3D MARKER TRACKING FOR HUMAN GAIT ANALYSIS

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

BY

CAN KÜÇÜK

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

DECEMBER 2005

Approval of Graduate School of Natural and Applied Science.

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet ERKMEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Uğur HALICI
Supervisor

Examining Committee Members:

Prof. Dr. Kemal LEBLEBİCİOĞLU	(METU, EE)
Prof. Dr. Uğur HALICI	(METU, EE)
Assist. Prof. İlkay ULUSOY	(METU, EE)
Assist. Prof. İlhan KONUKSEVEN	(METU, ME)
Umur AKINCI	(ASELSAN)

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.

Can KÜÇÜK

ABSTRACT

3D MARKER TRACKING FOR HUMAN GAIT ANALYSIS

Küçük, Can
M.S., Department of Electric and Electronics Engineering
Supervisor: Prof. Dr. Uğur Halıcı

December 2005, 85 pages

This thesis focuses on 3D marker tracking for human gait analysis. In KISS Gait Analysis System at METU, a subject's gait is recorded with 6 cameras while 13 reflective markers are attached at appropriate locations on his/her legs and feet. These images are processed to extract 2 dimensional (2D) coordinates of the markers in each camera. The 3 dimensional (3D) coordinates of the markers are obtained by processing the 2D coordinates of the markers with linearization and calibration algorithms. Then 3D trajectories of the markers are formed using the 3D coordinates of the markers. In this study, software which takes the 2D coordinates of markers in each camera and processes them to form the 3D trajectories of the markers is developed. Kalman Filter is used in formation of 3D trajectories. The results are found to be satisfactory.

Keywords: 3D Marker Tracking, Gait Analysis, Kalman Filter

ÖZ

İNSAN YÜRÜYÜŞ ANALİZİ İÇİN 3 BOYUTLU İŞARETLEYİCİ İZLEME

Küçük, Can
Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Uğur Halıcı

Aralık 2005, 85 sayfa

Bu tez, insan yürüyüş analizi için 3 boyutlu işaretleyici izleme üzerine odaklanmıştır. ODTÜ'deki KISS Yürüyüş Analiz Laboratuarı'nda, bacak ve ayaklarının uygun konumlarına ışık yansıtan işaretleyiciler yerleştirilen deneğin yürüyüşü 6 kamera ile kaydedilir. Bu görüntüler işaretleyicilerin her bir kameradaki 2 boyutlu (2B) koordinatlarını bulmak üzere işlenir. İşaretleyicilerin 2B koordinatlarının doğrusallaştırma ve kalibrasyon algoritmaları ile işlenmesiyle işaretleyicilerin 3 boyutlu (3B) koordinatları elde edilir. Daha sonra işaretleyicilerin 3B koordinatları kullanılarak işaretleyicilerin 3B yörüngeleri oluşturulur. Bu çalışmada, her bir kameradaki işaretleyicilerin 2B koordinatlarını alan ve onları işaretleyicilerin 3B yörüngelerini oluşturmak üzere işleyen bir yazılım geliştirilmiştir. 3B yörüngelerin oluşturulması sırasında Kalman Filtresi kullanılmıştır. Sonuçlar tatmin edici bulunmuştur.

Anahtar Kelimeler: 3 Boyutlu İşaretleyici İzleme, Yürüyüş Analizi, Kalman Filtresi.

To My Family

ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Dr. Uğur Halıcı for her supervision and guidance throughout this study.

Also thanks to ASELSAN Inc. Microelectronics, Guidance and Electro-Optics Division for support for this thesis.

I am thankful to all my family for their encouragement, support and patience in the preparation of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

CHAPTER 1

INTRODUCTION

1.1 GAIT ANALYSIS

Gait analysis is the study of walking and often referred to as "the quantitative description of all mechanical aspects of walking" [11]. Gait analysis is therefore a multidisciplinary field where people from the fields of engineering, physical medicine and rehabilitation, orthopedics, neurology and physiology are working on [1]. Main objective of gait analysis is to obtain quantitative information about the mechanics of the gait, including positions, angles, forces, etc. [12].

1.2 KISS GAIT ANALYSIS SYSTEM

KISS, Kinematic Support System, is the Gait Analysis System at METU. This is a 6-camera system with force plates to measure ground forces. To distinguish the markers placed on a subject whose gait is to be analyzed, the cameras are equipped with IR Filters which only let the IR light reach the CCD. Markers are passive balls wrapped with reflective material that reflects the IR light emitted from the IR LEDs placed at each camera.

Markers used in the experiments are shown in Figure 1.1 and the cameras equipped with IR Filters are shown in Figure 1.2.

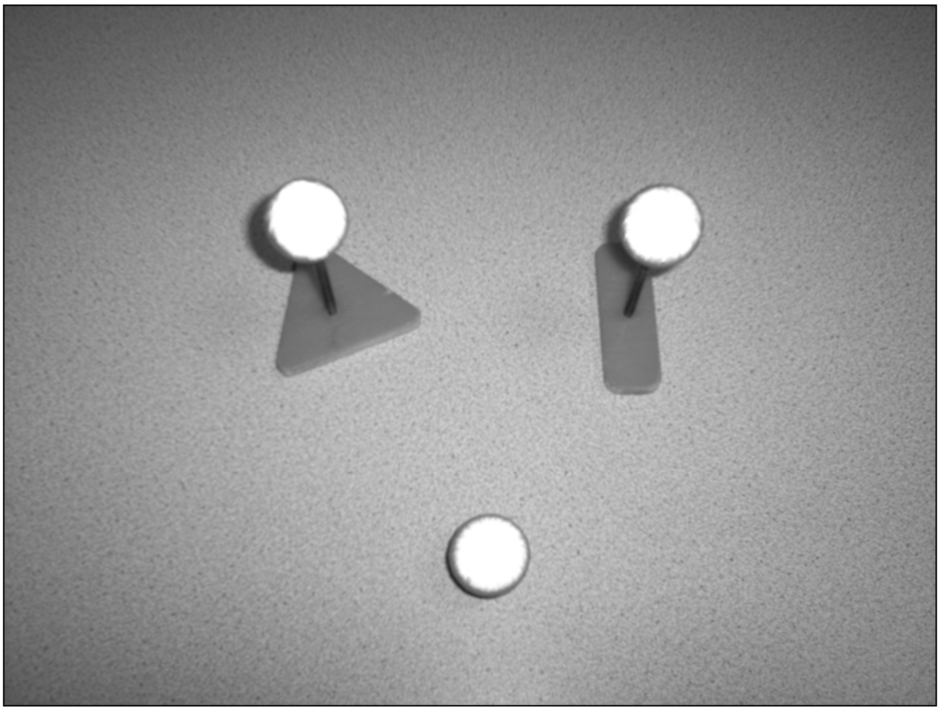


Figure 1.1 The Markers

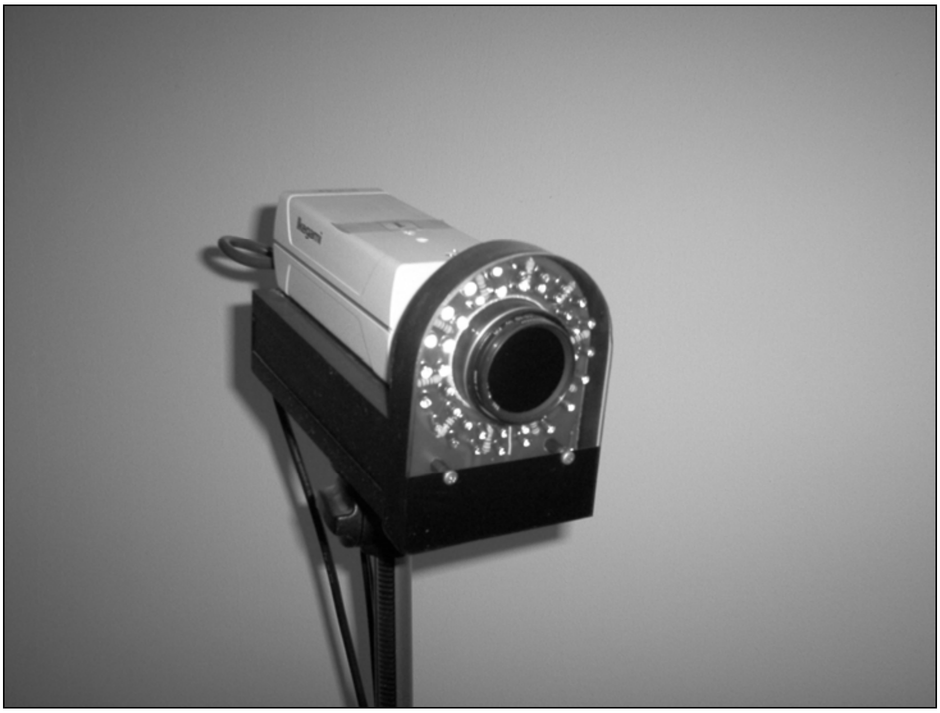


Figure 1.2 CCD Camera with IR LEDs

The marker positions on the subject's legs and feet are given in the Figure 1.3 (a) and (b), which show the side and front views, respectively. Figure 1.3 (c) shows the skeleton formed by markers.

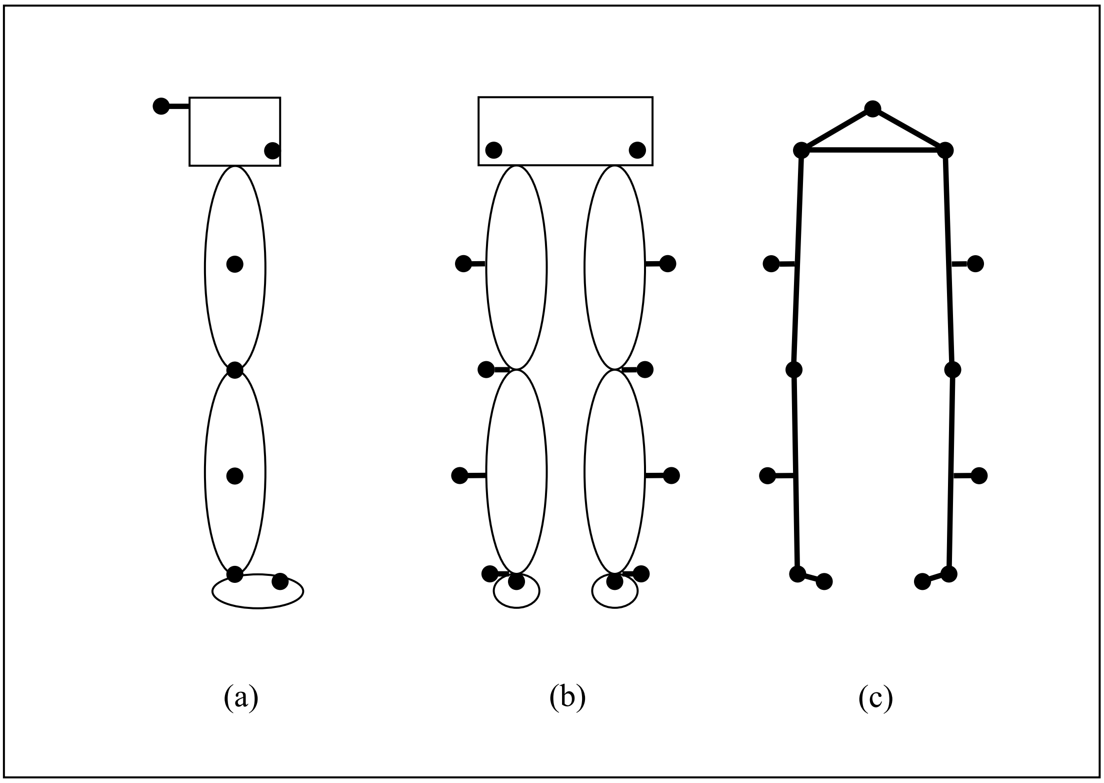


Figure 1.3 Marker Positions on the Legs and Feet from
(a) Side View and (b) Front View and (c) the Skeleton Formed by Markers

For the gait analysis experiment, 13 markers are placed on the subject's legs and feet. The subject walks through a path while the cameras are recording. At the same time, forces on the force plates on the floor are measured and recorded synchronously to the cameras. Recorded image sequences and measured force values are analyzed together using various gait analysis software.

There has been a lot of academic work done on this system at METU. The work presented in this thesis is also a continuation of the academic work for KISS.

H. C. Güler developed the mathematical tools in order to setup this laboratory. Mathematical formulations were presented to calibrate the cameras, to calculate 3D coordinates of the markers, anatomical angles between segments and the joint moments [1].

M. S. Shafiq prepared software which analyzes the 2D images to extract 2D marker position information, constructs 3D coordinates of the markers and then finds the 3D trajectories [3] [13].

Y. Karpat programmed and tested different camera calibration algorithms with the hardware of KISS [2].

B. Söylemez has evaluated and refined the gait analysis protocol used in the laboratory [11].

B. B. Kentel developed a database for KISS and confidence and prediction bands were constructed in order to interpret gait data [12].

1.3 SCOPE OF THE THESIS

This thesis focuses on 3D marker tracking for human gait analysis. In this study, software which takes the 2D marker positions of each of the 6 cameras and processes them to form the 3D marker trajectories is developed. The software uses the linearization parameters that were already generated for each camera to linearize the 2D marker positions and uses the calibration parameters that were already generated for each camera to form 3D marker positions from multiple images of the same marker from different cameras. 3D marker tracks are found from the 3D marker positions using the Kalman Filter.

1.4 ORGANIZATION OF THE THESIS

In the first chapter an introduction to gait analysis and the gait analysis system at METU is given. Previous work on KISS system is given also in this chapter.

In the second chapter the background information necessary to understand the methods applied is given briefly. Camera linearization, camera calibration, stereo vision, epipolar lines and Kalman Filter are explained briefly in this chapter.

The algorithms implemented in this thesis are given in the third chapter. These include the elimination of false detections in 2D images, linearization of 2D images, calibration of cameras, finding epipolar lines, matching points in 2D images, finding 3D points, finding 3D point groups, finding 3D point group initial tracks, and finally using Kalman Filter with 3D point group initial tracks.

Chapter 4 gives brief information about the graphical interface of the 3D Marker Tracking Software developed.

Chapter 5 gives the experimental results after the implementation of the methods mentioned.

Chapter 6 gives the conclusion and possible future work about the work done in this thesis.

CHAPTER 2

BACKGROUND

2.1 INTRODUCTION

This thesis is intended for forming 3D tracks of markers placed on a subject's legs and feet captured while the subject is walking in a gait analysis laboratory, specifically, KISS Gait Analysis Laboratory at METU. The input data is 2D marker positions in 6 cameras, linearization and calibration parameters for each of the 6 cameras. The output data is 3D marker tracks. This chapter gives the background information which is essential to understand the methods used in this thesis. Section 2.2 gives fundamental information about camera linearization and Section 2.3 briefly explains camera calibration. Stereo vision and epipolar lines are mentioned in Section 2.4 and Section 2.5, respectively. Section 2.6 is devoted to Kalman Filter.

2.2 CAMERA LINEARIZATION

Linearization is the process needed to correct the image coordinates which are displaced because of distortions [2]. In this thesis, the algorithm reported in [14] is used for camera linearization. A summary of the algorithm used is given in this section, while the details could be reached in the referenced material [2] [14].

Systematic errors, which are due to lens deformation, nonorthogonality of image axes and non-linear lens distortions, radial or tangential, result in displacement of the image coordinates. To correct the coordinates of the markers for nonlinearities of the systematic errors, linearization is applied to each marker coordinate in each camera. For the linearization process, linearization parameters must be generated for each camera. Otherwise, calculation of 3D point coordinates from 2D point coordinates will be affected by the lens distortions and so accuracy in 3D point coordinates will be decreased. After linearization, the measured 2D coordinates may be assumed to be the correct 2D coordinates [2].

In the linearization algorithm used in this thesis, a linearization grid in the form of *M* rows and *N* columns is produced. *MxN* circles covered with reflective material are placed on each corner of this grid. Firstly, a stick with reflective material on one end is mounted in the middle of the grid plane and the camera is placed in front of the grid such that the grid covers most of the image in the camera and the tip of stick with reflective material is observed in the middle of the grid. Then the stick is removed and the image of the grid is recorded for some duration. The average of the images recorded in this duration is used in a linearization algorithm to correct the lens distortion errors [2].

The linearization algorithm is based on finding a mapping between "the distorted grid", which is the average of the recorded images mentioned above, and "the optimal grid", which is representing the undistorted grid corresponding to the given distorted grid. This mapping is performed using a 4th order polynomial function and the 30 linearization parameters, 15 for horizontal and 15 for vertical direction, which are the coefficients of this mapping function. The procedure for finding these linearization parameters for the distorted grid is a three step procedure, starting first by forming the primary grid, then forming the optimal grid, and finally finding the parameters to change the distorted grid to the optimal grid. After finding these parameters, point coordinates in distorted images may be corrected using this 4th order polynomial with known coefficients [2].

The primary grid is formed by first finding the point which is the nearest point to the center of the distorted grid. As it can be assumed that distortions in the middle of the image are small compared to distortions in the outer parts of the image, the general horizontal and vertical distances in the grid can be taken as the distances between the points at the center of the grid. Therefore the average of the horizontal distances just at the left and right of the center point can be taken as the general horizontal distance in the grid and the average of the vertical distances just at the top and bottom of the center point as the general vertical distance in the grid. We form a new grid where the location of the center point is the same as the distorted grid but all the other points in the grid are located such that the general horizontal and vertical distances between the points are the calculated average horizontal and vertical distances, respectively. This new grid is called the primary grid [2].

The optimal grid is a scaled, rotated, and translated version of the primary grid. It represents the undistorted grid to which the distorted grid is tried to be converted. The transformation applied to the primary grid to obtain the optimal grid in homogeneous coordinates is

$$\begin{bmatrix} u_i'' \\ v_i'' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\alpha & s\sin\alpha & u_0 \\ -s\sin\alpha & s\cos\alpha & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i' \\ v_i' \\ 1 \end{bmatrix} \tag{2.1}$$

where

i=1, 2, ... MN where *MN* is the total number of points on the grid,

(u_i'',v_i'') are the point coordinates in the optimal grid corresponding to the point coordinates (u_i,v_i) in the distorted grid,

(u_i',v_i') are the point coordinates in the primary grid corresponding to the point coordinates (u_i,v_i) in the distorted grid,

s is a scale factor,

α is a rotation,

u_0 and v_0 are translations in two directions [2].

The optimal grid which is closest to the distorted grid is formed using least squares technique in finding the scale, rotation and translation factors given in Equation (2.1). After finding the optimum scale, rotation, and translation factors for this transformation, the optimal grid is formed. Details of this computation are explained in [2].

The optimal grid, which represents the undistorted grid to which the distorted grid is tried to be converted, is prepared and the mapping from the distorted grid to the optimal grid needs to be found.

It is assumed that the corrected coordinates are 4th order polynomial functions of the distorted coordinates. Assumed polynomial function is

$$\begin{aligned} u''' &= x_0 + x_1u + x_2v + x_3u^2 + x_4uv + x_5v^2 + x_6u^3 + x_7u^2v + x_8uv^2 + \\ &\quad x_9v^3 + x_{10}u^4 + x_{11}u^3v + x_{12}u^2v^2 + x_{13}uv^3 + x_{14}v^4 \\ v''' &= y_0 + y_1u + y_2v + y_3u^2 + y_4uv + y_5v^2 + y_6u^3 + y_7u^2v + y_8uv^2 + \\ &\quad y_9v^3 + y_{10}u^4 + y_{11}u^3v + y_{12}u^2v^2 + y_{13}uv^3 + y_{14}v^4 \end{aligned} \tag{2.2}$$

where

(u''',v''') are the point coordinates in the corrected image corresponding to the point coordinates *(u,v)* in the distorted image,

$x_0 \ldots x_{14}$ are the horizontal linearization parameters,

$y_0 \ldots y_{14}$ are the vertical linearization parameters [2].

Writing these equations for each point pair in the distorted grid and the optimal grid in matrix form,

$$\boldsymbol{Z} \,.\, \boldsymbol{x} = \boldsymbol{u} \tag{2.3}$$

and

$$\boldsymbol{Z} \,.\, \boldsymbol{y} = \boldsymbol{v} \tag{2.4}$$

where

$$Z = \begin{bmatrix} 1 & u_1 & v_1 & u_1^2 & u_1 v_1 & v_1^2 & . & . & v_1^4 \\ 1 & u_2 & v_2 & u_2^2 & u_2 v_2 & v_2^2 & . & . & v_2^4 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 1 & u_{MN} & v_{MN} & u_{MN}^2 & u_{MN} v_{MN} & v_{MN}^2 & . & . & v_{MN}^4 \end{bmatrix},$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ . \\ . \\ x_{14} \end{bmatrix}, \; y = \begin{bmatrix} y_0 \\ y_1 \\ . \\ . \\ y_{14} \end{bmatrix},$$

and

$$u = \begin{bmatrix} u_0'' \\ u_1'' \\ . \\ . \\ u_{MN}'' \end{bmatrix}, \; v = \begin{bmatrix} v_0'' \\ v_1'' \\ . \\ . \\ v_{MN}'' \end{bmatrix},$$

where

MN is the total number of points on the grid,

(u_i'',v_i'') are the point coordinates in the optimal grid corresponding to the point coordinates *(u_i,v_i)* in the distorted grid [2].

Solving Equations (2.3) and (2.4) for ***x*** and ***y***, 30 linearization parameters will be found, 15 for horizontal and 15 for vertical directions [2].

Using these parameters as the known constant coefficients for the mapping function given in Equation (2.2), point coordinates from each distorted image should be corrected before further analysis using these point coordinates [2].

2.3 CAMERA CALIBRATION

Basically cameras are devices which take a 2D snapshot of our 3D world. In Figure 2.1, it can be seen that both of the points $P_{g1}(x_1,y_1,z_1)$ and $P_{g2}(x_2,y_2,z_2)$ in the global coordinate system are represented by the same point, which is the point $P_i(u,v)$, in the image coordinate system. In order to be able to use the information from the cameras in measuring positions of real world points, the mapping from the 3D global system to the 2D image system has to be known. There are different techniques of calibration in the literature.

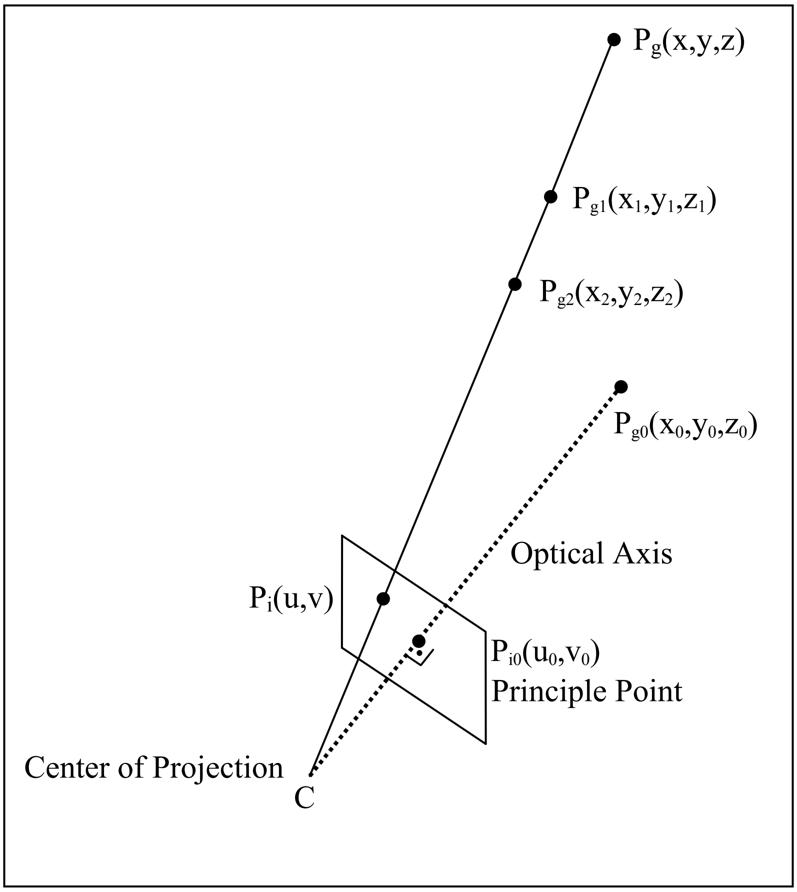


Figure 2.1 Camera Coordinate System and Object Coordinate System

The internal and external camera parameters are required to be known to be able to calculate the 3D coordinates of a point in space using its 2D coordinates in the image. The six parameters for the position and orientation of a camera coordinate system with respect to a global coordinate system are the external camera parameters and the internal camera parameters are the focal length of the lens, the two coordinates of the principal point and the aspect ratio of the image [1].

The Direct Linear Transformation (DLT) is a method used for camera calibration. DLT describes a linear mapping between the 3D object space and the 2D image space in terms of 11 parameters and these parameters implicitly include the internal and external camera parameters [1] [7]. In this study, the DLT method was used to calibrate the cameras.

The mapping between 3D global coordinate system and 2D image coordinate system is shown in Figure 2.2 [6].

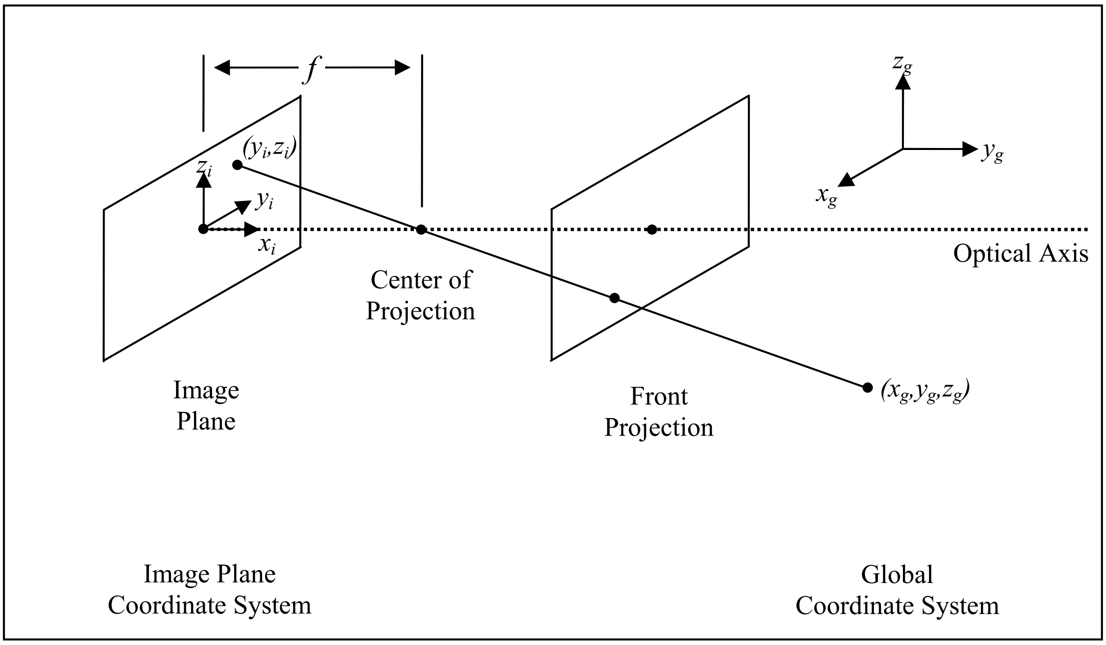


Figure 2.2 Object Point - Image Point Transformation

The transformation from 3D global coordinate system to 2D image plane coordinate system is given in homogeneous coordinates in [6] as

$$x_i = Ax_g \tag{2.5}$$

where

$$x_i = \begin{bmatrix} wy_i \\ wz_i \\ w \end{bmatrix},\ x_g = \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix},$$

and

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$$

where a_{ij} are the camera calibration parameters.

Due to the use of homogeneous coordinates, ***A*** matrix is not unique. It is possible to set a_{34} to 1 and use the remaining 11 parameters [6].

Note that there are notation differences between sources in the literature about the naming of the 11 calibration parameters. One notation is a_{ij} where *i=1,2,3* and *j=1,2,3,4* and one other notation is a_i where *i=1,2, ... 11.*

Rewriting the Equation (2.5) for a point in 3D global coordinates *(X,Y,Z)* and its 2D image coordinates *(u,v)* with $a_{34}=1$ [6] [1],

$$u = \frac{a_1X + a_2Y + a_3Z + a_4}{a_9X + a_{10}Y + a_{11}Z + 1} \tag{2.6}$$

$$v = \frac{a_5X + a_6Y + a_7Z + a_8}{a_9X + a_{10}Y + a_{11}Z + 1} \tag{2.7}$$

where the parameters a_1, a_2, ... a_{11} are the camera calibration constants (DLT parameters).

The 3D coordinates *(X,Y,Z)* are specified either in meters or millimeters, and *u* and *v* are the pixel coordinates of the point on the image plane [1]. The derivation of Equations (2.6) and (2.7) are given in [1].

There are modifications to the Equations (2.6) and (2.7) given above. DLT method was modified later into "Modified DLT" or MDLT in [20]. Or Equations (2.6) and (2.7) can be modified into Equations (2.8) and (2.9) to include the optical errors from the lens

$$u - \Delta u = \frac{a_1 X + a_2 Y + a_3 Z + a_4}{a_9 X + a_{10} Y + a_{11} Z + 1} \tag{2.8}$$

$$v - \Delta v = \frac{a_5 X + a_6 Y + a_7 Z + a_8}{a_9 X + a_{10} Y + a_{11} Z + 1} \tag{2.9}$$

where the parameters a_1, a_2, ... a_{11} are the standard camera calibration constants (standard DLT parameters), Δu and Δv are the terms for optical errors [8].

There are five additional calibration parameters, a_{12}, a_{13}, ... a_{16} in Δu and Δv in Equations (2.8) and (2.9). a_{12}, a_{13}, a_{14} are given to be related to the optical distortion while a_{15} and a_{16} to the de-centering distortion in [8] with reference to [19].

In this study, the additional calibration parameters, a_{12}, a_{13}, ... a_{16} are not used, only 11 DLT parameters, a_1, a_2, ... a_{11} in Equations (2.6) and (2.7) are used.

To find the DLT parameters, calibration points with known *(X,Y,Z)* coordinates are used. Each calibration point and the corresponding image point forms a point pair and each point pair yields 2 equations of 11 unknowns. Therefore, to find the 11 DLT parameters, at least six calibration points with known *(X,Y,Z)* coordinates are needed. More calibration points will decrease measurement errors [6] [1].

Equations (2.6) and (2.7) can be rewritten for each control point as:

$$X_k a_1 + Y_k a_2 + Z_k a_3 - u_k X_k a_9 - u_k Y_k a_{10} - u_k Z_k a_{11} + a_4 - u_k = 0 \qquad (2.10)$$

$$X_k a_5 + Y_k a_6 + Z_k a_7 - v_k X_k a_9 - v_k Y_k a_{10} - v_k Z_k a_{11} + a_8 - v_k = 0 \qquad (2.11)$$

where k = *1, 2 ... N*, and *N* is number of calibration points used. The *2N* equations given in Equations (2.10) and (2.11) can be written in a matrix form as follows:

$$Da = b \qquad (2.12)$$

where
D is a *2Nx11* matrix consisting of the known coordinates X_k, Y_k, Z_k, u_k, v_k,
a is the *11x1* vector of unknown DLT parameters,
b is a *2Nx1* vector consisting of the known coordinates u_k and v_k [1].

A least squares solution for Equation (2.12) was carried out using the pseudo-inverse technique:

$$a = (D^T D)^{-1} D^T b \qquad (2.13)$$

Information about the pseudo inverse technique can be found in [6].

Solution of Equation (2.13) yielded the unknown calibration parameters, which enabled the determination of the 3D coordinates of an arbitrary point in the space using its image coordinates.

2.4 STEREO VISION

Stereo vision is the case where a 3D point has a projection on the image planes of two cameras, as in shown in Figure 2.3. In this figure, *$P_1(u_1,v_1)$* and *$P_2(u_2,v_2)$* are the points in image planes of the two cameras, namely IP1 and IP2, corresponding to the point *P(x,y,z)* in 3D global coordinate system.

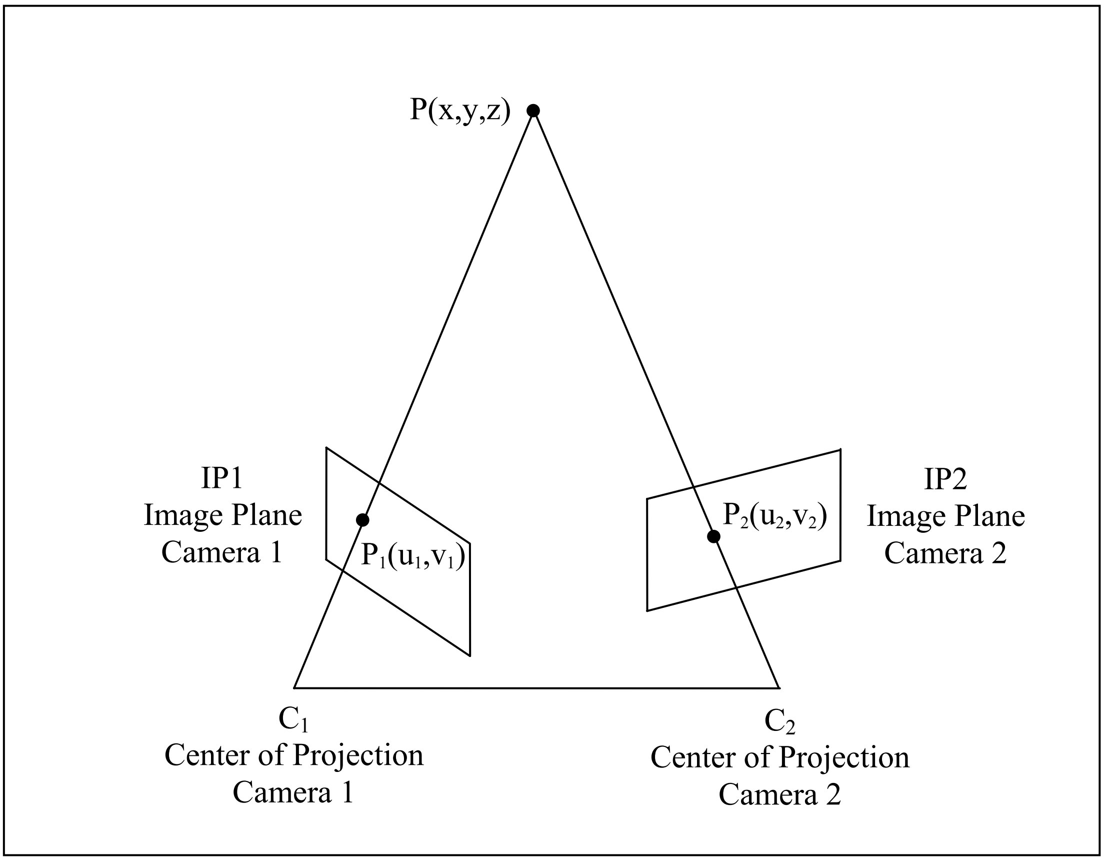


Figure 2.3 Stereo Vision

The problem of finding the corresponding points in two images in stereo vision will be explained in Section 2.5. In this section, stereo vision will be explained assuming that correspondence between points in two images has already been done. Therefore point pairs in two cameras corresponding to the same 3D point, like the points *$P_1(u_1,v_1)$* and *$P_2(u_2,v_2)$* in Figure 2.3, are available.

Equations (2.6) and (2.7) given in Section 2.3 can be rewritten in another from as

$$Xa_1 + Ya_2 + Za_3 - u_k Xa_9 - u_k Ya_{10} - u_k Za_{11} + a_4 - u_k = 0 \qquad (2.14)$$

$$Xa_5 + Ya_6 + Za_7 - v_k Xa_9 - v_k Ya_{10} - v_k Za_{11} + a_8 - v_k = 0 \qquad (2.15)$$

where

(X,Y,Z) are the 3D point coordinates,

(u_k,v_k), *k=1,2* are the 2D point coordinates in image *k* corresponding to this 3D point,

a_i, *i=1,2, ... 11* show the calibration parameters.

There are two equations for each point in each camera as given in Equations (2.14) and (2.15), making a total of 4 equations for two 2D point pairs *(u_1,v_1)* and *(u_2,v_2)* in two cameras corresponding to the 3D point *(X,Y,Z)*. These 4 equations may be written in matrix form as

$$Px = F \qquad (2.16)$$

where

$$P = \begin{bmatrix} a_{1,1} - u_1 a_{9,1} & a_{2,1} - u_1 a_{10,1} & a_{3,1} - u_1 a_{11,1} \\ a_{5,1} - v_1 a_{9,1} & a_{6,1} - v_1 a_{10,1} & a_{7,1} - v_1 a_{11,1} \\ a_{1,2} - u_2 a_{9,2} & a_{2,2} - u_2 a_{10,2} & a_{3,2} - u_2 a_{11,2} \\ a_{5,2} - v_2 a_{9,2} & a_{6,2} - v_2 a_{10,2} & a_{7,2} - v_2 a_{11,2} \end{bmatrix},$$

$$x = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \text{ and}$$

$$F = \begin{bmatrix} u_1 - a_{4,1} \\ v_1 - a_{8,1} \\ u_2 - a_{4,2} \\ v_2 - a_{8,2} \end{bmatrix}.$$

where

(X,Y,Z) are the 3D point coordinates,

(u_k,v_k), $k=1,2$ are the 2D point coordinates in image k corresponding to this 3D point,

$a_{i,k}$, $i=1,2, \dots 11$, $k=1,2$ show the calibration parameters a_i for the camera k [6].

From Equation (2.16), least squares estimate of $\boldsymbol{x}$, denoted by $\boldsymbol{x^e}$, may be found by using the pseudo inverse of $\boldsymbol{P}$, denoted by $\boldsymbol{P^+}$, as follows [6]:

$$x^e = P^+ F \tag{2.17}$$

Alternately, the elimination of one equation leaves three equations with three unknowns and a simple matrix inversion approach may be used to solve these equations [6].

2.5 EPIPOLAR LINES

Epipolar line concept, which is very useful in stereo vision, will be explained briefly in this thesis but more information can be found in [6]. In Figure 2.4 it can be seen that the points $P_1(x_1,y_1,z_1)$ and $P_2(x_2,y_2,z_2)$ in the 3D world coordinate system are projected to the same point, $P_0(u_0,v_0)$, in IP1, the image plane of camera 1. Thinking in reverse, the point $P_0(u_0,v_0)$ in IP1 may be the projection of $P_1(x_1,y_1,z_1)$ or $P_2(x_2,y_2,z_2)$ or any point on the line C_1P_1. The projection of the points $P_1(x_1,y_1,z_1)$ and $P_2(x_2,y_2,z_2)$ on IP2 is P_1' and P_2', respectively. The possible source points of the projection $P_0(u_0,v_0)$ in IP1, namely all the points on the line C_1P_1, forms a line when projected on IP2, namely the line $P_1'P_2'$. This line is called the "epipolar line" [6]. This epipolar line is formed by the points which are the possible correspondence points of the point $P_0(u_0,v_0)$ in IP1 when projected on IP2. The "epipolar plane" is the plane formed by the line C_1P_1 and the point C_2 [6]. In fact, the epipolar line is the intersection of the epipolar plane with IP2.

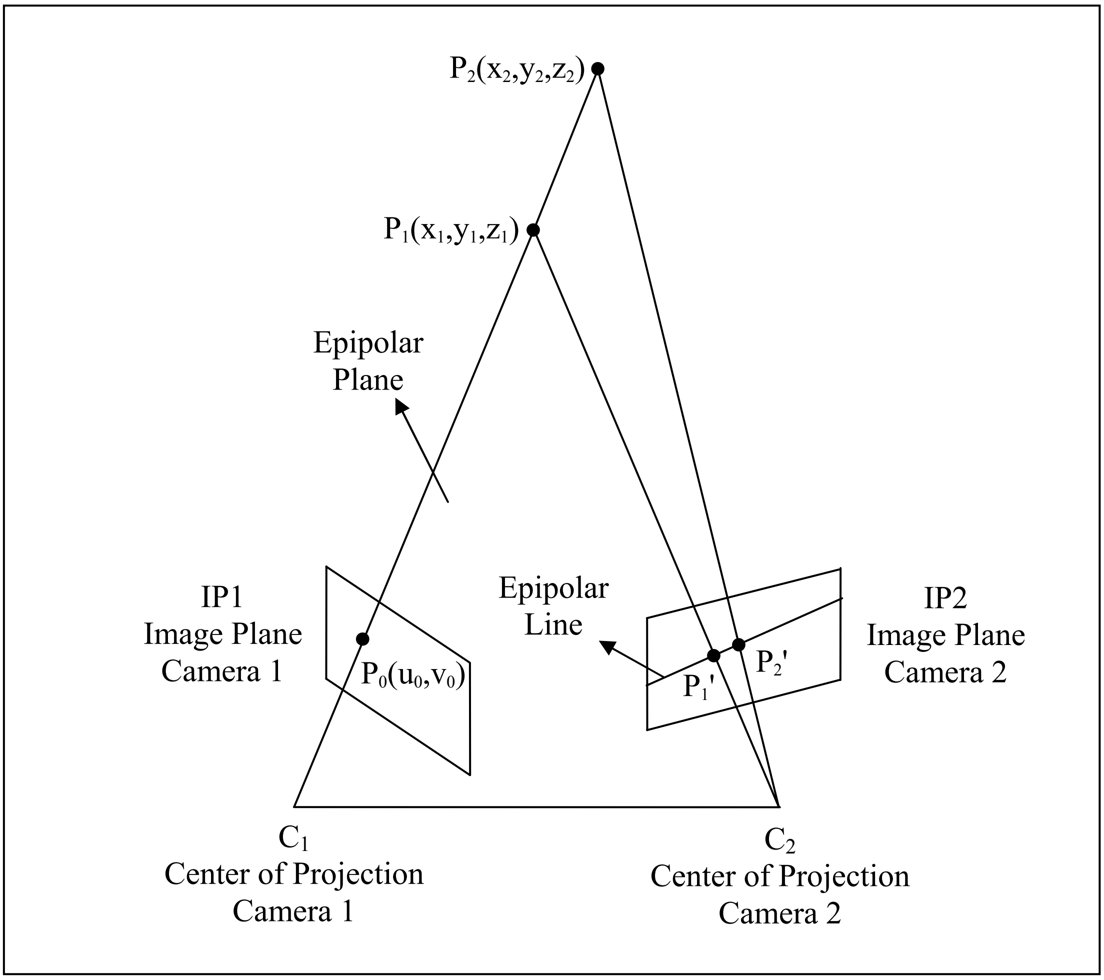


Figure 2.4 The Epipolar Line and the Epipolar Plane

This graphical explanation about epipolar lines and epipolar planes may also be shown using the equations given in Section 2.3 and Section 2.4, camera calibration and stereo vision sections.

Analyzing the Equation (2.16), it can be said that not all of the knowledge of corresponding image point locations is required to estimate ***x***. We may be able to determine ***x*** using only three of these four equations, assuming that the three equations are linearly independent. Therefore this equation defines an "over determined" situation [6].

The relationship given in Equation (2.16) may be written in homogeneous coordinates as

$$[-P|F]\hat{x} = 0 \tag{2.18}$$

where

$$\hat{x} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where X, Y, and Z are the 3D coordinates [6].

Defining the 4 x 4 matrix ***C*** as

$$C = [-P|F], \tag{2.19}$$

then

$$C\hat{x} = 0\,. \tag{2.20}$$

In [6] the basis for the epipolar line concept is given using the Equation (2.20) as follows:

This equation specifies that "every column of $\boldsymbol{C^T}$ (i.e., the rows of ***C***) is orthogonal to $\hat{x}$". The elements of the matrix ***C*** are functions of the calibration parameters for the two cameras and the corresponding image plane points *(u_1,v_1)* and *(u_2,v_2).* It can be seen from the Equations (2.18), (2.19) and (2.20) that "the columns of ***C*** must not span $\boldsymbol{R^4}$". Therefore the determinant of ***C*** must be zero; that is,

$$|C| = 0 \tag{2.21}$$

Expanding the determinant $|C|$ and simplifying the result yields the important result that u_1, v_1, u_2, and v_2 are constrained by the single equation

$$u_1u_2m_1+u_1v_1m_2+u_1v_2m_3+u_2v_1m_4+u_1m_5+u_2m_6+v_1m_7+v_2m_8+m_9=0 \tag{2.22}$$

Equation (2.22) is referred to as "a bilinear relationship" in the variables u and v [6]. The m_i coefficients in the above equation are functions of the camera calibration parameters. This relationship has the geometric significance that, "given a point in one image plane, i.e. (u_1,v_1), the set of corresponding image points in the second image plane are constrained to lie along a line" [6]. This can be seen by assuming a known point in one of the cameras and rewriting the Equation (2.22) for the point in the other camera. If (u_1,v_1) is a known point, Equation (2.22) can be written for the unknown point (u_2,v_2) as

$$(u_1m_1+ v_1m_4+m_6)\, u_2 + (u_1m_3+ m_8)\, v_2 + (u_1v_1m_2+u_1m_5+v_1m_7+m_9) = 0 \tag{2.23}$$

Equation (2.23) describes a line in 2D in terms of unknowns, u_2 and v_2. Thus, given a point in image plane of one camera, the region to be searched for the corresponding point in image plane of the other camera is not all the image plane, but only the line given by the Equation (2.23), that is the epipolar line [6]. Given a point in one camera, utilizing the epipolar line method reduces the search complexity to find the corresponding point in the image plane of the other camera as the search region is reduced to a line in the image plane rather than searching the entire image plane [21].

2.6 KALMAN FILTER

"The Kalman Filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown." [5].

R. E. Kalman published his paper [4] in 1960. Kalman Filter is examined academically and used a lot in many applications since then. Kalman Filter will not be examined with details in this thesis but just a summary will be given. Details of the Kalman Filter is given in [4], [5], [15], and many other sources in the literature.

"The Kalman Filter addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \tag{2.24}$$

with a measurement $z \in \Re^m$ that is

$$z_k = Hx_k + v_k. \tag{2.25}$$

The random variables w_k and v_k represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions $p(w) \sim N(0,Q), p(v) \sim N(0,R)$." [5].

A is the *nxn* matrix showing the relation of the previous state to the current state if no driving function or process noise exists, previous state being the state at time step *k-1* and current state being the state at time step *k*. ***B*** is the *nxl* matrix showing the relation of the control input $u \in \Re^l$ to the state. ***H*** is the *mxn* matrix showing the relation of the state to the measurement z_k [5].

In practice, the "process noise covariance" ***Q***, "measurement noise covariance" ***R***, "state transition matrix" ***A***, "control matrix" ***B***, and "measurement matrix" ***H*** might change with each time step, however here they are assumed to be constant [5] [10].

Defining the "a priori state estimate" at step *k* as $\hat{x}_k^- \in \Re^n$ and the "a posteriori state estimate" at step *k* as $\hat{x}_k \in \Re^n$; the "a priori estimate error", e_k^-, and the "a posteriori estimate error", e_k, is defined as

$$e_k^- = x_k - \hat{x}_k^- \tag{2.26}$$

$$e_k = x_k - \hat{x}_k \tag{2.27}$$

in [5].

Therefore, "a priori estimate error covariance" P_k^- and "a posteriori estimate error covariance" P_k are given as

$$P_k^- = E[e_k^- e_k^{-T}] \tag{2.28}$$

$$P_k = E[e_k e_k^T] \tag{2.29}$$

in [5].

"A posteriori state estimate $\hat{x}_k$ is a linear combination of a priori estimate $\hat{x}_k^-$ and the weighted difference between an actual measurement z_k and a measurement prediction $H\hat{x}_k$" with the relation given as

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \tag{2.30}$$

where

$(z_k - H\hat{x}_k^{-})$ is called "the measurement innovation", or "the residual", and K is called "the gain" or "the blending factor" [5].

The residual shows the discrepancy between the predicted measurement and the actual measurement. The gain is selected such that a posteriori error covariance P_k given in Equation (2.29) is minimized. The gain satisfying this minimization requirement is found as

$$K_k = P_k^{-} H^T (HP_k^{-} H^T + R)^{-1} \qquad (2.31)$$

The details of this calculation can be found in [16], [17], and [18].

Analyzing the Equations (2.30), it can be seen that the gain is the weighing factor between the predicted measurement and the actual measurement. When the gain is zero, then only the predicted measurement and when the gain is $\boldsymbol{H^{-1}}$ then only the actual measurement is used in the calculation of the a posteriori state estimate.

It can be concluded from the Equation (2.31) that the gain approaches zero when the a priori estimate error covariance approaches zero, that is, the predicted measurement is more important than the actual measurement and on the other hand the gain approaches $\boldsymbol{H^{-1}}$ when the measurement error covariance approaches zero, that is, the actual measurement is more important than the predicted measurement in the a priori state estimate.

Operation of the Kalman Filter may be divided into two stages: In the first stage, using the information from the current time step it makes estimates, which are called "a priori" estimates for the next time step. In the second stage, it takes the measurement, and makes "a posteriori" estimates which are corrected versions of "a priori" estimates. These two stages are called "time update" and "measurement update" stages, respectively [5].

The equations of the time update stage are [5]

$$\hat{x}_k^{\;-} = A\hat{x}_{k-1} + Bu_{k-1} \tag{2.32}$$

$$P_k^{\;-} = AP_{k-1}A^T + Q \tag{2.33}$$

The time update equations form the a priori estimates for the next time step, *k* and are using the a posteriori estimates for the time step *k-1*. That is, these equations project the estimates forward in time [5].

The equations of the measurement update stage are [5]

$$K_k = P_k^{\;-}H^T(HP_k^{\;-}H^T + R)^{-1} \tag{2.34}$$

$$\hat{x}_k = \hat{x}_k^{\;-} + K_k(z_k - H\hat{x}_k^{\;-}) \tag{2.35}$$

$$P_k = (I - K_kH)P_k^{\;-} \tag{2.36}$$

The measurement update equations form the a posteriori estimates for the time step *k*, using the a priori estimates for the same time step. That is, these equations correct the estimates using the actual measurement. In measurement update equations, no information from the previous time step, *k-1*, is used directly but this information is stored implicitly in the a priori estimates of the time step *k* [5].

Kalman Filter is a recursive algorithm. Time update and measurement update operation pair is performed recursively for each time step. This recursive algorithm has to be initialized somehow so that the prediction and correction operations can run correctly. There is plenty of information about how the initialization should be done and what the effects of initialization are in the literature, including [5] and [15].

CHAPTER 3

3D MARKER TRACKING
USING KALMAN ESTIMATOR

3.1 INTRODUCTION

In this chapter, the algorithm for finding the 3D trajectories of the markers from the 2D marker positions is given. The positions of the markers in each of the 6 cameras for each frame of the sequence are given as input to the algorithm. The algorithm finds the 3D trajectories of the markers using these 2D marker position data and the calibration and linearization parameters of each camera.

3.2 STEPS OF OUR ALGORITHM

The flow is shown in Figure 3.1. As shown in this figure, firstly 2D marker positions are analyzed and false detections in 2D marker positions are eliminated. Then the distortions in 2D positions of markers in each camera view are corrected using the linearization parameters. After correction of 2D point positions, epipolar lines are formed in each camera view using the calibration parameters. After 2D marker matching, matched 2D marker positions are used to form 3D marker positions using the calibration parameters again. The 3D markers that are very close to each other in fact correspond to the same real marker and these are grouped to find an estimate of the real 3D marker. After grouping, 3D tracks can be formed using these groups.

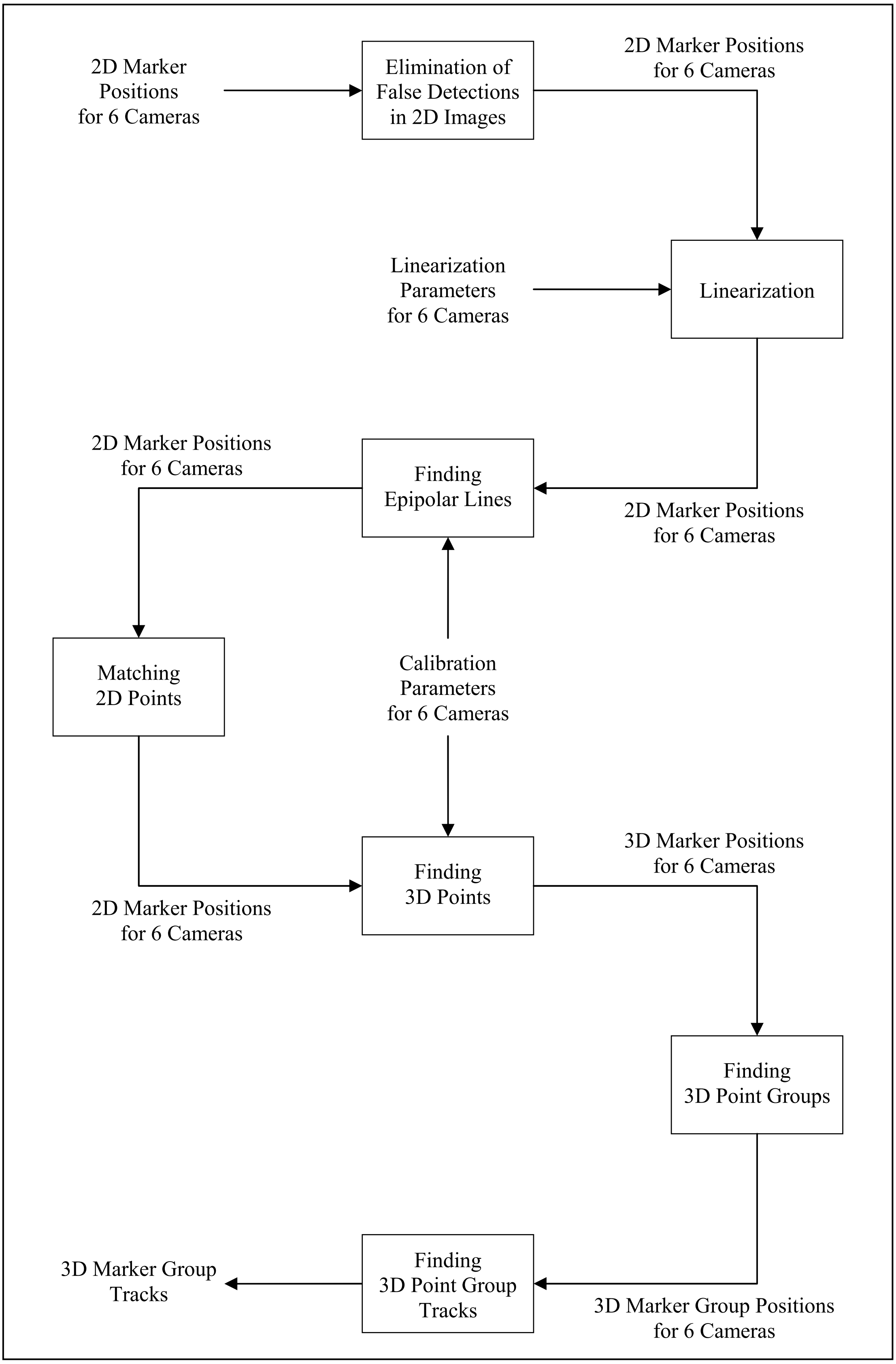

2D Marker Positions for 6 Cameras
Elimination of False Detections in 2D Images
2D Marker Positions for 6 Cameras
Linearization Parameters for 6 Cameras
Linearization
2D Marker Positions for 6 Cameras
Finding Epipolar Lines
2D Marker Positions for 6 Cameras
Matching 2D Points
Calibration Parameters for 6 Cameras
Finding 3D Points
3D Marker Positions for 6 Cameras
2D Marker Positions for 6 Cameras
Finding 3D Point Groups
3D Marker Group Tracks
Finding 3D Point Group Tracks
3D Marker Group Positions for 6 Cameras

Figure 3.1 The Steps of 3D Marker Tracking

3.3.1 ELIMINATION OF FALSE DETECTIONS IN 2D IMAGES

Elimination of false detections is needed because there are some false points in 2D images which are generally due to the detection of the LEDs on the other cameras. This can be seen from the Figure 3.2. This figure is formed by drawing all the marker positions throughout the sequence in one image. In the figure the false detections are clearly seen as they are stationary, that is they do not form a moving 2D track in the image through the sequence.

These false detections are eliminated using the fact that they are stationary. Firstly, all the marker positions throughout the sequence are scanned and the number of markers per pixel in the image is counted. If the number of markers per pixel of the image exceeds a determined constant value, this pixel is marked as a stationary point. Then all the marker coordinates are scanned and markers at that stationary point are deleted. But due to digitization and marker detection errors, these "stationary" points tend to move a few pixels through the frames in the sequence. Therefore the deletion operation is modified so that for each stationary marker position, the markers at the neighboring pixels are deleted. "Neighboring pixel" definition can be changed by a constant. Decreasing the number of neighboring pixels decrease the elimination rate of stationary points while increasing it may result in losing real "moving" marker points which are important. So a minimum of "neighboring distance" definition is made.

This elimination process must be avoided for only some camera configurations where a moving marker "seems" as stationary to a camera. But this is a very awkward situation because all the marker points tend to move through the sequence. But this portion of the algorithm may be taken out in such a case in another GAIT analysis system.

The results of this elimination can be seen in Figure 3.3. This figure is also formed by drawing all the marker positions throughout the sequence in one image. When compared with the Figure 3.2, it can be clearly seen that most of the stationary false detections are eliminated while moving marker positions are not effected from this elimination process, that is, no real markers are lost in this elimination process. But there are still some stationary points in Figure 3.3. They are distributed over a large area to be assumed to be stationary and trying to remove them will result in losing some real marker data, which is not accepted. Therefore these remaining false detections are left in the sequence.

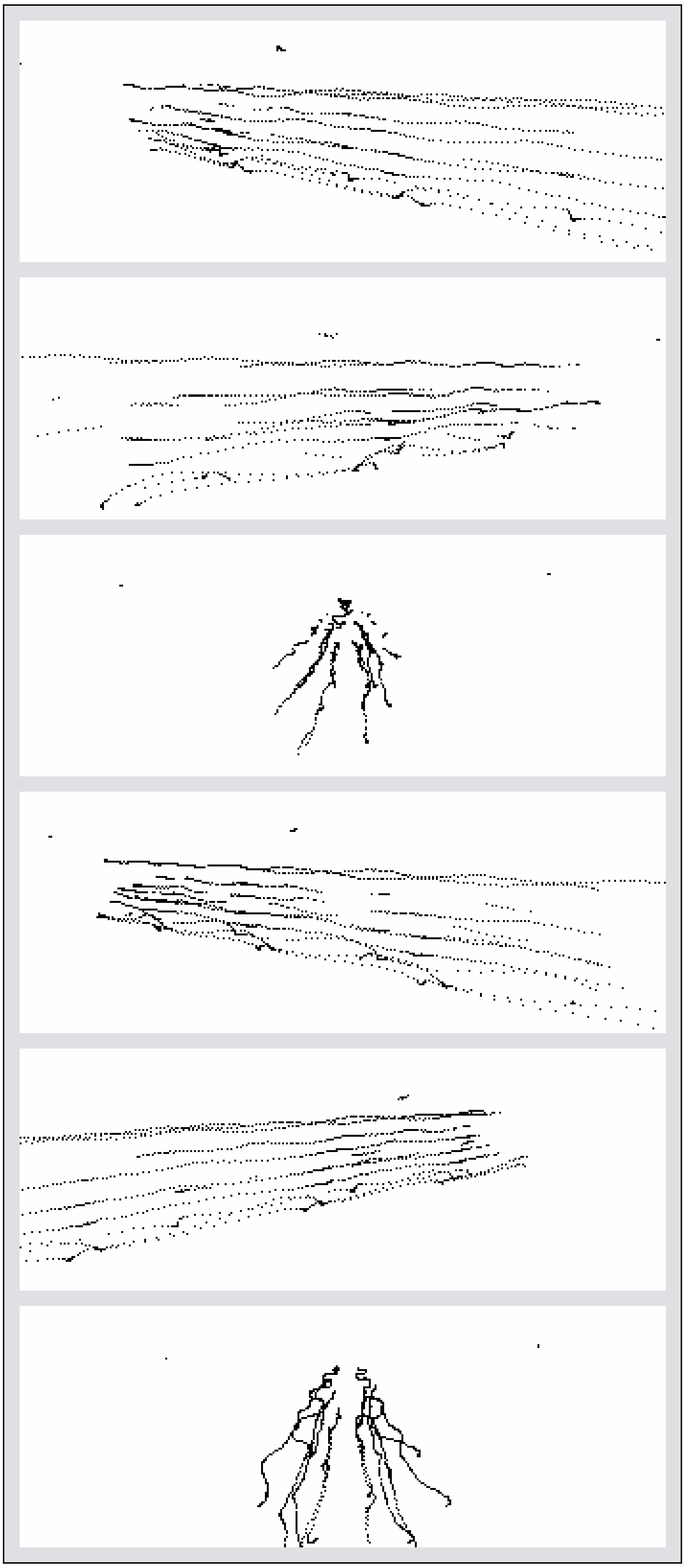


Figure 3.2 2D Markers in the Sequence Before Elimination of False Detections

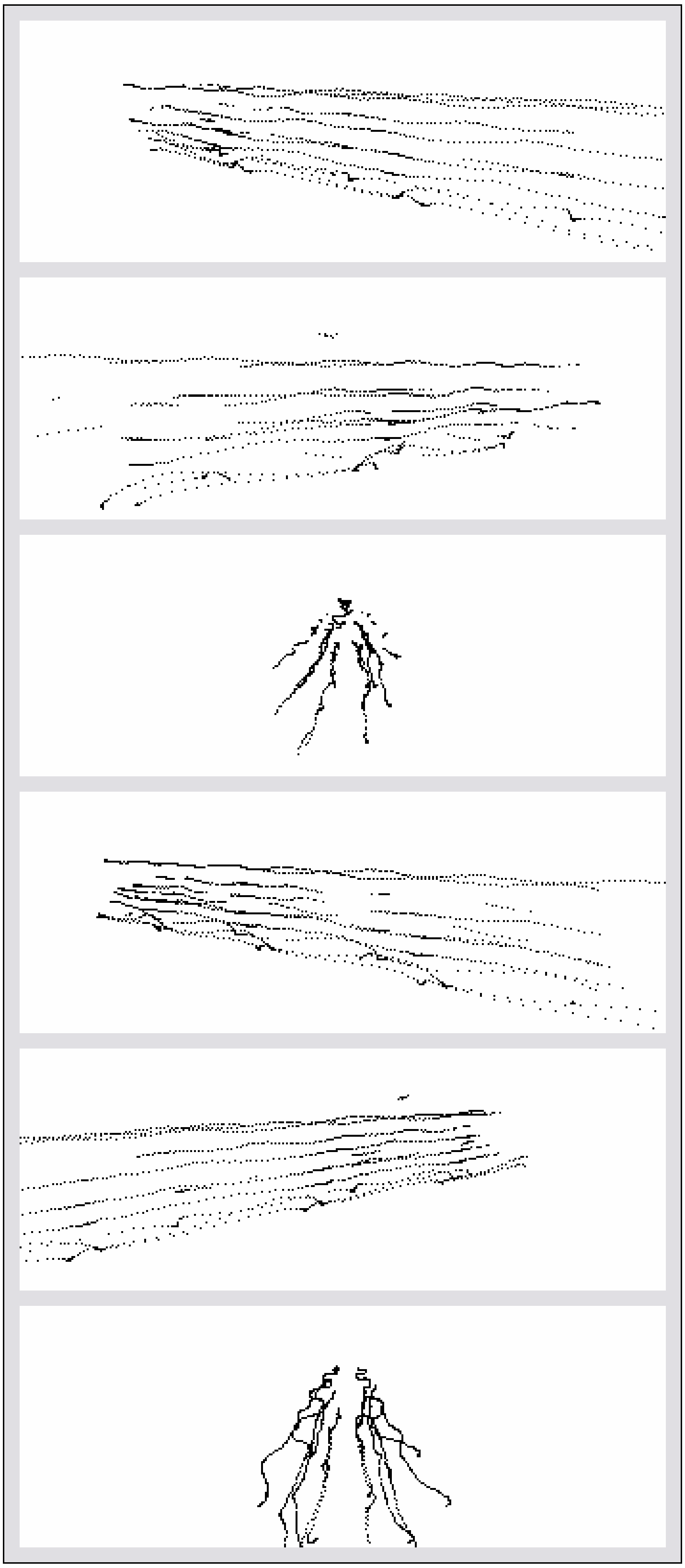


Figure 3.3 2D Markers in the Sequence After Elimination of False Detections

3.3.2 LINEARIZATION OF 2D IMAGES

The linearization process is explained in Section 2.2 and [2]. In the scope of this thesis, linearization parameters are assumed to be already determined. With this known linearization parameters, the 2D coordinates of the markers are corrected. The linearization process is applied after applying our method of elimination of false detections in 2D images.

As in explained in Section 2.2 and [2], it is assumed that the corrected coordinates are a polynomial function of the distorted coordinates. This relation between the corrected coordinates and the distorted coordinates was given in Equation (2.2) with the 30 linearization parameters, 15 for x coordinate and 15 for y coordinate, which are already known.

For each point in the image, the corrected point coordinates are found by using the Equation (2.2) with the known linearization parameters and the measured point coordinates which are distorted due to distortions [2].

The effect of linearization can be seen in Figure 3.4 and 3.5. The first figure shows marker positions in 6 camera images in a frame in the sequence before linearization, second figure shows the images in the same frame after linearization.

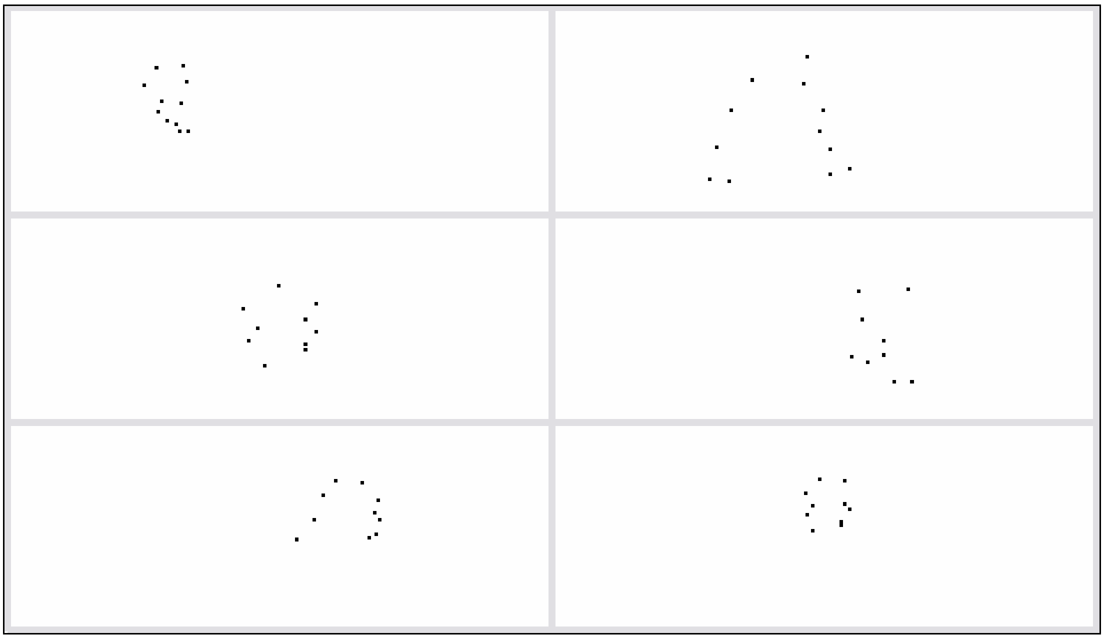


Figure 3.4 2D Marker Images of 6 Cameras at a Selected Frame Before Linearization

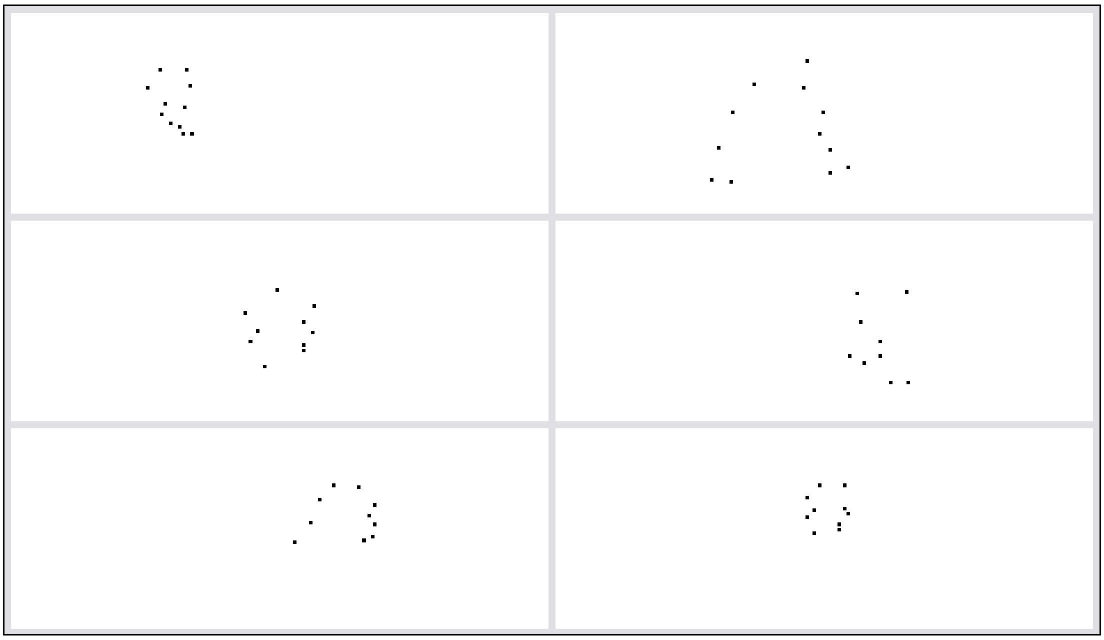


Figure 3.5 2D Marker Images of 6 Cameras at a Selected Frame After Linearization

The difference between the marker positions in Figure 3.4 and 3.5 can be seen when looked closer to the markers. The comparison of the 2D marker positions in the upper right camera shown in these figures is shown closer in Figure 3.6. In this figure, the plus signs show the positions before linearization and the squares show the positions after linearization.

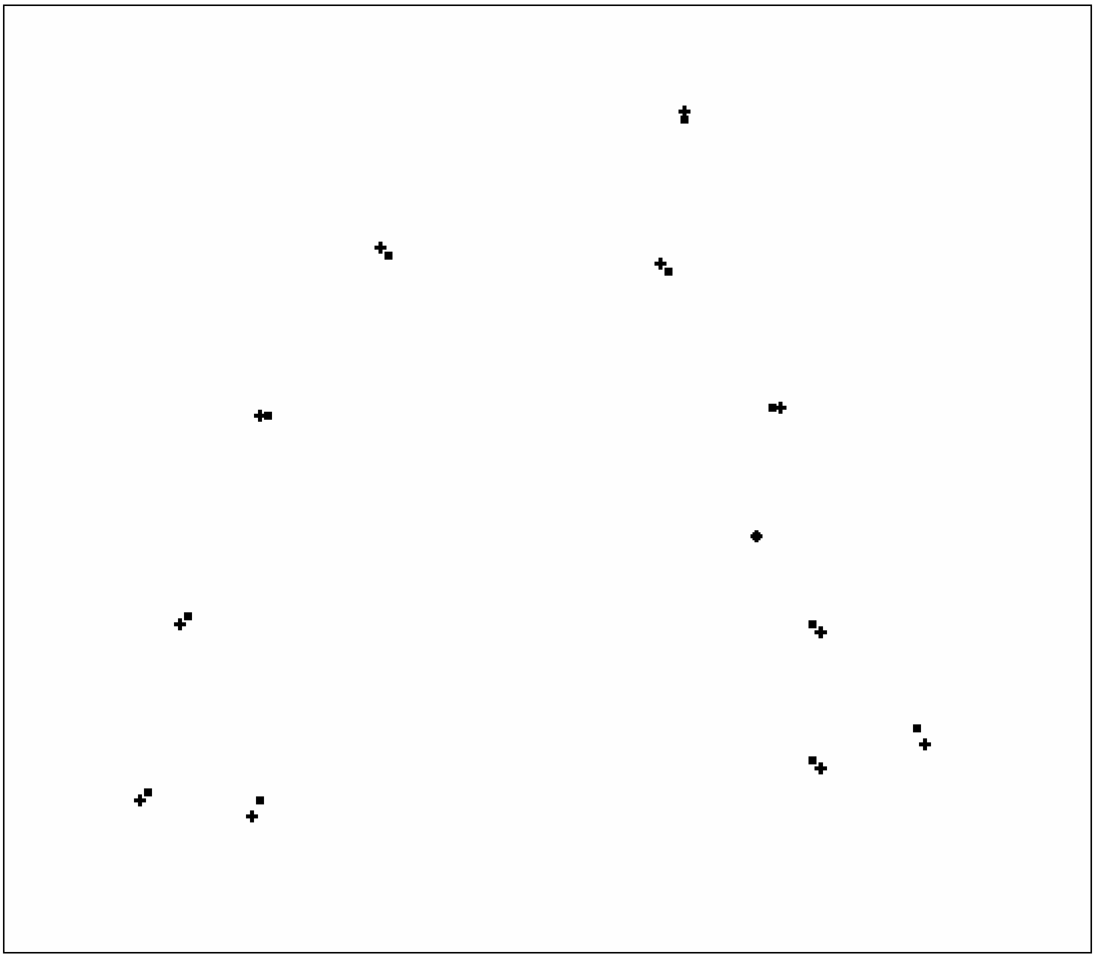


Figure 3.6 Close View of the Superimposed Camera Images Before and After Linearization at the Selected Frame in Figure 3.4 and 3.5

3.3.3 CALIBRATION OF CAMERAS

Calibration was mentioned in Section 2.3. In our application, calibration of the cameras is assumed to be already done. Our application takes the calibration parameters that are calculated in this calibration procedure. These calibration parameters are used in finding the epipolar lines and calculating the 3D position from the 2D positions of matched 2D points. These are explained in Sections 3.3.4 and 3.3.6, respectively.

3.3.4 FINDING EPIPOLAR LINES

In our first approach, all the points in the neighborhood of each epipolar line in an image were matched and a 3D point corresponding to this matching was created. In the second approach, to create a 3D point, finding a match between a point and an epipolar line is not enough and matching between three 2D points and the corresponding epipolar lines are looked for. Matching between three 2D points requires checking the matching between three possible pairs. This enables to filter out markers that are only seen by 2 of the 6 cameras. This is useful to filter out some ambiguous matches because most of the markers are seen by more than 2 cameras and therefore a correct marker will pass through this filter but a false match between unrelated points will not.

For a point $A(u_0,v_0)$ in camera i, there are $(n-1)$ epipolar lines in maximum, where n is the number of cameras in the system, because there may be at most one epipolar line in each of the cameras other than the camera i. The actual number of epipolar lines may not reach the maximum value $(n-1)$ because for the point $A(u_0,v_0)$ to have an epipolar line in camera j, the line connecting the projection center of the camera i to the point in 3D space should be in the visible area of the camera j. Generally, sights of all the cameras are adjusted so that these cameras see the entire calibration region. Even though a point which is detected to be in the calibration region by a camera will be in the area of vision of all the other cameras, the detection may or may not exist. So the number of matches is not predetermined.

Assuming that all the cameras are placed such that all of them can see the calibration region completely and that each point is detected correctly, each point in camera i will have one epipolar line in camera j, where $j \neq i$ resulting in $(n-1)$ epipolar lines in total in camera j. Therefore, the number of epipolar lines in an image in a frame is the total number of points in all other cameras in that frame.

There are many epipolar lines in an image. This can be seen in Figure 3.7. In this figure, epipolar lines of the second camera on all other cameras at a selected frame in the sequence are seen. It can be seen easily that many epipolar lines seem to match points correctly.

In our application, epipolar lines have to be stored for each image. This is performed by storing the gain m and offset n values describing the line given in Equation (2.23). For each camera in the system, all points in the camera are taken as the source point and epipolar lines corresponding to these source points in all the target cameras (i.e. all the other cameras) are found using Equation (2.23).

The epipolar line is represented by the formula $y=mx+n$ or $v=mu+n$ in the target image plane. Equation (2.22) may be used to calculate v_2 for a given u_2 after giving the constant source plane pixel positions (u_1,v_1) and the m_i coefficients in this equation, which are the known calibration parameters. This method is used to find v_2 values corresponding to two u_2 values, the first value being zero, corresponding to the left edge of the image plane of the target camera, IP2, and the second value being the maximum horizontal value in the target image plane, corresponding to the right edge of the image plane of the target camera, IP2.

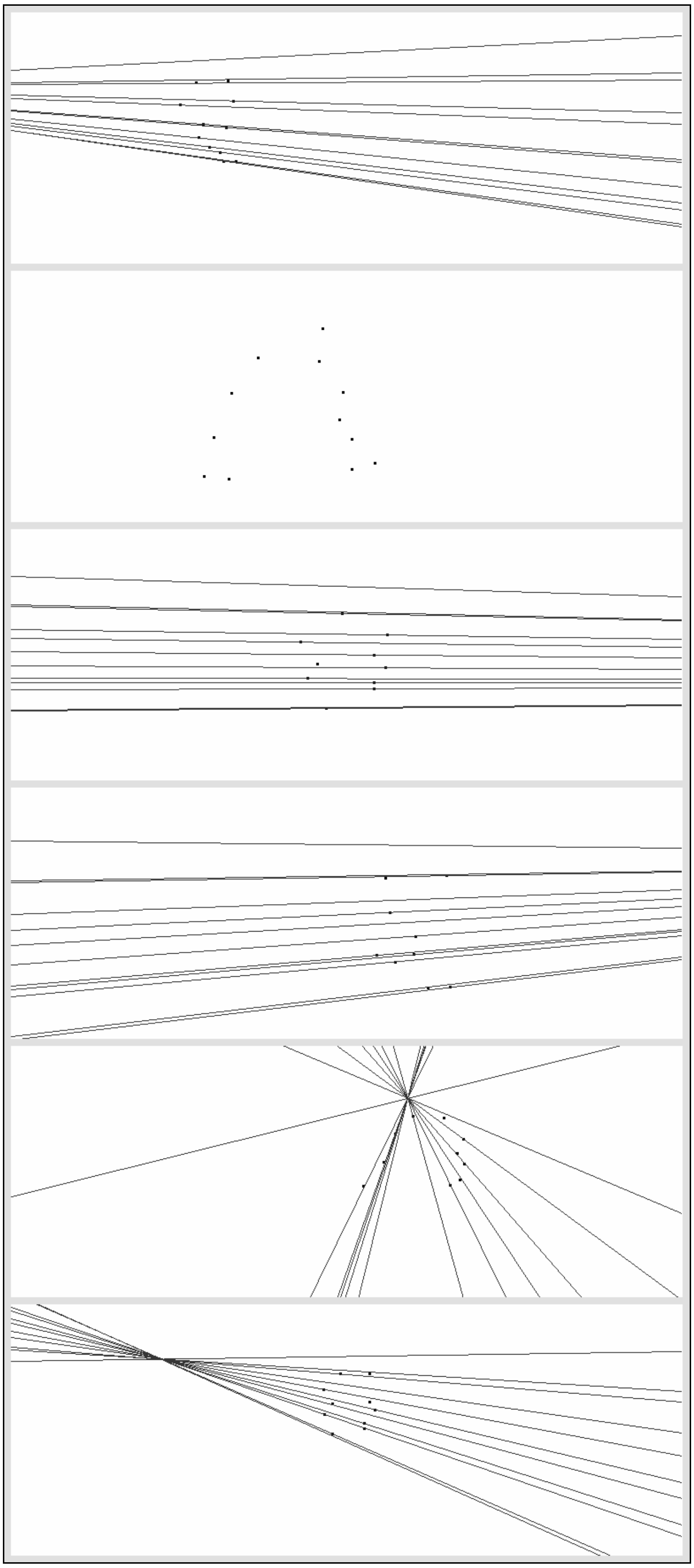


Figure 3.7 Epipolar Lines of the Second Camera on all Other Cameras at a Selected Frame

The calculated v_2 values may be invalid for that image plane but as only gain, m, and offset, n, are calculated and stored, the line segment which is in the valid area (image plane) will be reconstructed using these m and n values. Invalid v_2 values may be negative values or values larger than the image height. This can be seen in Figure 3.8 where P_3P_4 or P_5P_6 are also valid epipolar lines as well as P_1P_2.

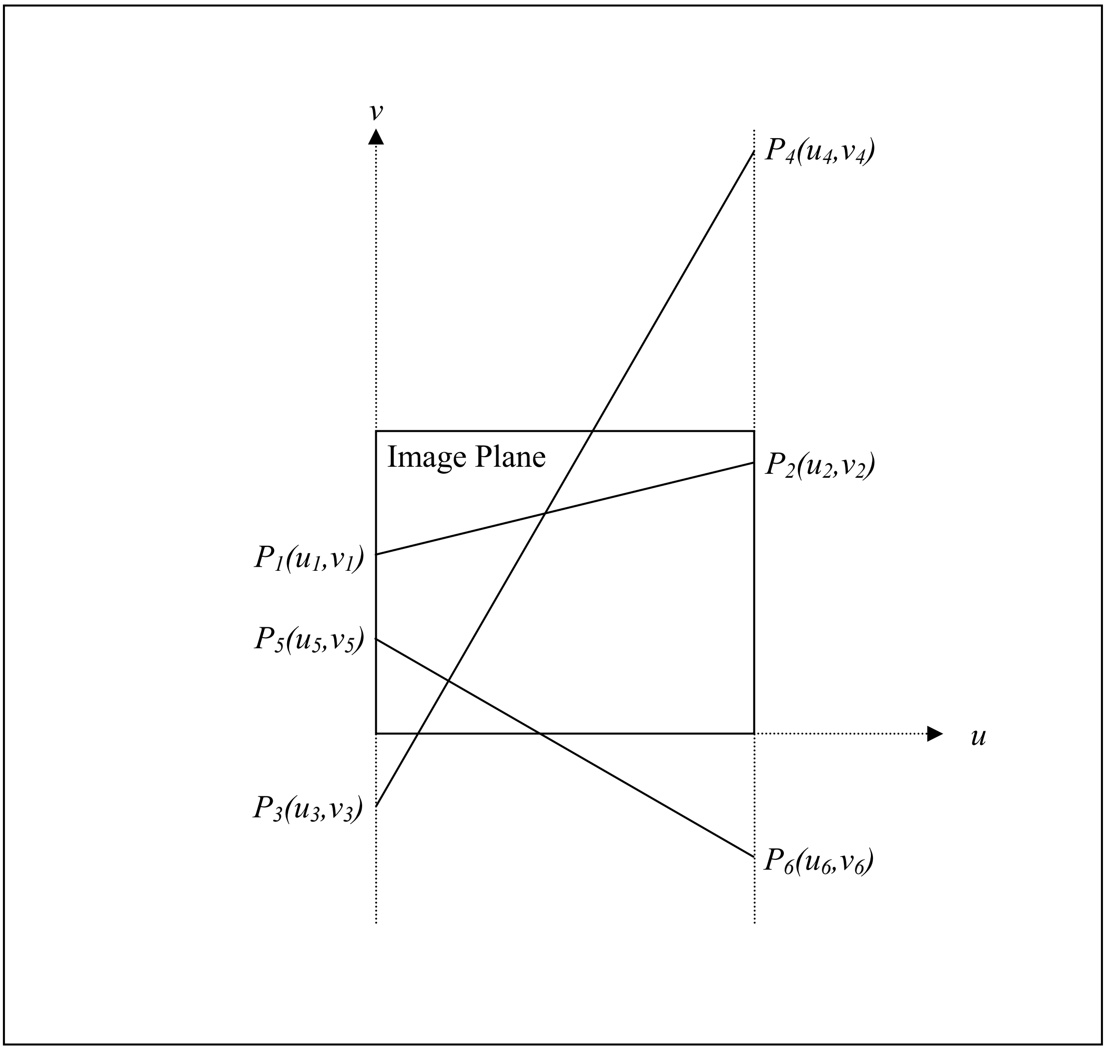


Figure 3.8 Examples of Valid Epipolar Lines on an Image Plane

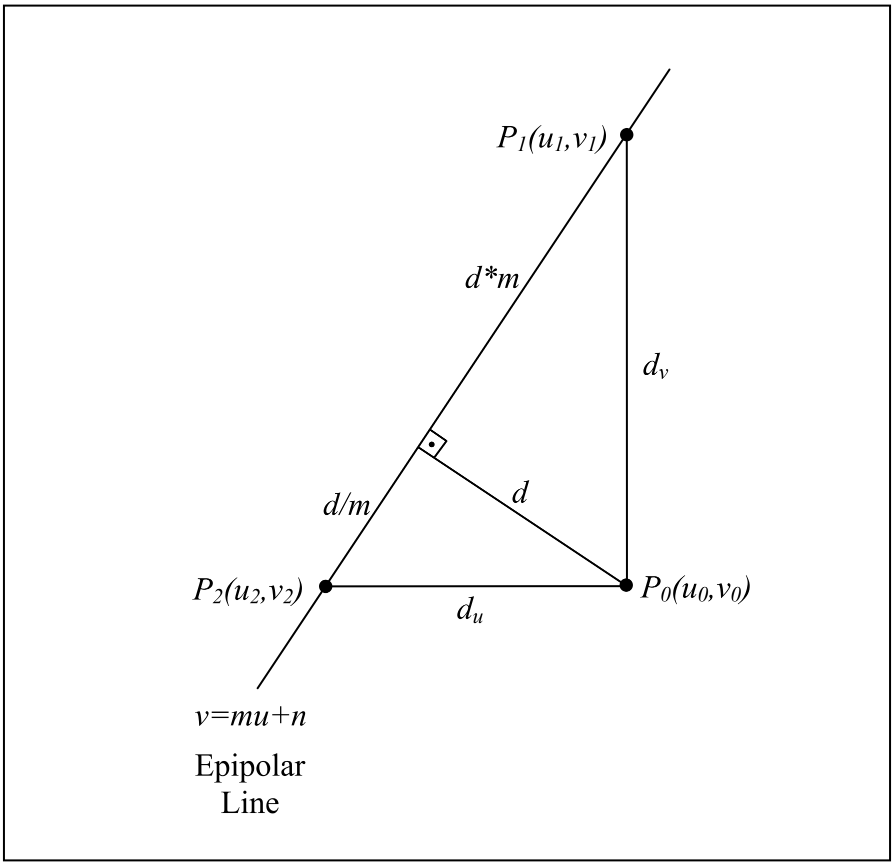


Figure 3.9 Epipolar Line Neighborhood

When a point in the target plane, say point $P_0(u_0,v_0)$, is analyzed for being on an epipolar line or not, the distance of this point to the epipolar line has to be calculated. Theoretically, if two 2D points in two camera images are the corresponding points of the same 3D point, then each of these points must lie exactly on the epipolar line of the other point. But due to digitization and marker detection errors, points may be very close to the epipolar line if they are not lying on it. Therefore an epipolar line neighborhood distance s is defined and the points closer to the epipolar line than this neighborhood distance are assumed to be lying on the epipolar line. In Figure 3.9, a point and an epipolar line are drawn and the distance of the point to the epipolar line d is shown. We have to find if d is smaller than the fixed neighborhood distance s or not. But the horizontal and vertical distances of the point to the epipolar line, d_u and d_v respectively, can also be used to determine if the point is close enough to the epipolar line.

Firstly, the points P_1 and P_2 in the figure are found. These points are found using the information that these are the horizontal and vertical projections of the point $P_0(u_0,v_0)$ on the epipolar line and that these points are lying on the line defined with the equation $v=mu+n$ or $u=(v-n)/m$, that is, coordinates of these points should satisfy these equations. Therefore $P_1(u_1,v_1)$ and $P_2(u_2,v_2)$ will be found as

$$u_1 = u_0 \tag{3.1}$$

$$v_1 = mu_0+n \tag{3.2}$$

$$u_2 = (v_0-n)/m \tag{3.3}$$

$$v_2 = v_0 \tag{3.4}$$

and therefore

$$d_u = u_2-u_0 = [(v_0-n)/m]-u_0 \tag{3.5}$$

$$d_v = v_1-v_0 = mu_0+n-v_0 \tag{3.6}$$

for the point $P_0(u_0,v_0)$.

As the neighborhood condition is defined by a constant s, that is, a point is assumed to be lying on the epipolar line if the distance of that point to the epipolar line is smaller than s, this neighborhood distance can be written in horizontal and vertical directions using Figure 3.9 as

$$s_u = \sqrt{s^2 + (\frac{s}{m})^2} \tag{3.7}$$

$$s_v = \sqrt{s^2 + (s.m)^2} \tag{3.8}$$

Then the distance of the point is compared with these distances and the point is said to be on the epipolar line if $d_u<s_u$ or $d_v<s_v$.

3.3.5 MATCHING POINTS IN 2D IMAGES

Algorithm for matching points in 2D images starts after all the epipolar lines in all frames of the sequence is calculated and stored as gain, *m*, and offset, *n*. Matching 2D points is performed to find the point in other camera corresponding to the point in selected camera. Only after this matching is done, these two points corresponding to the same 3D point is used to find the position of this 3D point according to Equation (2.16). If false matching is performed, a false 3D point is assumed to be detected, which makes tracking in 3D space more difficult and increases the number of mistakes in forming tracks or may lead the track to a wrong direction and lose the real movement.

Our matching algorithm does not guarantee the one-to-one matching between the points and the epipolar lines. That is, as the search algorithm does not check if the epipolar line has already been matched to another point, an epipolar line may be matched to more than one point at the end of searching all points in the image. There is no easy way to determine which point is the real match to the epipolar line therefore all matches are accepted including the multiple matches. Possible false matches are eliminated through the method of only accepting three points which are matched in pairs. This guarantees that in order to generate a point in 3D space, there must be at least three cameras where this 3D point was seen and detected. This method will decrease false 3D point generation and will not affect the true points as most of the true markers are seen and detected by more than two cameras.

Firstly, all the points in each frame of each camera are searched for an epipolar line the point lies on. In Section 3.3.4 the definition for a point be "lying on an epipolar line" or be "close to an epipolar line" was given. When a point P_1 from camera *i* is detected to be on the epipolar line of point P_2 from camera *j* on the image plane of camera *i*, these two points are temporarily matched and progress to a second operation where a third point in that frame to match the two points matched temporarily are found. This is done by analyzing the points in the cameras other than *i* and *j*. If the epipolar lines corresponding to this third point P_3 in the image of the camera *k*, on the image planes of cameras *i* and *j* is in the neighborhood of the points P_1 and P_2, respectively, and the epipolar lines corresponding to P_1 and P_2, on the image plane of camera *k* is in the neighborhood of the point P_3 then P_1, P_2, and P_3 are matched. This triple point matching criterion does not use any priority system with respect to the distance of the point to the epipolar line since the neighborhood limits are need to be kept very small not to have many false alarms and that the differences in distances of the points to the epipolar lines in this small region may not contain much information about the correct matching between epipolar line and the points and may as well be caused by measurement noise.

This triple point matching is shown in the Figure 3.10. In this figure, IP_i is the image plane of camera *i*, P_i is the point in camera *i* which is the candidate of matching, EL_m^n is the epipolar line corresponding to the point P_m in the image plane IP_n. As can be seen in the figure, each point has a corresponding epipolar line in the image planes of the other cameras. For these three points, namely P_1, P_2, and P_3, to be matched, all should be in the neighborhood of the two epipolar lines corresponding to the two other points. That is,

$$P_i \textit{ should be in the neighborhood of } EL_m^n \textit{ for } i=1..3,\ n=1..3,\ m=1..3,\ m\neq i,\ n\neq i \qquad (3.9)$$

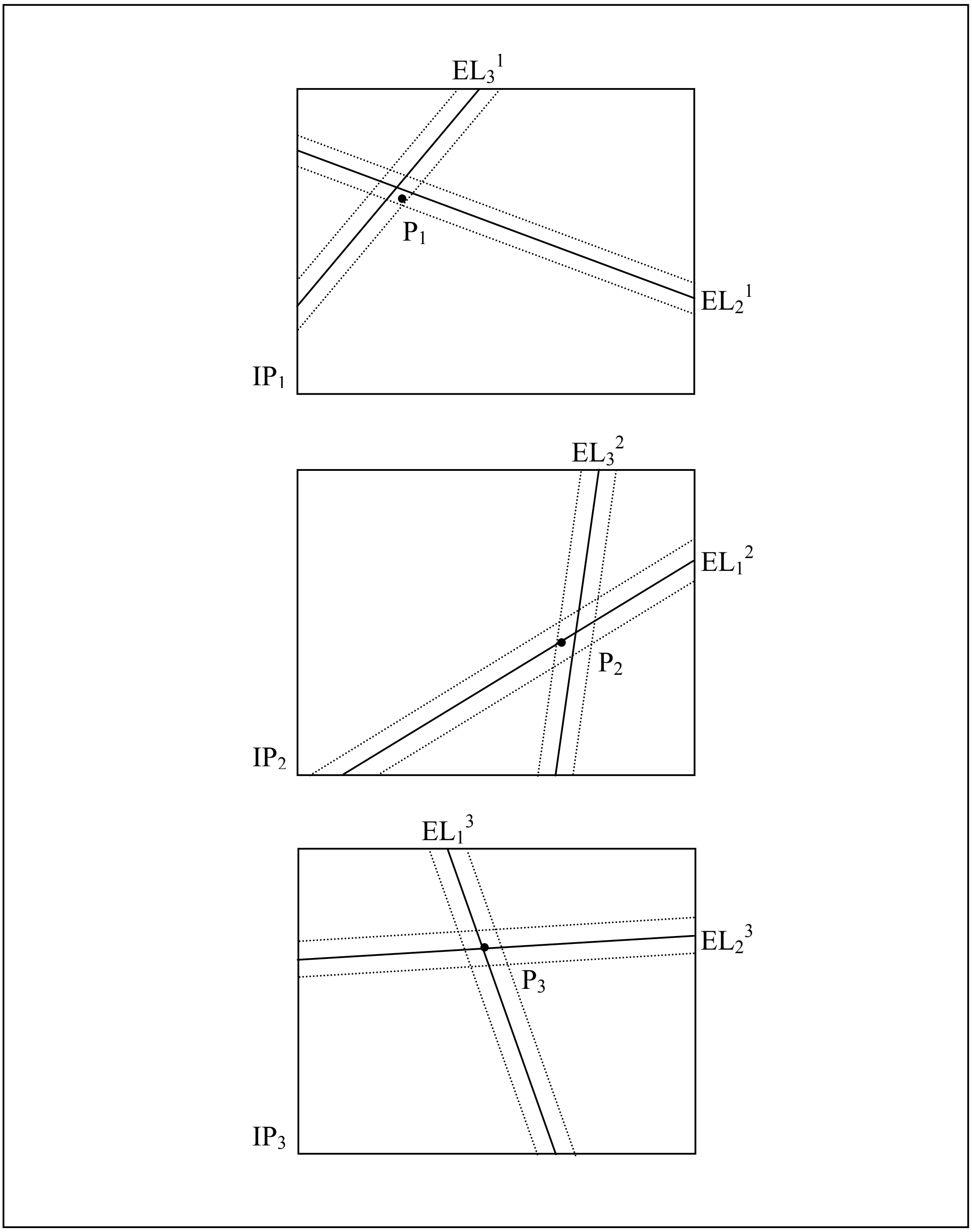


Figure 3.10 Matching 2D Points in Groups of Three Points

3.3.6 FINDING 3D POINTS

After matching 2D point triples in three cameras, the 3D point corresponding to these matched 2D points is possible according to Equation (2.16). But Equation (2.16) can be used to find the 3D point corresponding to two 2D points in two cameras which are thought to be the projection of the same 3D point on the image planes of these two cameras. In our case, there are three points in three cameras which are thought to be the projection of the same 3D point on the image planes of these three cameras. There may be two solutions for this situation.

One solution may be finding the least squares estimation to the equation set written using Equation (2.14) and Equation (2.15) for each of the three cameras, that is, the extended version of Equation (2.16) for three cameras. This equation set is written in matrix form as

$$Px = F \tag{3.10}$$

where

$$P = \begin{bmatrix} a_{1,1} - u_1 a_{9,1} & a_{2,1} - u_1 a_{10,1} & a_{3,1} - u_1 a_{11,1} \\ a_{5,1} - v_1 a_{9,1} & a_{6,1} - v_1 a_{10,1} & a_{7,1} - v_1 a_{11,1} \\ a_{1,2} - u_2 a_{9,2} & a_{2,2} - u_2 a_{10,2} & a_{3,2} - u_2 a_{11,2} \\ a_{5,2} - v_2 a_{9,2} & a_{6,2} - v_2 a_{10,2} & a_{7,2} - v_2 a_{11,2} \\ a_{1,3} - u_3 a_{9,3} & a_{2,3} - u_3 a_{10,3} & a_{3,3} - u_3 a_{11,3} \\ a_{5,3} - v_3 a_{9,3} & a_{6,3} - v_3 a_{10,3} & a_{7,3} - v_3 a_{11,3} \end{bmatrix},$$

$$x = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \text{ and}$$

$$F = \begin{bmatrix} u_1 - a_{4,1} \\ v_1 - a_{8,1} \\ u_2 - a_{4,2} \\ v_2 - a_{8,2} \\ u_3 - a_{4,3} \\ v_3 - a_{8,3} \end{bmatrix}$$

where $a_{i,k}$, *i=1,2, ... 11*, *k=1,2,3,* show the calibration parameters a_i for the camera *k*.

Solving this vector equation using pseudo-inverse technique, a solution for the 3D point positions, (X,Y,Z), will be found. By this way, only one 3D point will be generated for the three 2D points that are the projection of the same 3D point on the image planes of the three cameras.

Yet one other solution may be solving Equation (2.16) for each pair of 2D points in the 2D point triple. Solving Equation (2.16) for each pair of 2D points using pseudo-inverse technique, a solution for the 3D point positions, (X,Y,Z), will be found for each pair. This leads to having three 3D points corresponding to each point triple in 2D coordinates.

One another solution may be not storing the three 3D points found in the solution above and store only one 3D point for these three points using averaging or any other way. This leads to smaller number of points in 3D frames.

The second solution is selected for our application because many points are seen by more than three cameras and a group of 3D points already form in 3D frames around the real position of the marker. As grouping is performed in 3D frames as in explained in Section 3.3.7, grouping is left to 3D domain.

3.3.7 FINDING 3D POINT GROUPS

For each point triple in 2D images, three 3D points are generated. If a point in 3D world is seen and detected by more than three cameras, there will be more than three points in the 3D image, which should ideally be located at the same 3D position. But due to measurement and marker detection errors, these 3D points corresponding to the same 3D point will group like a cloud in the 3D frames.

Figure 3.11 shows 3D points created in a selected frame in the sequence. A closer look into this figure is shown in Figure 3.12, where it is clearly seen that 3D points come close forming 3D "clouds".

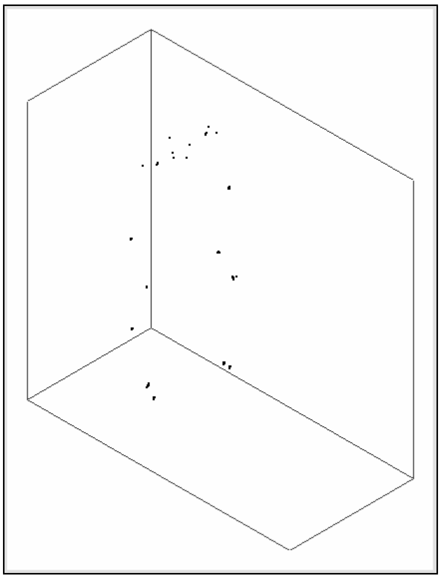


Figure 3.11 3D Points at a Selected Frame Without Grouping

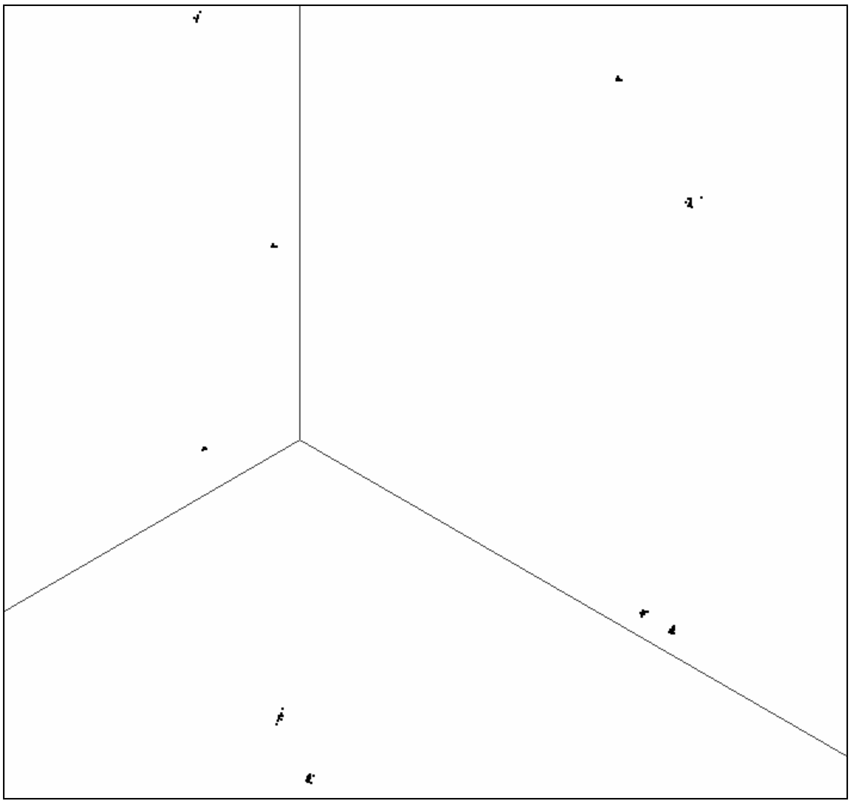


Figure 3.12 Closer View of Figure 3.11

By defining a neighborhood distance, these "cloud" of 3D points may be unified to have one 3D point which can be assumed to be near to the real 3D position of the marker. The cloud of points are marked and the average x, y, and z values for the group is calculated. Scanning all the points in 3D frames, if there are no groups in the neighborhood distance, a new group is formed, if there is a group, this point is also added to the group and the average values of the group is updated.

Grouping the 3D points which are very near to each other, the number of 3D points can be decreased. From this point on, tracking is performed on 3D groups but not on the 3D points.

The selected frame after grouping is shown in Figure 3.13. In this figure, false detections are easily seen. The skeleton formed by the markers given in Figure 1.3 is manually drawn onto the snapshot. Note that the image in this figure is formed by decreasing the scales in all three dimensions asymmetrically to be able to see the whole volume easily.

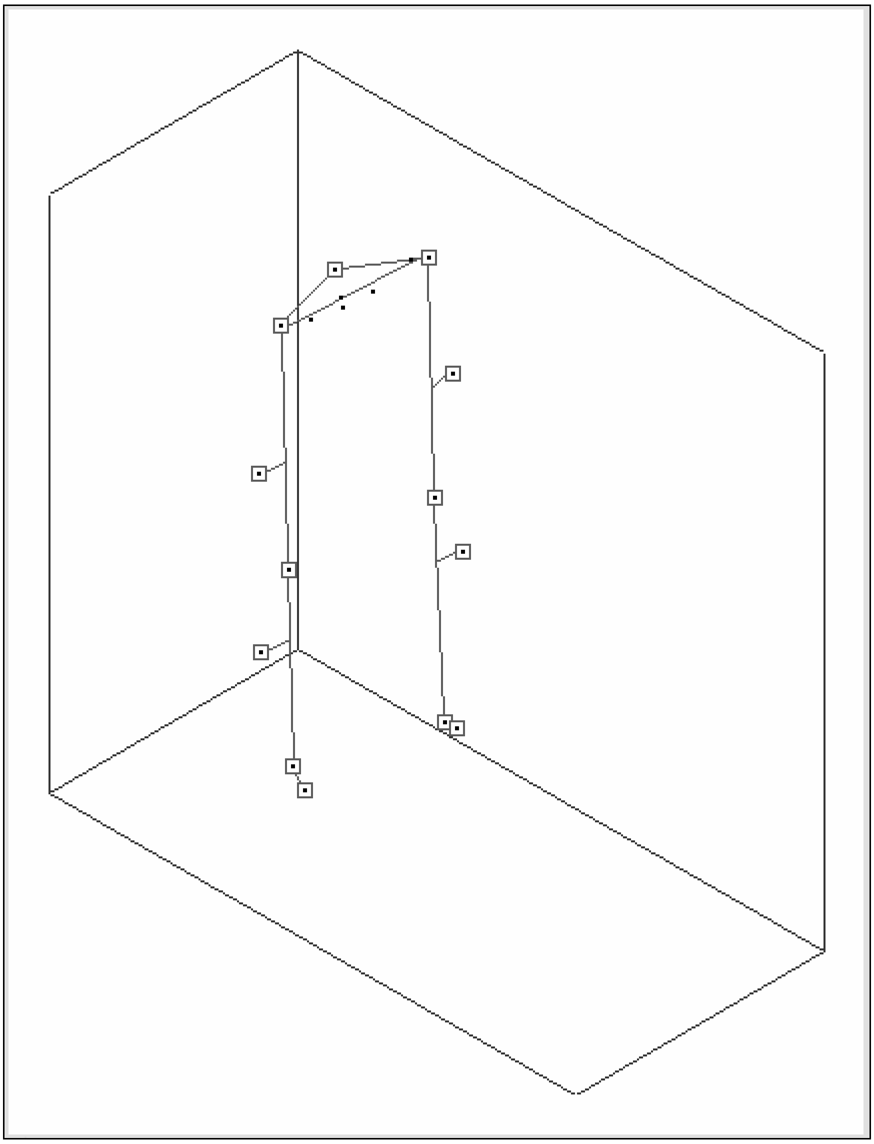


Figure 3.13 3D Points at a Selected Frame After Grouping

3.3.8 FINDING 3D POINT GROUP INITIAL TRACKS

After grouping the 3D points in the sequence, 3D tracks are to be formed. The sequence is processed one frame at a time until the formation of the tracks, that is, no temporal information is considered. To form the tracks using the 3D point groups, temporal information is also considered to find the correspondences of 3D point groups through the frames, that is, correspondences in time.

Our first method for placing the 3D points on continuous tracks was using linear extrapolation and taking the nearest neighbor. In this method, firstly, the predicted position of each 3D point group in the following frame is found. When predicting the position of the point in the following frame, the point is assumed to move the same amount as it moved between the previous frame and the present frame. This means that the point is assumed to have constant velocity. This predicted position is found by performing linear extrapolation using the position of that point and the position of the previous point of the track. This method is shown in Figure 3.14. The figure is prepared for the 2D case in order to be able to show the method easily and extension to 3D case is trivial. The equations of this prediction are given below:

$$
\begin{aligned}
x'_{i+1} &= x_i + (x_i - x_{i-1}),\\
y'_{i+1} &= y_i + (y_i - y_{i-1}),\\
z'_{i+1} &= z_i + (z_i - z_{i-1})
\end{aligned}
\qquad (3.11)
$$

where

(x_i,y_i,z_i) is the position of the point in frame i,

$(x_{i-1},y_{i-1},z_{i-1})$ is the position of the point in frame $i-1$, and

$(x'_{i+1},y'_{i+1},z'_{i+1})$ is the predicted position of the point in frame $i+1$.

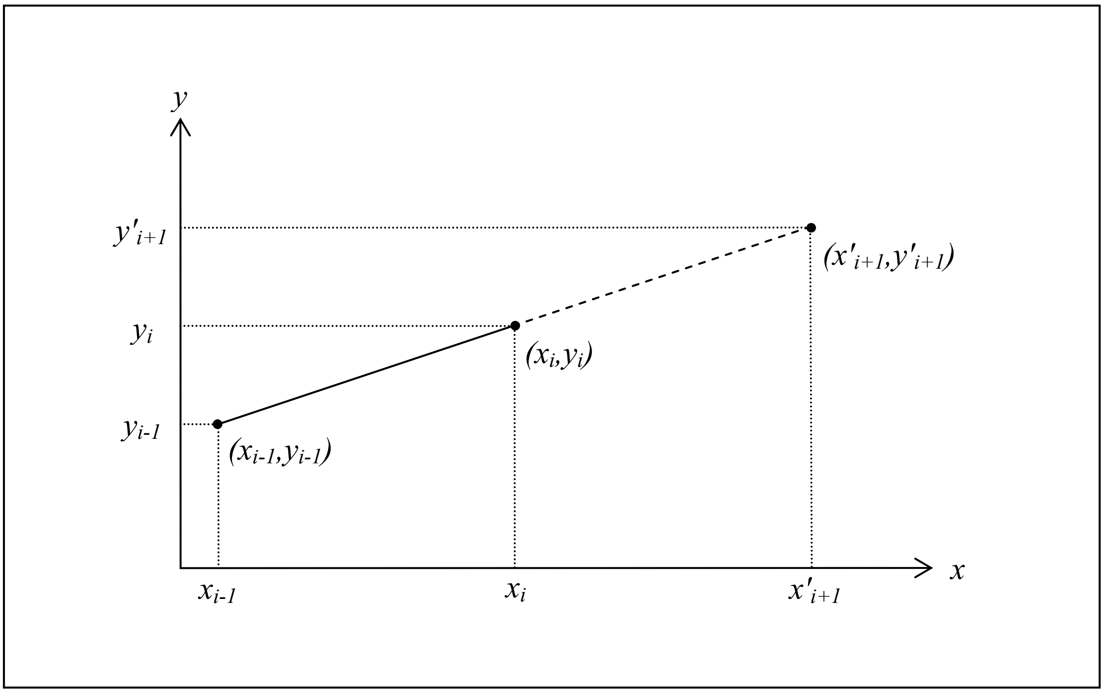


Figure 3.14 Prediction Method of Finding 3D Point Group Initial Tracks

This method assumes the points to have constant velocity. The difference in this equation gives the average velocity of the point from frame *i-1* to frame *i*. The equation shows that this difference is added to the position of the point in the present frame to find the position in the following frame. This difference is updated for each frame in order to update the velocity.

This constant velocity prediction cannot be performed for the first point in the track as there is no previous point of the track. For the first point, the predicted point in the following frame is taken as the same point.

After the predicted point is found in the following frame, the next frame is searched and the nearest point to this predicted point is found. Then the distance of the nearest point to the predicted point is compared to a constant measure of neighboring distance to determine if it will be taken as the next point or the track will end at this point. If the distance of the nearest point in three dimensions is smaller than this constant value, the point is selected as the next point of the current track. Otherwise, the track is marked as ending at that frame.

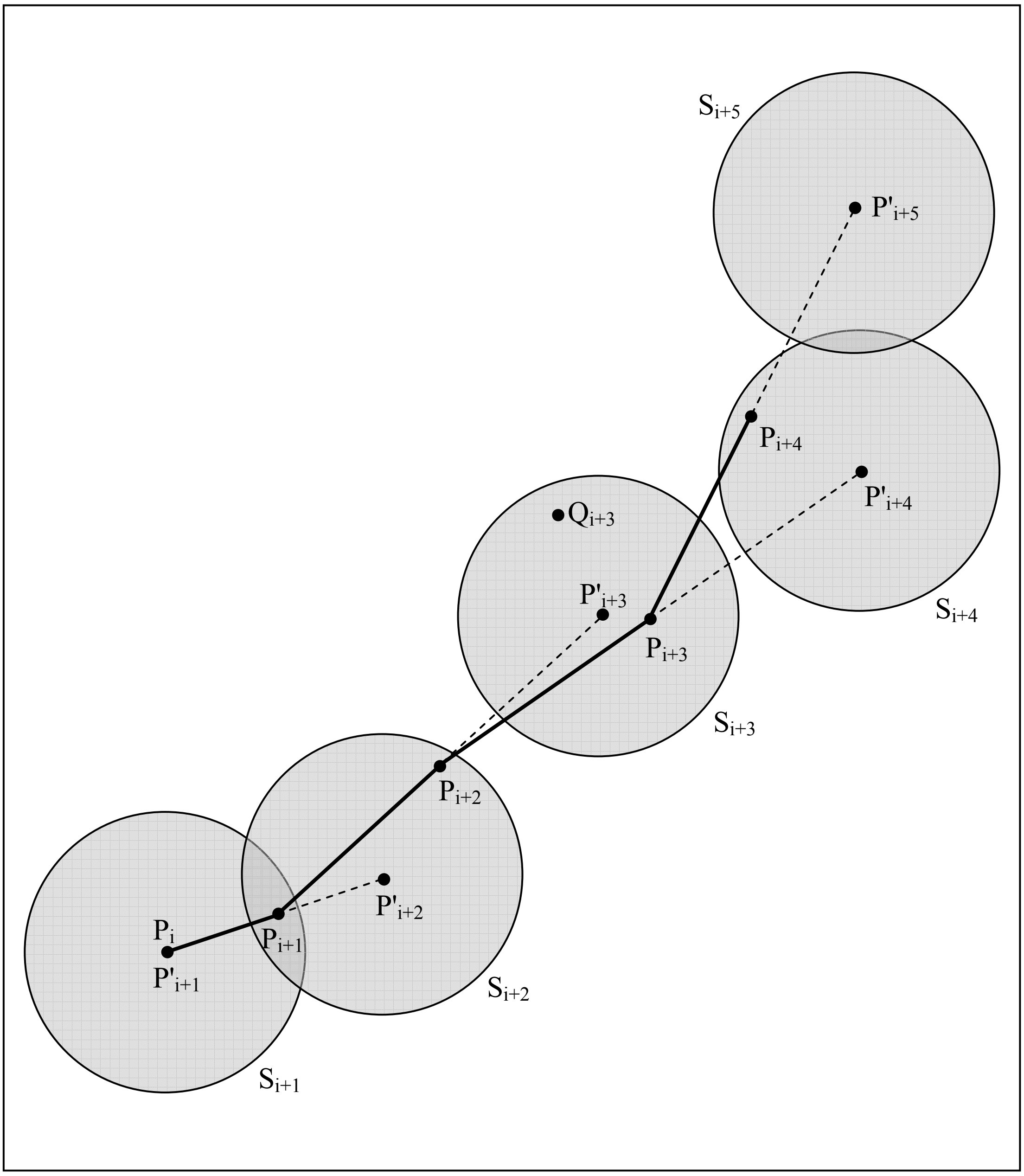


Figure 3.15 Formation of 3D Point Group Initial Tracks

Figure 3.15 shows the steps of this method. The figure is prepared for the 2D case to show easily and extension to 3D case is trivial. In the figure, subscripts show the frame numbers, P_k and Q_k are points in frame k, P'_k is the predicted point in frame k, and S_k is the search region in frame *k*. The track starts from point P_i in frame *i*. As this is the starting point of the track, there are no points of the track in the previous frame to use in the extrapolation operation to find the predicted point in the following frame. Therefore this starting point is taken as the predicted point for the following frame, P'_{i+1}, and the search region S_{i+1} is formed directly around this point. In frame *i+1*, all the points are analyzed if they are in the search region S_{i+1}. The point P_{i+1} is found to be in the search region S_{i+1}. As this point is the only point in the search region, it is taken as the point of the track in frame *i+1*. Using the points P_i and P_{i+1}, the point P'_{i+2} is found and the search region S_{i+2} is formed around this point for frame *i+2*. Then P_{i+2} is found in S_{i+2} and then P'_{i+3} is found and S_{i+3} is formed. In frame *i+3*, there are two points in the search region S_{i+3}, the points P_{i+3} and Q_{i+3}. As P_{i+3} is closer to the predicted point P'_{i+3} than the point Q_{i+3}, P_{i+3} is taken as the point in track in frame *i+3*. Similarly, P'_{i+4}, S_{i+4}, P_{i+4}, P'_{i+5} are found in order and finally S_{i+5} is formed. No points are found in S_{i+5} and therefore *i+4* is the frame that the track ends in. In the figure, the track is shown with bold lines while the prediction extrapolations for the following frame are shown with dotted lines.

This basic temporal matching algorithm did not give satisfactory results. Data association was sometimes problematic and tracks may be lost easily. The velocity of the point is found from only the previous two points and therefore the noise in these two measurements may lead to a wrong velocity assumption and the track may be lost easily. This also makes estimating the position of a missing point in one or more frames and then catching the track again as the velocity assumption using only two frames is not enough to estimate the position after a missing point. If the number of missing points is greater than 1, the track is lost probably.

3.3.9 USING KALMAN FILTER WITH 3D POINT GROUP INITIAL TRACKS

Because of the problems encountered in tracking using a simple nearest neighbor method using the 3D groups, it is necessary to use a better tracking method which will know the direction and speed of the track and estimate the position of the next point in the track so that false point associations will not be seen. Another advantage of such a clever tracking, even if one or more points in the track is not detected, due to occlusion or detection algorithms applied beforehand, the track may be continued for a few frames to catch a point continuing the track. So that the algorithm may be able to handle separated tracks without the help of an operator, which was included in our aims to increase the performance of the tracker.

In this thesis, Kalman Filter is used as an estimator in tracking 3D markers groups. A summary of the Kalman Filter was given in Section 2.6.

Intel® Open Source Computer Vision Library (OpenCV) [9] is used in this thesis work as the Kalman Filter. In this thesis, notation in OpenCV is used as Kalman Filter notation [5] [9] [10].

Kalman Filter or Kalman Estimator will be used in this application. The fundamental equations of the Kalman Filter were given in Section 2.6 in Equations (2.24) and (2.25).

Three independent Kalman Filters are created for x, y, and z directions.

The "states" of the estimators are selected to be the position and the velocity of the marker, that is, the three Kalman Filter has the states

$$\begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \begin{bmatrix} y \\ \dot{y} \end{bmatrix}, \begin{bmatrix} z \\ \dot{z} \end{bmatrix} \tag{3.12}$$

where $\dot{x}$ shows the derivative of x, $\dot{y}$ shows the derivative of y, $\dot{z}$ shows the derivative of z.

The transition equations without the control input function and noise elements are

$x_k = x_{k-1} + \dot{x}_{k-1}$ and $\dot{x}_k = \dot{x}_{k-1}$,

$$y_k = y_{k-1} + \dot{y}_{k-1} \text{ and } \dot{y}_k = \dot{y}_{k-1},$$

$$z_k = z_{k-1} + \dot{z}_{k-1} \text{ and } \dot{z}_k = \dot{z}_{k-1}. \tag{3.13}$$

Therefore, the common transition matrix, which is the vector ***A*** in Equation (3.12), of all three Kalman Filters is

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \tag{3.14}$$

In our application, the control input, ***u*** in Equation (2.24), is not used, therefore the matrix ***B*** in that equation is set to zero.

The measurement is directly the position, therefore the measurement matrix, that is, the matrix ***H*** in Equation (2.25) is

$$\begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{3.15}$$

These values for Kalman Filter mean that a constant velocity model is used in modeling marker movements.

In Section 2.6, the operation cycle of the Kalman Filter was mentioned. In each cycle, Kalman Filter makes a prediction about the state of the system based on previous measurements, and then given the present measurement, it updates its prediction about the state. In OpenCV, the first stage of this cycle is performed with the *cvKalmanPredict* function, and after updating the measurement matrix in the Kalman Filter, the second stage is performed with the *cvKalmanCorrect* function.

As Kalman Filter is a recursive algorithm, it must be initialized before the start of the operation cycles. Process noise covariance, measurement noise covariance and initial a posteriori error covariance are given to the Kalman Filter algorithm before the start of recursive operation. Then Kalman Filter is initialized at the start points of each 3D point group initial track. This initializes the start point of the Kalman Filter.

Kalman Filter is used to track 3D point groups, therefore initialization with the use of 3D point group initial tracks is performed. All initial tracks that have more than or equal to a predetermined number of 3D point groups is taken to initialize three Kalman Filters, one for each direction. This enables the Kalman Filter to adapt to the initial track for this number of 3D point groups so that following the initialization period, the Kalman Filter can track the 3D point groups without leading from this initial track.

After Kalman Filter initialization on the track, the Kalman Filter tracks the 3D point groups independent of the 3D point group initial tracks even if the point group initial track continues to track. This separation from the initial track has to be performed as soon as possible so that the Kalman Filter does not make the same "mistakes" that the initial track does while proceeding through the frames. If the Kalman Filter does not leave the initial track, than it will be only used for filtering, not for estimating, and this would not help the tracking be more robust.

Track gate is the region in the next frame that the next point in the track is predicted to be in. In tracking algorithms, the track gate concept is very important as it is used for filtering out the data that is not likely to be an element of the present track and thus reducing search region and eliminating some association problems.

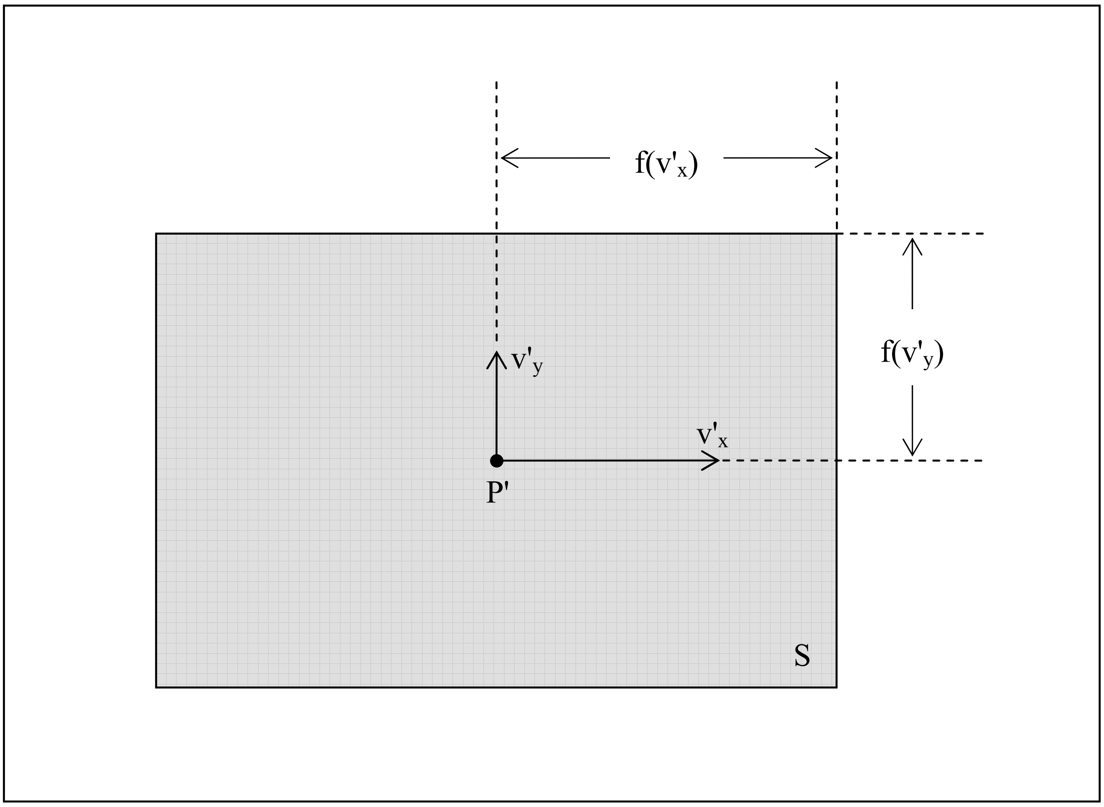


Figure 3.16 Kalman Filter Track Gates

The track gate used is shown in Figure 3.16. The figure is prepared for the 2D case to show easily and extension to 3D case is trivial. In the figure, *P'* is the predicted point for any frame, v'_x and v'_y are the predicted velocities of the predicted point *P'* in that frame, in x and y directions respectively, and *S* is the search region or the track gate for that frame. The size of the track gate is made dependent on the predicted velocities. This dependence is given as

$$\mathrm{f(x)} = \begin{cases} m.x & if \quad x \geq k \\ m.k & if \quad x < k \end{cases} \tag{3.16}$$

where *m* and *k* are constants.

The dependence of the track gate size on the predicted velocity is needed in order to be able to handle large velocities. If the track gate is formed with a constant size which is large enough to cover the position of the point moving with a large velocity, wrong points will be taken into the track in regions where the point moves with a slower velocity. There are some frames where there are no points of a specific track because of occlusion or elimination in previous processes. Allowing the track to have occluded points is generally better than allowing the track to cover a wrong point. Another problem with large velocities is that the error in predicting the velocity or the error in predicting the position may be also large in points moving with large velocities. Therefore to cover the points with larger velocity, the track gate size is made dependent on the predicted velocity of the point in regions where the point moves with large velocities. If the track gate size is a constant multiple of the predicted velocity, another problem arises with small predicted velocities. The track gate size will be small where the predicted velocity is small but this leads to losing the track if the point starts to move after being stationary for a period of time. The markers close to the heel and toe are the markers with quite varying velocities. They are stationary for some period of gait and then they start to move and reach large velocities after which they slow down to stationary position again. In a typical gait cycle, there are stationary and moving states of heel and toe [22]. Therefore the track gate size should allow a quick start of movement from the stationary state. This constraint is satisfied with the condition in Equation (3.16) which guarantees a minimum value, *m.k*, for f(x).

We find independent track gate lengths for the three Kalman Filters corresponding to three directions and these are used to find the 3D track gate. This is because the character of movement is different in three directions.

Each frame represented one cycle of the Kalman Filter. At each frame, a track gate is formed around the predicted value of the next 3D point group position. If there is a 3D point group in this track gate, it is taken as the measurement and Kalman Filter is correcting its prediction. This corrected prediction is used to form the track gate for the next frame. These cycles are repeated through the frames of the sequence until no 3D point groups are found in the track gate.

If there are no 3D point groups in the track gate, the predicted value, that is the center of the track gate, is given the system as if it was the measurement so that one more cycle can be performed. In this second cycle, if a 3D point group is found in the track gate, the Kalman Filter can continue. This case is the situation where one 3D point group is lost in a frame in the track because of occlusion of any other reason. In this case, Kalman Filter is enough for most of the cases to continue the track in spite of this missing 3D point group because of the velocity information inherent in it, enabling it to predict the next state using the latest velocity value in the state definition given in Equation (3.12).

In the case of having no point groups in the track gate, putting the predicted value instead of a lost 3D point group may work for a few frames but if the track is broken for a large number of 3D point groups, Kalman Filter will eventually have to stop. The stopping condition, i.e. the end of a track, is determined by counting the number of consecutive missing 3D point groups. Otherwise, Kalman Filter will continue to the end of the sequence.

In Figure 3.17 through 3.19, 3D tracks formed in a sample gait are shown.

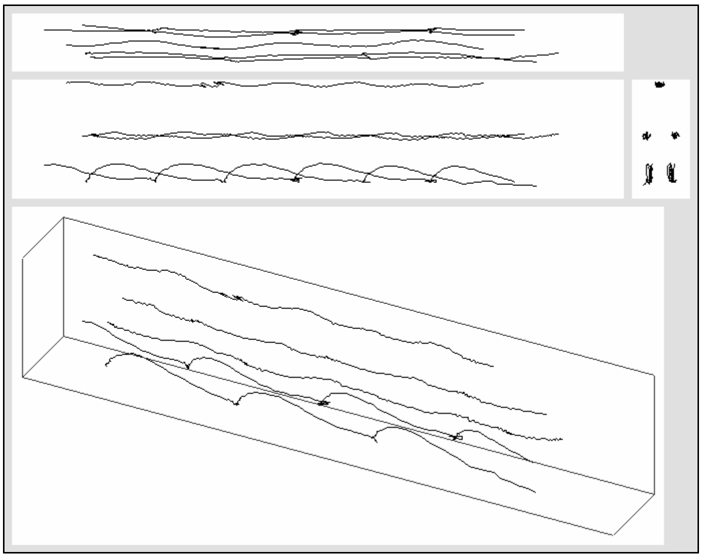


Figure 3.17 Tracks A, F, G, J, and K in a Sample Gait

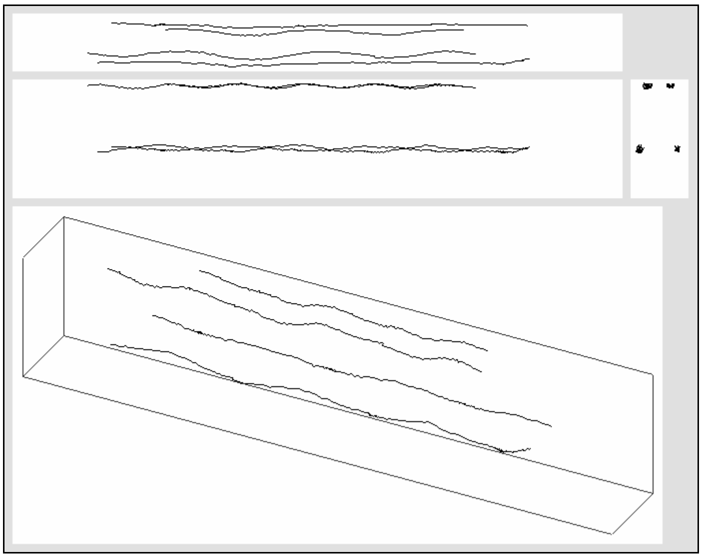


Figure 3.18 Tracks B, C, H, and I in a Sample Gait

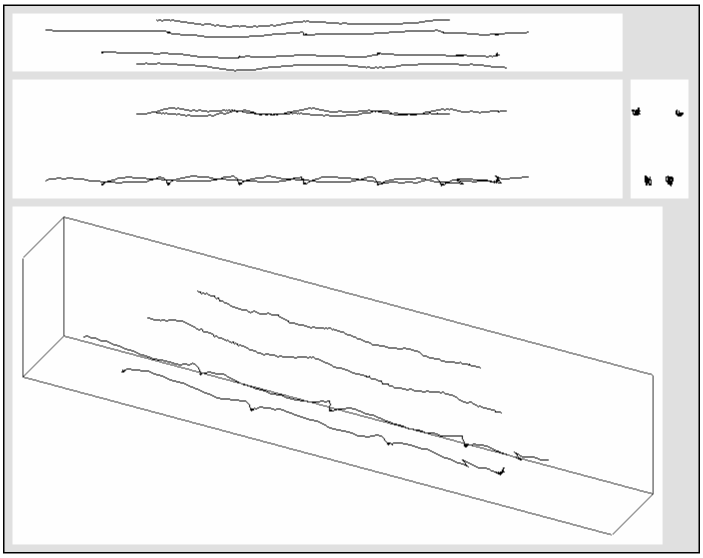
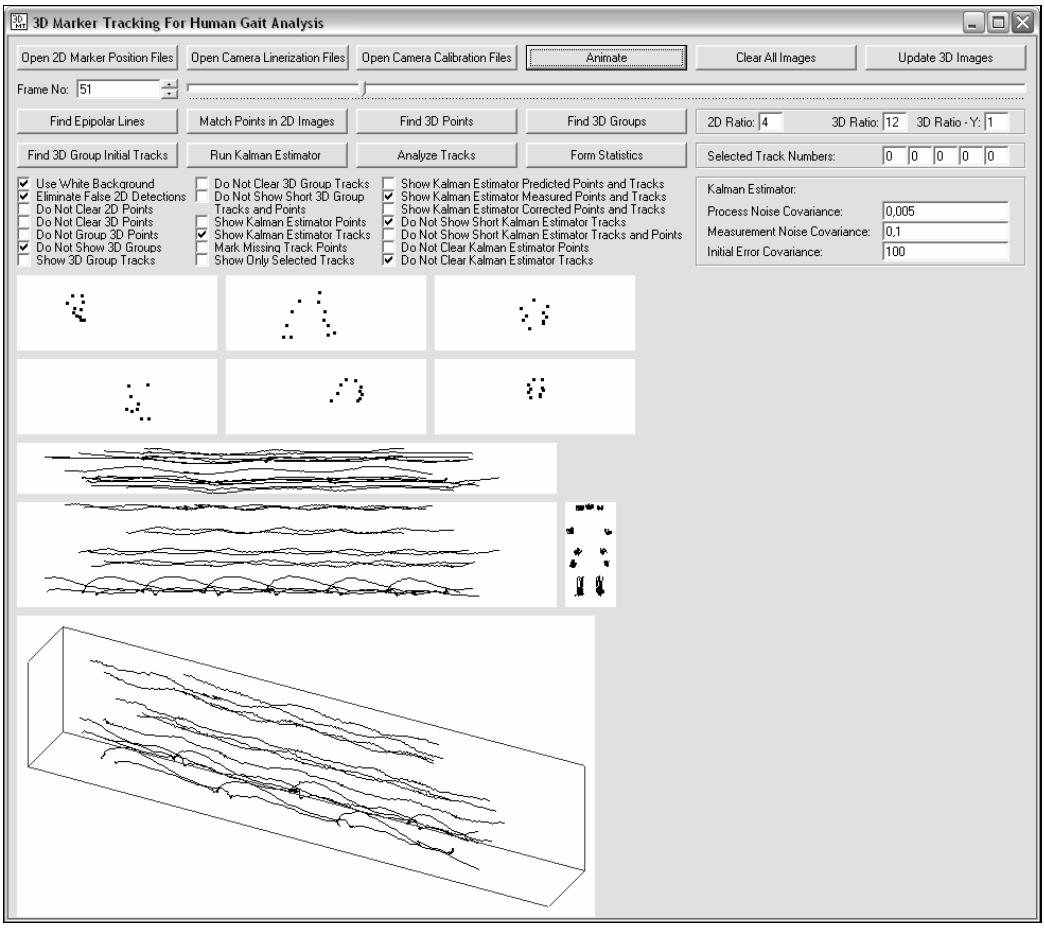


Figure 3.19 Tracks D, E, L, and M in a Sample Gait

CHAPTER 4

THE 3D MARKER TRACKING SOFTWARE

In this thesis work, 3D marker tracking software is developed. This software takes files of 2D marker positions for each of the 6 cameras as input and finds 3D trajectories corresponding to each marker as output. A general view from the software is shown in Figure 4.1.


3D Marker Tracking For Human Gait Analysis
Open 2D Marker Position Files
Open Camera Linerization Files
Open Camera Calibration Files
Animate
Clear All Images
Update 3D Images
Find Epipolar Lines
Match Points in 2D Images
Find 3D Points
Find 3D Groups
Find 3D Group Initial Tracks
Run Kalman Estimator
Analyze Tracks
Form Statistics

Frame No: 51

2D Ratio: 4 3D Ratio: 12 3D Ratio - Y: 1

Selected Track Numbers: 0 0 0 0 0

☑ Use White Background
☑ Eliminate False 2D Detections
☐ Do Not Clear 2D Points
☐ Do Not Clear 3D Points
☐ Do Not Group 3D Points
☑ Do Not Show 3D Groups
☐ Show 3D Group Tracks

☐ Do Not Clear 3D Group Tracks
☐ Do Not Show Short 3D Group Tracks and Points
☐ Show Kalman Estimator Points
☑ Show Kalman Estimator Tracks
☐ Mark Missing Track Points
☐ Show Only Selected Tracks

☐ Show Kalman Estimator Predicted Points and Tracks
☑ Show Kalman Estimator Measured Points and Tracks
☐ Show Kalman Estimator Corrected Points and Tracks
☑ Do Not Show Short Kalman Estimator Tracks
☐ Do Not Show Short Kalman Estimator Tracks and Points
☐ Do Not Clear Kalman Estimator Points
☑ Do Not Clear Kalman Estimator Tracks

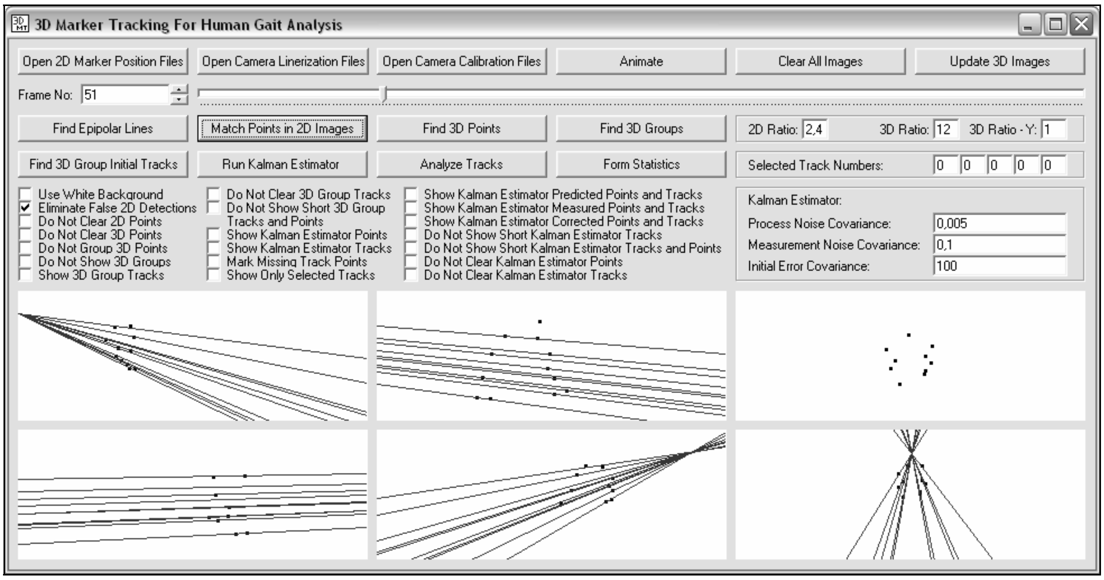
Kalman Estimator:
Process Noise Covariance: 0,005
Measurement Noise Covariance: 0,1
Initial Error Covariance: 100

Figure 4.1 A General View from the 3D Marker Tracking Software

This software has a user friendly interface. While loading files, predetermined files at predetermined locations are automatically selected and loaded unless the user selects different files. Process sequence is embedded into the software and selections of the buttons show the user what the next step is and the user only needs to press the "Enter" key. After processing is finished, viewing selections can be made using the checkboxes. The 6 2D images show the markers from each camera view. Three 3D images are the top, right, and front views of the 3D volume. The fourth is a trimetric 3D image. The sizes of the 2D and 3D images can be decreased by factors given by the user to be able to see the whole sequence on the screen without scrolling. For example, to get the snapshot in Figure 4.1, 2D image sizes are decreased by a factor of 4 and 3D image sizes are decreased by a factor of 12. Kalman Estimator inputs may be changed using the graphical interface by the user.

The sequence of the software is given below with brief explanations of the steps and corresponding buttons.

- "Open 2D Marker Position Files" Button:
  Opens 6 files corresponding to the 6 cameras containing 2D marker positions and forms 2D images in the graphical interface.
- "Open Camera Linearization Files" Button:
  Opens 6 files corresponding to the 6 cameras containing 30 linearization parameters for each camera and performs linearization to all 2D points in the sequence in all cameras.
- "Open Camera Calibration Files" Button:
  Opens 6 files corresponding to the 6 cameras containing 11 calibration parameters for each camera.
- "Find Epipolar Lines" Button:
  Finds epipolar lines all through the sequence. After this operation, Clicking on an image will draw epipolar lines corresponding to the points in that image on the other 5 images. The software graphical interface will look like Figure 4.2 after clicking on the upper right image.


3D Marker Tracking For Human Gait Analysis
Open 2D Marker Position Files
Open Camera Linerization Files
Open Camera Calibration Files
Animate
Clear All Images
Update 3D Images
Find Epipolar Lines
Match Points in 2D Images
Find 3D Points
Find 3D Groups
Find 3D Group Initial Tracks
Run Kalman Estimator
Analyze Tracks
Form Statistics

Frame No: 51

2D Ratio: 2,4 3D Ratio: 12 3D Ratio - Y: 1

Selected Track Numbers: 0 0 0 0 0

- ☐ Use White Background
- ☑ Eliminate False 2D Detections
- ☐ Do Not Clear 2D Points
- ☐ Do Not Clear 3D Points
- ☐ Do Not Group 3D Points
- ☐ Do Not Show 3D Groups
- ☐ Show 3D Group Tracks
- ☐ Do Not Clear 3D Group Tracks
- ☐ Do Not Show Short 3D Group Tracks and Points
- ☐ Show Kalman Estimator Points
- ☐ Show Kalman Estimator Tracks
- ☐ Mark Missing Track Points
- ☐ Show Only Selected Tracks
- ☐ Show Kalman Estimator Predicted Points and Tracks
- ☐ Show Kalman Estimator Measured Points and Tracks
- ☐ Show Kalman Estimator Corrected Points and Tracks
- ☐ Do Not Show Short Kalman Estimator Tracks
- ☐ Do Not Show Short Kalman Estimator Tracks and Points
- ☐ Do Not Clear Kalman Estimator Points
- ☐ Do Not Clear Kalman Estimator Tracks

Kalman Estimator:
Process Noise Covariance: 0,005
Measurement Noise Covariance: 0,1
Initial Error Covariance: 100

Figure 4.2 View of the Software after Finding Epipolar Lines and Clicking on the Upper Right Image

- "Match Points in 2D Images" Button:

  Performs the process of matching 2D points in order to find 3D positions corresponding to these matched 2D points.

- "Find 3D Points" Button:

  Calculates 3D positions corresponding to matched 2D points and forms the 3D images.

- "Find 3D Groups" Button:

  Groups the 3D points that are very close to each other.

- "Find 3D Group Initial Tracks" Button:

  Finds the initial tracks that are formed by linear extrapolation using previous two points and finding the nearest neighbor.

- "Run Kalman Estimator" Button:

  Runs the Kalman estimator starting from the 3D group initial tracks.

There are other functions or selections on the software for viewing or analyzing purposes. These functions and some of the selections are given below:

- "Analyze Tracks" Button: Selects only long tracks as valid tracks and assigns numbers to these tracks so that each track can be viewed separately.
- "Form Statistics" Button: Calculates the average and variance statistics of the tracks.
- "Animate" Button: Animates the sequence so that movements can be seen easily.
- "Clear All Images" Button: Clears all 2D and 3D images.
- "Update 3D Images" Button: Updates the 3D images after a change in one of the 3D image size factors defined by "3D Ratio" and "3D Ratio - Y".
- "Frame No" Box: Allows the user to see and change which frame is shown in the 2D and 3D images.
- "2D Ratio" Box: Allows the user to change the factor that all dimensions of the 2D image are decreased by.
- "3D Ratio" Box: Allows the user to change the factor that all dimensions of the 3D image are decreased by.
- "3D Ratio - Y" Box: Allows the user to change the factor that the Y axis of the 3D image is decreased by.
- "Selected Track Numbers" Boxes: Allows the user to change the tracks that are shown in 3D images.
- "Process Noise Covariance", "Measurement Noise Covariance" and "Initial Error Covariance" Boxes: Allows the user to change these inputs of the Kalman Estimator.
- "Eliminate False 2D Detections" Selection: Allows the user to enable or disable elimination of false detections in 2D images.

For the user, there are also many viewing selections which can be used in a large number of combinations.

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter, results of the algorithm are given. Firstly, markers are named according to Figure 5.1 (a) so that fixed names can be used for each marker, each track and each segment between markers. Segments are named using the two markers at each end, i.e. "A-B" is the name for the segment between markers "A" and "B". In Figure 5.1 (b), the skeleton formed by markers is shown.

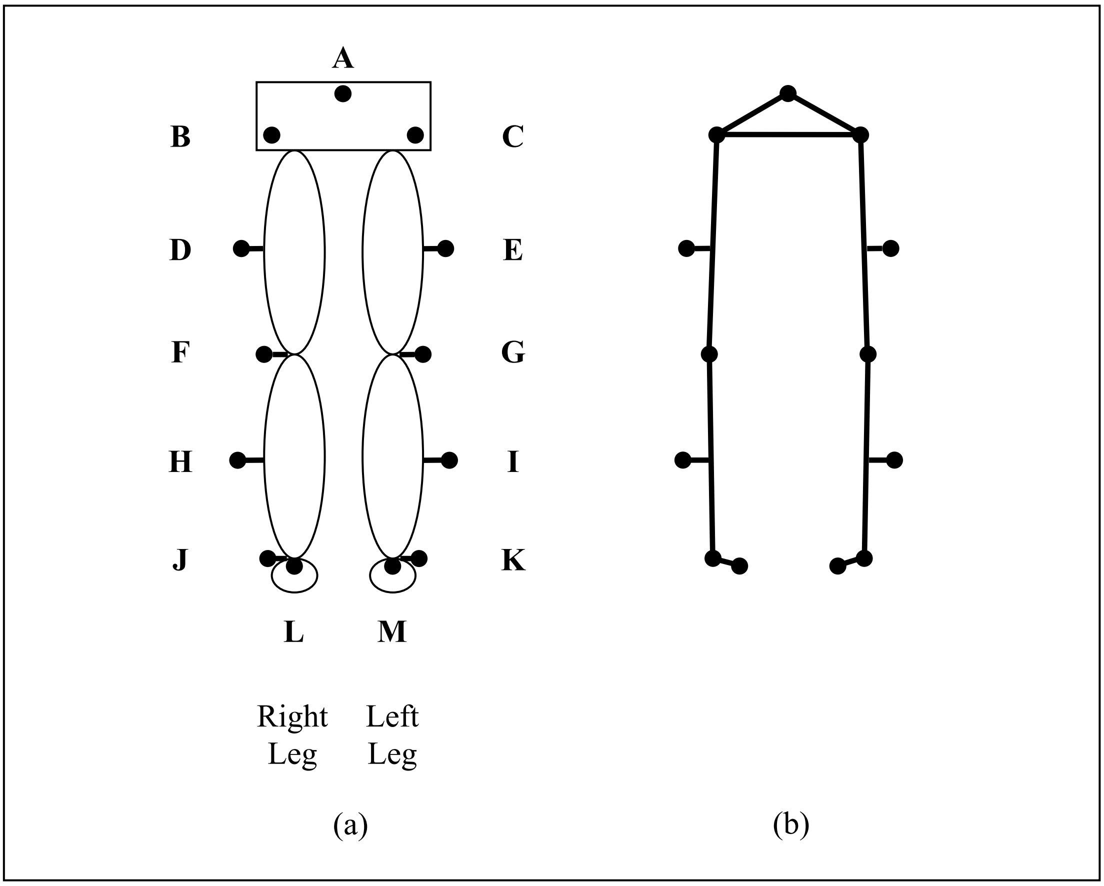


Figure 5.1 (a) Marker Names (b) the Skeleton Formed by Markers

The lengths of the segments D-F, E-G, F-H, G-I, H-J, and I-K are expected to be fixed because of their positions on the body. Other segments may have variable lengths because of the changes in the angle of the joints they are connected to.

Our principle aim was finding marker tracks throughout the sequence without the need for an operator. If the tracks are lost easily, a track corresponding to a marker will be divided into parts and an operator will have to connect them in order to have a complete track for a marker. In our algorithm, if there are no markers in the track gate for a frame, the predicted position of the track at that frame is used as if it was a real measurement, that is, as if there was a marker detected at the predicted point position. This predicted position is found by the Kalman Filter. This makes the prediction of the marker position at a frame more accurate than prediction with linear extrapolation using two previous point positions. In Figures 5.2 and 5.3, two examples of application of this method on a sample gait data are seen. In Figure 5.2, a snapshot of a small part of the 3D track named F in Figure 5.1 is shown and in Figure 5.3, a snapshot of a small part of the 3D track named J in Figure 5.1 is shown. In these figures, dots show the real measurements and large crosses show the predicted position of the marker which is used as if it was a real measurement because no markers are found in that track gate. The points of the track in adjacent frames are connected with lines. There are 6 occurrences where the predicted position of the marker is used as if it was a real measurement in Figure 5.2 and 3 occurrences in Figure 5.3.

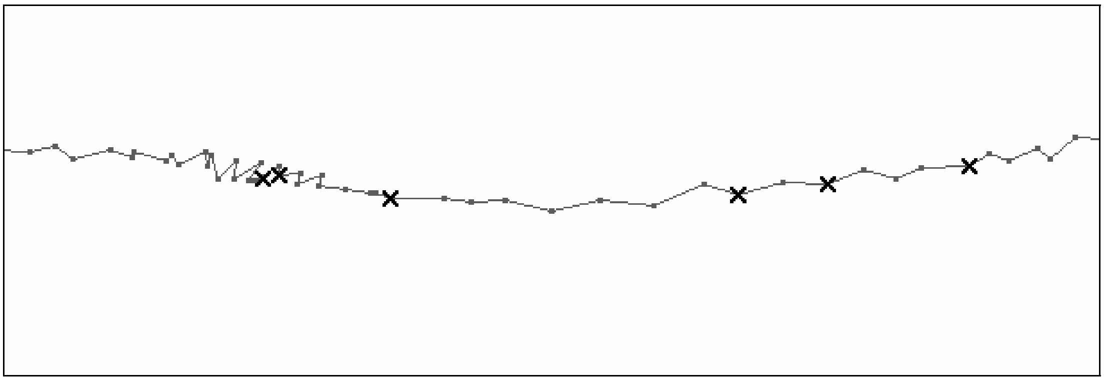


Figure 5.2 A Part of Track F with Missing Measurements

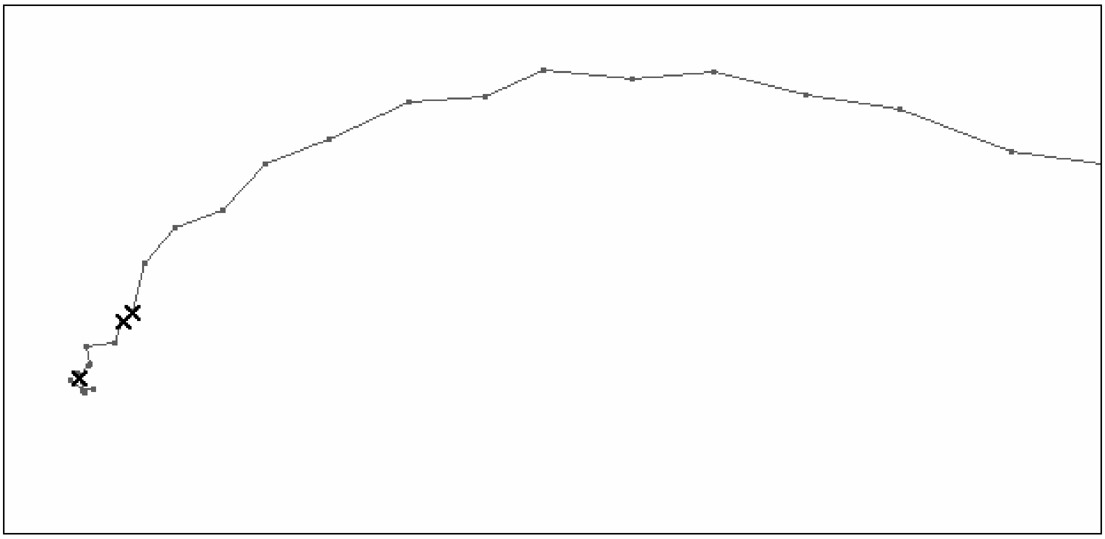


Figure 5.3 A Part of Track J with Missing Measurements

As can be seen from these figures, a linear extrapolation would not be enough to predict the position of the marker in the following frame as the noise in the measurements may lead to a wrong "instant velocity". This will result in a wrong position prediction and afterwards wrong placement and size of the track gate. But with the Kalman Filter, measurements are filtered and making more accurate position and velocity prediction is possible.

To predict the point position and velocity each frame, Kalman Filter is used instead of linear extrapolation which uses previous two points. As mentioned in Section 2.6, two important inputs of the Kalman Filter are "the process noise covariance" and "the measurement noise covariance". These can be seen in Kalman equations which were given in Equations (2.24) and (2.25). These constants effect the prediction and therefore the tracking performance.

In Tables 5.1 through 5.12, the average length and variance of the length of each segment shown in Figure 5.1 are given. In these figures, the standard deviation of the length and the ratio of the standard deviation of the length to the average length are also shown for each segment. The tables are prepared using three different values for the "process noise covariance" (PNC) and three different values for the "measurement noise covariance" (MNC) of the Kalman Filter. Entries named as "measured" are the measurements and entries named as "corrected" are the a posteriori estimates of the Kalman Filter for that frame.

In these tables, variances corresponding to Kalman Filter "corrected" tracks are seen to be smaller or equal to the variances of the "measured" tracks for the segments whose lengths are expected to be fixed, i.e. the segments D-F, E-G, F-H, G-I, H-J, and I-K, except very few cases. This means that the results obtained by applying Kalman Filter are more reliable.

Table 5.1 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment A-C for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	265,528	265,528	265,528
Corrected Average	264,813	264,927	265,041
Measured Variance	197,787	197,787	197,787
Corrected Variance	168,471	170,515	170,087
Measured Standard Deviation	14,064	14,064	14,064
Corrected Standard Deviation	12,980	13,058	13,042
Measured Standard Deviation / Average	0,053	0,053	0,053
Corrected Standard Deviation / Average	0,049	0,049	0,049

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	242,508	243,180	234,431
Corrected Average	242,049	241,918	234,246
Measured Variance	1625,499	1643,426	2158,151
Corrected Variance	1553,492	1725,190	2158,032
Measured Standard Deviation	40,317	40,539	46,456
Corrected Standard Deviation	39,414	41,535	46,455
Measured Standard Deviation / Average	0,166	0,167	0,198
Corrected Standard Deviation / Average	0,163	0,172	0,198

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	234,431	233,431	232,462
Corrected Average	234,231	233,262	232,215
Measured Variance	2158,151	2167,013	2142,683
Corrected Variance	2162,269	2187,642	2171,741
Measured Standard Deviation	46,456	46,551	46,289
Corrected Standard Deviation	46,500	46,772	46,602
Measured Standard Deviation / Average	0,198	0,199	0,199
Corrected Standard Deviation / Average	0,199	0,201	0,201

Table 5.2 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment A-B for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	271,232	271,232	271,232
Corrected Average	271,098	271,091	271,152
Measured Variance	187,049	187,049	187,049
Corrected Variance	158,829	160,302	161,779
Measured Standard Deviation	13,677	13,677	13,677
Corrected Standard Deviation	12,603	12,661	12,719
Measured Standard Deviation / Average	0,050	0,050	0,050
Corrected Standard Deviation / Average	0,046	0,047	0,047

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	254,290	254,650	247,500
Corrected Average	254,290	254,090	247,740
Measured Variance	1444,373	1451,592	2172,098
Corrected Variance	1397,271	1537,680	2145,826
Measured Standard Deviation	38,005	38,100	46,606
Corrected Standard Deviation	37,380	39,213	46,323
Measured Standard Deviation / Average	0,149	0,150	0,188
Corrected Standard Deviation / Average	0,147	0,154	0,187

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	247,500	246,875	246,240
Corrected Average	247,644	246,990	246,308
Measured Variance	2172,098	2181,707	2188,758
Corrected Variance	2155,423	2172,354	2185,829
Measured Standard Deviation	46,606	46,709	46,784
Corrected Standard Deviation	46,427	46,609	46,753
Measured Standard Deviation / Average	0,188	0,189	0,190
Corrected Standard Deviation / Average	0,187	0,189	0,190

Table 5.3 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment C-E for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	300,782	300,782	300,782
Corrected Average	300,605	300,630	300,588
Measured Variance	190,060	190,060	190,060
Corrected Variance	198,929	195,227	192,632
Measured Standard Deviation	13,786	13,786	13,786
Corrected Standard Deviation	14,104	13,972	13,879
Measured Standard Deviation / Average	0,046	0,046	0,046
Corrected Standard Deviation / Average	0,047	0,046	0,046

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	300,782	300,782	300,782
Corrected Average	300,588	300,563	300,613
Measured Variance	190,060	190,060	190,060
Corrected Variance	194,906	193,981	190,157
Measured Standard Deviation	13,786	13,786	13,786
Corrected Standard Deviation	13,961	13,928	13,790
Measured Standard Deviation / Average	0,046	0,046	0,046
Corrected Standard Deviation / Average	0,046	0,046	0,046

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	300,782	300,782	300,782
Corrected Average	300,588	300,588	300,605
Measured Variance	190,060	190,060	190,060
Corrected Variance	191,945	192,031	192,476
Measured Standard Deviation	13,786	13,786	13,786
Corrected Standard Deviation	13,854	13,858	13,874
Measured Standard Deviation / Average	0,046	0,046	0,046
Corrected Standard Deviation / Average	0,046	0,046	0,046

Table 5.4 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment B-D for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	284,372	284,372	284,372
Corrected Average	284,759	284,672	284,620
Measured Variance	92,476	92,476	92,476
Corrected Variance	96,546	95,169	94,737
Measured Standard Deviation	9,616	9,616	9,616
Corrected Standard Deviation	9,826	9,755	9,733
Measured Standard Deviation / Average	0,034	0,034	0,034
Corrected Standard Deviation / Average	0,035	0,034	0,034

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	284,372	284,372	284,372
Corrected Average	284,423	284,416	284,423
Measured Variance	92,476	92,476	92,476
Corrected Variance	92,237	91,435	93,143
Measured Standard Deviation	9,616	9,616	9,616
Corrected Standard Deviation	9,604	9,562	9,651
Measured Standard Deviation / Average	0,034	0,034	0,034
Corrected Standard Deviation / Average	0,034	0,034	0,034

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	284,372	284,372	284,372
Corrected Average	284,387	284,387	284,387
Measured Variance	92,476	92,476	92,476
Corrected Variance	92,365	92,365	92,162
Measured Standard Deviation	9,616	9,616	9,616
Corrected Standard Deviation	9,611	9,611	9,600
Measured Standard Deviation / Average	0,034	0,034	0,034
Corrected Standard Deviation / Average	0,034	0,034	0,034

Table 5.5 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment E-G for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	238,576	238,576	238,576
Corrected Average	238,896	238,752	238,712
Measured Variance	14,322	14,322	14,322
Corrected Variance	11,519	11,034	11,227
Measured Standard Deviation	3,784	3,784	3,784
Corrected Standard Deviation	3,394	3,322	3,351
Measured Standard Deviation / Average	0,016	0,016	0,016
Corrected Standard Deviation / Average	0,014	0,014	0,014

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	238,576	238,576	238,576
Corrected Average	238,624	238,608	238,592
Measured Variance	14,322	14,322	14,322
Corrected Variance	12,078	13,198	13,875
Measured Standard Deviation	3,784	3,784	3,784
Corrected Standard Deviation	3,475	3,633	3,725
Measured Standard Deviation / Average	0,016	0,016	0,016
Corrected Standard Deviation / Average	0,015	0,015	0,016

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	238,576	238,576	238,576
Corrected Average	238,568	238,560	238,552
Measured Variance	14,322	14,322	14,322
Corrected Variance	14,473	14,471	14,423
Measured Standard Deviation	3,784	3,784	3,784
Corrected Standard Deviation	3,804	3,804	3,798
Measured Standard Deviation / Average	0,016	0,016	0,016
Corrected Standard Deviation / Average	0,016	0,016	0,016

Table 5.6 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment D-F for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	260,796	260,796	260,796
Corrected Average	261,322	261,138	261,000
Measured Variance	13,240	13,240	13,240
Corrected Variance	11,555	11,592	11,539
Measured Standard Deviation	3,639	3,639	3,639
Corrected Standard Deviation	3,399	3,405	3,397
Measured Standard Deviation / Average	0,014	0,014	0,014
Corrected Standard Deviation / Average	0,013	0,013	0,013

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	260,796	260,796	260,796
Corrected Average	260,875	260,888	260,836
Measured Variance	13,240	13,240	13,240
Corrected Variance	11,953	12,098	12,352
Measured Standard Deviation	3,639	3,639	3,639
Corrected Standard Deviation	3,457	3,478	3,515
Measured Standard Deviation / Average	0,014	0,014	0,014
Corrected Standard Deviation / Average	0,013	0,013	0,013

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	260,796	260,796	260,796
Corrected Average	260,862	260,875	260,875
Measured Variance	13,240	13,240	13,240
Corrected Variance	13,089	13,242	13,242
Measured Standard Deviation	3,639	3,639	3,639
Corrected Standard Deviation	3,618	3,639	3,639
Measured Standard Deviation / Average	0,014	0,014	0,014
Corrected Standard Deviation / Average	0,014	0,014	0,014

Table 5.7 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment G-I for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	152,390	152,390	152,390
Corrected Average	152,881	152,791	152,537
Measured Variance	19,525	19,525	19,525
Corrected Variance	17,787	17,930	17,570
Measured Standard Deviation	4,419	4,419	4,419
Corrected Standard Deviation	4,217	4,234	4,192
Measured Standard Deviation / Average	0,029	0,029	0,029
Corrected Standard Deviation / Average	0,028	0,028	0,027

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	152,390	152,390	152,390
Corrected Average	152,559	152,401	152,373
Measured Variance	19,525	19,525	19,525
Corrected Variance	18,324	18,029	19,139
Measured Standard Deviation	4,419	4,419	4,419
Corrected Standard Deviation	4,281	4,246	4,375
Measured Standard Deviation / Average	0,029	0,029	0,029
Corrected Standard Deviation / Average	0,028	0,028	0,029

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	152,390	152,390	152,390
Corrected Average	152,384	152,395	152,379
Measured Variance	19,525	19,525	19,525
Corrected Variance	19,546	19,505	19,436
Measured Standard Deviation	4,419	4,419	4,419
Corrected Standard Deviation	4,421	4,416	4,409
Measured Standard Deviation / Average	0,029	0,029	0,029
Corrected Standard Deviation / Average	0,029	0,029	0,029

Table 5.8 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment F-H for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	168,722	168,722	168,722
Corrected Average	169,528	169,256	169,089
Measured Variance	30,077	30,077	30,077
Corrected Variance	31,440	30,313	29,616
Measured Standard Deviation	5,484	5,484	5,484
Corrected Standard Deviation	5,607	5,506	5,442
Measured Standard Deviation / Average	0,033	0,033	0,033
Corrected Standard Deviation / Average	0,033	0,033	0,032

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	168,722	168,722	168,722
Corrected Average	168,928	168,839	168,733
Measured Variance	30,077	30,077	30,077
Corrected Variance	28,677	29,360	29,871
Measured Standard Deviation	5,484	5,484	5,484
Corrected Standard Deviation	5,355	5,418	5,465
Measured Standard Deviation / Average	0,033	0,033	0,033
Corrected Standard Deviation / Average	0,032	0,032	0,032

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	168,722	168,722	168,722
Corrected Average	168,733	168,728	168,739
Measured Variance	30,077	30,077	30,077
Corrected Variance	29,883	30,173	30,169
Measured Standard Deviation	5,484	5,484	5,484
Corrected Standard Deviation	5,467	5,493	5,493
Measured Standard Deviation / Average	0,033	0,033	0,033
Corrected Standard Deviation / Average	0,032	0,033	0,033

Table 5.9 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment I-K for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	321,006	320,793	320,570
Corrected Average	323,158	322,145	321,380
Measured Variance	71,149	54,695	43,839
Corrected Variance	73,346	50,218	38,940
Measured Standard Deviation	8,435	7,396	6,621
Corrected Standard Deviation	8,564	7,086	6,240
Measured Standard Deviation / Average	0,026	0,023	0,021
Corrected Standard Deviation / Average	0,027	0,022	0,019

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	320,570	320,570	320,578
Corrected Average	320,832	320,626	320,539
Measured Variance	43,839	43,839	43,592
Corrected Variance	38,112	38,503	39,999
Measured Standard Deviation	6,621	6,621	6,602
Corrected Standard Deviation	6,173	6,205	6,324
Measured Standard Deviation / Average	0,021	0,021	0,021
Corrected Standard Deviation / Average	0,019	0,019	0,020

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	320,578	319,821	319,821
Corrected Average	320,333	319,611	319,611
Measured Variance	43,592	44,155	44,155
Corrected Variance	41,729	44,106	44,067
Measured Standard Deviation	6,602	6,645	6,645
Corrected Standard Deviation	6,460	6,641	6,638
Measured Standard Deviation / Average	0,021	0,021	0,021
Corrected Standard Deviation / Average	0,020	0,021	0,021

Table 5.10 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment H-J for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	319,343	319,343	319,343
Corrected Average	321,203	320,636	320,175
Measured Variance	37,966	37,966	37,966
Corrected Variance	34,081	30,923	29,872
Measured Standard Deviation	6,162	6,162	6,162
Corrected Standard Deviation	5,838	5,561	5,466
Measured Standard Deviation / Average	0,019	0,019	0,019
Corrected Standard Deviation / Average	0,018	0,017	0,017

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	319,160	319,160	319,160
Corrected Average	319,670	319,460	319,310
Measured Variance	17,314	17,314	17,314
Corrected Variance	15,793	16,517	15,454
Measured Standard Deviation	4,161	4,161	4,161
Corrected Standard Deviation	3,974	4,064	3,931
Measured Standard Deviation / Average	0,013	0,013	0,013
Corrected Standard Deviation / Average	0,012	0,013	0,012

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	319,160	319,160	319,160
Corrected Average	319,160	319,180	319,200
Measured Variance	17,314	17,314	17,314
Corrected Variance	16,431	16,453	16,493
Measured Standard Deviation	4,161	4,161	4,161
Corrected Standard Deviation	4,054	4,056	4,061
Measured Standard Deviation / Average	0,013	0,013	0,013
Corrected Standard Deviation / Average	0,013	0,013	0,013

Table 5.11 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment K-M for Different Values of PNC and MNC

	PNC:	0,005		
	MNC:	0,100	0,050	0,025
Measured Average		129,581	132,011	132,330
Corrected Average		130,430	132,553	132,585
Measured Variance		337,824	148,266	133,476
Corrected Variance		301,997	113,757	113,549
Measured Standard Deviation		18,380	12,176	11,553
Corrected Standard Deviation		17,378	10,666	10,656
Measured Standard Deviation / Average		0,142	0,092	0,087
Corrected Standard Deviation / Average		0,133	0,080	0,080

	PNC:	0,050		
	MNC:	0,100	0,050	0,025
Measured Average		132,330	132,330	132,354
Corrected Average		132,319	132,298	132,206
Measured Variance		133,476	133,476	132,886
Corrected Variance		120,441	125,210	129,805
Measured Standard Deviation		11,553	11,553	11,528
Corrected Standard Deviation		10,975	11,190	11,393
Measured Standard Deviation / Average		0,087	0,087	0,087
Corrected Standard Deviation / Average		0,083	0,085	0,086

	PNC:	0,500		
	MNC:	0,100	0,050	0,025
Measured Average		132,354	132,809	132,809
Corrected Average		132,175	132,617	132,638
Measured Variance		132,886	74,579	74,579
Corrected Variance		134,895	78,832	78,680
Measured Standard Deviation		11,528	8,636	8,636
Corrected Standard Deviation		11,614	8,879	8,870
Measured Standard Deviation / Average		0,087	0,065	0,065
Corrected Standard Deviation / Average		0,088	0,067	0,067

Table 5.12 Average Length, Variance and Standard Deviation of the Length and Ratio of Standard Deviation of the Length to Average Length of the Segment J-L for Different Values of PNC and MNC

PNC:	0,005		
MNC:	0,100	0,050	0,025
Measured Average	153,034	153,034	153,034
Corrected Average	153,691	153,450	153,383
Measured Variance	57,618	57,618	57,618
Corrected Variance	93,652	72,342	61,189
Measured Standard Deviation	7,591	7,591	7,591
Corrected Standard Deviation	9,677	8,505	7,822
Measured Standard Deviation / Average	0,050	0,050	0,050
Corrected Standard Deviation / Average	0,063	0,055	0,051

PNC:	0,050		
MNC:	0,100	0,050	0,025
Measured Average	153,804	153,804	153,804
Corrected Average	154,178	154,103	153,972
Measured Variance	64,624	64,624	64,624
Corrected Variance	51,717	55,046	58,176
Measured Standard Deviation	8,039	8,039	8,039
Corrected Standard Deviation	7,191	7,419	7,627
Measured Standard Deviation / Average	0,052	0,052	0,052
Corrected Standard Deviation / Average	0,047	0,048	0,050

PNC:	0,500		
MNC:	0,100	0,050	0,025
Measured Average	153,804	153,804	153,804
Corrected Average	153,841	153,785	153,813
Measured Variance	64,624	64,624	64,624
Corrected Variance	62,861	64,354	64,060
Measured Standard Deviation	8,039	8,039	8,039
Corrected Standard Deviation	7,929	8,022	8,004
Measured Standard Deviation / Average	0,052	0,052	0,052
Corrected Standard Deviation / Average	0,052	0,052	0,052

One measure of ability of tracking in spite of a missing marker is the number of occurrences where a marker is not found in the track gate and predicted value of the Kalman Filter is used as if it was the measurement and the track continues afterwards finding a marker in the track gate in the following frame. The number of occurrences of using predicted position instead of measured position, that is, the number of occurrences of missing point prediction with three different process noise covariance (PNC) and three different measurement noise covariance (MNC) values are given in Table 5.13 for the segments shown in Figure 5.1. The entries in this table give how many times the track is overcoming a problem of missing point that may have lead to losing the track and continues afterwards.

Table 5.13 Number of Occurrences of Missing Point Prediction for Different Values of PNC and MNC

PNC:	0,005			0,050			0,500		
MNC:	0,100	0,050	0,025	0,100	0,050	0,025	0,100	0,050	0,025
A	0	0	0	1	1	8	8	8	8
B	2	2	2	2	2	2	2	2	2
C	3	3	3	3	3	3	3	3	3
D	3	3	3	3	3	3	3	3	3
E	2	2	2	2	2	2	2	2	2
F	6	6	6	6	6	6	6	6	6
G	2	2	2	2	2	2	2	2	2
H	13	13	13	13	13	13	13	13	13
I	0	0	0	0	0	0	0	0	0
J	5	5	5	3	3	3	3	3	3
K	2	0	0	0	0	0	0	0	0
L	10	10	10	10	10	5	5	5	5
M	1	1	1	1	1	2	2	2	2

CHAPTER 6

CONCLUSION

This thesis is on 3D marker tracking for human gait analysis. Firstly 2D marker positions are analyzed and false detections in 2D marker positions are eliminated. Then the distortions in 2D positions of markers in each camera view are corrected using the linearization parameters. After correction of 2D point positions, epipolar lines are formed in each camera view using the calibration parameters. After 2D marker matching, matched 2D marker positions are used to form 3D marker positions using the calibration parameters again. The 3D markers that are very close to each other are grouped to find an estimate of the real 3D marker. After grouping, 3D tracks are formed using these 3D groups.

The algorithm of elimination of false detection points in 2D images was successful and useful as it eliminated the false stationary 3D points which are generally out of the calibration frame. The application software is adjusted to show all the 3D points generated in the output images but these false 3D points may sometimes be in extreme points in 3D space, making it impossible to see the real markers of the gait: if the 3D space to be viewed is very large, real markers may be stuck to a very small area. This was eliminated using the 2D stationary marker elimination method, which worked quite well.

The linearization and calibration algorithms have already been developed and used in this system. The laboratory has the needed infrastructure (hardware and software) for linearization and calibration of the cameras. There are sets of calibration and linearization parameters for each set of marker 2D position data. This makes the measurement from the system be correct as linearization and calibration are performed frequently.

Grouping points in 3D coordinate system was also important to ease the problem. As the 3D points generally group in a "cloud" in 3D space, association of a 3D point with a 3D marker point is not very problematic because all the points in the "cloud" are associated with the same 3D marker point, but separated a little because of the digitization and marker detection processes.

3D group initial tracks did not work satisfactorily. Finding the nearest neighbor of the point in a frame predicted with linear extrapolation using the two points in the previous two frames did not give satisfactory results. To overcome the large error in "instant velocity" of the marker found using previous two points, the area of neighborhood has to be increased but increasing the area of neighborhood results in association problems for some 3D groups of different tracks that are very close in some portions of the gait. This emerged the need for more accurate position and velocity prediction, and therefore Kalman Filter estimation is used.

Kalman Filter worked well with the problem. If the algorithms up to Kalman Filtering are working well, then Kalman Filter makes the tracking algorithm be more robust. Less operator interaction to connect the tracks is required in this system.

The variances corresponding to Kalman Filter "corrected" tracks are seen to be smaller or equal to the variances of the "measured" tracks for the segments whose lengths are expected to be fixed, except very few cases. This means that the results obtained by applying Kalman Filter are more reliable.

In the algorithm presented in this thesis, constraints of the body are not used. An algorithm using a body model and fitting the measured marker positions onto this body model would give better results. In this way, false detections may be discarded and data association errors may be handled more easily. Also, different walking speeds of the subjects affect the performance of the algorithm because of the fixed minimum value for the track gate, which is independent of the predicted velocity.

REFERENCES

[1] Güler, H. C., *Biomechanical Modeling of Lower Extremity and Simulation of Foot During Gait*, PhD Thesis, Mechanical Engineering Department, Middle East Technical University, Turkey, 1998.

[2] Karpat, Y., *Development and Testing of Kinematic Data Acquisition Tools for a Gait Analysis System*, MSc Thesis, Mechanical Engineering Department, Middle East Technical University, Turkey, 2000.

[3] Shafiq, M. S., *Motion Tracking in Gait Analysis*, MSc Thesis, Mechanical Engineering Department, Middle East Technical University, 1998.

[4] Kalman, R. E., A New Approach to Linear Filtering and Prediction Problems, *Transaction of the ASME — Journal of Basic Engineering*, March 1960, pp 35-45.

[5] Welch, G., Bishop G., An Introduction to the Kalman Filter, *Technical Report TR95-041*, Department of Computer Science, University of North Carolina, 1995, Updated April 5, 2004.

[6] Schalkoff, R. J., *Digital Image Processing and Computer Vision*, John Wiley & Sons, Inc., Singapore, 1989, pp 18-69, 459-461.

[7] Abdel-Aziz, Y.I., Karara, H.M., Direct linear transformation from comparator coordinates into object space coordinates, *Proceedings of the Symposium on Close-Range Photogrammetry*, Falls Church, VA: American Society of Photogrammetry, 1971, pp 1-18.

[8] http://kwon3d.com/theory/dlt/dlt.html

[9] http://www.intel.com/technology/computing/opencv/

[10] The Documentation in the OpenCV Library

[11] Söylemez, B., *An Investigation on the Gait Analysis Protocol of the "KISS" Motion Analysis System*, MSc Thesis, Mechanical Engineering Department, Middle East Technical University, Turkey, 2002.

[12] Kentel, B. B., *Evaluation and Interpretation of Normal Gait*, MSc Thesis, Mechanical Engineering Department, Middle East Technical University, Turkey, 2002.

[13] Shafiq, M. S., Tümer, T., Güler, H. C., Marker Detection and Trajectory Generation Algorithms for a Multicamera Based Gait Analysis System, *Mechatronics 11,* 2001, pp 409-437.

[14] Fischer, J. E., *Reconstruction of Three-Dimensional Surfaces from Picture Series with Different Resolution Grades*, PhD Thesis, University of Stuttgart, 1993.

[15] Bar-Shalom, Y., Li, X., *Estimation and Tracking Principles, Techniques, and Software*, Artech House, Inc., 1993, pp 207-257.

[16] Maybeck, P. S., *Stochastic Models, Estimation, and Control, Volume 1*, Academic Press, Inc., 1979.

[17] Brown, R. G., Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering, Second Edition*, John Wiley & Sons, Inc., 1992.

[18] Jacobs, O. L. R., *Introduction to Control Theory, 2nd Edition*, Oxford University Press, 1993.

[19] Walton, J. S., *Close-range cine-photogrammetry: A generalized technique for quantifying gross human motion*, Unpublished Ph.D. Dissertation, Pennsylvania State University, University Park, 1981.

[20] Hatze, H., High-precision three-dimensional photogrammetric calibration and object space reconstruction using a modified DLT approach, *J. Biomech 21*, 1988, pp 533-538.

[21] Hwang, J. J., Hall. E. L., Matching of featured objects using relational tables from stereo images, *Computer Graphics and Image Processing*,

1982, 20, 22-42.

[22] Whittle, M. W., *Gait Analysis: an introduction, Third Edition*, Butterworth-Heinemann, 2002.