 \\
x_2(p_2^TX) - (p_2^TX) = 0 \\
x'_1(p^T_{12}X) - x'_2(p^T_{11}X) = 0 \\
x'_1(p^T_{22}X) - (p^T_{21}X) = 0 \\
x'_2(p^T_{22}X) - (p^T_{22}X) = 0 \\
x'_1(p^T_{22}X) - x'_2(p^T_{11}X) = 0
\end{align*}
\tag{7.5}
\]

Here, \( P = \begin{bmatrix} p_1^T \\ p_2^T \\ p_3^T \end{bmatrix} \)

Clearly, equations (7.5) are linear in terms of \( X \) and only two of them are linearly independent. Therefore, matrix \( A \) in equation (7.3) is obtained to be

\[
A = \begin{bmatrix}
x_1 p_2^T - p_1^T \\
x_2 p_2^T - p_2^T \\
x'_1 p_3^T - p_1^T \\
x'_2 p_3^T - p_2^T
\end{bmatrix}
\tag{7.6}
\]

Here, matrix \( A \) includes two equations from the two images. Also, one should note that the solution \( X \) from equation (7.3) can be found only up to a scale and it is a 4 x 1 vector.
7.3 Obtaining Extrinsic Parameters

As stated before, in this study the extrinsic parameters of the cameras are obtained by using the Bouguet camera calibration toolbox. In order to do that, the stereo calibration function of the Bouguet camera calibration toolbox is used. The results of a typical experiment which is performed in this thesis are shown below.

Extrinsic parameters (position of right camera with respect to left camera):

Rotation vector: \( \mathbf{om} = [0.04423 \ 0.14274 \ -0.01012] \pm [0.02868 \ 0.02372 \ 0.00344] \)

Translation vector: \( \mathbf{T} = [-262.09198 \ -7.84430 \ 39.35428] \pm [1.79250 \ 1.85734 \ 10.86413] \)

Here, one should note that the numerical errors are approximately three times the standard deviations. In addition to that, it is seen that the rotation matrix is given as a rotation vector which is 3 x 1 vector. The corresponding rotation matrix can be obtained by using the function called Rodrigues in the Bouguet camera calibration toolbox.

After converting the rotation vector to a rotation matrix, the following rotation matrix is obtained.

\[
R = \begin{bmatrix}
0.9898 & 0.0132 & 0.1420 \\
-0.0069 & 0.9990 & -0.0448 \\
-0.1424 & 0.0433 & 0.9889 \\
\end{bmatrix}
\]

The next step is to find the 3D coordinates of a point, which is viewed by two cameras, by the triangulation method.
7.4 Performing Triangulation

In this thesis, in order to find the 3D coordinates of a point, the MATLAB® function stereo triangulation which exists in the Bouguet camera calibration toolbox, has been used. This function requires the camera calibration parameters and the image points from the left and right views to be inputted. In other words, the focal lengths of the left and right cameras, the principal points, the translation and the rotation matrices must be entered as inputs. Furthermore, one can also enter the distortion coefficients in order to obtain better results.

The function yields two sets of 3D coordinates, namely $X_R$ and $X_L$ which are related by the equation

$$X_R = R \times X_L + T$$

(7.7)

where, $X_L$ and $X_R$ are the 3D coordinates of the point in left and right camera frames, and $R$ and $T$ is rotation matrix and translation vector as it is specified before.

7.5 Case Studies

In this section, case studies, where the 3D coordinates of a point is obtained via the developed programs, are given. In these case studies, errors, associated with the distance between two points, are also investigated.
Figure 42: Left View of a Scene with Detected Corner Points

Figure 43: Right View of a Scene with Detected Points
Now, consider the six points on the pattern which exist on the wall in Figure 42 and Figure 43. Note that the edge of each square is 60 mm. The points which are selected on the pattern and their coordinates are shown in Figure 44 and Figure 45.

Figure 44: Left View with Selected Few Corner Points

Figure 45: Right View with Selected Few Corner Points
Firstly, recall that the camera calibration matrix $K$ is defined via the equation.

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

The calibration matrix for the left camera has been obtained to be

$$K_{left} = \begin{bmatrix} 859.2126 & 0 & 263.2093 \\ 0 & 859.2126 & 220.8987 \\ 0 & 0 & 1 \end{bmatrix}$$

whereas, the calibration matrix for the right camera has been obtained to be

$$K_{right} = \begin{bmatrix} 925.6289 & 0 & 294.9676 \\ 0 & 925.6289 & 249.6408 \\ 0 & 0 & 1 \end{bmatrix}$$

The translation vector between the cameras

$$T = \begin{bmatrix} -262.092 \\ -7.844 \\ 39.354 \end{bmatrix}$$

And the rotation matrix between the cameras is given by
\[ R = \begin{bmatrix} 0.9898 & 0.0132 & 0.1420 \\ -0.0069 & 0.9990 & -0.0448 \\ -0.1424 & 0.0433 & 0.9889 \end{bmatrix} \]

The 3D coordinates of the six selected points that have been obtained are given in Table 5.

Table 5: Image Coordinates and 3D Point Coordinates

<table>
<thead>
<tr>
<th>No</th>
<th>Left Image Coordinate</th>
<th>Right Image Coordinate</th>
<th>3D coordinate with respect to the left camera coordinate frame</th>
<th>3D coordinate with respect to the right camera coordinate frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[335,45]²</td>
<td>[369,80]²</td>
<td>[296.4,-499.2,2975.8]²</td>
<td>[185,-649.8,2957.4]²</td>
</tr>
<tr>
<td>2</td>
<td>[354,46]²</td>
<td>[387,80]²</td>
<td>[357.2,-492.2,2949.2]²</td>
<td>[241.6,-641.8,2922.8]²</td>
</tr>
<tr>
<td>3</td>
<td>[336,64]²</td>
<td>[370,98]²</td>
<td>[300.2,-443.8,3014.6]²</td>
<td>[195,-596,2997.6]²</td>
</tr>
<tr>
<td>4</td>
<td>[355,63]²</td>
<td>[387,96]²</td>
<td>[359.4,-441.2,2970.2]²</td>
<td>[247.4,-592,2945.4]²</td>
</tr>
<tr>
<td>5</td>
<td>[336,81]²</td>
<td>[371,116]²</td>
<td>[301.8,-387.4,3032.6]²</td>
<td>[200,-540.6,3017.8]²</td>
</tr>
<tr>
<td>6</td>
<td>[355,81]²</td>
<td>[388,114]²</td>
<td>[362.2,-386.2,3005.6]²</td>
<td>[255.8,-538.8,2828.4]²</td>
</tr>
</tbody>
</table>

Suppose, now, that we have two different points \( M_1 = [X_1, Y_1, Z_1]^T \) and \( M_2 = [X_2, Y_2, Z_2] \). The distance between these two points, \( D_{12} \) is given by

\[
D_{12} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2} \quad (7.8)
\]

Table 6 shows the calculated distances (calculated by using the equation (7.8)) between the six points whose coordinates are given in Table 5.
Table 6: Estimated Distance between Points

<table>
<thead>
<tr>
<th>Points used</th>
<th>Calculated distance (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>66.7</td>
</tr>
<tr>
<td>1-3</td>
<td>67.7</td>
</tr>
<tr>
<td>3-4</td>
<td>74.1</td>
</tr>
<tr>
<td>2-4</td>
<td>55.2</td>
</tr>
<tr>
<td>3-5</td>
<td>59.2</td>
</tr>
<tr>
<td>5-6</td>
<td>66.2</td>
</tr>
<tr>
<td>4-6</td>
<td>65.5</td>
</tr>
</tbody>
</table>

It is known that the dimension of an edge of square on the pattern is 60 mm. Therefore, we can find the errors associated with the estimated dimensions as shown in Table 7.

Table 7: Error Percentage between Estimated Distance and Real Distance Values

<table>
<thead>
<tr>
<th>Points used</th>
<th>Error in distance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>11.2</td>
</tr>
<tr>
<td>1-3</td>
<td>12.9</td>
</tr>
<tr>
<td>3-4</td>
<td>24.1</td>
</tr>
<tr>
<td>2-4</td>
<td>8</td>
</tr>
<tr>
<td>3-5</td>
<td>1.3</td>
</tr>
<tr>
<td>5-6</td>
<td>10.2</td>
</tr>
<tr>
<td>4-6</td>
<td>9.1</td>
</tr>
</tbody>
</table>
Now, 10 more points are selected on the scene in Figure 46 and Figure 47 and show them in 3D graph leading to Figure 48.

Figure 46: Left View with Selected Corner Points

Figure 47: Right View with Selected Corner Points
Figure 48: Original View of 3D scene

Figure 49: x-y View of 3D Scene
Figure 50: x- z View of 3D Scene

Figure 51: y- z View of 3D Scene
If one compares the x-y view of the 3D scene with images, it can be seen that they are very similar, having almost the same structure.
The kinematic analysis of a slider crank mechanism is presented in this chapter. As it is known, the degree of freedom of the mechanism is one. The mechanism which is analyzed is shown in Figure 52. This mechanism has been already built for the TUBITAK project 112M110 [50]. Hence, it was readily available to use in this study. The mechanism is actuated by a motor which is located at point, O₁, at which the crank and the fixed link are connected. The slider moves in a track on the fixed link.
One should note that the points $O_1$, $O_2$ and $O_3$ are the origins of the body fixed reference frames for the crank, the connecting rod and the slider, respectively. Note also that these points coincide with the centers of the 3 revolute joints in the mechanism.

In Figure 52, one can observe that there are several red markers on the moving links. The locations of the markers have been selected somewhat arbitrarily. The vision system is used to obtain the positions of the markers for each frame. As has been stated before, the cameras used in this thesis are 30 fps cameras, i.e., there is a $(1/30)$ second time difference between two consecutive frames.

### 8.1 Positions of the Markers

In order to obtain the marker positions, a calibrated vision system, which has been developed before, will be used. As stated before, the extrinsic parameters of the cameras, with respect to each other, are fixed and known during the recording of the mechanism. The intrinsic parameters, on the other hand, are obtained by using the self–calibration method. Since it is shown that the equal eigenvalue theorem gives better results, this method is used to find the intrinsic parameters. Unlike the previous case where the two methods are compared, here, 48 frames are used to perform self–calibration in order to obtain more accurate results. The results of the self–calibration process have been presented below for the two cameras.

The calibration matrix for the left camera is given by

$$K_{\text{left}} = \begin{bmatrix} 710.4835 & 0 & 305.3316 \\ 0 & 710.4835 & 240.0807 \\ 0 & 0 & 1 \end{bmatrix}$$
whereas, the calibration matrix for the right camera is

\[
K_{right} = \begin{bmatrix}
726.1441 & 0 & 321.1462 \\
0 & 726.1441 & 236.0546 \\
0 & 0 & 1
\end{bmatrix}
\]

When the intrinsic and extrinsic parameters are known, the triangulation method can be used to determine the coordinates of a point in the scene. Hence, using the triangulation method, the center points of the markers are obtained in the left and right camera reference frame. One should recall that the lens distortion effects are neglected in this thesis.

As can be seen in Figure 52, there are 4 markers on each of the 3 moving links. Hence, totally, there are 12 markers on the mechanism.

Consider, now, the following notation.

- \( l \) \( \Rightarrow \) Number of links in the mechanism.
- \( j \) \( \Rightarrow \) Number of joints in the mechanism.
- \( NM_i \) \( \Rightarrow \) Number of markers on link \( i \) of the mechanism.
- \( NM_L \) \( \Rightarrow \) Total number of markers on the mechanism, i.e.,

\[
NM_L = NM_1 + NM_2 + \ldots + NM_l
\]  \hspace{1cm} (8.1)

Note that there are no markers on link 1 (which is the fixed link), i.e., \( NM_1 = 0 \)
Each link in the mechanism is expressed such that fixed link is link 1, crank is link 2, connecting rod is link 3 and the slider is link 4.

Hence, for the mechanism we have

\[ N \sum M = 0 + 4 + 4 + 4 = 12 \]

A marker is labelled as \( M_{i,j} \) where \( i \) represents the link number on which the marker exists and \( j \) is the number of the marker. In the light of this notation, the Figure 53 shows the marker labels for each of the markers.

![Figure 53: Markers on the mechanism](image)

In this thesis, the number of frames, NF, used for the kinematic analysis of the slider crank mechanism is 188 which corresponds to 6 periods of the motion of the mechanism. Therefore, one obtains 188 position data for each marker. After obtaining
the position data, a curve fitting process is performed after excluding the outliers, if there are any.

**8.1.1 Curve Fitting Process for the Position Data**

Curve fitting is a process that yields the parameters of a curve which fits a set of points in the best possible manner. In this thesis, the MATLAB® curve fitting toolbox is used for this purpose. In this study, independent variable is the frame number or time (time is used in this thesis) and the dependent variable is the position data. In order to obtain the curve which fits a given set of points, a least squares method is used, i.e., the sum of the squares of the errors are minimized. Hence, the function to be minimized is given by:

\[ SSE = \sum_{t=r_0}^{t=r_{\text{max}}} (p(t) - p_f(t))^2 \]  

(8.2)

where, \( p(t) \) is the original data, position of marker, to be fitted, \( t \) is the independent variable, time; and \( p_f(t) \) is the curve fitting the data. One should also note that \( p(t) - p_f(t) \) represents the residual, which is the difference between the actual value and the estimated value. The values of the residual and the SSE that are obtained are very useful to determine the best fit. In order to determine the best fit, one should examine the obtained fit visually and check numerical fit results. Firstly, the curve should represent the general trend of the data. The residual graph also gives an idea about the curve. Hence, after examining the curve visually, one should also examine the numerical results. SSE and the adjusted R-square are two good indicators of the curve fitted; and MATLAB® is able to provide these indicators as a result of the curve fitting operation. As stated before, SSE is the sum of the squares of the error. Hence, a value closer to zero indicates a better curve. The adjusted R-square is another indicator which can be used when the number of coefficients for the fit is increased. An adjusted R-square value closer to one means a better curve, too. In this study, the smoothing spline method of curve fitting is used for markers’ position data. Figure 54 and Figure 55
show two examples of the curve fitting process for the x, y and z coordinates of the markers M_{2,3} and M_{3,1} in the left camera reference frame.

Figure 54: Curve Fit for position of marker M_{2,3}
Figure 55: Curve Fit for position of marker M3,1
8.2 Distances between the Markers

After finding the positions of the markers, one can obtain the distances between the markers. The distance between the markers $M_{j,i}$ and $M_{k,l}$ is given by

$$D = \sqrt{(x_{j,i} - x_{k,l})^2 + (y_{j,i} - y_{k,l})^2 + (z_{j,i} - z_{k,l})^2}$$ (8.3)

The following table shows the measured distances between different markers and the corresponding distances obtained by the vision system. One should note that the distances between the markers are measured by using a Vernier.

<table>
<thead>
<tr>
<th>Markers</th>
<th>Measured Distance, $D$ (mm)</th>
<th>$D$ by Vision (mm)</th>
<th>Percent Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{2,1}$-$M_{2,2}$</td>
<td>37</td>
<td>38.68</td>
<td>4.55</td>
</tr>
<tr>
<td>$M_{2,1}$-$M_{2,3}$</td>
<td>67.1</td>
<td>72.06</td>
<td>7.39</td>
</tr>
<tr>
<td>$M_{2,2}$-$M_{2,3}$</td>
<td>58</td>
<td>62.58</td>
<td>7.89</td>
</tr>
<tr>
<td>$M_{2,3}$-$M_{2,4}$</td>
<td>142.5</td>
<td>149.69</td>
<td>5.05</td>
</tr>
<tr>
<td>$M_{3,1}$-$M_{3,3}$</td>
<td>67.6</td>
<td>71.94</td>
<td>6.42</td>
</tr>
<tr>
<td>$M_{3,2}$-$M_{3,3}$</td>
<td>47</td>
<td>52.17</td>
<td>11.00</td>
</tr>
<tr>
<td>$M_{3,3}$-$M_{3,4}$</td>
<td>70.5</td>
<td>73.93</td>
<td>4.87</td>
</tr>
<tr>
<td>$M_{3,2}$-$M_{3,4}$</td>
<td>107.4</td>
<td>116.47</td>
<td>8.44</td>
</tr>
<tr>
<td>$M_{3,1}$-$M_{3,4}$</td>
<td>138.35</td>
<td>145.63</td>
<td>5.26</td>
</tr>
<tr>
<td>$M_{4,1}$-$M_{4,2}$</td>
<td>17.6</td>
<td>18.68</td>
<td>6.15</td>
</tr>
<tr>
<td>$M_{4,1}$-$M_{4,3}$</td>
<td>91.3</td>
<td>93.99</td>
<td>2.94</td>
</tr>
<tr>
<td>$M_{4,2}$-$M_{4,4}$</td>
<td>90.5</td>
<td>93.42</td>
<td>3.22</td>
</tr>
<tr>
<td>$M_{4,3}$-$M_{4,4}$</td>
<td>19.2</td>
<td>20.02</td>
<td>4.27</td>
</tr>
</tbody>
</table>
8.3 Slider Crank Mechanism

In Figure 56, a slider crank mechanism is illustrated.

Figure 56: Illustration of Slider Crank Mechanism

\( \mathcal{O}_mX_mY_mZ_m \) : fixed coordinate system, or the mechanism coordinate system, with origin at \( \mathcal{O}_m \).

\( \overline{AB} = a_2 \)

\( \overline{BC} = a_3 \)

The vertical distance between horizontal lines passing through A and C = \( a_1 \)

Note that \( a_1, a_2, a_3 \) are the dimensions of the slider crank mechanism.

\( \theta_2, \theta_3, s_4 \) : joint variables of the slider crank mechanism

\( M_{i,j} \) : \( j \)th marker on link \( i \) where \( j = 1, 2, \ldots \) and \( i = 2, 3, 4 \)
\((L_{i,j}, y_{i,j})\) : polar coordinates of marker \(M_{i,j}\) with respect to a body fixed frame attached to link \(i\).

\((X_{i,j}, Y_{i,j})\) : \(x,y\) coordinates of marker \(M_{i,j}\) with respect to the \(O Xu Y u Z u\) coordinate system.

The loop closure equation (LCE) of the slider crank mechanism is given by

\[
a_2 e^{i\theta_2} + a_4 e^{i\theta_4} = s_4 + a_1 i
\]  

(8.4)

which yields the following two scalar equations, obtained from the real and imaginary parts of equation (8.4):

\[
s_4 - a_3 \cos(\theta_3) - a_2 \cos(\theta_2) = 0
\]  

(8.5)

\[
a_2 - a_3 \sin(\theta_3) - a_2 \sin(\theta_2) = 0
\]  

(8.6)

The kinematic dimensions of the slider crank mechanism used are given by

\(a_1 = 160\) mm

\(a_2 = 102\) mm

\(a_3 = 350\) mm

The polar coordinates of the markers are used, on the other hand, are presented in Table 9. Hence, for a given value of the input \(\theta_2\), one can solve \(\theta_3\) from equation (8.6) and insert values of \(\theta_2\) and \(\theta_3\) into equation (8.5) to obtain a corresponding value of \(s_4\). One should note that there are at most 2 different solutions for the unknowns \(\theta_2\) and \(s_4\) for a given \(\theta_2\). Therefore, one should select the correct closure among these 2 possible solutions.
Table 9: Polar Coordinates of the Markers

<table>
<thead>
<tr>
<th>Marker</th>
<th>Polar Coordinates (L&lt;sub&gt;i,j&lt;/sub&gt; [mm], γ&lt;sub&gt;i,j&lt;/sub&gt; [°])</th>
</tr>
</thead>
<tbody>
<tr>
<td>M&lt;sub&gt;2,1&lt;/sub&gt;</td>
<td>84.7 mm, -14.6°</td>
</tr>
<tr>
<td>M&lt;sub&gt;2,2&lt;/sub&gt;</td>
<td>84.0 mm, 10.6°</td>
</tr>
<tr>
<td>M&lt;sub&gt;2,3&lt;/sub&gt;</td>
<td>64.5 mm, 12.7°</td>
</tr>
<tr>
<td>M&lt;sub&gt;2,4&lt;/sub&gt;</td>
<td>95.1 mm, 1.8°, -62.2 mm (z-direction)</td>
</tr>
<tr>
<td>M&lt;sub&gt;3,1&lt;/sub&gt;</td>
<td>78.9 mm, 0°</td>
</tr>
<tr>
<td>M&lt;sub&gt;3,2&lt;/sub&gt;</td>
<td>116.9 mm, 10.2°, -26.1 mm (z-direction)</td>
</tr>
<tr>
<td>M&lt;sub&gt;3,3&lt;/sub&gt;</td>
<td>146.2 mm, -0.1°</td>
</tr>
<tr>
<td>M&lt;sub&gt;3,4&lt;/sub&gt;</td>
<td>216.7 mm, -0.3°</td>
</tr>
<tr>
<td>M&lt;sub&gt;4,1&lt;/sub&gt;</td>
<td>50.6 mm, 18.3°</td>
</tr>
<tr>
<td>M&lt;sub&gt;4,2&lt;/sub&gt;</td>
<td>60.9 mm, 30.9°</td>
</tr>
<tr>
<td>M&lt;sub&gt;4,3&lt;/sub&gt;</td>
<td>140.4 mm, 4.5°</td>
</tr>
<tr>
<td>M&lt;sub&gt;4,4&lt;/sub&gt;</td>
<td>145.1 mm, 11.9°</td>
</tr>
</tbody>
</table>

Note that, the positions of the markers are obtained in the camera reference frames. Hence, one should find the rotation matrix that relates the mechanism and camera reference frames.

8.4 Rotation Matrix Relating the Camera and Mechanism Reference Frames

\[ O_cX_cY_cZ_c \quad \Rightarrow \quad \text{Camera reference frame fixed the ground (the position and orientation unknown)} \]

\[ O_mX_mY_mZ_m \quad \Rightarrow \quad \text{Mechanism reference frame fixed to a suitable point on the ground, i.e., link 1 of the mechanism. Preferably, the origin should be at a revolute joint associated with link 1. Furthermore, } X_m, Y_m \text{ axes lie on the plane of motion of the mechanism such that } Z_m \text{ is perpendicular to the plane of motion. Note that } X_m, Y_m \text{ axes will correspond to the real and imaginary axes while} \]
writing the LCE. Hence, the \( \theta \) angles will be measured relative to the \( X_m \) axis.

\[
[mR_c] \quad \Rightarrow \quad \text{Rotation matrix which relates the coordinates of a vector expressed in the camera and mechanism frames via the equation}
\]

\[
\vec{V}_m = [mR_c] \vec{V}_c 
\]  

(8.7)

where,
\[
\vec{V}_m, \vec{V}_c \quad \Rightarrow \quad \text{Coordinates of a vector \( \vec{V} \) expressed in the mechanism and camera frames, respectively.}
\]

Since the cameras are stationary, we can use the motion of the slider in order to find \( X_m \) axis of the mechanism reference frame. For this purpose, while the mechanism moves five frames are taken into consideration and the coordinates of the two markers, namely, \( M_{4,1} \) and \( M_{4,2} \), are determined in each of the five frames. The unit vector in \( X_m \) direction is calculated via the following equation.

\[
\vec{u}_{m}^{'}(n) = \frac{\vec{r}(n) - \vec{r}(n+1)}{||\vec{r}(n) - \vec{r}(n+1)||} 
\]  

(8.8)

where,
\[
\vec{r}: \text{position vector of the markers in camera reference frame}
\]

\( n\): frame number, \( n = 1, 2, 3, 4 \).
In order to find the \( Z_m \) axis, the markers \( M_{2,1}, M_{2,2} \) and \( M_{2,3} \) on the crank (link 2) are used together with the following notation.

\[
\vec{r}_{12} : \text{vector from marker } M_{2,1} \text{ to marker } M_{2,2}
\]

\[
\vec{r}_{12} : \text{vector from marker } M_{2,1} \text{ to marker } M_{2,3}
\]

The unit vector which is perpendicular to the crank can be found by the following formula:

\[
\vec{u}_2'(n) = \frac{\vec{r}_{12}(n) \times \vec{r}_{13}(n)}{|\vec{r}_{12}(n) \times \vec{r}_{13}(n)|}
\] (8.9)

The mean value for \( \vec{u}_1' \) and \( \vec{u}_2' \) can be calculated as

\[
\vec{u}_1' = \frac{\sum_{n=1}^{N} \vec{u}_1'(n)}{N} = \vec{u}_1
\] (8.10)

\[
\vec{u}_2' = \frac{\sum_{n=1}^{N} \vec{u}_2'(n)}{N} = \vec{u}_2
\] (8.11)

where, \( N \) is the total number of unit vectors.

Finally, \( \vec{u}_2 \) can be found as follows:

\[
\vec{u}_2 = \frac{\vec{u}_2 \times \vec{u}_2}{|\vec{u}_2 \times \vec{u}_2|}
\] (8.12)

Therefore, the rotation matrix, \([^m R_c]\), is found to be

\[
[^m R_c] = [\vec{u}_1 \quad \vec{u}_2 \quad \vec{u}_3]
\] (8.13)

### 8.5 Position Vectors of the Markers

Consider, now, the following notation.
\[ \overrightarrow{O_cM_{i,j}}(t) \quad \Rightarrow \quad \text{Vector } \overrightarrow{O_cM_{i,j}}, \text{ i.e. vector from the origin } O_c \text{ to marker } M_{i,j}, \text{ at time (or, frame) } t. \]

\[ \overrightarrow{O_cO_m} \quad \Rightarrow \quad \text{Vector from } O_c \text{ to } O_m \text{ (constant)} \]

\[ \overrightarrow{O_mO_i} \quad \Rightarrow \quad \text{Vector from } O_m \text{ to } O_i, \text{ origin of the body fixed frame of link } i, \text{ at time(or, frame) } t \]

One may write the following equation at time \( t \), for the marker \( M_{2,j} \), which is the \( j^{th} \) marker located on link 2:

\[
\overrightarrow{O_cM_{2,j}}(t) = \overrightarrow{O_cO_m} + \overrightarrow{O_mO_2}(t) + [L_{2,j} \cos(\theta_2 + \gamma_{2,j}) \overrightarrow{t_m} + L_{2,j} \sin(\theta_2 + \gamma_{2,j}) \overrightarrow{m}] 
\]

\( (8.14) \)

In equation (8.14), all of the vectors must be expressed in the \( x_m y_m z_m \) frame since the last two vectors of the equation are expressed in the \( x_m y_m z_m \) frame. Therefore, one can rewrite the equation (8.14) as follows:

\[
[mR_c] \overrightarrow{(O_cM_{2,j})_c}(t) = [mR_c] \overrightarrow{(O_cO_m)_c} + [mR_c] \overrightarrow{(O_mO_2)_m}(t) + [L_{2,j} \cos(\theta_2 + \gamma_{2,j}) \overrightarrow{m}] 
\]
+ L_{2,j} \sin(\theta_2 + \gamma_{2,j})j_m]}

(8.15)

where,

\( (O_cM_{2,j})_c(t) \Rightarrow O_cM_{2,j}(t) \) vector expressed in the \( x_cy_cz_c \) reference frame (as obtained from the vision system)

\( (O_cO_m)_c \Rightarrow O_cO_m^c \) vector expressed in \( x_cy_cz_c \) reference frame (as obtained from the vision system)

\( (O_mO_{2,j})_m(t) \Rightarrow O_mO_{2,j}(t) \) vector expressed in the \( x_my_mz_m \) reference system

One should note that \( O_mO_{2,j}(t) \) is 0 since the origin of the body fixed reference frame of link 2 and the origin of the mechanism reference frame are coincident.

A similar equation can also be written for the maker \( M_{3,j} \) on link 3, yielding

\[ [^mR_c] (O_cM_{3,j})_c(t) = [^mR_c] (O_cO_m)_c + (O_mO_{3,j})_m(t) + [L_{3,j} \cos(\theta_3 + \gamma_{3,j})j_m] + \]

\[ + L_{3,j} \sin(\theta_3 + \gamma_{3,j})j_m] \]

(8.16)

A similar equation can also be written for the maker \( M_{4,j} \) on link 4, with a few changes, leading to the equation
\[ [mR_c](O_mO_4)_m(t) = [mR_c](O_cO_m)_c + (O_mO_4)_m(t) + [L_{4,j}\cos(y_{4,j})]\vec{t}_m \\
+ L_{4,j}\sin(y_{4,j})\vec{j}_m \]

(8.17)

where,

\[ (O_mO_4)_m(t) = \begin{bmatrix} S_4 \\ a_1 \\ 0 \end{bmatrix} \]

By using equations (8.15), (8.16) and (8.17) together with the LCE, one can find a set of joint variables, \((\theta_2, \theta_3, s_4)\), by using any marker on any link.

8.5.1 Curve Fitting Process to Joint Variables

After finding a set of joint variables, \((\theta_2, \theta_3, s_4)\), one should make sure that the joint variables calculated by the vision system and the joint variables calculated by using the encoder have the same period at steady state. For this purpose, a Fourier series, with a pre-determined period, is fitted to the set of variables. The Fourier series function that is fitted to a joint variable is described by the function

\[ g_{fit}(t) = a_0 + \sum_{n=1}^{N} a_n \cos(nwt) + b_n \sin(nwt) \]

(8.18)

where,

w: frequency,

N: number of harmonics

\(a_0, a_n\ and b_n\): undetermined coefficients.
Here, one should note that $N$ is a design parameter which is selected according to the number of extremes of the data which is going to be fitted. An explanation of reasoning used in order to select $N$ is given in Appendix D. According to this reasoning, the minimum of $N$ should be equal to the number of extremes [49].

While selecting the period to be used in the fits, both the vision data and the encoder data should be examined. When the vision data is examined, it is observed that a period is completed in 32 frames after the mechanism reaches steady state, i.e., the period is $\frac{32}{30} = 1.066$ seconds. When the encoder data is examined, on the other hand, it is observed that a period is completed in 1.065 seconds after the mechanism reaches steady state. (Figure 57). Since the encoder gives more frequent data, a period is taken to be 1.065 seconds for the Fourier series fit for both the encoder and vision data.

![Figure 57: Encoder Data for a specific time interval](image)
The curve fitted to the joint variable $\theta_3$, with a period of 1.065 seconds is given in Figure 58.

Since $\theta_3$ has 4 extremes, $N$ should be taken to be at least 2. In Figure 58, the graphs on the left correspond to $N = 2$ while the graphs on the right correspond to $N = 3$. It can be observed from Figure 58 that the two fits, for $N = 2$ and $N = 3$, are quite close to each other, implying that the results are dependable. One can apply the same process for $s_{14}$. $\theta_2$, however, is different from $\theta_3$ and $s_{14}$ since it does not have any extremes as seen in Figure 59. Hence, one should determine time derivative of $\theta_2$ to find the velocity data and apply the curve fitting process to the velocity data. Then, the integral of the curve fitted should yield $\theta_2$ and time derivative of it will yield the acceleration data. In addition, the goodness of the curve fitted is evaluated according to the rules presented for the curve fitting process in the previous sections.
8.6 Case Studies

In this section, kinematic analysis of the slider crank mechanism shown in Figure 52 is performed by using the equations, presented in the previous sections, and the results are presented. Figure 60 shows the definitions of the joint variables of the slider crank mechanism which is analyzed.
The set of joint variables \((\theta_2, \theta_3, s_{14})\) is calculated by using different number of markers. After finding the joint variables, \((\theta_2, \theta_3, s_{14})\), the error in the joint variable \(\theta_2\) is calculated via the equation:

\[
\text{Percent error in } \theta_2 = e_2(t) = \frac{(\theta_2)_{\text{vision}} - (\theta_2)_{\text{encoder}}}{(\theta_2)_{\text{encoder}}} \times 100 \tag{8.19}
\]

where \((\theta_2)_{\text{vision}}\) and \((\theta_2)_{\text{encoder}}\) denote the \(\theta_2\) values obtained via the vision system data and via the encoder data, respectively. Here, it is assumed that encoder data yields actual value of \(\theta_2\).
Equation (8.19) defines the error in $\theta_2$. One can easily write down similar equations to define the errors in $\theta_3$ and $s_{14}$ as well.

In the forthcoming sections, the graphs of the joint variables $(\theta_2, \theta_3, s_{14})$ obtained by the vision system; obtained by the vision system and the encoder; the graphs of the errors in the joint variables $(\theta_2, \theta_3, s_{14})$; and the graphs of the velocity and acceleration variables, i.e., first and second time derivation of the joint variables, are presented for different number of markers. The aforementioned results have been obtained by using different sets of markers.

**8.6.1 Case 1**

In this case, the kinematic analysis of the slider crank is performed by using only one marker. One should note that the Cartesian coordinates of the markers with respect to the body fixed reference frames and the dimensions of the slider crank mechanism are assumed to be known. Also, the coordinates of the origins of the body fixed reference frames are obtained by using the vision system. The results for marker M2,1 are presented in Figures 61 and 62. The results for the remaining markers are given in Appendix E.
Figure 61: Results of the kinematic analysis by using marker M2,1
Figure 62: Results of the kinematic analysis by using marker M2,1 [continued]
8.6.2 Case 2

In this case, the kinematic analysis of the slider crank is performed by using two markers which are selected randomly. One should note that, similar to case 1, the Cartesian coordinates of the markers with respect to the body fixed reference frames and the dimensions of the slider crank mechanism are assumed to be. A single value for the joint variable vector \((\theta_2, \theta_3, s_{14})\) is calculated for each marker. Then, average values of the joint variables are calculated via the equations

\[
\bar{\theta}_2 = \frac{\sum_{i=1}^{NM} \theta_2^i}{NM} \quad (8.20)
\]

\[
\bar{\theta}_3 = \frac{\sum_{i=1}^{NM} \theta_3^i}{NM} \quad (8.21)
\]

\[
\bar{s}_4 = \frac{\sum_{i=1}^{NM} s_4^i}{NM} \quad (8.22)
\]

where \(\theta_2^i, \theta_3^i\) and \(s_4^i\) denote the \(i^{th}\) value of \(\theta_2, \theta_3\) and \(s_4\), respectively; and NM is the number of markers. Note that for case 2, NM = 2.

Clearly, the average values obtained via (8.20) – (8.22) will not satisfy the LCE. Therefore, in order to find the values of the joint variables which do satisfy the LCE, a minimization problem which minimizes the objective function

\[
w_1 \left( \frac{\theta_2 - \bar{\theta}_2}{\bar{\theta}_2} \right)^2 + w_2 \left( \frac{\theta_3 - \bar{\theta}_3}{\bar{\theta}_3} \right)^2 + w_3 \left( \frac{s_4 - \bar{s}_4}{\bar{s}_4} \right)^2 \quad (8.23)
\]

subject to the constraints
\[ s_4 + a_3 \cos(\theta_3) - a_2 \cos(\theta_2) = 0 \]
\[ a_1 + a_3 \sin(\theta_3) - a_2 \sin(\theta_2) = 0 \]

is solved. Here, \( w_1, w_2 \) and \( w_3 \) are weighting coefficients, which are assumed as unity.

The obtained \((\theta_2, \theta_3, s_{14})\) values which minimizes the objective function are presented in Figure 63 and 64. The results for the remaining marker pairs are given in Appendix F.
Figure 63: Results of the kinematic analysis by using markers M2,1 and M2,2
Figure 64: Results of the kinematic analysis by using markers M2,1 and M2,2
[continued]
8.6.3 Case 3

Case 3 is very similar to case 2, the only difference is that the number of markers used for the kinematic analysis is three, leading to \( NM = 3 \). The procedure that is followed is the same as case 2. The results for the marker triple \( M_{2,3} \), \( M_{3,4} \) and \( M_{4,1} \) are presented in Figures 65 and 66. The results for the remaining marker triples are given Appendix G.
Figure 65: Results of the kinematic analysis by using the marker triple M2,3, M3,4 and M4,1
Figure 66: Results of the kinematic analysis by using the marker triple M2,3, M3,4 and M4,1 [continued]
8.6.4 Case 4

In this case, the number of markers used for the kinematic analysis is 4. The procedure that is followed is the same as in case 2 and case 3. The results are presented in Figures 67 and 68.
Figure 67: Results of the kinematic analysis by using markers $M_{2,2}$, $M_{2,3}$, $M_{3,4}$ and $M_{4,1}$
Figure 68: Results of the kinematic analysis by using markers M2,2, M2,3, M3,4 and M4,1 [continued]
8.7 Comparison of Results and Discussion

Let’s define the average percent error and percent rms errors associated with $\theta_2(t)$ via the following two equations.

Average percent error in $\theta_2 = e_{\theta_2} = \frac{\int_{t_{initial}}^{t_{final}} [e_{\theta_2}(t)] dt}{t_{final} - t_{initial}}$ (8.24)

Rms value of error $e_{\theta_2}(t) = (e_{\theta_2})_{rms} = \sqrt{\frac{\int_{t_{initial}}^{t_{final}} [e_{\theta_2}(t)]^2 dt}{t_{final} - t_{initial}}}$ (8.25)

The average and rms errors associated with $\theta_3$ and $s_{14}$ are defined in a similar manner. Table 10 and Table 11, average and rms errors thus obtained are shown.
Table 10: Average Errors of Joint Variables

<table>
<thead>
<tr>
<th>Number of Markers</th>
<th>Markers</th>
<th>Average Percent Error of θ₂ (%)</th>
<th>Average Percent Error of θ₃ (%)</th>
<th>Average Percent Error of s₁₄ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M₂₁</td>
<td>5,3</td>
<td>0,4</td>
<td>3,2</td>
</tr>
<tr>
<td>1</td>
<td>M₂₂</td>
<td>5,3</td>
<td>0,3</td>
<td>2,5</td>
</tr>
<tr>
<td>1</td>
<td>M₂₃</td>
<td>6,7</td>
<td>0,7</td>
<td>4,4</td>
</tr>
<tr>
<td>1</td>
<td>M₂₄</td>
<td>17,9</td>
<td>1,5</td>
<td>12,0</td>
</tr>
<tr>
<td>1</td>
<td>M₃₁</td>
<td>12,1</td>
<td>0,7</td>
<td>7,3</td>
</tr>
<tr>
<td>1</td>
<td>M₃₂</td>
<td>13,0</td>
<td>0,8</td>
<td>8,3</td>
</tr>
<tr>
<td>1</td>
<td>M₃₃</td>
<td>15,1</td>
<td>1,2</td>
<td>8,3</td>
</tr>
<tr>
<td>1</td>
<td>M₃₄</td>
<td>11,6</td>
<td>0,6</td>
<td>5,6</td>
</tr>
<tr>
<td>1</td>
<td>M₄₁</td>
<td>14,2</td>
<td>1,7</td>
<td>13,5</td>
</tr>
<tr>
<td>1</td>
<td>M₄₂</td>
<td>13,3</td>
<td>1,1</td>
<td>9,1</td>
</tr>
<tr>
<td>1</td>
<td>M₄₃</td>
<td>12,3</td>
<td>1,0</td>
<td>10,0</td>
</tr>
<tr>
<td>1</td>
<td>M₄₄</td>
<td>13,5</td>
<td>1,0</td>
<td>10,2</td>
</tr>
<tr>
<td>2</td>
<td>M₂₁-M₂₂</td>
<td>1,3</td>
<td>0,2</td>
<td>1,3</td>
</tr>
<tr>
<td>2</td>
<td>M₂₁-M₂₃</td>
<td>7,8</td>
<td>0,4</td>
<td>3,8</td>
</tr>
<tr>
<td>2</td>
<td>M₂₃-M₂₄</td>
<td>11,1</td>
<td>0,7</td>
<td>6,2</td>
</tr>
<tr>
<td>2</td>
<td>M₃₂-M₃₃</td>
<td>6,1</td>
<td>0,5</td>
<td>4,2</td>
</tr>
<tr>
<td>2</td>
<td>M₃₃-M₃₄</td>
<td>8,2</td>
<td>0,7</td>
<td>6,6</td>
</tr>
<tr>
<td>2</td>
<td>M₄₁-M₄₂</td>
<td>12,0</td>
<td>1,3</td>
<td>11,4</td>
</tr>
<tr>
<td>2</td>
<td>M₂₃-M₃₄</td>
<td>7,1</td>
<td>0,6</td>
<td>3,9</td>
</tr>
<tr>
<td>2</td>
<td>M₂₂-M₄₁</td>
<td>9,9</td>
<td>0,6</td>
<td>4,6</td>
</tr>
<tr>
<td>2</td>
<td>M₃₄-M₄₁</td>
<td>4,5</td>
<td>0,8</td>
<td>6,0</td>
</tr>
<tr>
<td>3</td>
<td>M₂₃-M₃₄-M₄₁</td>
<td>5,9</td>
<td>0,4</td>
<td>4,2</td>
</tr>
<tr>
<td>3</td>
<td>M₂₂-M₂₃-M₃₄</td>
<td>5,0</td>
<td>0,4</td>
<td>2,7</td>
</tr>
<tr>
<td>4</td>
<td>M₂₂-M₂₃-M₃₄-M₄₁</td>
<td>4,6</td>
<td>0,3</td>
<td>1,3</td>
</tr>
<tr>
<td>Number of Markers</td>
<td>Markers</td>
<td>Rms Values of Error Percentage in $\theta_2$</td>
<td>Rms Values of Error Percentage in $\theta_3$</td>
<td>Rms Values of Error Percentage in $s_{14}$</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------</td>
<td>----------------------------------------------</td>
<td>----------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>1 M$_{2,1}$</td>
<td>7,0</td>
<td>0,5</td>
<td>4,1</td>
<td></td>
</tr>
<tr>
<td>1 M$_{2,2}$</td>
<td>9,5</td>
<td>0,4</td>
<td>3,1</td>
<td></td>
</tr>
<tr>
<td>1 M$_{2,3}$</td>
<td>9,1</td>
<td>0,9</td>
<td>5,9</td>
<td></td>
</tr>
<tr>
<td>1 M$_{2,4}$</td>
<td>20,8</td>
<td>1,9</td>
<td>13,5</td>
<td></td>
</tr>
<tr>
<td>1 M$_{3,1}$</td>
<td>18,6</td>
<td>0,9</td>
<td>9,5</td>
<td></td>
</tr>
<tr>
<td>1 M$_{3,2}$</td>
<td>15,9</td>
<td>1,0</td>
<td>9,8</td>
<td></td>
</tr>
<tr>
<td>1 M$_{3,3}$</td>
<td>18,2</td>
<td>1,5</td>
<td>10,3</td>
<td></td>
</tr>
<tr>
<td>1 M$_{3,4}$</td>
<td>19,1</td>
<td>0,8</td>
<td>6,7</td>
<td></td>
</tr>
<tr>
<td>1 M$_{4,1}$</td>
<td>16,1</td>
<td>2,4</td>
<td>17,2</td>
<td></td>
</tr>
<tr>
<td>1 M$_{4,2}$</td>
<td>16,4</td>
<td>1,2</td>
<td>10,6</td>
<td></td>
</tr>
<tr>
<td>1 M$_{4,3}$</td>
<td>15,4</td>
<td>1,1</td>
<td>11,6</td>
<td></td>
</tr>
<tr>
<td>1 M$_{4,4}$</td>
<td>17,5</td>
<td>1,1</td>
<td>12,1</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{2,1}$-M$</em>{2,2}$</td>
<td>1,8</td>
<td>0,3</td>
<td>1,9</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{2,1}$-M$</em>{2,3}$</td>
<td>10,6</td>
<td>0,4</td>
<td>4,5</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{2,3}$-M$</em>{2,4}$</td>
<td>15,1</td>
<td>0,9</td>
<td>8,3</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{3,2}$-M$</em>{3,3}$</td>
<td>13,2</td>
<td>0,7</td>
<td>5,6</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{3,3}$-M$</em>{3,4}$</td>
<td>9,4</td>
<td>0,9</td>
<td>8,0</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{4,1}$-M$</em>{4,2}$</td>
<td>16,4</td>
<td>1,5</td>
<td>13,2</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{2,3}$-M$</em>{3,4}$</td>
<td>8,5</td>
<td>0,7</td>
<td>4,9</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{2,2}$-M$</em>{4,1}$</td>
<td>12,6</td>
<td>0,7</td>
<td>5,1</td>
<td></td>
</tr>
<tr>
<td>2 M$<em>{3,4}$-M$</em>{4,1}$</td>
<td>5,9</td>
<td>1,0</td>
<td>8,0</td>
<td></td>
</tr>
<tr>
<td>3 M$<em>{2,3}$-M$</em>{3,4}$-M$_{4,1}$</td>
<td>8,6</td>
<td>0,5</td>
<td>5,2</td>
<td></td>
</tr>
<tr>
<td>3 M$<em>{2,2}$-M$</em>{2,3}$-M$_{3,4}$</td>
<td>6,1</td>
<td>0,5</td>
<td>3,3</td>
<td></td>
</tr>
<tr>
<td>4 M$<em>{2,2}$-M$</em>{2,3}$-M$<em>{3,4}$-M$</em>{4,1}$</td>
<td>5,5</td>
<td>0,4</td>
<td>1,7</td>
<td></td>
</tr>
</tbody>
</table>
When the results obtained by using only one marker are compared, it is observed that the errors associated with markers M2,4 is a bit higher than the others. There are two reasons for that. First of all, marker M2,4 is in a different plane than the other markers. Hence, there is a depth difference, and it is farther, with respect to the camera, than the remaining markers on link 2. It is clear that there are measures errors associated with the vision system. Furthermore, when a marker is farther from the camera, the error increases. Another reason for the error is that marker M2,4 cannot always be viewed by the right camera. This marker can only be viewed by the right camera during half of the period. Therefore, the errors associated with marker M2,4 are large. Same comments are valid for markers M3,1, M3,2 and M3,3. These markers cannot be viewed by cameras at all times because generally link 2 and link 1, which is the fixed link, obscure these markers. This leads to position data for the markers M3,1, M3,2 and M3,3.

The next thing that is observed is that the markers on the slider lead to high error values, too. The first reason is that, the slider doesn’t move on a line since there is a very small clearance between slider and the track. In other words, the slider can move in the track in an upward direction, slightly, because of the forces applied by link 3. Furthermore, while using the flashback to synchronize the encoder with the camera, the flashback has obstructed the markers on the slider in some of the frames. This is another source of error.

Another source of error associated with all of the markers is that there is extensive blur in some of the frames. In fact, sometimes some of the markers cannot be detected as circles because of the high blur. In this case, the centroid of the blurred area is taken to be the center of the marker.

Another point is that the encoder yields the relative position (with respect to the first position of the mechanism). So, it is assumed that the first calculated data set for θ2 by using the vision system at time t = 0 sec is correct. The encoder values at t ≠ 0 is determined relative to the first value at the time t = 0 sec. Hence, if there is an error in
the \( \theta_2 \) value at \( t = 0 \), this error will affect the results. In addition to that, obviously, the encoder does not give precise results. For instance, it is observed that according to the encoder the period of the mechanism is not constant. However, according to the vision system data, the period of the mechanism is constant after two period. There is also some error involved in the rotation matrix (between the camera reference frame and the mechanism reference frame).

When the results obtained by using two markers are considered, one can state that, generally, the average error values are less than the errors obtained by using only one marker. However, it is observed that sometimes the average error value obtained by using two markers is between the average error values obtained by using these markers one by one. Also, one can observe that the errors are quite small when markers \( M_{2,1} \) and \( M_{2,2} \) are used.

When the results obtained by using three markers are considered, one can notice that the average error is lower than the ones obtained by using one or two markers except for the case where markers \( M_{2,1} \) and \( M_{2,2} \) are used.

The last case corresponds to the usage of 4 markers. As expected, it is seen that the average error values in all joint variables are lower than the ones where three markers are used. Therefore, one can conclude that when the number of markers used is increased, the average error values decrease.

If the average error values of the joint variables are compared with each other, one observes that the error associated with \( \theta_2 \) is higher, while the error associated with \( \theta_3 \) are lower. This is an expected result since angular speed of link 2 is higher than the angular speed of link 3. The maximum speed of \( \theta_2 \) is about 9 rad/s, while the maximum speed of \( \theta_3 \) is about 3.5 rad/s (See Figure 69). Recall that the cameras used in this thesis
are 30 fps cameras. Hence, using higher speed and higher resolution cameras will definitely reduce the errors.

The velocities $\dot{\theta}_2, \dot{\theta}_3, s'_{14}$ and the accelerations $\ddot{\theta}_2, \ddot{\theta}_3, s''_{14}$ obtained by using the encoder data and by using the vision system data for one marker only are presented in Figure 69. The results obtained by using the encoder and vision system are quite consistent for $s'_{14}$ and $s''_{14}$; less consistent for $\dot{\theta}_2$ and $\dot{\theta}_3$ and least consistent for $\dot{\theta}_2$ and $\ddot{\theta}_2$. 
Figure 69: Velocities and Accelerations obtained via the vision system and via the encoder
CHAPTER 9

CONCLUSION

This thesis consists of two main parts. In the first part, a vision system consisting of 2 inexpensive, commercial cameras has been developed. In this first part, the aim is to calibrate the cameras without using any calibration patterns. Here, it is assumed that the extrinsic parameters of the cameras (with respect to each other) are fixed and known during the recording. One needs to determine the intrinsic parameters of the cameras as well. In this thesis, self-calibration methods are used to calculate the intrinsic parameters. Two methods, which are the simplified Kruppa equations method and the equal eigenvalue method, are compared. It is shown that the equal eigenvalue method is more convenient to use if there exists noise. Hence, in this thesis the equal eigenvalue method is used to calculate the intrinsic parameters. Since only the intrinsic parameters are calculated and the extrinsic parameters are known, the vision system developed in this thesis is a pre-calibrated one. In order to realize the self-calibration methods, the cameras are turned around a fixed point independently in order to obtain the views of the scene at different orientations. The positions of the cameras are then fixed and the scene is recorded. Once the intrinsic and the extrinsic parameters are obtained, the coordinates of any 3D point in the scene can be calculated in any camera reference frame by using triangulation method.

In the second part of the thesis, kinematic analysis of a slider crank mechanism is realized by using the vision system that has been developed. Markers are attached on different links of the mechanism. By tacking the markers, position, velocity and acceleration analysis are performed. The results are compared with the results that are obtained by using an encoder which is attached to the input link of the mechanism. It is demonstrated that when the number of markers used is increased, better results can be obtained.

The result obtained in this study may be improved extensively by using cameras with better specifications. As it is known, the slider crank is a planar mechanism. The methods developed in this thesis may be easily extended to the kinematic analysis of
a spatial mechanism. In addition to that, gait analysis of human or animals can be realized by using the developed vision system without using a calibration pattern. And finally, the scene may be reconstructed by using only uncalibrated images. In other words, the intrinsic and extrinsic parameters may be obtained by using self-calibration methods only.
REFERENCES


[31] Bouguet, J. Y., “Camera Calibration Toolbox for MATLAB®” [online],
http://www.vision.caltech.edu/bouguetj/calib_doc/, last accessed date: 2/12/2013
Goethe University, 13 September 2010
[33] Tsai Roger Y., “A Versatile Camera Calibration Technique for High – Accuracy
3D Machine Vision Metrology Using Off – the Shelf TV Cameras and Lens”, IEEE
[34] Zhang Zhengyou, “Camera Calibration with One – Dimensional Objects”, IEEE
[35] Strum Peter, “Critical Motion Sequences for the Self – Calibration of Cameras
and Stereo Systems with Variable Focal Length” The 10th British Machine Vision
Conference (BMVC’ 99), pp. 63-72
[37] Triggs Bill, “Autocalibration and the Absolute Quadric”, International
Conference on Computer Vision & Pattern Recognition (CVPR’ 97), pp. 609-614, 1997
[38] Heyden Anders, Aström Kalle, “Euclidean Reconstruction from Image Sequences
with Varying and Unknown Focal Length and Principal Point”, Computer Vision and
Pattern Recognition, pp. 438-443, 1987
[39] Wilcox Walter, Physic 2435 Lecture Notes – Chapter 34, Baylor University,
https://bearspace.baylor.edu/Walter_Wilcox/www/courses/phy2435/chap34xxa.pdf
[40] Dr. Hasan Demirel, EE – 583 Digital Image Processing Lecture Notes, Eastern
Mediterranean University, http://faraday.ee.emu.edu.tr/ee583/Lectures/EE%20583-
Lecture01.pdf
[41] Alexandrov Alexander, “Corner Detection Overview and Comparison”,
Computer Vision 558 Notes, May 3, 2002
[42] Collins Robert, “Harris Corner Detector Reading: T&V Section 4.3”, CSE 486, Penn State


APPENDIX A

IMAGE AND IMAGE PROCESSING

A.1 Digital Image

A digital image is an electronic snapshot of a physical scene. It can be sampled as a grid of dots or imaging elements (pixel). Each pixel represents the color at a single point. So, each pixel has a particular location and value. Basically, a digital image is a numeric representation which consists of spatial coordinates and a range of colors.

Figure 70: An Example of Digital Image
A digital image can be represented as an $M \times N$ matrix. $M$ and $N$ can be any positive integers and they indicate the numbers of pixels in each direction. For example, in a 640x480 digital image, 640 is the number of pixels in the horizontal direction and 480 is the number of pixels in the vertical direction. However, $(x, y)$ represents the $x^{th}$ pixel.
on the vertical axis and the y<sup>th</sup> pixel on the horizontal axis. The color of a pixel is represented by \( f(x, y) \). Therefore, a digital image may be represented by the matrix as:

\[
F_p = \begin{bmatrix}
    f_{11} & f_{12} & \cdots & f_{1N} \\
    f_{21} & f_{22} & \cdots & f_{2N} \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{M1} & f_{M2} & \cdots & f_{MN}
\end{bmatrix}
\]  

(A.1)

where, \( f_{ij} \) is the color value, or intensity value, of pixel \((i, j)\)

Images can be achieved, examined, altered, displayed, transmitted or printed by digitizing them.

### A.2 Image Types

There are basically three types of images depending on its color, or intensity level, of its pixels. These images are, namely, binary image, grayscale image and full color image.

#### A.2.1 Binary Image

Binary images can be called black and white images. Binary images are made up of pixels each of which holds two discrete numbers, 0 or 1. Pixels with 0 value are displayed as black while pixels with 1 value are displayed as white. (See Figure 72).
Binary images are used in many applications because they are the simplest form of image to process. Binary images are especially useful for finding the silhouette of objects in the scene. Other useful applications of binary image are:

1. Identifying objects
2. Identifying orientation of objects
3. Interpreting text.

### A.2.2 Grayscale Image

In this type of image, image is composed of shades of gray. Black is the weakest intensity, while white is the strongest one. Like binary images, black pixels are represented by the integer value of 0 and white pixels are represented by the integer value of 1. Gray pixels can be represented by any number between 0 and 1. However, the gray level is generally indicated as a power of 2, i.e.,

\[ L = 2^k \]  

(A.2)
Here,

L: the number representing the gray level

k: the number of bits to store one pixel

Most image file formats support a minimum of 8 – bit grayscale which provides $2^8 = 256$ levels of gray per pixel. Some image formats support 16 – bit grayscale which provides $2^{16} = 65536$ levels of gray.

Figure 73: Representation of Grayscale Image

Grayscale images have similar application areas with binary images.
A.2.3 Full Color Image

The colors are identified by the response of a visual system to the presence of light at various wavelengths. For full color image, the image is stored as an array of $M \times N \times 3$. There are various reasons for specifying the colors numerically. First of all, identifying a color numerically makes accurate color reproduction easier. Digital imaging causes some color reproduction problems. One can ensure that everyone observes the same color with the help of numerically defined colors. Another reason is that few color names are widely recognized by English speakers. Hence, sometimes it is not possible to agree on appropriate color names. Therefore, some color models are standardized. In Figure 74, an example for a full color image is given.

Figure 74: Full Color Image
A.2.3.1 RGB Color Space

In this model, three primary colors are used to define color. These primary colors are red, green and blue which have single wavelengths. Magenta, cyan and yellow are called secondary colors of the RGB color space. You can see these colors in Figure 75.

![Primary and Secondary Colors of the RGB Model](image)

Figure 75: Primary and Secondary Colors of the RGB Model

Here, magenta is the combination of red and blue. Cyan is the combination of green and blue. Finally, yellow is the combination of red and green. The RGB model is based on the 3D Cartesian coordinate system, where the color of subspace of interest is the color tube shown in Figure 76.
As seen in Figure 76, the center of the cube, which is the point (0, 0, 0), represents the black color, while the point (1, 1, 1) represents the white color. In addition to that, some colors can only be obtained by subtractive color matching in the RGB color model.

A.2.3.2 CMY Color Space

CMY color model designates the cyan, magenta and yellow color model. The CMY color space uses these colors as the primary colors to define other colors. CMY color model is suitable for some printers and devices.
The conversion from RGB to CYM can be performed as follows:

\[
\begin{bmatrix}
C \\
M \\
Y
\end{bmatrix} = 
\begin{bmatrix}
1 \\
1 \\
1
\end{bmatrix} - 
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]  \hspace{1cm} (A.3)

Here, 1 represents the white color.
A.2.3.3 HSV Color Space

The HSV color model uses hue, saturation and value which are three important indicators in describing color for this model. So, HSV color model basically describes colors (i.e., hues) in terms of their shade, (i.e., saturation), and their brightness, (i.e., value). These three concepts are explained below.

1. Hue represents the purity of the color. It is expressed as a number from 0 to 360 degrees. Hue represents red, yellow, green, cyan, blue, magenta colors. Each color fills a part which is 60 degrees.
2. Saturation represents the measure of the degree to which a pure color is diluted by white light. It can be called the density of color. It ranges from 0 % to 100 %.
3. Value describes the brightness of the color from 0 % to 100 %.

These concepts can be seen in the following two figures.

Figure 78: HSV Color Circle [40]
A.3 Conversion between Image Types

Conversion between image types is an important function for image processing. MATLAB® has various functions for conversion between image types with the help of the Image Processing Toolbox. In this thesis, full color images are converted to grayscale images or binary images. Therefore, only conversion to grayscale and conversion to binary image (from full color image) are discussed in the following parts.
A.3.1 Conversion from Full Color Image to Grayscale Image

Images are expressed as a matrix of size M x N x 3 if they are in the full color image form. So, in order to convert a full color image to a grayscale one, the image should be expressed as an M x N matrix. If the image is given in the RGB color model, the conversion can be performed by using the following formula:

\[
\text{Gray} = 0.2989 \text{R} + 0.5870 \text{G} + 0.1140 \text{B} \quad (A.4)
\]

Here; R, G, B are the primary colors of RGB color model.

If the image is given in the HSV color model, the grayscale image of it can be obtained by eliminating hue and saturation values.

In this thesis, the conversion from full color image to grayscale image is performed by using MATLAB® which has various functions for this purpose. Among these functions, the “rgb2gray” function is chosen to perform the conversion.

A.3.2 Conversion from Grayscale Image to Binary Image

As explained earlier, a binary image can be expressed as a matrix of the size M x N, which contains only 1’s and 0’s. Therefore, a threshold function is needed to perform the conversion from grayscale to binary image. This threshold function may be described as follows:

\[
B(x, y) = \begin{cases} 
1 & \text{if } I(x, y) > T \\ 
0 & \text{if } I(x, y) < T 
\end{cases} \quad (A.5)
\]
Here,

\( B \): the value for binary image for the pixel \((x,y)\)

\( T \): the level of threshold

\( I \): grayscale value of pixel \((x,y)\)

### A.3.3 Conversion from Full Color Image to Binary Image

In order to convert a full color image to a binary image, firstly a conversion from full color image to a grayscale one should be performed. After obtaining the grayscale image, the conversion from grayscale image to binary image part, which is explained before, should be applied.

### A.4 Feature Points Detection

After performing conversion, it may be necessary to determine various features in the image. Surfaces, edges or corner points can be helpful for any purpose in computer vision. In this thesis, corner points are obtained from images. Therefore, in the following part, only corner points are discussed.

### A.4.1 Corner Points

Corners are the intersection points of two edges which have different orientations. Corners and edges are useful in order to define shapes in image planes. Corner points ease the matching process between images, pattern recognition and measurement since they are stable when viewpoint of the camera changes. In order to detect corner points,
there are different kinds of detectors such as the Harris corner detector or the Susan corner detector. Generally, the corners are located in the regions with large intensity changes in every direction. So, the corners can be identified by observing intensity values within a small window. If the window is on a corner point, shifting the window in any direction will result in large change in appearance or intensity.

A.4.1.1 Harris Corner Detector [42]

The Harris corner detector uses square or rectangular windows to search for the corner points. (See Figure 80).

If the function

$$m(\Delta x, \Delta y) = \sum_{(x,y)} w(x,y) \left( I(x,y) - I(x + \Delta x, y + \Delta y) \right)^2$$

(A.6)
assumes high values for any direction at some point, the point can be considered as a distinct point.

Here, \((\Delta x, \Delta y)\): shifting direction

\[ w(x, y): \text{the window function} \]

\[ I: \text{the intensity values at a point} \]

For constant patches, this function’s value should be near 0. However, for distinctive patches, the function should have large values. Therefore, there will be no change in all directions for flat regions, there will be no change in edge direction if the detector is on the edge and there will be a significant change in all directions if the detector is on a corner point.

Now, let us define the C matrix as follows.

\[
C = \begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2
\end{bmatrix}
\]  

(A.7)

Here, \(I_x\) and \(I_y\) are image gradients in the horizontal and vertical directions

As it is seen, \(C\) is a symmetric matrix. Also, it has two nonnegative significant eigenvalues. A corner is detected if the minimum of these two eigenvalues is larger than a threshold value. Instead of calculating the eigenvalues, the product of eigenvalues can be compared via following expression:

\[
\text{Det}(C) - k \cdot \text{trace}(C^2) = \lambda_1 \cdot \lambda_2 - k(\lambda_1 + \lambda_2)^2
\]

(A.8)
where, \( k \): a small number

\[ \lambda_1, \lambda_2 : \text{the eigenvalues of matrix } C \]

If this expression is maximized, the detector is on the corner point.

**A.4.1.2 Susan Corner Detector**

Unlike the Harris corner detector, the Susan corner detector uses a circle as a search window.

Near a corner point, univalue segment assimilating nucleus (USAN) significantly decreases and attains a local minimum at the corner point. For Susan corner detector, brightness difference threshold can be utilized for deciding if a pixel is in the circular

---

Figure 81: Susan Corner Detector [41]
mask belongs to USAN and geometrical threshold can be used to decide if a local minimum is a corner point.

In this thesis, Harris corner detector is used during this thesis. MATLAB® has various functions to find corner points. “Detectcornerpoints” or “Corner” functions of MATLAB® can be used to detect the corner points in the images.
APPENDIX B

SINGULAR VALUE DECOMPOSITION (SVD) [30]

The Singular Value Decomposition of an M x N matrix A with M ≥ N can be written as the product of an M x N orthogonal matrix U, an N x N diagonal matrix D with, non-negative diagonal elements, and the transpose of an N x N orthogonal matrix V such that

\[ A = UDV^T = \sum_{i=1}^{N} d_i U_i V_i^T \]  

(B.1)

Here, \( d_i \) is \( D(i, i) \) and \( U_i, V_i \) are the \( i \)th columns of matrices U and V.

Here, one is interested in computing \( \frac{\partial U}{\partial a_{ij}} \), \( \frac{\partial V}{\partial a_{ij}} \) and \( \frac{\partial D}{\partial a_{ij}} \) for every element \( a_{ij} \) of the matrix A. If we take derivative of equation (B.1), with respect to \( a_{ij} \), the following equation is obtained.

\[ \frac{\partial A}{\partial a_{ij}} = \frac{\partial U}{\partial a_{ij}} DV^T + U \frac{\partial D}{\partial a_{ij}} V^T + UD \frac{\partial V^T}{\partial a_{ij}} \]  

(B.2)

One can show that \( \frac{\partial a_{kl}}{\partial a_{ij}} = 0 \) while \( \frac{\partial a_{ij}}{\partial a_{ij}} = 1 \) for \( \forall (k, l) \neq (i, j) \)

Since U is an orthogonal matrix, one can write the following equation:

\[ UU^T = I \rightarrow \frac{\partial U^T}{\partial a_{ij}} U + U^T \frac{\partial U}{\partial a_{ij}} = \Omega_U^{ij} + \Omega_U^{ij} = 0 \]  

(B.3)

where \( \Omega_U^{ij} \) is given by
\[ \Omega_u^{ij} = U^T \frac{\partial U}{\partial a_{ij}} \]  

(B.4)

It is clearly seen that \( \Omega_u^{ij} \) is an anti–symmetric matrix. Similarly, an anti–symmetric matrix, \( \Omega_v^{ij} \), for V can be written as

\[ \Omega_v^{ij} = \frac{\partial v^T}{\partial a_{ij}} V \]  

(B.5)

By multiplying equation (B.2) by \( U^T \) and V from the left and right sides, respectively and combining it with equations (B.4) and (B.5), the following equation is obtained.

\[ U^T \frac{\partial A}{\partial a_{ij}} V = \Omega_u^{ij} D + \frac{\partial D}{\partial a_{ij}} + D \Omega_v^{ij} \]  

(B.6)

Since \( \Omega_u^{ij} \) and \( \Omega_v^{ij} \) are anti–symmetric matrices, all diagonal elements of them must be equal to zero. Also, the diagonal elements of \( \Omega_u^{ij} D \) and \( D \Omega_v^{ij} \) are equal to zero because D is a diagonal matrix. Therefore, equation (B.6) yields the derivative of singular values as:

\[ \frac{\partial d_k}{\partial a_{ij}} = u_{ik} v_{jk} \]  

(B.7)

where \( u_{ik} \) and \( v_{jk} \) designate elements of matrices U and V.
The elements of matrices $\Omega_{U}^{ij}$ and $\Omega_{V}^{ij}$ can be computed by solving a set of $2 \times 2$ linear system, which are derived from the off–diagonal elements of the matrices in equation (B.6) as follows:

\begin{align*}
d_l \hat{\Omega}_{U}^{ij} + d_k \hat{\Omega}_{U}^{kl} &= u_{ik} v_{jl} \\
d_k \hat{\Omega}_{U}^{ij} + d_l \hat{\Omega}_{U}^{kl} &= -u_{il} v_{jk} \\
\end{align*}

(B.8)

Here, index ranges are $k = 1……N$ and $l = i+1……N$. Thus, the parameters, defining the non – zero elements of $\Omega_{U}^{ij}$ and $\Omega_{V}^{ij}$, can be calculated by solved corresponding $2 \times 2$ linear system.

After computing $\Omega_{U}^{ij}$ and $\Omega_{V}^{ij}$, $\frac{\partial U}{\partial a_{ij}}, \frac{\partial V}{\partial a_{ij}}$ are computed as follows:

\begin{align*}
\frac{\partial U}{\partial a_{ij}} &= U \hat{\Omega}_{U}^{ij} \\
\frac{\partial V}{\partial a_{ij}} &= -V \hat{\Omega}_{V}^{ij} \\
\end{align*}

(B.9)
APPENDIX C
MORE CONSTRAINED MINIMIZATION ALGORITHM [1]

The statement is:

Minimization \( \| Ax \| \) subject to \( \| x \| = 1 \) and \( x = G\hat{w} \) for a given matrix \( G \) and some unknown vector \( \hat{w} \).

The condition that \( x = G\hat{w} \) for some \( \hat{w} \) means nothing more than that \( x \) lies in the span of the columns of \( G \). If \( G = UDV^T \) where \( D \) has \( r \) non-zero entries, then let \( U' \) be the matrix consisting of the first \( r \) columns of \( U \). By the way, one should specify that \( G \) has rank \( r \). Then, \( G \) and \( U' \) have the same column space. The solution can be found by setting \( x' \) to be unit vector that minimizes \( \| AU'x' \| \), then setting \( x = U'x' \).

If \( \hat{w} \) is required, it can be solved by \( G\hat{w} = x = U'x' \). The solution can be obtained as:

\[
\hat{w} = G^+x = G^+U'x'
\]  
(C.1)

where, \( G^+ \) is the pseudo-inverse of \( G \).

Summarization of complete algorithm:

1) Compute the SVD of \( G \) as \( G = UDV^T \), where non-zero values of \( D \) appear first down the diagonal.

2) Take \( U' \) as the matrix which is formed by first \( r \) column of \( U \).

3) Find the unit vector \( x' \) that minimizes \( \| AU'x' \| \) as:

\( x' \) is the last column of \( V \), where \( AU' = UDV^T \) is the SVD of \( AU' \).
4) Find required solution as $x = U'x'$.

5) If $\hat{w}$ is demanded, it can be computed as $\hat{w} = V'D'^{-1}x'$, where $V'$ is formed by first $r$ columns of $V$ and $D'$ is the upper $r \times r$ block of $D$. 
APPENDIX D

PROOF FOR ORDER OF FOURIER FITS

Fourier series equation is given by

\[ g^{fit}(t) = a_0 + \sum_{n=1}^{N} a_n \cos(nwt) + b_n \sin(nwt) \]  \hspace{1cm} (D.1)

When the variables \( t \) is changed as \( \theta = wt \), the equation (D.1) becomes

\[ g^{fit}(\theta) = a_0 + \sum_{n=1}^{N} a_n \cos(n\theta) + b_n \sin(n\theta) \]  \hspace{1cm} (D.2)

The critical points of \( g^{fit}(t) = 0 \) are obtained by solving the equation \( \frac{dg^{fit}(t)}{dt} = 0 \)

In case where the variable is \( \theta \), one should solve the following equation to find critical points.

\[ \left[ \frac{dg^{fit}(\theta)}{d\theta} \right] \frac{d\theta}{dt} = 0 \]  \hspace{1cm} (D.3)

If the derivative of equation (D.2) is taken and substituted into (D.3), one can obtain
\[ [\sum_{n=1}^{N} - na_n \sin(n\theta) + \sum_{n=1}^{N} n b_n \cos(n\theta)] [w] = 0 \] (D.4)

Also, one can write the multiple angle formula as follows:

\[ \sin(n\theta) = \sum_{k=0}^{n} \binom{n}{k} \cos^{k}(\theta) \sin^{n-k}(\theta) \sin[(n - k)(\pi/2)] \] (D.5)

\[ \cos(n\theta) = \sum_{k=0}^{n} \binom{n}{k} \cos^{k}(\theta) \sin^{n-k}(\theta) \cos[(n - k)(\pi/2)] \] (D.6)

And one can write the tangent half angle formula as follows:

\[ \sin(\theta) = (2 * t)/(1 + t^2) \] (D.7)

\[ \cos(\theta) = (1 - t^2)/(1 + t^2) \] (D.8)

Substituting the equations (D.5), (D.6), (D.7) and (D.8) into (D.4), a polynomial equation of degree 2N in \( t \) is obtained. Therefore, it is proven that the number of critical points of \( g^{it}(t) = 0 \) is 2N at most.
APPENDIX E

RESULTS OBTAINED BY USING ONLY ONE MARKER

Results obtained by using one marker only are presented starting from the next page in this section.
Figure 82: Results of the kinematic analysis by using marker M₂,₂
Figure 83: Results of the kinematic analysis by using marker $M_{2,2}$ [continued]
Figure 84: Results of the kinematic analysis by using marker M2,3
Figure 85: Results of the kinematic analysis by using marker M\textsubscript{2,3} [continued]
Figure 86: Results of the kinematic analysis by using marker M₂,₄
Figure 87: Results of the kinematic analysis by using marker M2,4 [continued]
Figure 88: Results of the kinematic analysis by using marker M₃₁.
Figure 89: Results of the kinematic analysis by using marker M3,1 [continued]

189
Figure 90: Results of the kinematic analysis by using marker M3,2
Figure 91: Results of the kinematic analysis by using marker M₃₂ [continued]
Figure 92: Results of the kinematic analysis by using marker M3,3
Figure 93: Results of the kinematic analysis by using marker M3.3 [continued]
Figure 94: Results of the kinematic analysis by using marker $M_{3,4}$
Figure 95: Results of the kinematic analysis by using marker M_{3,4} [continued]
Figure 96: Results of the kinematic analysis by using marker $M_{4,1}$
Figure 97: Results of the kinematic analysis by using marker M4,1 [continued]
Figure 98: Results of the kinematic analysis by using marker $M_{4,2}$
Figure 99: Results of the kinematic analysis by using marker M4,2 [continued]
Figure 100: Results of the kinematic analysis by using marker M4,3
Figure 101: Results of the kinematic analysis by using marker M43 [continued]
Figure 102: Results of the kinematic analysis by using marker M_{4,4}
Figure 103: Results of the kinematic analysis by using marker M_{4.4} [continued]
APPENDIX F

RESULTS OBTAINED BY USING TWO MARKERS

Results obtained by using two markers are presented starting from the next page in this section.
M₂₁ and M₂₃

Figure 104: Results of the kinematic analysis by using markers M₂₁ and M₂₃

206
Figure 105: Results of the kinematic analysis by using markers M2,1 and M2,3 [continued]
M_{2,3} and M_{2,4}

Figure 106: Results of the kinematic analysis by using markers M_{2,3} and M_{2,4}
Figure 107: Results of the kinematic analysis by using markers M_{2,3} and M_{2,4} [continued]
M₃,₂ and M₃,₃

Figure 108: Results of the kinematic analysis by using markers M₃,₂ and M₃,₃
Figure 109: Results of the kinematic analysis by using markers $M_{3,2}$ and $M_{3,3}$ [continued]
Figure 110: Results of the kinematic analysis by using markers M₃₃ and M₃₄
Figure 111: Results of the kinematic analysis by using markers M3,3 and M3,4 [continued]
Figure 112: Results of the kinematic analysis by using markers $M_{4,1}$ and $M_{4,2}$
Figure 113: Results of the kinematic analysis by using markers M₄₁ and M₄₂
[continued]
M2,3 and M3,4

Figure 114: Results of the kinematic analysis by using markers M2,3 and M3,4
Figure 115: Results of the kinematic analysis by using markers M_{2,3} and M_{3,4} [continued]
M$_{2,2}$ and M$_{4,1}$

Figure 116: Results of the kinematic analysis by using markers M$_{2,2}$ and M$_{4,1}$
Figure 117: Results of the kinematic analysis by using markers M_{2,2} and M_{4,1}  
[continued]
Figure 118: Results of the kinematic analysis by using markers M_{3,4} and M_{4,1}
Figure 119: Results of the kinematic analysis by using markers $M_{3,4}$ and $M_{4,1}$
[continued]
APPENDIX G

RESULTS OBTAINED BY USING THREE MARKERS

Results obtained by using three markers are presented starting from the next page in this section.
**M_{2,2}, M_{2,3} and M_{3,4}**

Figure 120: Results of the kinematic analysis by using the marker triple $M_{2,2}$, $M_{2,3}$ and $M_{3,4}$
Figure 121: Results of the kinematic analysis by using the marker triple M_{2,2}, M_{2,3} and M_{3,4} [continued]