PERFORMANCE IMPROVEMENT OF A 3-D CONFIGURATION RECONSTRUCTION ALGORITHM FOR AN OBJECT USING A SINGLE CAMERA IMAGE

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

BY

SİBEL ÖZKILIÇ

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MECHANICAL ENGINEERING

DECEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences.

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. Kemal İder

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. Bülent E. Platin

Supervisor

Examining Committee Members

Prof Dr. Turgut Tümer (Chairman)

Prof Dr. Bülent E. Platin

Prof. Dr. Tuna Balkan

Prof. Dr. Abdülkadir Erden

Dr. Buğra Koku

ABSTRACT

PERFORMANCE IMPROVEMENT OF A 3-D CONFIGURATION RECONSTRUCTION ALGORITHM FOR AN OBJECT USING A SINGLE CAMERA IMAGE

ÖZKILIÇ, Sibel

M.S., Department of Mechanical Engineering

Supervisor: Prof. Dr. Bülent E. Platin

December 2003, 92 pages

Performance improvement of a 3-D configuration reconstruction algorithm using a passive secondary target has been focused in this study. In earlier studies, a theoretical development of the 3-D configuration reconstruction algorithm was achieved and it was implemented by a computer program on a system consisting of an optical bench and a digital imaging system. The passive secondary target used was a circle with two internal spots.

In order to use this reconstruction algorithm in autonomous systems, an automatic target recognition algorithm has been developed in this study. Starting from a pre-captured and stored 8-bit gray-level image, the algorithm automatically detects the elliptical image of a circular target and determines its contour in the scene. It was shown that the algorithm can also be used for partially captured elliptical images.

Another improvement achieved in this study is the determination of internal camera parameters of the vision system.

Keywords: Camera calibration, thresholding, binarization, ellipse recognition, lens distortion, target recognition.

ÖZET

ÜÇ BOYUTLU UZAYDA CİSİMLERİN KONUMLARININ TEK KAMERA GÖRÜNTÜSÜ KULLANILARAK BELİRLENMESİ İÇİN GELİŞTİRİLMİŞ YÖNTEMİN PERFORMANS İYİLEŞTİRİLMESİ

ÖZKILIÇ, Sibel
Yüksek Lisans Tezi, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Bülent E. Platin
Aralık 2003, 92 sayfa

Bu çalışmada, cisimlerin 3 boyutlu uzaydaki konumlarını pasif ikincil bir hedef kullanılarak belirleyen bir yöntemin iyileştirilmesi üzerine odaklanılmıştır. Bu çalışmaya temel oluşturan önceki çalışmalarda, 3 boyutlu uzayda cisimlerin konumlarını belirleyen bir yöntemin kuramı geliştirilmiş ve bu yöntem bir bilgisayar programı yardımıyla, bir optik ölçme sistemi ve sayısal görüntüleme sistemi üzerinde uygulanmıştır. Kullanılan pasif ikincil hedef, içinde iki benek olan bir daireden oluşmaktadır.

Geliştirilen bu yöntemi herhangi bir dış müdahale olmaksızın çalışan otonom sistemlerde uygulayabilmek için, bir otomatik hedef belirleme yöntemi geliştirilmiştir. Bu yöntem optik sistemce çekilmiş ve depolanmış 8 bitlik bir resimde dairesel hedefin görüntüsü olan elipsi bulmakta, kenarlarını otomatik olarak belirlemektedir. Yöntemin kısmi olarak görüntülenmiş elipsler için de kullanılabildiği gösterilmiştir.

Çalışmada geliştirilen başka bir iyileştirme de görüntüleme sisteminin iç kamera parametrelerinin belirlenmesidir.

Anahtar kelimeler: Kamera kalibrasyonu, eşik belirleme, siyah-beyazlaştırma, elips tanıma, mercek bozuklukları, hedef tanıma.

ACKNOWLEDGEMENTS

I am grateful to my supervisor Prof. Dr. Bülent E. Platin, for his guidance and valuable comments during the preparation of this study and also for his efforts during the preparation of the manuscript.

I would like to express my thanks to my room mates, Özge İnce, Ezgi Ozan, Derya Gülhan for their valuable support for two years. Many thanks go to Seda Sırakaya, for her never ending moral support. I also would like to thank to Kerem Altun for his suggestions and support. I especially thank to Varlık Kılıç for his helps in order to avoid the problems on the set-up.

Finally, I would like to express my greatest thanks to my family, Süleyman, Yasemin, Mustafa Cem Özkılıç for their sincere support and reliance.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

a	The parameter vector of an ellipse when the ellipse equation is written in polynomial form.
a, b, c, d, e, f	The coefficients of the ellipse when the ellipse equation is written in polynomial form.
C(x,y)	The representation for the general conic equation
d_0	The distance between the origin of the camera coordinate system to the image plane coordinate system
d	The distance of the origin of the image plane coordinate system to the target coordinate system.
$E_i^{(average)}$	The average reconstruction error in the configuration parameters i.
f	Focal length of the camera.
F(**a**,**x**)	The general conic representation of the ellipse.
$H_{wc}^{(w,c)}$,$H_{ci}^{(c,i)}$, $H_{it}^{(I,t)}$	Homogeneous transformation matrices.
J(T)	The criterion function for threshold selection.
k_1,k_2	Radial lens distortion parameters.
l_1, l_2	The tangent lines to the ellipse.
L(x,y)	The parameter for the line connecting the pair of points on the ellipse.
M	maximum score in accumulation space.
p_1,p_2	Tangential lens distortion parameters.
q	Quality factor.
r	Radius of the circular area.
R_x( ), R_y ( ), R_z ( )	Basic rotation matrices.
T	The threshold value for binarization.

$T_x(\ )$, $T_y(\ )$, $T_z(\ )$	Basic translation matrices.
u_0^{ud}, v_0^{ud}	Undistorted image coordinates.
δu_0, δv_0	The total lens distortion.
$\delta u_j^{(r)}$, $\delta v_0^{(r)}$	The radial lens distortion.
$\delta u_j^{(t)}$, $\delta v_0^{(t)}$	The tangential lens distortion.
x_0, y_0	Image plane coordinates of the target center.
x_i, y_i	Coordinates of any point in the image plane in the image coordinate system.
X_v, Y_v, Z_v	Coordinates of an image point lying on the actual image plane expressed in the virtual image coordinate system.
$x_v.y_v$	Coordinates of the projected image point on the virtual plane coordinate system.
x_s, y_s	Coordinates of the center of the outer spot of the image of the target in the image plane coordinates.
x_s', y_s'	Coordinates of the center of the outer spot of the image of the target in the virtual image plane coordinates.
λ	A constant in ellipse equation.
ϕ	Tilt angle of the camera about x-axis in the world coordinate system.
θ	Pan angle of the camera about y-axis in the world coordinate system.
α	The rotation of the target about x-axis in the target plane.
β	The rotation of the target about y-axis in the target plane.
γ	In-plane rotation of the target about z-axis in the target plane.
μ_1, μ_2	The means of the Gaussian distribution.
σ_1, σ_2	The variances of the Gaussian distribution.

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 General

In many robotic applications that aim positioning or recognition tasks, vision is one of the most effective and flexible techniques to sense the environment. Similar to the human vision system, machine vision systems may give very rich information about the environment. Therefore, the intelligence on the environment begins with the interpretation of this visual information. Mechanisms involved and the experience in human perception of environment through vision may be used in this interpretation step.

In practical applications, there are basically two types of vision systems; namely, the “stereo vision” and the “monocular vision”. Stereo vision systems use two cameras to view an object. In this type of vision systems, the need for synchronization of two cameras brings an additional cost to the hardware. The interpretation of two images of a same scene is another challenging task to perform in their processing phase. On the other hand, unlike stereo vision, the depth information cannot directly be obtained from a single image with monocular vision. So, some a priori information about the object is required for this purpose.

With the help of the developing technology, the cost for setting these visual systems decreases and their use spreads in many application areas, like automated production lines, manipulated guidance, and autonomous vehicles. Especially, in tool selection applications, where the configuration information of an object needs not to be accurate but only the recognition of the object is essential, low-cost vision systems can be easily implemented.

However, in robotic applications like manipulator guidance, since the accuracy of configuration parameters are very crucial, the accuracy of the configuration reconstruction technique used becomes very important next to the importance of the quality of the image as well as the quality of the lens.

Vision applications, in which the aim is the positioning can be analyzed in two parts, namely “pattern recognition” and “camera calibration”. Pattern recognition is the science that concerns with the description or classification (recognition) of measurements. On the other hand, pattern recognition may be characterized as an information reduction, information mapping, or information labeling process [1].

Historically, two major approaches to pattern recognition are seen as the statistical (or decision theoretic) and the syntactic (or structural) approaches. Recently, the emerging technology of neural networks has provided a third approach, especially for “black box” implementations of pattern recognition algorithms. The structure of a typical pattern recognition system is shown in Figure 1.1.

Pattern recognition techniques are based on geometrical and/or textural features of patterns. A pattern can be as basic as a set of measurements or observations, perhaps represented in vector or matrix notation [1]. As seen in Figure 1.1, the pattern data is converted into a measured data after sensing it with a sensor or transducer. This measured data is generally preprocessed to filter for producing a more useful data free of any unnecessary details and/or noise so that it can be used in the feature extraction step. In the feature extraction step, the purpose is to reduce the data by measuring certain “features” or “properties” that distinguish one pattern from the others. These features (or, more precisely, the values of these features) are then passed to a classification algorithm that evaluates the evidence presented and makes a final decision about class of the pattern [2].

Camera calibration in the context of three-dimensional machine vision is a process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3-D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters) [3]. While external camera parameters consist of three rotational and three translational positions, the intrinsic camera parameters can be pronounced as the effective focal length, scale factor, image center, and lens distortion.

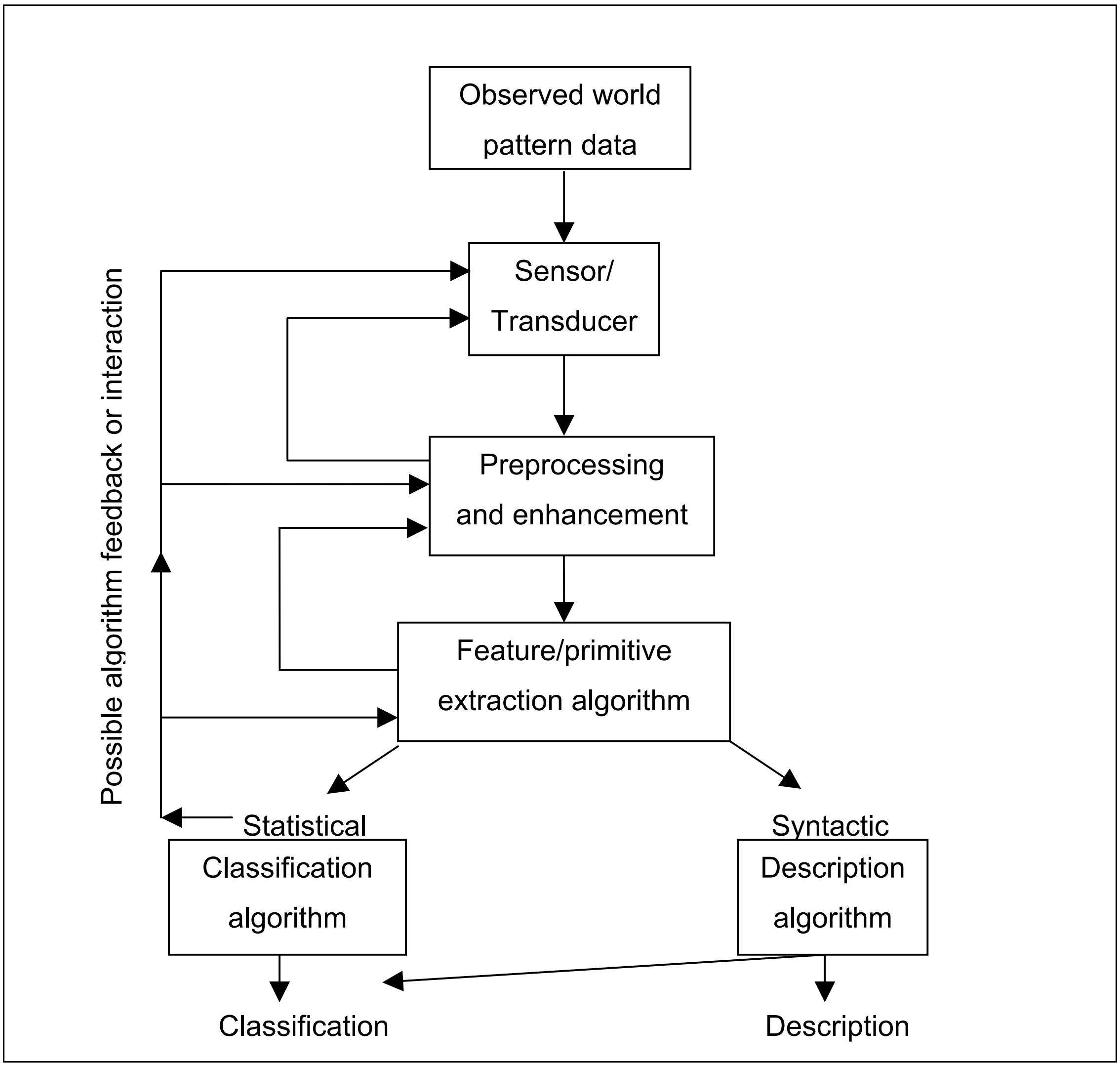


Figure 1.1 Typical pattern recognition system structures [1]

In camera calibration, the transformation from 3-D world coordinates to 2-D image coordinates is determined by solving the unknown parameters of the camera model [4]. Depending on accuracy requirements, the model is typically based on either orthographic or perspective projection. Orthographic transformation can be considered as the roughest projection approximation assuming that objects in 3-D space are orthogonally projected on the image plane. It is more suitable for vision applications where objects are positioned very far away from the camera and/or requirements of the geometric accuracy are somewhat low. Due to its linearity, it provides a simpler and computationally less expensive solution than perspective projection, which is in turn a nonlinear mapping. However, for 3-D motion estimation and reconstruction problems, the perspective projection gives an idealized mathematical framework, which is actually quite accurate to model high quality camera systems.

In order to describe the mechanism involved in perspective imaging, let us consider an ideal pinhole O at a fixed distance in front of an image plane (Figure 1.2). Assume that an enclosure is provided that only light coming through the pinhole can reach the image plane. Since light travels along straight lines, each point on the image corresponds to a particular direction defined by a ray from that point through the pinhole. Thus, the perspective projection is achieved.

In Figure 1.2, x- and y-axes of the coordinate system are parallel to the image plane. The optical axis is defined to be perpendicular from the pinhole to the image plane along z-axis. Note that the introduced Cartesian coordinate system has

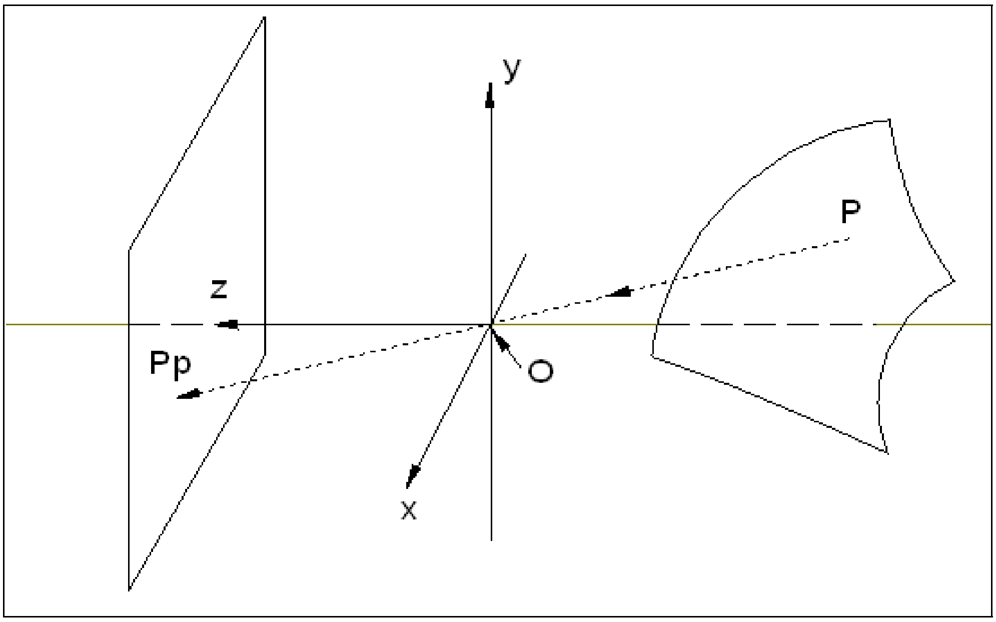


Figure 1.2 Perspective projection model

its origin at the pinhole O. The image of a point P on the object is P’ on the image plane, when there is no other obstructing object lies on the ray from P to the pinhole O. Performing this imaging principle for all points lying on the surface of the object provides the familiar perspective projection of this object on the image plane [5].

The automation degree in a given vision application is also another important point in actual applications. A true autonomous system needs to interpret the image information without any supervisor in spite of the changes in environmental conditions and/or in camera settings.

In this study, the vision application is achieved on a monocular vision set-up for the configuration determination task modeled by perspective projection. The major aim is to develop and implement an algorithm, which will automatically interpret a given image information. This algorithm will be used to improve the performance of a previously developed configuration reconstruction algorithm by Kılınç [6] and Acar [7] in the Mechanical Engineering Department of Middle East Technical University.

1.2 Objectives of the Study

This study is based on a previously developed 3-D configuration reconstruction algorithm, which determines the external camera parameters of a secondary target using its single gray-level image. In this algorithm, the target was recognized by a supervisor manually, so it was not an autonomous algorithm.

The major objective of this study is to develop an automatic target recognition task and its integration to the existing configuration reconstruction algorithm.

The improvements, which will be aimed in this study, are listed below as relative merits comparing the previous study and the present study:

Binarization:

- In the previous study [7], the threshold was taken as a constant value for all scenes and the lighting was manually adjusted accordingly to get a full ellipse in the binarized image.
- In this study, an automatic binarization algorithm is aimed that calculates the best threshold for every image without any need for a lighting adjustment.

Pattern recognition:

- In the previous study [7], the ellipse was recognized by a supervisor on the captured image, manually.
- In the present study, the development of a pattern recognition technique is aimed to detect the center of the ellipse automatically on the image without any need for supervisor assistance.

Segmentation of the ellipse area:

- In the previous study [7], a supervisor was needed to segment the ellipse area, manually.
- In this study, an automatic segmentation of the ellipse area is aimed.

Software platform:

- In the previous study [7], a software developed in DOS environment, and it is not a user-friendly program.
- In this study, a user-friendly GUI, in which all the image processing and camera calibration tasks achieved in this study and in the previous study are integrated, is aimed. In the aimed software, there exist both single tasks used in the image processing steps and complete target recognition task. Also an option for the continuous processing of automatic 3-D configuration reconstruction of the object exits in the interface.

Calibrating the internal parameters of the camera:

- In the previous study [7], only the image center of the camera was determined beforehand.
- In this study, a method for the determination of the lens distortion parameters and its implementation by a software program are aimed.

1.3 Proposed Method for Autonomous Reconstruction of 3-D Position of an Object

In reaching the aims listed above, the following sets of consecutive tasks are proposed to form a resulting autonomous system. The process will begin with capturing a gray scale image of a scene that contains a circular secondary target. Then, a threshold will be determined automatically to binarize this image. After this binarization, an edge detection will be performed followed by an edge thinning. Then, an ellipse fitting procedure will be performed on the whole image. All pixels of the scene will be labeled according to their probability of being an ellipse center as a result of this ellipse fitting algorithm. A contour detection algorithm will be performed around the center, which has the highest possibility of being the center of an ellipse. After the segmentation of the target area containing the contour of the ellipse, the contour information on the ellipse and its spots will be generated for use in the reconstruction algorithm. In case of a partial capture of an ellipse, the object recognition should still be possible up to some degree of partition.

The advantages of the proposed 3-D configuration reconstruction algorithm can be summarized as follows:

- Monocular vision
- Simple non-iterative solution in external calibration
- Determination of all 6 configuration parameters
- Uniqueness of the solution
- Solution for the recognition of the partial images
- Low importance on lighting conditions
- No need for any structural lighting
- No point to point correspondence or knowledge of the world coordinate points of the target
- No perfect geometry for the target

1.4 Outline of the Study

The background of the 3-D configuration reconstruction algorithm is presented in Chapter 2. In addition to the method used, the experimental set-up and results of experiments are also briefly explained.

In Chapter 3, some improvements developed to make the algorithm autonomous are given. Automatic thresholding and ellipse detection algorithms are explained and their performances are discussed. The ellipse detection algorithm is performed for some artificial scenes that also contain some regular geometric shapes other than ellipse and on real images grabbed with a camera under different lighting conditions. Results obtained for scenes with partially captured ellipses are also given.

In Chapter 4, results of the automatic object recognition algorithm are presented.

In Chapter 5, definitions of the internal camera parameters studied in this thesis are explained and a method for the detection of lens distortion parameters is presented.

Chapter 6 presents the summary and conclusion for the work done in this study and ends up with some recommendations for future work.

Appendix A contains a brief summary of some basic image processing techniques used in this study.

In Appendix B, the aim and the structure of the computer program developed in this study are explained. Also, a user manual for the computer program is presented.

CHAPTER 2

PREVIOUS WORK

2.1 Overview

There are various studies on the external camera calibration. Chatterjee and Roychowdhury [8], proposed a coplanar calibration method based on nonlinear optimization method, which also computed the lens distortion parameters. They also compared the result of their method with results of three other camera calibration studies. Han and Rhee [9] used a circular pattern containing two internal spots, one of which was located on the center of the circular pattern and aligned with the optical axis. They determined three rotation parameters, the image plane distance, and the distance between the origin of the camera, and the origin of the world coordinate system. Tsai [3] used the radial alignment constraint to reduce the dimensionality of the unknown parameter space. In this study, the tangential distortion was ignored and radial distortion parameters, effective focal length, scale factor, and all six external camera parameters are determined. Zhang [10] proposed a method for the determination of external camera parameters and radial lens distortion. The proposed procedure consisted of a closed form solution, followed by a nonlinear refinement based on maximum likelihood criterion. Ahn, Rauh and Kim [11] used circular coded targets for the automation of an optical 3-D measurement system in which they used stereo vision and multiple imaging. Heikkila [4] suggested a calibration procedure that utilizes circular control points and performs mapping from world coordinates into image coordinates and backward from image coordinates to 3-D plane coordinates. The camera model, which was used in this study, allowed least-squares optimization with the distorted image coordinates. Özdemir [12] and Çetin [13] studied locating a mobile robot in a structural environment by using a single camera. In these studies, the straight-line correspondences were used.

Olgac, Gan and Platin [14] used a circular secondary target and orthographic projection model but also made the assumption that the target center lies on the optical axis of the camera. However, the use of a full circular target made the determination of the in-plane rotation inherently impossible. Platin et al. [15] and Olgac et al.[16] applied the same method by replacing the orthographic projection model with a perspective projection model. The major disadvantages of these algorithms were the constraints on the target configuration and the requirements of a priori knowledge of some configuration variables. This method was further improved by Kılınç [6] by adding two internal spots to the circular secondary target such that all six configuration parameters can be reconstructed. The method developed by Kılınç [6] was implemented on an experimental set-up and tested by Acar [7] and satisfactory results for the reconstruction of all 6 external parameters were obtained.

Since this study aims some improvements on the implementation of the existing 3-D configuration reconstruction algorithm [6,7], this algorithm will be explained in this Chapter and the experimental set-up used to implement this method will also be presented.

As far as the configuration reconstruction algorithm is concern, only its general solution and some results of its implementation will be summarized here. Their details can be found in [6] and [7]. Steps of the computer algorithm in the previous study beginning from the image capture to the determination of the 3-D configuration parameters will also be given and some results obtained from the experimental investigations will be presented.

The previous work that this study is based on consisted of two main parts:

a) Development of an algorithm to theoretically determine the configuration of an object in the workspace with respect to a base plate [6].
b) Implementation of this algorithm via a computer program and investigation of its performance on an experimental vision set-up [7].

In monocular vision, in order to determine the distance between a camera and a rigid body, some a priori information about the body is required. So, a planar secondary target was assumed to be mounted on the rigid body. The plane on which the secondary target is mounted were called as the “target plane”. Determining the configuration of the secondary target with respect to a world coordinate system would also mean determining the configuration of the rigid body with respect to the same coordinate system.

Kılınç [6] and Acar [7] used a circular planar target of radius R with two internal spots as a passive secondary target (Figure 2.1). One of these spots was located at the center of the circle and this spot was used to determine the location of true center of the circle in the image plane. The other spot was placed (at a known distance) r_f away from the center. Both spots were identical circles with radii of r_0. While the boundary of the elliptical image of the projected circular target was used to determine the two rotation parameters and the depth information, the image of the center spot was used for two translation parameters, and the image of the second spot was used for the remaining rotation parameter.

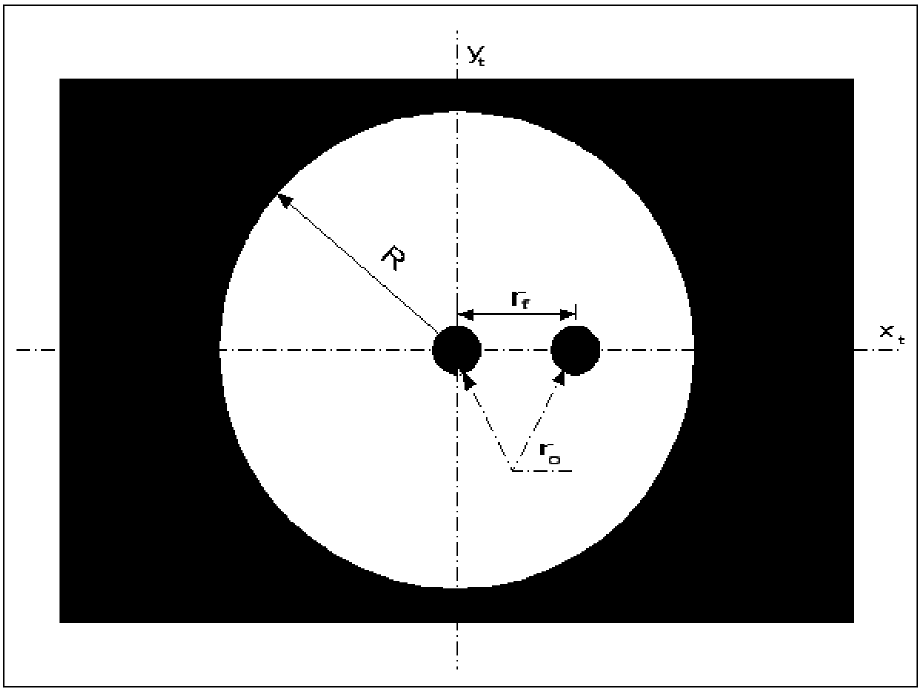


Figure 2.1 Circular secondary target

2.2 3-D Configuration Reconstruction Algorithm

The reconstruction parameters can be seen in Figure 2.2. World coordinates of any point P lying on the target plane can be expressed in terms of augmented matrices by the following vector equation:

$$R_{wp}^{(w)} = H_{wt}^{(w,t)} R_{tp}^{(t)} = H_{wc}^{(w,c)} H_{ci}^{(c,i)} H_{it}^{(I,t)} R_{tp}^{(t)} \tag{2.1}$$

where

$R_{wp}^{(w)}$: The augmented matrix of the vector from the origin of the world coordinate system to the point P in the world coordinates.

$R_{tp}^{(t)}$: The augmented matrix of the vector from the origin of the target coordinate system to the point P in the target plane coordinates.

$H_{wc}^{(w,c)}$: The configuration of the camera with respect to the world coordinate system.

$H_{ci}^{(c,i)}$: The configuration of the image plane with respect to the camera coordinate system.

$H_{it}^{(I,t)}$: The configuration of the target plane with respect to the image plane

Among the transformation matrices used in Equation (2.1), $H_{wc}^{(w,c)}$ is determined by a camera calibration and $H_{ci}^{(c,i)}$ is known for a given lens setting. Therefore, the solution can be obtained if the remaining $H_{it}^{(I,t)}$ can be determined. This transformation matrix can be decomposed to a series of basic rotation and translation matrices as follows:

$$H_{it}^{(I,t)} = T_z(d_o)\, R_x(-\Phi)\, R_y(-\theta)\, T_z(d)\, R_x(\alpha)\, R_y(\beta)\, R_z(\gamma) \tag{2.2}$$

where

$T_z(d_o)$: Basic translation matrix for a translation d_o along z_i-axis in the image plane coordinate system

$R_x(-\Phi)$: Basic rotation matrix for a rotation $-\Phi$ about x_i-axis in the image plane coordinate system

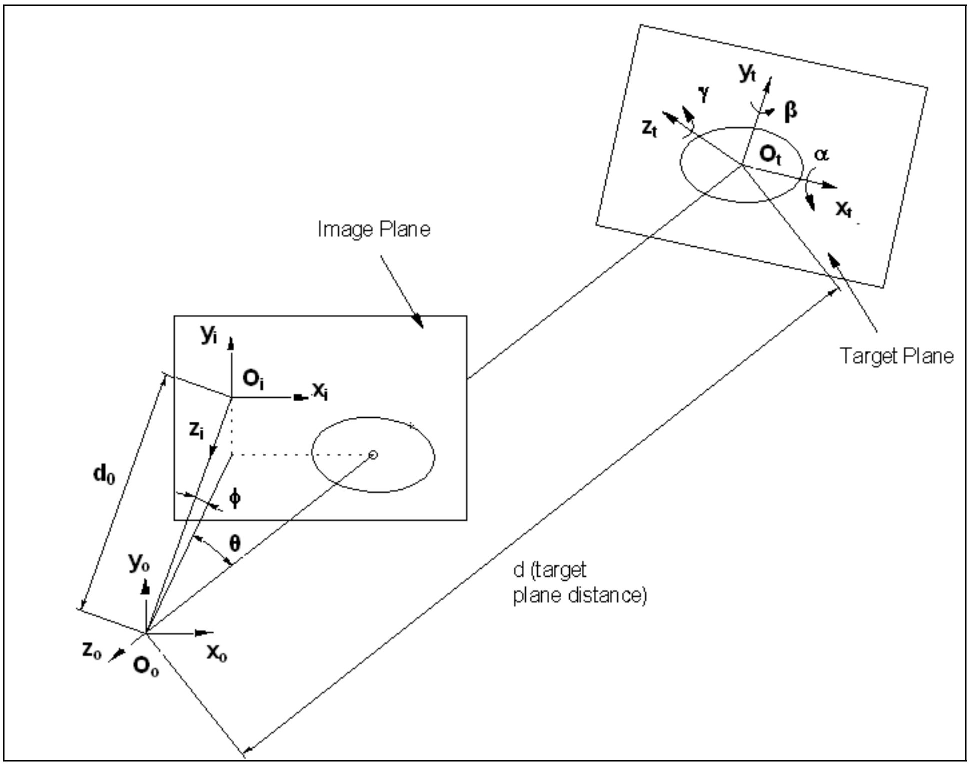


Figure 2.2 3-D configuration reconstruction parameters

$R_y(-\theta)$: Basic rotation matrix for a rotation $-\theta$ about y_i-axis in the image plane coordinate system

$R_x(\alpha)$: Basic rotation matrix for a rotation α about x_t-axis in the target plane coordinate system

$R_y(\beta)$: Basic rotation matrix for a rotation β about y_t-axis in the target plane coordinate system

$R_z(\gamma)$: Basic rotation matrix for a rotation γ about z_t-axis in the target plane coordinate system

$T_z(d)$: Basic translation matrix for a translation d along z_t-axis in the target plane coordinate system

The purpose of the reconstruction algorithm is to compute values of α, β, γ, d, Φ, θ (Figure 2.2) for a given elliptical image in the image plane of the camera. Therefore, reconstruction parameters (Figure 2.2) can be defined as follows:

Φ : Tilt angle of the camera (from which the vertical offset is calculated)

θ : Pan angle of the camera (from which the horizontal offset is calculated)

d : Distance of the image plane to the target plane along the optical axis

α : Tilt angle of the target (rotation about horizontal x_t -axis)

β : Yaw angle of the target (rotation about y_t-axis)

γ : In-plane rotation of the target about z_t-axis

All details of the general solution are given in [6,7]. Here, a general algorithm [7] will be presented as if it can be implemented directly.

- First, ϕ and θ angles are determined from the centroid of the center spot.
- Then d, α, and β parameters are determined from the contour information for the image ellipse.
- Finally, the in-plane rotation γ of the target is determined from the centroid of the outer spot.

The image of the central spot of the target represents the image of the true center of the circle on the image plane. Its centroid will be the image of the origin of the image coordinate system in the target centered solution. If one can determine ϕ and θ, then the direction in which the target center is shifted can be found.

The image coordinates of the true center of the target are given as [6]:

$$x_0 = d_0 \tan\theta / \cos\phi \quad (2.3a)$$

$$y_0 = -d_0 \tan\phi \quad (2.3b)$$

Since these coordinates (x_0,y_0) are known for a given image, ϕ and θ can be found as:

$$\phi = \tan^{-1}(-y_0 / d_0) \quad (2.4a)$$

$$\theta = \tan^{-1}(-x_0 \cos\phi / d_0) \quad (2.4b)$$

Once ϕ and θ are determined, then the direction in which the target center is shifted can be found. It is possible to define a virtual image plane of a fictitious camera whose optical axis (z_{v1}-axis) goes through the target center (Figure 2.3). A coordinate system is defined in this virtual image plane and is denoted by ($O_{v1}x_{v1}y_{v1}z_{v1}$). If one finds the coordinates of the image points in the virtual image plane, then it will be possible to apply the target centered solution to find d, α, β with respect to the virtual plane.

Since ϕ and θ are known, it is possible to write the transformation matrix $H_{vi}^{(v,i)}$ from the virtual image plane to the actual image plane as follows

$$H_{vi}^{(v,i)} = T_z(d_0)\, R_y(\phi)\, R_x(\theta)\, T_z(-d_0) \tag{2.5}$$

Then the coordinates of any point lying on the image plane can be expressed in the virtual image plane coordinate system by the following homogeneous transformation as

$$\begin{bmatrix} X_v \\ Y_v \\ Z_v \\ 1 \end{bmatrix} = H_{vi}^{(v,i)} \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix} \tag{2.6}$$

where (X_v, Y_v, Z_v) are the coordinates of the image point lying on the actual image plane expressed in the virtual image coordinate system whereas x_i and y_i are the coordinates of this image point in the actual image coordinate system. On the other hand, the projection of a point in the actual image plane on the virtual image plane can be achieved using the following perspective transformation:

$$\begin{bmatrix} x_v \\ y_v \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} (X_v d_o)/(d_o - Z_v) \\ (Y_v d_o)/(d_o - Z_v) \\ 0 \\ 1 \end{bmatrix} \tag{2.9}$$

where

x_v : The horizontal coordinate of the projected point on the virtual plane coordinate system.

y_v : The vertical coordinate of the projected point on the virtual plane coordinate system.

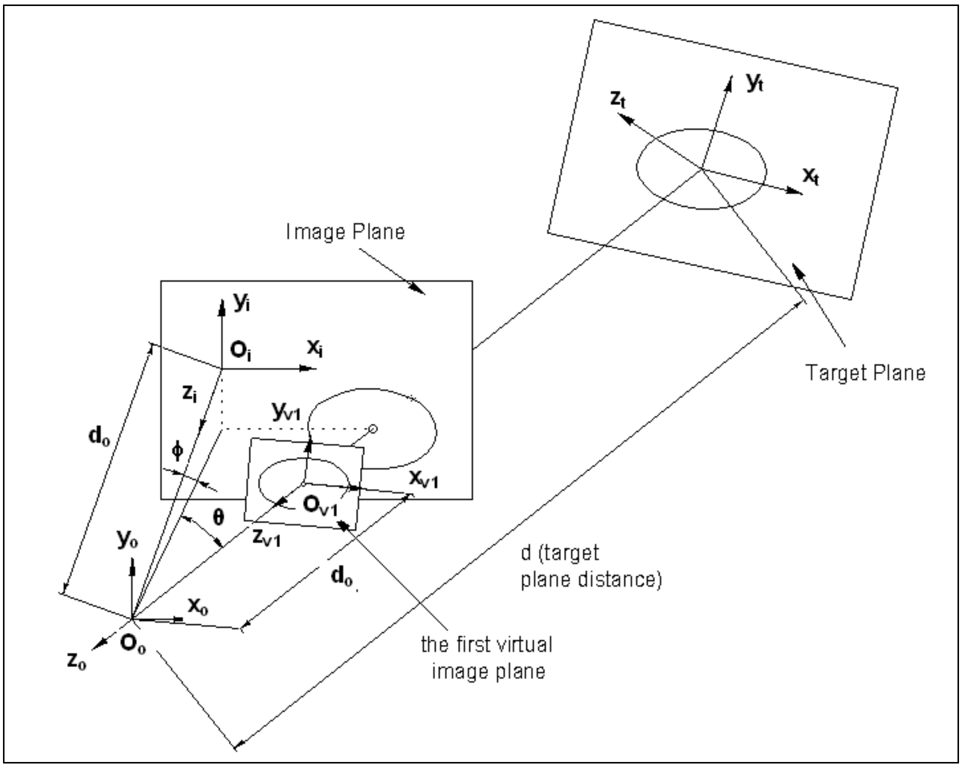


Figure 2.3 The virtual image plane

Finally, the virtual image plane coordinates of the projection of an image point projected onto the virtual image plane are obtained as:

$$x_v = \frac{d_o(x_i \cos\theta + y_i \sin\theta \sin\phi - d_o \sin\theta \cos\phi)}{x_i \sin\theta - y_i \sin\phi \cos\theta + d_o \cos\theta \sin\phi} \tag{2.7a}$$

$$y_v = \frac{d_o(d_o \sin\phi + y_i \cos\phi)}{x_i \sin\theta - y_i \sin\phi \cos\theta + d_o \cos\theta \sin\phi} \tag{2.7b}$$

After finding (x_v, y_v), the solution for d, α, and β are available for a target centered case [6].

In order to get in plane rotation of the target, a new virtual image plane is defined such that it is parallel to the target plane and its origin lying on the line O_oO_t (Figure 2.4). Also to ease the solution procedure, one more constraint to the location of the virtual image plane is added. It is positioned such that the center of the image spot lies on the intersection line between the actual image plane and the virtual image plane.

Once the location of virtual image plane is determined, it is possible to write down the transformation matrix from this second virtual image plane to the actual image plane as:

$$H_{vi}^{(v,i)} = R_y(-\beta)\, R_x(-\alpha)\, T_z(d_x)\, R_y(\theta)\, R_x(\phi)\, T_z(-d_o) \tag{2.8}$$

where the parameter d_x is used to locate the center of the image spot to lie on the intersection of the second virtual image plane and the actual image plane. The virtual image plane will be parallel to the target plane and as a consequence z_t will be also parallel to z_{v2}.

When the basic transformation matrices are substituted, elements of this transformation matrix can be obtained as:

$$H_{vi}^{(v,i)}[1,1] = \cos\beta\cos\theta + \sin\beta\cos\alpha\sin\theta \tag{2.9a}$$

$$H_{vi}^{(v,i)}[1,2] = \sin\beta\sin\alpha\cos\phi + \cos\beta\sin\theta\sin\phi - \sin\beta\cos\alpha\cos\theta\sin\phi \tag{2.9b}$$

$$H_{vi}^{(v,i)}[1,3] = -\sin\beta\sin\alpha\sin\phi + \cos\beta\sin\theta\cos\phi - \sin\beta\cos\alpha\cos\theta\cos\phi \tag{2.9c}$$

$$\begin{aligned} H_{vi}^{(v,i)}[1,4] = {} & d_o\sin\beta\sin\alpha\sin\phi - d_o\cos\beta\sin\theta\cos\phi \\ & + d_o\sin\beta\cos\alpha\cos\theta\cos\phi - d_x\sin\beta\cos\alpha \end{aligned} \tag{2.9d}$$

$$H_{vi}^{(v,i)}[2,1] = -\sin\alpha\sin\theta \tag{2.9e}$$

$$H_{vi}^{(v,i)}[2,2] = \cos\alpha\cos\phi + \sin\alpha\cos\theta\sin\phi \tag{2.9f}$$

$$H_{vi}^{(v,i)}[2,3] = -\cos\alpha\sin\phi + \sin\alpha\cos\theta\cos\phi \tag{2.9g}$$

$$H_{vi}^{(v,i)}[2,4] = d_o\cos\alpha\sin\phi - d_o\sin\alpha\cos\theta\cos\phi + d_x\sin\alpha \tag{2.9h}$$

$$H_{vi}^{(v,i)}[3,1] = \sin\beta\cos\theta - \cos\beta\cos\alpha\sin\theta \tag{2.9i}$$

$H_{vi}^{(v,i)}$ [3,2] = - cosβ sinα cosϕ + sinβ sinθ sinϕ + cosβ cosθ cosα sinϕ (2.9j)

$H_{vi}^{(v,i)}$ [3,3] = cosβ sinα sinϕ + sinβ sinθ cosϕ + cosβ cosθ cosα cosϕ (2.9k)

$H_{vi}^{(v,i)}$ [3,4] = - d_o sinα cosβ sinϕ - d_o sinθ sinβ cosϕ

- d_o cosβ cosθ cosα cosϕ + d_x cosα cosβ (2.9l)

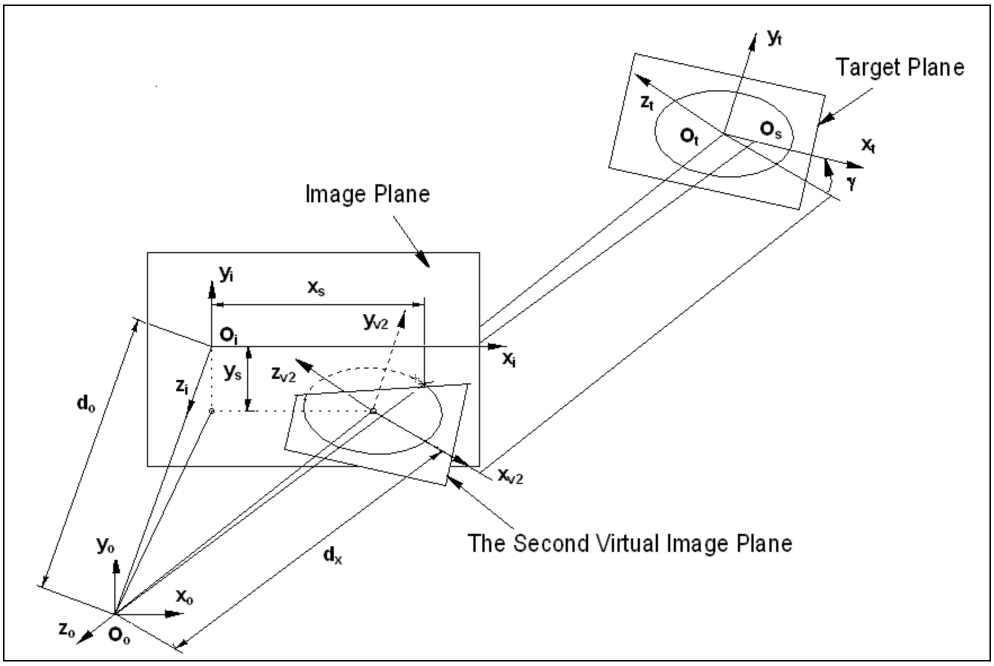


Figure 2.4 The virtual image plane for the calculation of in-plane rotation

$H_{vi}^{(v,i)}$ [4,1] = 0 (2.9m)

$H_{vi}^{(v,i)}$ [4,2] = 0 (2.9n)

$H_{vi}^{(v,i)}$ [4,3] = 0 (2.9o)

$H_{vi}^{(v,i)}$ [4,4] = 1 (2.9p)

If the matrix $H_{vi}^{(v,i)}$ is represented by its elements h_{ij} where i denotes the row and j denotes the column as

$$\begin{bmatrix} X_v \\ Y_v \\ Z_v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix} \quad (2.10)$$

then, for the image center (x_s, y_s) of the outer spot lying on the second virtual image plane, the key point is to impose $Z_v = 0$ explicitly and to solve for d_x using $x_i = x_s$ and $y_i = y_s$. This is equivalent to write:

$$\begin{bmatrix} h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 0 \\ 1 \end{bmatrix} = 0 \tag{2.11}$$

Observing the elements h_{31}, h_{32} and h_{34}, it is seen that the only unknown is d_x, since x_s and y_s can be determined using the centroid of the image of the spot. Therefore, d_x can be calculated as:

$$d_x = \frac{-(h_{31})x_s - (h_{32})y_s + (h_{34})}{\cos\beta\cos\alpha} \tag{2.12}$$

Now, the virtual image coordinates of the center of the outer spot which are going to be denoted by x_s' and y_s' can be determined by substituting d_x in the expressions of h_{14} and h_{24};

$$x_s' = (h_{11})\, x_s + (h_{12})\, y_s + (h_{14}) \tag{2.13a}$$

$$y_s' = (h_{21})\, x_s + (h_{22})\, y_s + (h_{24}) \tag{2.13b}$$

As the virtual image plane and the target plane are parallel, γ can be calculated using a double argument arctangent function as follows:

$$\gamma = \tan_2^{-1}\left(\frac{y_s'}{x_s'}\right) \tag{2.14}$$

One important property of this method is that it does not recall the off-set distance of the outer spot denoted by r_f in Figure 2.2. Thus the in-plane rotation can be reconstructed just with the knowledge that the spot is located on the x_t axis.

2.3 Experimental Setup

The experimental set-up consists of two parts; namely, a vision system and a positioning system on which the target and the camera are mounted. The vision set-up composed of a Charge Injected Device (CID) camera and a frame grabber.

The frame grabber used to grab images is a Data Translation DT3851 Series Board. The board is built around the onboard graphics processor, a Texas Instruments TMS34020 [17]. The graphics processor controls video input, video display, and DT-connect transfers, and performs a limited amount of image processing. The frame grabber has 640Hx480V non-interlaced 8-bit graphic format for single monitor operation. The image is grabbed in Windows environment and stored in a file.

The Charge Injected Device (CID) structure has a fundamentally different principle of operation, and readout technique from Charge Coupled Device (CCD`s), providing useful performance advantages. Each pixel in the CID array is individually addressed during readout. Scanning routines are implemented via electronic switching of row and column electrodes, which are fabricated in a thin, clear polysilicon matrix over the surface of the array. While CCD`s transfer collected charge out of the pixels during readout (hence erasing the image stored on the sensor), CID`s do not transfer charge from site to site in the array. Instead, readout is accomplished by transferring or “shifting” the collected “charge packet” within an individually addressed pixel, and sensing displacement values across the electrodes at the site. CID Technologies Inc.’s CID2250D model used as the CID camera in the previous study and this study. It has a 512Hx512V CID35 solid state sensor with a continuous zoom.

There are two lenses used with this camera:

- Cosmicar CCVT lens having a focal length of 25 mm maximum aperture ratio 1:1.4 and 1 inch image size.

- RS lens having a focal length of 16 mm, maximum aperture ratio 1:1.4, and 2/3 inch image size.

Figure 2.5 Positioning unit

The positioning unit (Figure 2.5) has two stations; first station is basically a camera-positioning unit, and second station is a target-positioning unit [7]. The camera-positioning unit has translational freedoms in x-, y-, and z-axes, and a rotational freedom around y-axis. In each of the translational axes, the travel distance is 50 mm with a resolution of 0.01 mm. The rotary table has 360° continuous travels with a resolution of 0.002°. On the other hand, the target-positioning unit has two translational freedoms in x-axis and z-axis. It has a travel distance 50 mm with 0.01 mm resolution in x-axis. The maximum travel distance in z-axis is 300 mm and it has a resolution of 0.01 mm. It has also the rotational freedoms in two axes; namely about x-axis and y-axis. The x-axis has a continuous rotation of 360° with 0.002° resolutions. The y-axis has 360° continuous travel with a resolution of 6 arc minutes.

2.4 Object Recognition

In the previous study of Acar [7], first the image of the target was grabbed and stored in a file. Then a computer program that works under DOS environment and compiled with Borland C++ 3.1 was executed. In the beginning, the name and path of the image file was written, afterwards the image was seen on the screen and it was binarized with a constant threshold value of 128. Secondly, the ellipse area was segmented by a supervisor with the cursor. Starting from the near edge of the rectangle outside the ellipse region, the algorithm moved through the edge pixels. After finding one edge pixel, the contour was followed, until the starting point was reached. The contour pixels that would be used in the reconstruction algorithm were determined by the MaxMinCross type pixel selection method [7]. The same contour following algorithm was also used to determine the contour of the spots whose centers were determined by using the moments. The information obtained was used in the 3-D configuration reconstruction algorithm and all six external parameters were supplied as the output.

2.5 Results of the 3-D Configuration Reconstruction Algorithm

In the experiments performed by Acar [7], it was possible to measure the first two rotations R_x and R_y using the optical bench directly, but not the third rotation R_z, it was taken to be a constant value and compared with the reconstructed value. Three translations T_x, T_y, T_z could be measured directly from the set-up. When rotation around R_x or combined rotations around R_x and R_y were present, it was not possible to measure T_x and T_y directly. Hence these two axes were analyzed separately.

Different data sets mainly depend on minimum and maximum target distances, and reconstruction parameters were determined. It was reported that the method works properly within the following limits:

$10° < | R_x | < 80°$

$10° < | R_y | < 80°$

$500 \text{ mm} < T_z < 1300 \text{ mm}$

Results of the 3-D configuration reconstruction obtained in [7] can be summarized as:

- T_x and T_y are constructed successfully in the whole range of rotations, because they are constructed using the central spot.
- Reconstruction errors for T_z are low in the limits of rotations, given above. However, as the target moves away from the camera, this error increases.
- R_x and R_y are determined successfully in the limits. Reconstruction errors increase when the rotations get closer to the ultimate limits.
- R_z is constructed successfully in the whole range of rotations, because it is constructed using the location of the outer spot.

The average reconstruction errors were found as follows:

$E_{Tx}^{(average)} \approx 0.5$ mm $\qquad E_{Rx}^{(average)} \approx 0.4°$

$E_{Ty}^{(average)} \approx 0.5$ mm $\qquad E_{Ry}^{(average)} \approx 0.4°$

$E_{Tz}^{(average)} \approx 1.5$ mm $\qquad E_{Rz}^{(average)} \approx 0.5°$

CHAPTER 3

IMAGE PROCESSING AND ANALYSIS

3.1 General

As it is explained in the previous Chapter, the 3-D configuration reconstruction algorithm developed in previous studies [6,7] utilizes the contour data of the elliptical image of a circular target and images of two inner spots. In order to get the contour information from an image, some image processing methods should be applied. In this study, first a binarization is performed on the image using a threshold value calculated for each image automatically. Three versions of this binarization algorithm are developed and compared for different lighting conditions and lens settings. The second and third steps of the image processing are an edge detection with a Sobel operator and an edge thinning. Next steps are the detection of the ellipse center and segmentation of the ellipse area. The final step of the image processing is the determination of the ellipse contour with a contour following algorithm.

3.2 Image Binarization

Binarization is a technique performed to convert a gray level image to a binary image. It is usually used to distinguish an object from its background. Picture cells in which the gray-level is above a threshold give rise to ones in the corresponding position of the binary image, and those below it give rise to zeros (or vice versa) [5]. In order to make the segmentation robust to variations in a scene,,the algorithm used should be able to select an appropriate threshold value automatically using samples of the image intensity present in the image. An automatic thresholding must analyze the gray-level values and use the knowledge about the application and its environment to select the most appropriate threshold value [18].

In a digitized image, a gray-level histogram giving the number of picture cells having a particular gray-level can be created. An example histogram is given in Figure 3.1. For the application in this study, the only a priori knowledge about the scene is a white circle on a black background, so it is decided to use a histogram modeling method.

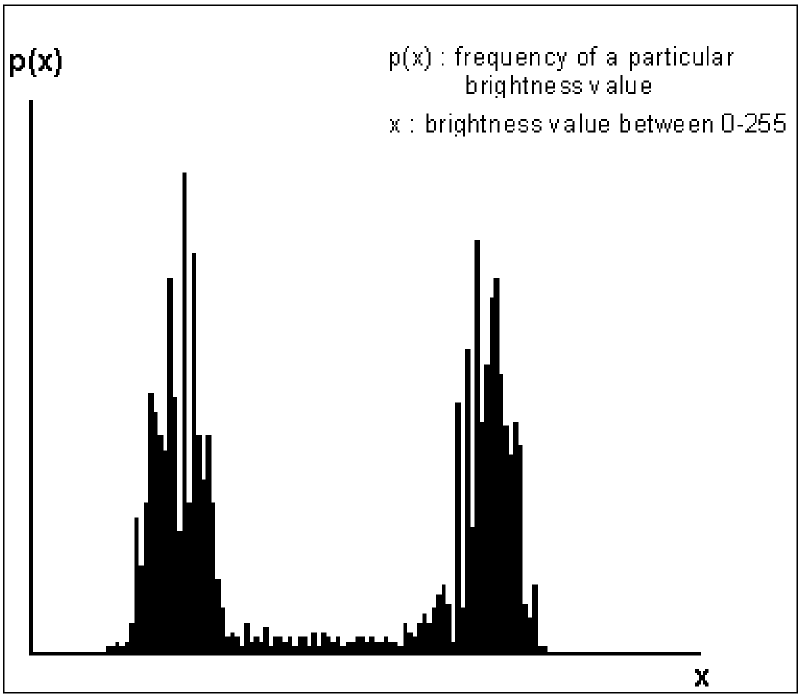


Figure 3.1 An example histogram

There exist many studies concerning the thresholding problem [19-22]. In most of these studies, the binarization is considered as a classification problem and a criterion function is used for the decision making. The threshold, which is the value that minimizes or maximizes this criterion function, depends strongly on the method used. One of the oldest studies on this topic is “Otsu method” [19], which selects a threshold that minimizes the resulted weighted sum of the within-group variances for two classes of pixels. Another approach is the “maximum entropy sum” proposed by

Kapur [20], which is based on the information measure between two classes. The method developed by Sahoo, Wilkins and Yeager [20] includes both Kapur's maximum entropy sum method and the entropic correlation method of Cheng and Don [21].

One of the most effective studies on this topic is Kittler and Illingworth's "Minimum Error Thresholding" method [22]. According to this method, if gray-level distributions of pixels for an object and its background are known or can be estimated, then an optimal, minimum error threshold can be obtained using results of the statistical decision theory.

3.2.1 Minimum Error Thresholding Method

In this method, the histogram is modeled as a mixture of two Gaussian distributions having respective means and variances (μ_1,σ_1^2) & (μ_2,σ_2^2) and respective proportions P_1 and P_2. The principal idea behind this method is to minimize the following criterion function J(T) related to the average pixel classification error rate.

$$J(T) = 1 + 2[\, P_1(T)\log\sigma_1(T)+P_2(T)\log\sigma_2(T)] - 2[\, P_1(T)\log P_1(T)+P_2(T)\log P_2(T)] \qquad (3.1)$$

where T is an trial threshold value, $P_1(T)$ and $P_2(T)$ are cumulative brightness of two classes of pixels. Trying various brightness values between 0 and 255, the best threshold value is selected as the one that gives a minimum for the criterion function.

This technique can be applied to a whole image and a single threshold value can be found automatically. However, under uneven lighting conditions, spatially varying thresholding can be used as well, in which the image is divided into small regions, and for each region, a local threshold value may be calculated.

In this study, first a global thresholding (GT) is performed. However, for some non-uniformly lighted images, the GT is proven to be not too satisfactory.

Therefore, two separate local thresholding (LT) techniques are also developed using the minimum error thresholding method. In these LT techniques, a 640Hx480V pixels image is divided into 300 local sub-images of 32x32 pixel size each and then LT values for each of these sub-areas are found. However, in the LT method, there appears to be a problem of discontinuity between the divided regions, once these regions are patched together. In order to avoid this type of discontinuities, local threshold values are smoothened with a 2x2 mask of [1 1; 1 1] within the neighboring regions and then 64x64 pixel sub-regions are binarized with this averaged threshold value (LT1).

In the second LT method (LT2), local regions are selected such that half of the new region is overlapped with its neighbors (regions on the left and above). By this application, the regions are not strictly separated from each other and the initial trial brightness values are selected using those threshold values calculated for neighboring regions. As a result, a total of 1200 threshold values are obtained for a 640Hx480V pixels image. Again, these values are averaged with a mask of [1 1; 1 1]. Results of these three methods are shown with some sample images captured in different lighting and set-up conditions in Figures 3.2-3.7.

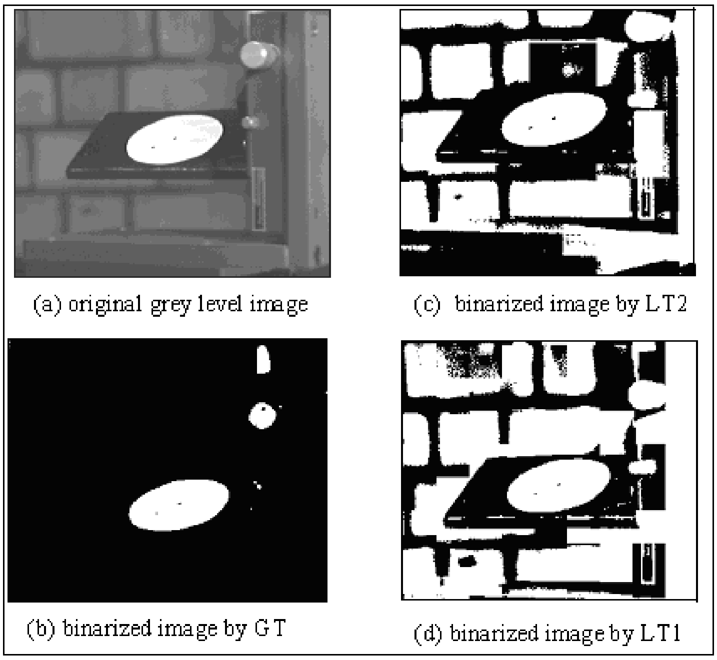


Figure 3.2 Image captured with a 25 mm lens in controlled lighting conditions

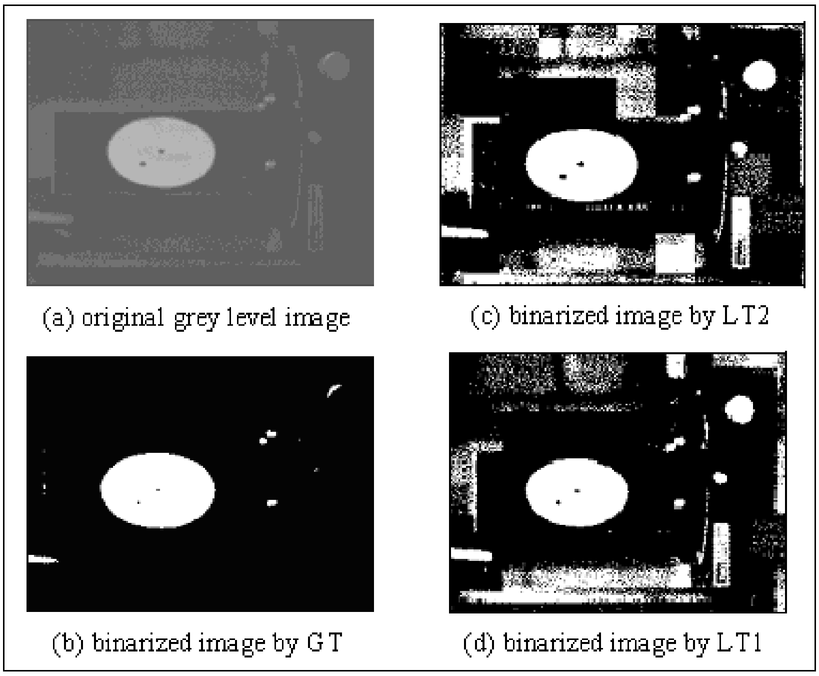


Figure 3.3 Image captured with a 25 mm lens in normal lighting conditions of the room

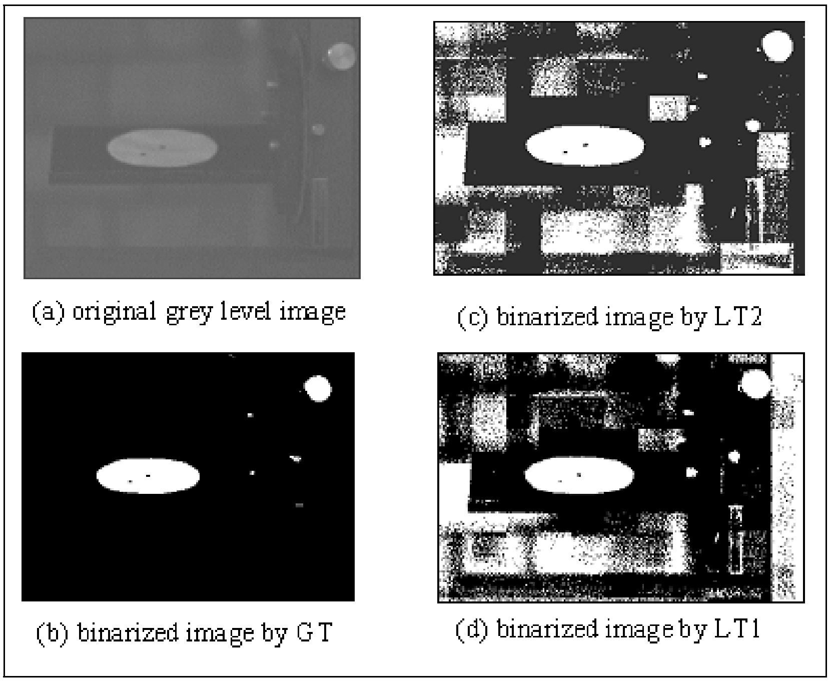
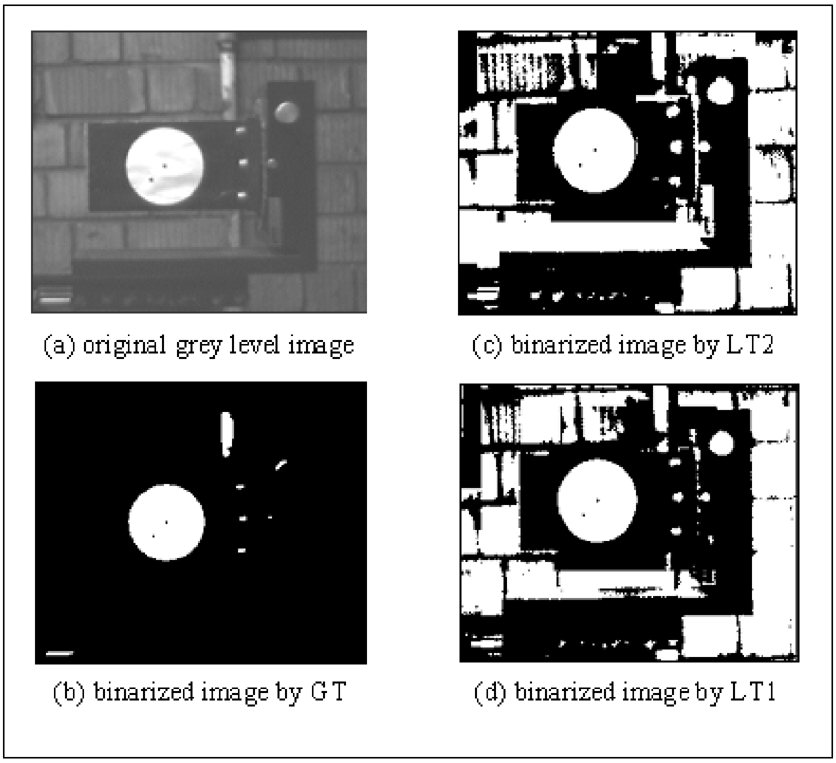
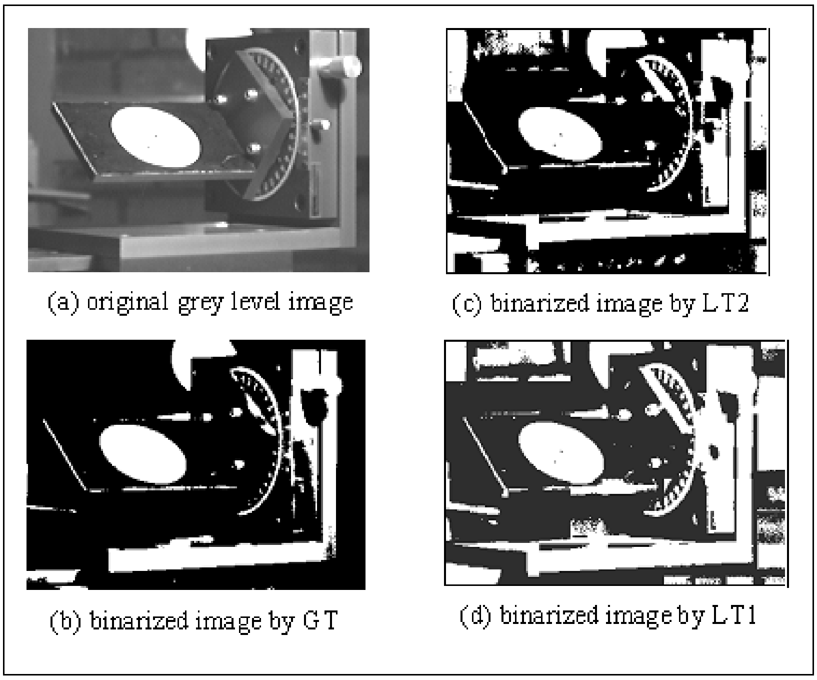


Figure 3.4 Image captured with a 25 mm lens in normal lighting conditions of the room rotated 70° in x-axis.



(a) original grey level image
(c) binarized image by LT2
(b) binarized image by GT
(d) binarized image by LT1

Figure 3.5 Image captured with a 16 mm lens in controlled lighting conditions



(a) original grey level image
(c) binarized image by LT2
(b) binarized image by GT
(d) binarized image by LT1

Figure 3.6 Image captured with a 25 mm lens with extra lighting on the target

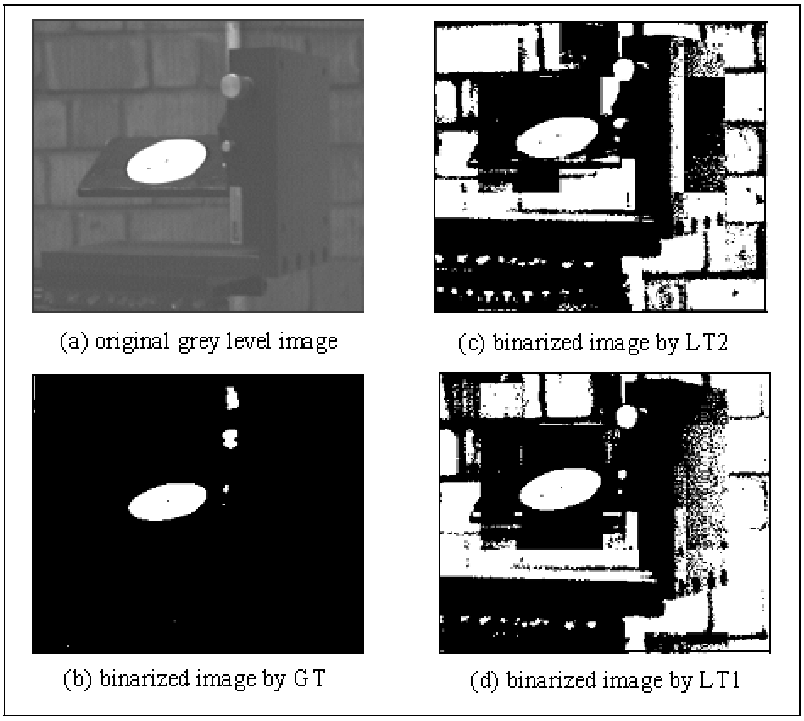


Figure 3.7 Image captured with a 16 mm lens in normal lighting conditions without any control

3.2.2 Comparison of Binarization Algorithms

By a simple eye inspection, only some properties of the target (spots and boundaries) can be checked. In order to compare performances of various methods, the homogeneity of distributions is checked. So, the addition of the variance of each pixel from the mean of the class is compared for the minimum value. It is assumed that smaller the addition of variances is, better the object-background separation gets. This criterion is only performed on a small region containing the target, not on the whole image. Group variances of the binarized images (Figure 3.2-3.7) according to binarization method are given in Table 3.1.

Table 3.1 Group variances of the binarized image according to binarization method

Figure	Variance ($x10^7$)		
	GT	LT1	LT2
3.2	1.71	2.58	2.55
3.3	0.09	4.72	2.55
3.4	0.12	3.03	1.21
3.5	1.20	3.76	2.22
3.6	3.79	1.85	1.70
3.7	0.67	2.29	1.70

As seen in Table 3.1, the smallest group variances are obtained in Figures 3.2, 3.3, 3.4, 3.5 and 3.7, all with global thresholding methods. However, in Figure 3.6, in which the image captured under an extra lighting condition on top of the target, the global thresholding shows the worse performance according to the performance criteria. Also, in the resulting globally thresholded image (Figure 3.6b), internal spots of the target cannot be recognized. The LT technique works better for this kind of lighted images.

Another important result for global thresholding is given in Figure 3.7b. The target is small because of the use of a wide-angle lens. In this case, the second spot cannot be seen when the image is binarized using the global threshold method. Although from performance results in Table 3.1, LT2 method seems to work well, the target edges are smoother in the first LT method.

However, the most important cases are in Figure 3.3 and in Figure 3.4, since they resemble a normal indoor environment. As seen in both Table 3.1 and in the binarized images, the GT method is best for these cases. Moreover, the GT method turns out to be more efficient in terms of computation time, which becomes an important factor especially for the real-time applications.

As a result, it is decided to use the global threshold method in this study. This binarization is the first step of the target recognition algorithm and it is performed to separate objects from their common background without any critical information loss on the image. As explained above, there is a possibility of the vanishing inner spots on the target. But these two inner spots are important for the 3-D configuration reconstruction algorithm. So, they should also be determined in the target recognition. In order to avoid this problem, after the determination of the ellipse area, a second binarization is performed on the restricted area containing the ellipse to guarantee the recognition of these spots together with the ellipse itself. When the ellipse area is recognized in the image at the end of the target recognition, a rectangular area containing the target is segmented. The size of this segmented area is 1.5 times larger then the ellipse on both horizontal and vertical directions. The second binarization mentioned above is achieved on this segmented area.

3.3 Ellipse Detection

Ellipse is one of the important geometric shapes in image processing studies. Therefore, various methods [23-31] were developed for its recognition. These methods can mainly be classified as “clustering” and “least squares fitting based” methods. Clustering methods are based on mapping sets of points to the parameter space, such as the Hough transform (see the Appendix A for details). On the other hand, least squares fitting methods focus on determining the set of parameters that minimize some distance measure between the data points and ellipse.

Criteria used for the ellipse detection algorithm developed in this study are:

- Computational load
- Capability of using edge detected image data
- Capability of determining partially grabbed ellipses up to some portion
- Accuracy

3.3.1 Hough Transform Based Algorithms

These methods differ in the parameterization step of the ellipse, which is important in reducing the parameters needed for the definition of the ellipse, i.e., parameters used in the accumulation space. This parameterization step will affect the computational load of the method. Some recently developed Hough transform based methods can be summarized as follows:

Lei and Wong [23] determined a symmetry axis for each pair of points on the contour of the ellipse, and develops an accumulation space for the candidates of symmetry axes. Two perpendicular axes that take the maximum number of votes were selected as the major and the minor axes. After finding these axes, another accumulation array was made for the lengths of these axes. The orientation of the ellipse would be given by the orientation of these axes. In this procedure, all pixels on the contour were used for determining the symmetry axes, so the algorithm cannot be used for partially occluded ellipses. Sewisy and Leberl [24] detected midpoints of edge points pairs with the same y-coordinates, defined a vertical line, and similarly using pairs with the same x-coordinates, defined a horizontal line using the accumulation method. They suggested to use the intersection of these two lines to determine the center of the ellipse. After finding the edge points of the ellipse, three edge points were used for ellipse fitting. The algorithm is suitable for partially occluded ellipses. In the study of Yip, Tam and Leung [25], the ellipses and circles were detected, using a 2-D array. Two pairs of edge points with the same edge orientation were selected. Using these points, the vertices of the ellipse were calculated as a function of five parameters of the ellipse. A 2-D accumulation space was obtained using these vertices. Four peaks in the accumulation space would correspond to four vertices of the ellipse. After selecting these vertices, five parameters of the ellipse were determined. It can be used for partially occluded ellipses, but the experiments showed that the accuracy of the extracted parameters was not as high as in a complete ellipse case. Guil and Zapata [26] used tangents of nonparallel edge points for a focusing algorithm and also recommended some checkpoints for erroneous pairings that could also be used for other algorithms. Yoo and Sethi [27] made use of poles and pole definitions of the ellipse. The algorithm is capable of detecting partially visible ellipses, overlapping ellipses, and groups of concentric ellipses. McLaughlin [28] used a randomized Hough transform after determining the center and linearizing the equation of the ellipse. Bennett, Burridge and Saito [29] suggested a single pass algorithm, which could extract any group of ellipse parameters, or characteristics that might be computed from those parameters without detecting all five parameters.

3.3.2 Least Squares Fit Based Algorithms

The most popular least squares fitting algorithm was proposed by Fitzgibbon, Pilu and Fisher [30]. This algorithm can be summarized as follows. Having representing the general conic by an implicit second order polynomial as

$$F(\mathbf{a},\mathbf{x}) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{3.2}$$

where $\mathbf{a} = [\ a\ \ b\ \ c\ \ d\ \ e\ \ f\ ]^T$ is the parameter vector and $\mathbf{x} = [\ x\ \ y\ ]^T$ is the position vector, one constraints the parameter vector **a** so that the conic represented is forced to be an ellipse. The appropriate constraint is well known, namely that the discriminant b^2-4ac is to be smaller than zero. Fitzgibbon changed this inequality constraint into an equality constraint as $b^2-4ac = -1$ and performed a least squares fitting.

Halir and Flusser [31] improved the algorithm in [30] against some singularities. Hough transform based methods have a high robustness to occlusion and no requirement for pre-segmentation, unlike least square algorithms. However, the computational load and memory use are high in these methods as compared to least squares fitting based algorithms. Those algorithms that reduce the dimensions of the accumulation space can be used to avoid this disadvantage.

In this study, Bennett’s algorithm is implemented, since it avoids the object-based pre-segmentation, has a 2-D accumulation space, and finally has a lower complexity.

3.3.3 Bennett's Approach to Characterizing Ellipses

In this approach, a parameterization is obtained for a family of ellipses, which are tangent to two line segments. Using these line segments, a general quadratic equation for a conic is represented as

$$C(x,y) = L^2(x,y) - \lambda\, l_1(x,y)\, l_2(x,y) = 0 \tag{3.3}$$

where $L(x,y) = 0$ is the line connecting two points P_1 and P_2, $l_1(x,y) = 0$ and $l_2(x,y) = 0$ are the respective tangent lines to these points (Figure 3.8), and λ is a constant. The equation for a conic could also be written in the following form:

$$\mathbf{x}^T\mathbf{A}\mathbf{x} + 2\mathbf{k}^T\mathbf{x} + c = 0 \tag{3.4}$$

where

$$\mathbf{k} = [\,f \;\; g\,]^T \tag{3.5a}$$

and

$$\mathbf{A} = \begin{bmatrix} \mathbf{a} & \mathbf{h} \\ \mathbf{h} & \mathbf{b} \end{bmatrix} \tag{3.5b}$$

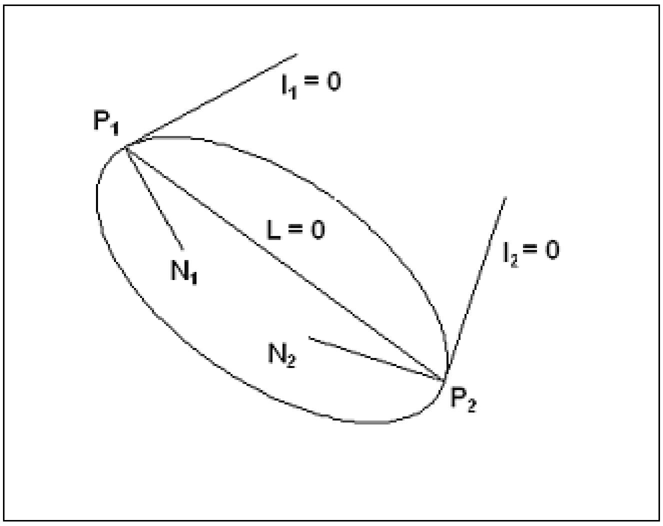


Figure 3.8 Pairs of points on an ellipse and their tangents

where parameters a, b, c, f, g, h are linear functions of λ and also depend on coordinates of each pair of edge points as well as their orientations. For such a conic to represent an ellipse, the **A** matrix should be positive definite. For each pair of edge points, the range of λ, that gives a positive definite **A** matrix is determined and ellipse centers are accumulated according to these ranges of λ.

In order to use Equation (3.3), positions of the edge pixels and their edge gradients in the image are needed. Using these values, tangents to edges at these points can be determined.

In the original algorithm, pairs of points P_1 and P_2 are selected randomly from a predetermined area. It is 25Hx25V pixels area in a 100Hx100V pixels area in [29]. In this study, the image is segmented into its connected points, and these edge pairs for the ellipse fitting are selected among the family of connected points.

3.3.4 Case Studies for Ellipse Detection

The ellipse detection algorithm is tested by using various scenes, some are generated artificially by using a paint program (Figures 3.9-3.10, 3.13-3.14) and the rest are grabbed with a CID camera and frame grabber set-up (Figures 3.11, 3.15-3.16). These scenes can be grouped as follows:

- Scenes containing symmetrical shapes other than ellipse, like rectangles or squares (Figure 3.9, 3.13)
- Scenes containing other fully captured ellipses (Figure 3.9, 3.11)
- Scenes containing partially captured ellipses (Figure 3.10, 3.13c, 3.13d)
- Real images containing some other irregular objects from the environment (Figure 3.11, 3.15-16)

It is obvious that larger the area that contains the necessary points for ellipse fitting is, better the ellipse fitting results are. However, scanning a larger area brings a greater computational load, so before the ellipse fitting process, images are segmented to their connected points.

Centers are accumulated in a 2-D array using all connected points as pairs with all possible combinations. In order to determine the characteristics of the accumulation for an ellipse and for other regular or irregular objects, the 3-D graphics of accumulation space created for various cases are observed. In these graphs, scales on the bottom plane of the graphs are the pixels on each analyzed image whereas the vertical scale is the accumulation score. Also in the graphics, some very lower scores are not included, in order to avoid the difficulty for the visual analyses of the graphs.

Before expressing some comments on these graphics, terms that will be used in the remaining part of this study will be explained. While observing these 3-D graphs, the sharpness of some peaks draws attention. In order to compare peaks with respect to this property, their degrees of sharpness are described as the "quality factor". Another feature that will be used during the comparison of peaks is their symmetry. Some local symmetrical features up to some distance from the center of each local region are observed for peaks. The distance where this symmetry vanishes is given a name as the "distance of symmetry".

Using these terms, some important observations on these accumulation spaces can be summarized as follows:

O1. Peaks corresponding to ellipses have a higher quality factor with respect to other symmetrical shapes (Figure 3.9).

O2. There are some small peaks around the maximum peak in the ellipse area (Figure 3.9). By the accumulation of some group of pixels on the high curvature side of the ellipse contour, some smaller ellipses can be obtained. Centers of these smaller ellipses are accumulated in these side peaks.

O3. Side peaks may have higher scores than other partial ellipses in the image (Figure 3.10).

O4. When more than half of an ellipse is captured, (like the ellipse in the bottom of Figure 3.10), the peak score determined in that region is located at the center of ellipse. If an ellipse is grabbed in an asymmetrical manner (such as the ellipse on the upper corner in Figure 3.10), the maximum peak in that region does not point out the center of ellipse. However, there is a peak at the actual center of the ellipse, as well.

O5. If there are some other objects with curvatures greater than the ellipse in the image, their accumulation score is greater than the ellipse score, but again the quality factor of the ellipse is greater than the quality factors of those objects. (Figure 3.11)

O6. Peaks for ellipses are more symmetrical than other peaks, so symmetry may be another property that should be added to comparison criteria.

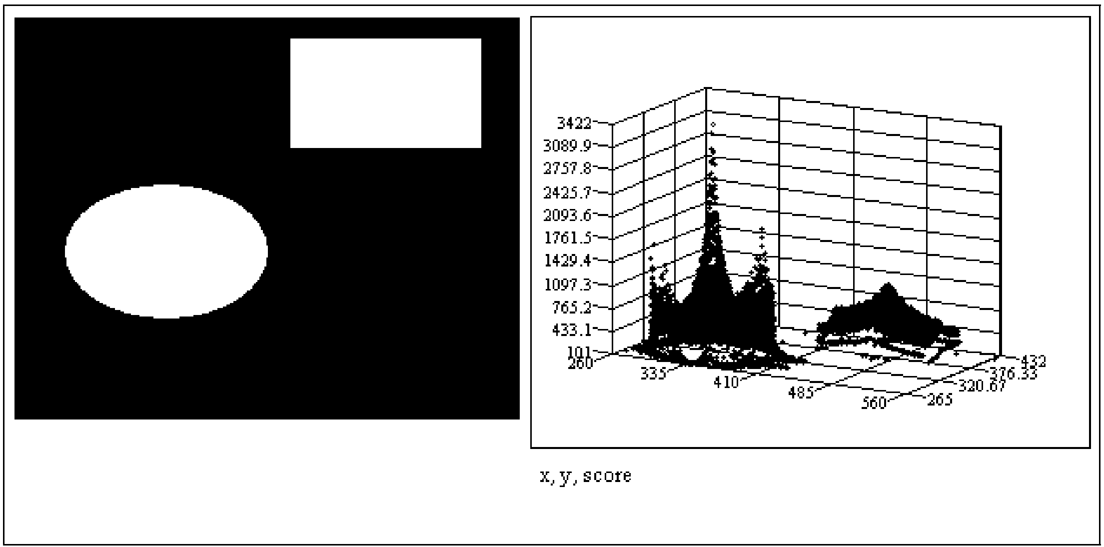


Figure 3.9 An artificial image containing an ellipse and an additional rectangular shape (above), and its 2-D accumulation space of candidate ellipse centers with their scores (right).

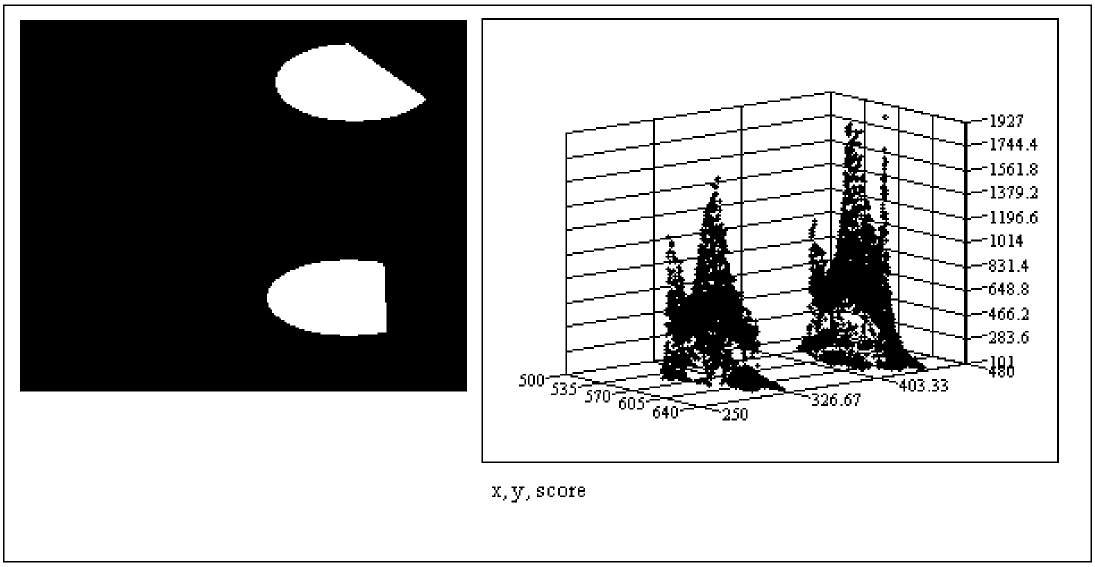


Figure 3.10   An artificial image containing two partial ellipses, and its 2-D accumulation space of candidate ellipse centers with their scores (right)

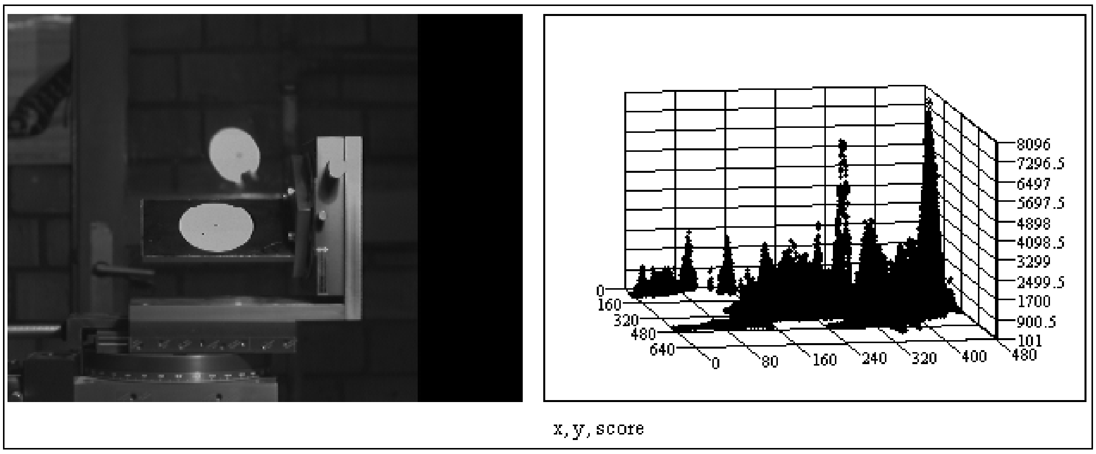


Figure 3.11   An image grabbed using the experimental set-up, and its 2-D accumulation space of candidate ellipse centers with their scores (right)

Measurement of quality factor and symmetry:

After suppressing the local maximum around each peak in the accumulation space, some imaginary contours are drawn around this maximum in the accumulation space. In Figures 3.12a and 3.12b, an example ellipse and its 3-D accumulation space can be seen. In Figure 3.12c and 3.12d, contour plots of the accumulation space are shown. Also in Figure 3.12d, in order to visualize the method, some example square contours are drawn around the maximum score. As it can be seen from Figure 3.12b, the maximum score is about 3600. So one must draw these square contours, until all the points on the contour have their score less than half of the maximum score (approximately 1800 for this case). Other scores and their coordinates are added one by one and at the end of addition of each contour, the area moments of contours are calculated. During this addition, when the weight of points that spoils the symmetry is added, the centroid of the area shifts from the starting coordinate (the coordinate of the maximum peak). The radius of the circle, which is tangent to this last contour, is defined as the “distance of symmetry”.

During this extension process, the contour, whose score is half of the maximum score, is also determined. The “quality factor” is calculated as the ratio of the maximum score to the area of circle, which is tangent to this contour. For example, if edges of the square contour which has the half score of the maximum score is 2r pixels long, and the maximum score is M, then the quality factor for that center candidate is calculated as

$$q = M / (r^2\pi) \qquad (3.6)$$

In order to test these criteria on example scenes, a series of images (Figure 3.13-3.16) are created and their symmetry and quality factors are measured and classified. Table 3.2 summarizes this information.

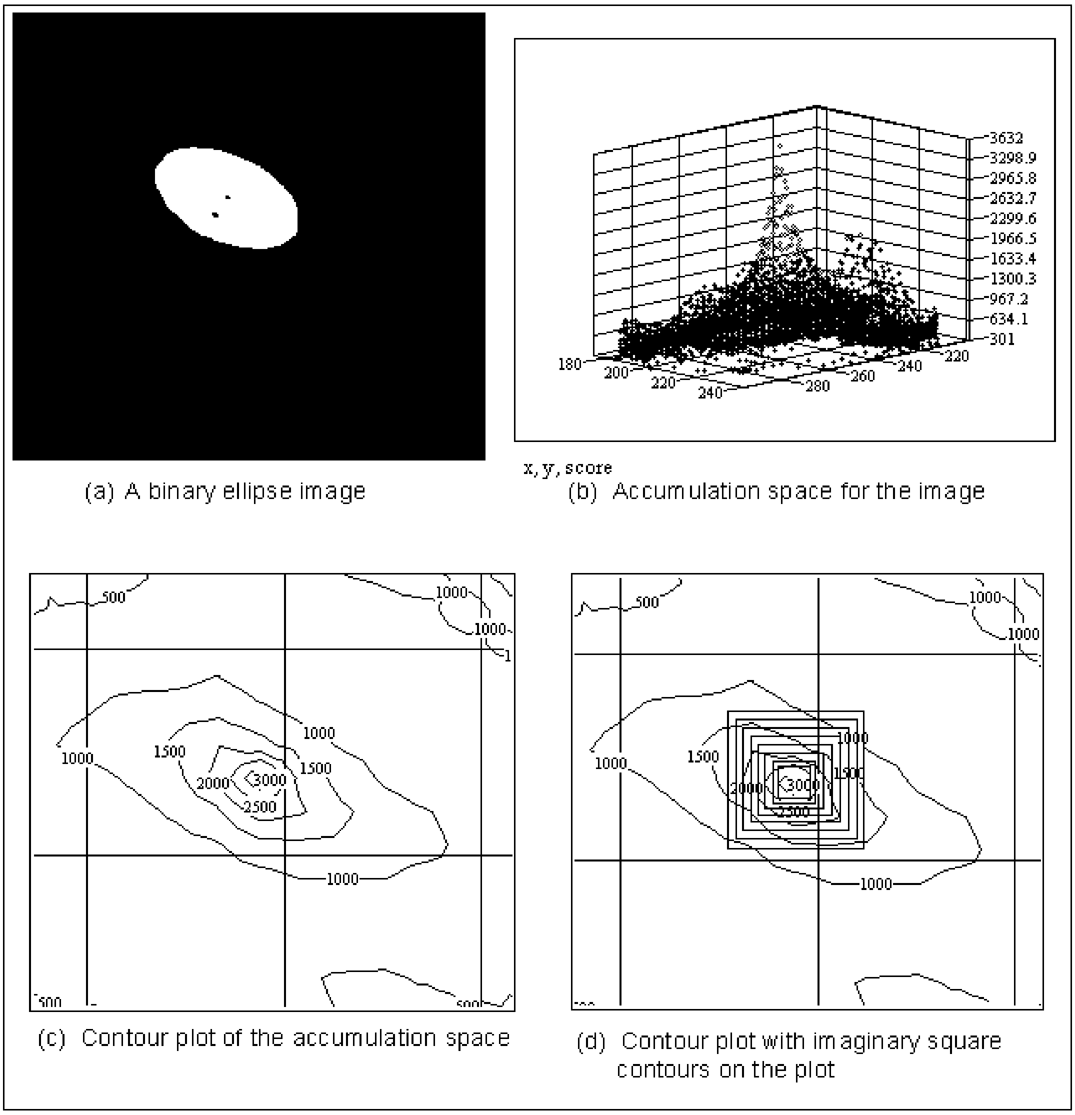


Figure 3.12 An example figure for quality factor calculation

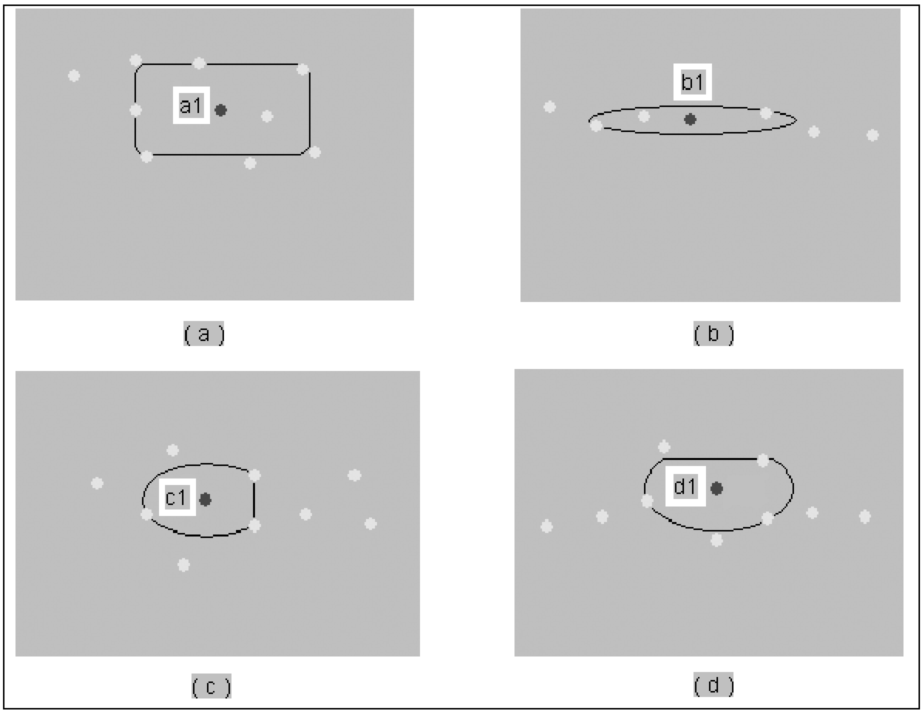


Figure 3.13 Some artificial pictures for quality factor and symmetry analysis

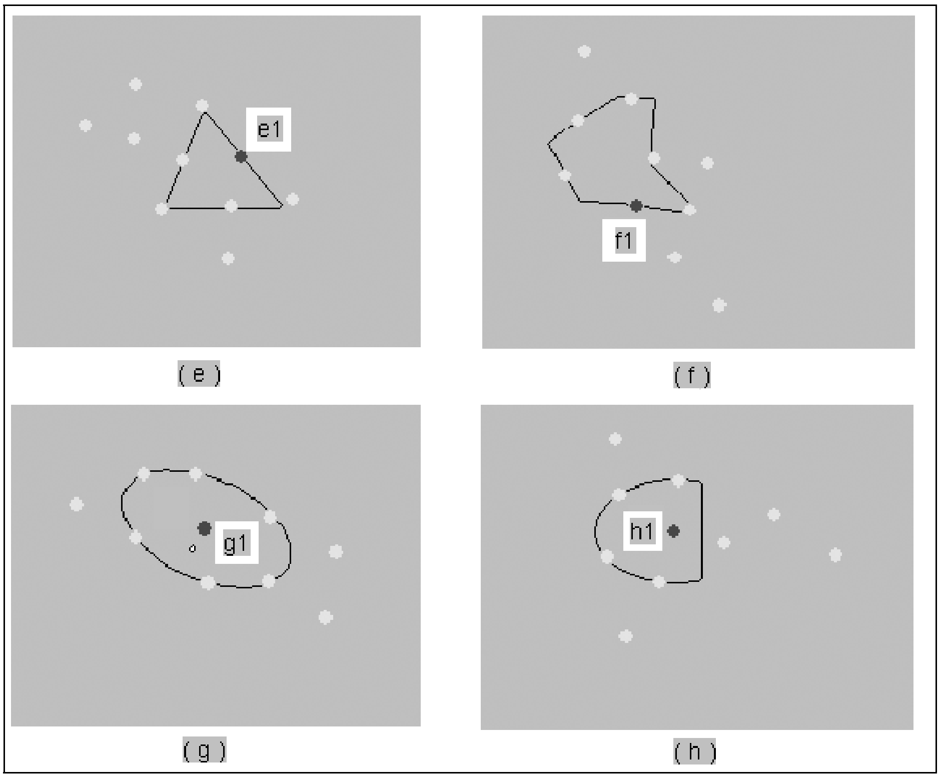


Figure 3.14 Some artificial pictures for quality factor and symmetry analysis

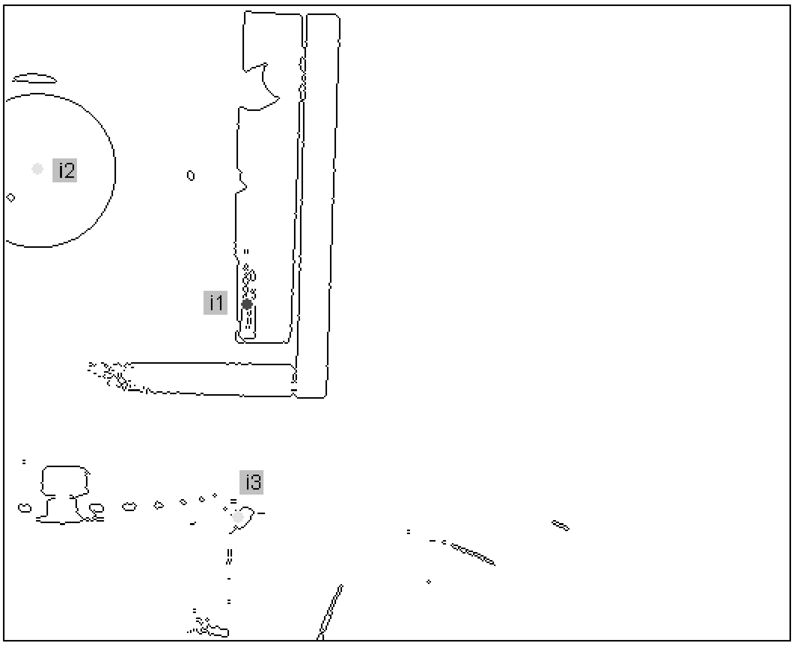


Figure 3.15 An edge thinned partial target image from the vision set-up for quality factor and symmetry analysis

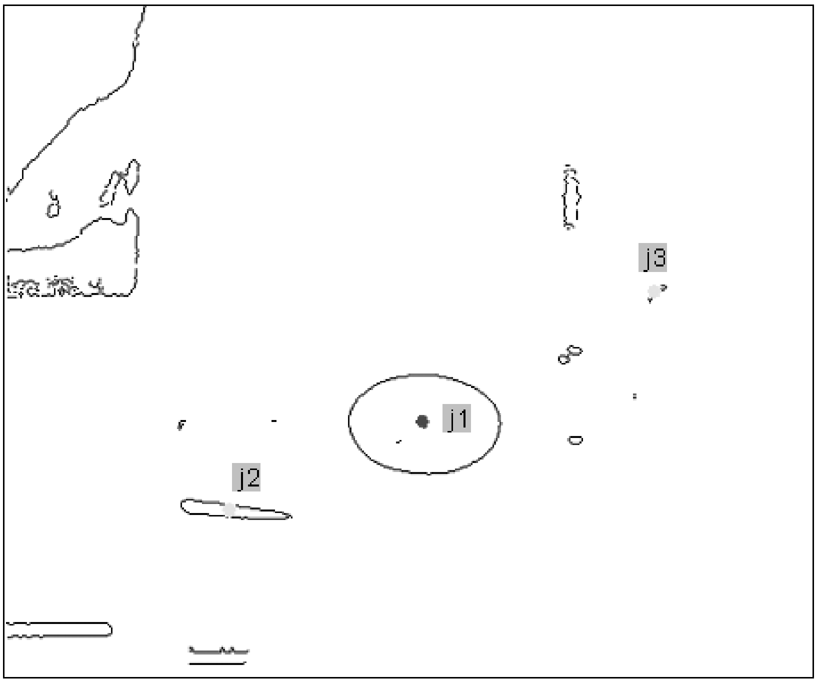


Figure 3.16 An edge thinned target image from the vision set-up for quality factor and symmetry analysis

Table 3.2 Quality and distance of symmetry analysis of artificial pictures

Point	Quality factor (1/pixel^2)	Distance of symmetry (pixels)	Score	Presence of ellipse
a1	4	90	1337	not an ellipse
b1	213	184	2682	full ellipse
c1	82	30	2315	partial ellipse
d1	42	21	2116	partial ellipse
e1	896	9	2815	not an ellipse
f1	138	12	1733	not an ellipse
g1	128	207	3632	full ellipse
h1	337	13	1694	partial ellipse
i1	134	16	6753	not an ellipse
i2	142	17	1787	partial ellipse
i3	579	23	1820	not an ellipse
j1	122	97	3443	full ellipse
j2	221	37	2773	not an ellipse
j3	597	36	1876	not an ellipse

Examining Table 3.2, one can conclude that, full ellipses have a greater quality factor with respect to other symmetrical shapes, like the one in Fig. 3.13 (a1).

As seen in Figure 3.14, the quality factors of the irregular shapes (e1 and f1) are higher than those for ellipses. Because, pixels with the half of the maximum score are closer to the center pixel as a result of asymmetric characteristics of these irregular shapes.

It is also observed that the distance of symmetry decreases as the cut portion of an ellipse increases (see h1 in Fig. 3.14).

Using these observations, in order to make a classification, the decision boundaries for the quality factor and distance of symmetry features of an ellipse can be given as

$$50\ 1/\text{pixel}^2 < \text{quality factor} < 500\ 1/\text{pixel}^2$$
$$35\ \text{pixels} < \text{distance of symmetry}$$

As a result, in order to classify points on the accumulation space as a center of an ellipse or not, points which have the highest score are selected. These points are candidates for the center of a possible ellipse in the image. For each center candidate, the quality factor and the distance of symmetry are calculated. The center candidates, which have the quality factor and distance of symmetry values between the limits given above, are selected. From the selected ones, the ellipse center is the one with the highest score. If there is no candidate for ellipse center, which has the quality factor and distance of symmetry values between these limits, there is still a possibility of having a partial ellipse in the image. So, the decision boundary for the distance of symmetry can be extended in order to recognize a partial ellipse.

CHAPTER 4

TARGET RECOGNITION

4.1 Overview

In the previous chapter, some image processing and pattern recognition algorithms are explained and results obtained by their applications on a set of artificial and real scenes are presented. In this chapter, a complete target recognition algorithm and results obtained by its application will be presented. The flow chart of the algorithm can be seen in Figure 4.1.

As briefly introduced in Chapter 1, a typical pattern recognition algorithm starts with the measured output of a sensor in a system and continues with the preprocessing and enhancing this data. The preprocessing is the filtering or transforming raw input data to aid its computational feasibility and feature extraction and to minimize the noise [1]. The next step in the recognition algorithm is the extraction of features, which will be utilized in a classification algorithm to end up with the recognition of observed data. Features are some extractable measurements, which may be symbolic, numerical, or both.

The feature selection is a process of choosing an input to the pattern recognition system and involves judgment. It is often useful to develop a geometric viewpoint of features, especially in a statistical pattern recognition case. Features are arranged in a d-dimensional feature vector, which yields a multidimensional measurement space or feature space. Often, a classification is accomplished by partitioning the feature space into regions for each class.

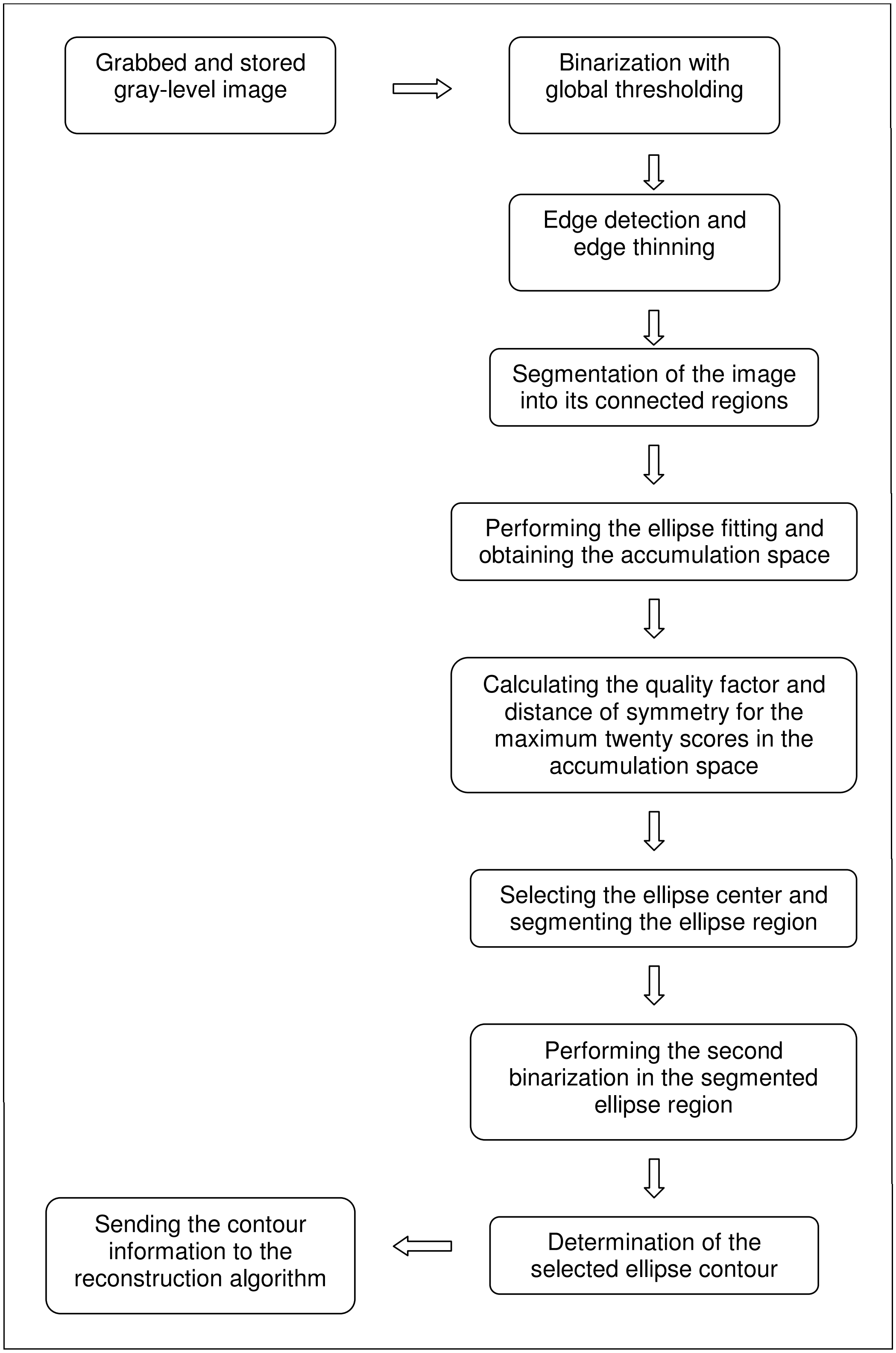


Figure 4.1 Flow chart for target recognition algorithm

In this study, a pattern recognition algorithm is developed to recognize an elliptical image of a circular target used in the 3-D configuration reconstruction algorithm. It begins with a previously captured and stored gray-level image. In the preprocessing step, an automatic binarization with global threshold, an edge detection and an edge thinning are performed.

The edge detection is achieved by the convolution of the image array with a 3x3 Sobel operator, which is actually a first derivative operator. Since after the binarization, the gray-level image is transformed into an image with stepwise edges, a first derivative operator is decided to be suitable for the detection of local intensity changes in the binarized image. Some details of the edge detection process and also some common algorithms are given in Appendix A.

After the edge detection, it is seen that edges are usually thicker than one pixel size, and therefore an edge thinning is needed in order to reduce the computational load for rest of the processing. Some details for the definition of edge thinning and also some algorithmic principles can be found in Appendix A.

The ellipse-fitting algorithm employed in this study uses pairs of image pixels belonging to the boundary of a region. This region is expected to be elliptical, and therefore the algorithm tries to fit an ellipse to these point pairs. In the fitting algorithm, local tangents to edges of these two points are utilized. In other words, local slopes of edges at these points are used. The slope information needed is obtained during the edge detection step. Center coordinates of all such fitted ellipses and scores of each center are accumulated in a space called as the "accumulation space". The selection of pixel pairs becomes very crucial for the selection of features. It is seen that, if pixel pairs are selected as connected pairs, regions in the accumulation space become more separated and their properties become more perceivable. For this reason, it is best to segment the image into its connected regions, before applying the ellipse-fitting algorithm.

When an accumulation space is observed, it is seen that regions containing elliptical boundaries can be classified using the values defining the quality factor and size of the symmetry region. So the feature space can be partitioned into decision regions by the following features: Ellipse fitting score, quality factor, and size of the symmetry region for each ellipse center candidate.

The algorithm is applied to some real and artificial scenes and a series of quantitative values for objects of regular and irregular shapes are obtained and presented in the previous chapter. Using these results, some decision boundaries for the classification are determined.

Following the determination of the ellipse center, an outward search is started from this center to reach a point on the ellipse contour. Once such a point is found, a contour following algorithm [7] is used to determine the complete contour of the ellipse.

As explained in Chapter 3, a global thresholding may result in elliptical images with vanishing internal spots under some lighting environments. To remedy this undesirable performance, following the detection of an elliptical region, this region is segmented from the original image and a second binarization is performed only on this segmented region. Since the target is a white circle with a black background and with two black inner spots, the selection of a new threshold for this segmented area makes two classes of background and foreground pixels more separated. After this second binarization, a new ellipse contour is determined for use in the 3-D configuration reconstruction algorithm.

The entire image processing steps and the rest of the 3-D configuration reconstruction algorithm is implemented by a software, which is presented in Appendix B.

4.2 Results of the Recognition Algorithm

After checking the performance of the image processing algorithms separately, the target recognition algorithm is applied to the images captured using the vision set-up. As an example, target recognition algorithm is presented step by step through the Figures 4.2a-4.2e. The original image can be seen in Figure 4.2a, and its binarized image in Figure 4.2b. In Figures 4.2c and 4.2d, images obtained after edge detection and edge thinning processes are shown, respectively. In these last two images, the background is painted in black color and the edge pixels are in white color. For all edge pixels in Figure 4.2c, the magnitudes of the edge gradients in horizontal and vertical direction are stored in text files. These files are used to calculate the slopes of edge pixels in the ellipse fitting step. As seen in the figure, both the contour of the ellipse and the contour of the spots are visible after the edge thinning process.

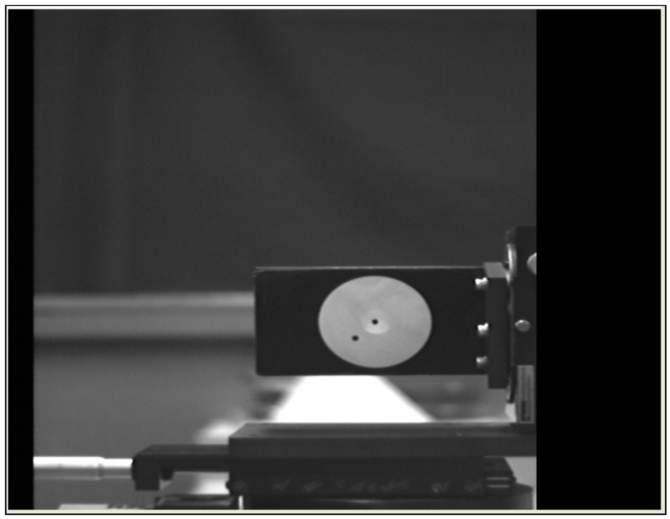


Figure 4.2a Image of the target before processing

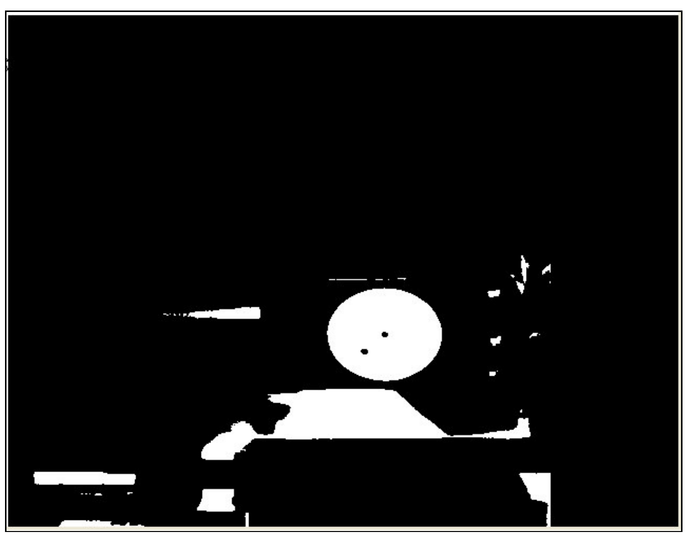


Figure 4.2b Image of the target after binarization

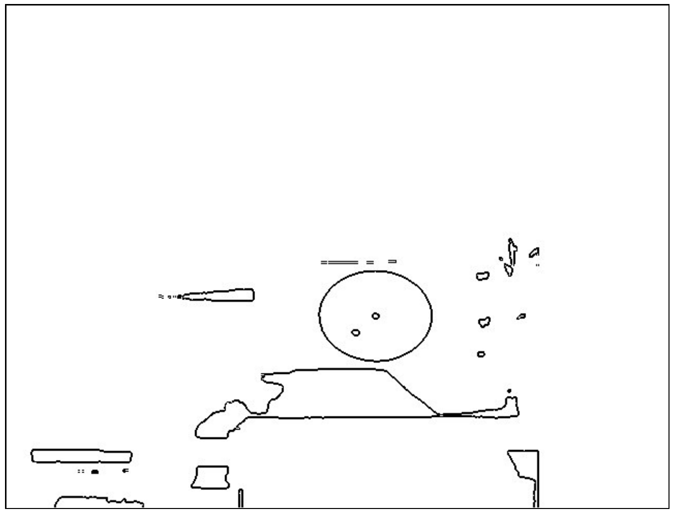


Figure 4.2c Image of the target after the edge detection

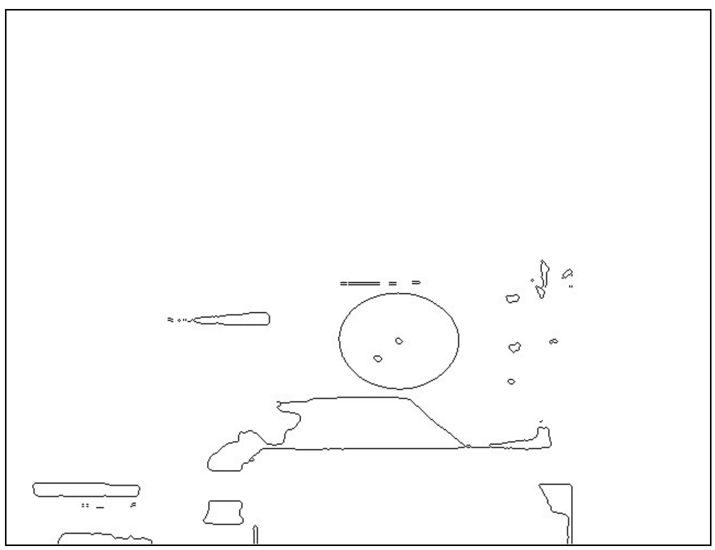


Figure 4.2d Image the target after the edge thinning

Using only edge pairs of connected components, an ellipse fitting is performed and features like quality factor and distance of symmetry for the top twenty center candidates in the accumulation space are calculated. The center candidate, which is within the decision boundaries of quality factor and distance of symmetry is the center of the ellipse and marked with a spot and called as “e1”. The quality factor for the point “e1” is 409 1/pixel^2, and the distance of symmetry is 53 pixels. The outer contour of the ellipse is determined and a rectangular area containing this contour is segmented. A second binarization is performed on this area. The binary segmented area is shown on the left upper corner of Figure 4.2e. In this figure, again the colors are inverted, and the background is shown in white and the object pixels are in black color. After the second binarization, the ellipse contour is determined once more, and the resultant contour is ready to be used in the 3-D configuration reconstruction algorithm.

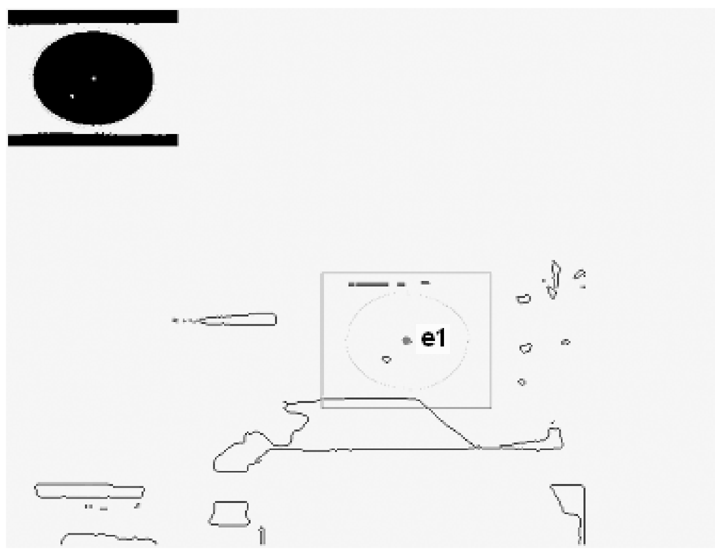


Figure 4.2e Image of the target after target recognition

In the second example, a binary image (Figure 4.3a) and its resultant image after target recognition (Figure 4.3b) are shown. Again, three features of the objects (accumulation score, quality factor, and distance of symmetry) in the image are calculated. While summarizing the properties of the accumulation space in the previous chapter, it is pointed out that although the ellipse centers do not have the greatest scores in the accumulation space for some images, the ellipse can be recognized properly according to their quality factor and distance of symmetry features. The ellipse in Figure 4.3a is an example for this kind of images. The quality factor for the point “b1” on the ellipse center (Figure 4.3b) is 126 1/pixel^2 and the distance of symmetry is 35 pixels. After the segmentation of the ellipse area, the ellipse contour is determined.

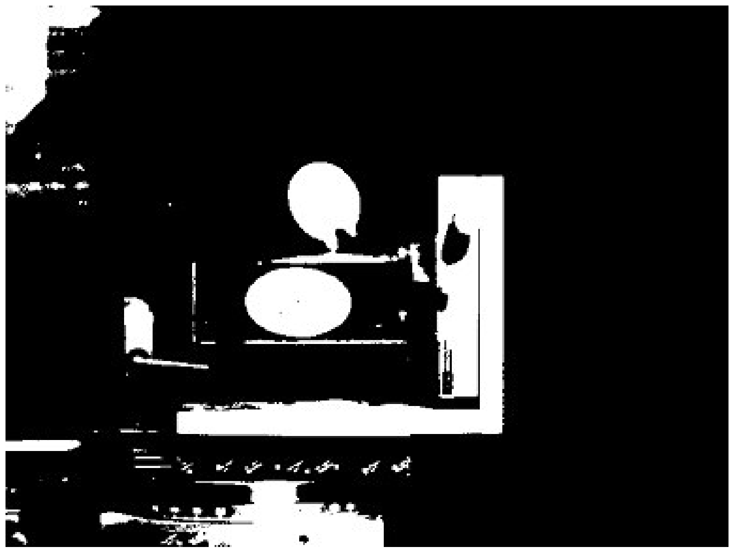


Figure 4.3a Image of the target after binarization

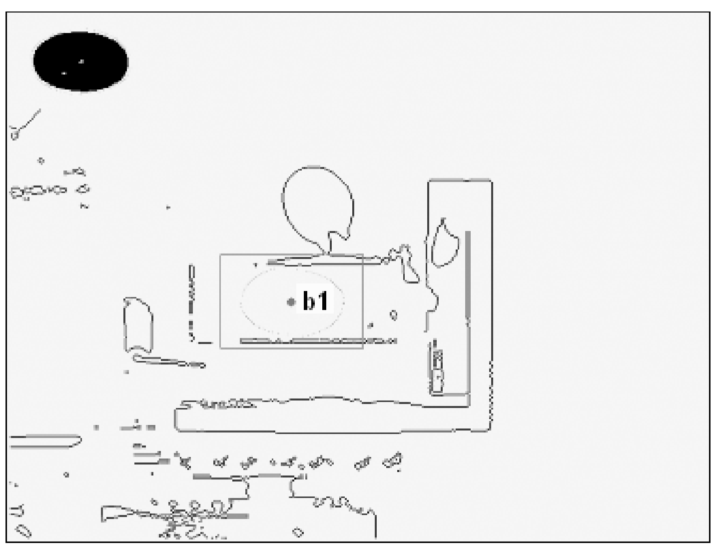


Figure 4.3b Image of the target after target recognition

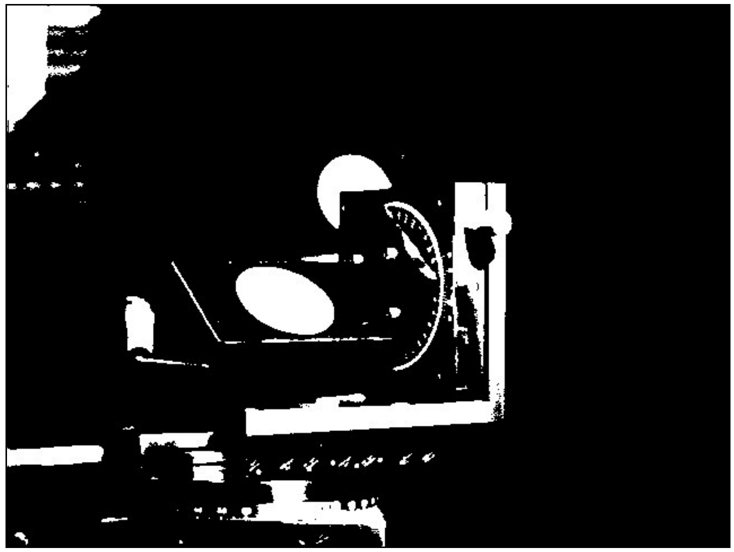


Figure 4.4a Image of the target after binarization

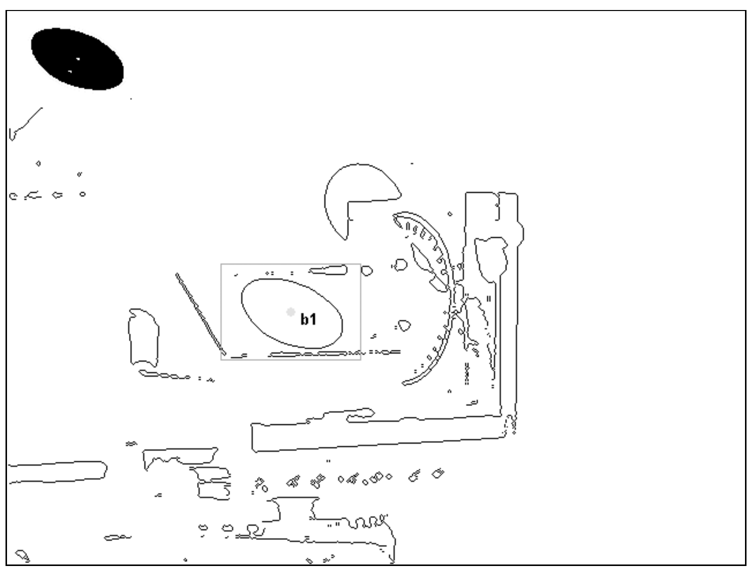


Figure 4.4b Image of the target after target recognition

In Figure 4.4a, another binary image is given. During the initial global binarization, inner spots are lost, so a second binarization seems to be necessary after the segmentation of the ellipse area. The target recognition is performed and the ellipse area is recognized correctly as seen in Figure 4.4b. After a second binarization in this area, inner spots can also be recognized. The quality factor for the ellipse center “b1” is 113 1/pixel^2 and the distance of symmetry is 75 pixels. Even though this point does not have the highest score in the accumulation space; it has proper quality factor and distance of symmetry values as in the previous example.

The images in the case studies have size for 640Hx480V pixels^2. When the size of the target is greater the quarter of the image area, the decision boundaries for the quality factor may not be utilized. As the size of the ellipse area approaches the half of the image area, the peak in the accumulation space on the ellipse center vanishes.

As a result of these case studies, it can be concluded that the target recognition algorithm developed in this study can be used for the recognition of ellipses. The accumulation score, quality factor and distance of symmetry are the features that are used in the recognition. The decision boundaries for these features are given in the previous chapter. If the ellipse recognition cannot be achieved with in these boundaries, they may be extended in order to recognize a partial ellipse. After the recognition of the ellipse, segmentation of the area that contains the ellipse and the second binarization on this area is achieved. At the end of the target recognition, contour of the ellipse and the contour of the inner spots are detected. This information is utilized for the 3-D configuration reconstruction algorithm.

CHAPTER 5

INTERNAL CAMERA CALIBRATION

5.1 Overview

In this chapter, the aim of internal camera calibration and previous studies on this subject will be explained. Then, a proposed method for the determination of lens distortion parameters will be presented.

The problem of interior orientation can be defined as to determine the internal geometry of the camera. The geometry is represented by a set of camera parameters [32]:

a) <u>Camera constant</u> for the distance of the image plane from the center of projection.
b) <u>Principal point</u> for the location of the origin of the image plane coordinate system.
c) <u>Lens distortion coefficients</u> for the changes in the image plane coordinates caused by optical imperfections in the camera.
d) <u>Scale factors</u> for distances between the rows and columns.

The interior orientation problem is a problem of compensating errors in constructing a camera so that the bundle of rays inside the camera obeys the assumptions of the perspective projection. Camera parameters are also called as the “intrinsic parameters”, as opposed to the “extrinsic parameters” for the exterior orientation of the camera.

The “camera constant” is not the same as the “focal length” of the lens all the times except when the lens is focused at infinity; then the camera constant is equal to the focal length. Otherwise, the camera constant is always less than the focal length. On the other hand, the “principal point” is where the optical axis intersects the image plane.

In the previous studies, the calibration for the determination of the optical center is done geometrically by the rotation method [7,33]. Also, an iterative algorithm is proposed in [6] for the determination of the image plane distance, but it is not included in the solution of the reconstruction algorithm.

In this study, a method for the determination of the lens distortion is also proposed.

5.2 Studies on the Internal Camera Calibration

Tsai and Lenz [3,34], uses the concept of “radial alignment constraint” to decompose calibration parameters into two groups. The first group consists of only extrinsic parameters; i.e., the relative motion and translation between 3-D camera and world coordinate system. The second group of calibration parameters, i.e., radial lens distortion parameters and the focal length, can be obtained by solving perspective projection equations with a few iterations for optimization, using an initial guess obtained by solving two unknowns.

Devernay and Faugeras [35] use 3-D line segments, which can be city scenes, interior scenes, or aerial views containing buildings and man-made structures. Edge extraction and polynomial approximation are applied on these images in order to detect possible 3D edges present in the scene, and then they look for the distortion parameters that minimize the curvature of the 3D segments projected to the image. After they find a first estimate of the distortion parameters, they perform another polynomial approximation on the corrected edges. This way, straight-line segments that were broken into several line segments due to distortion become one single line segment, and outliners are implicitly eliminated. They continue this iterative process until they fall into a stable minimum of the distortion error after a polynomial approximation step.

Another method is measuring the distortion in an image of a perfect circle into an ellipse. Chatterjee and Roychowdhury [8] use one of these methods to determine the initial estimate. They deal with the coplanar camera calibration and discuss the camera calibration model consisting of extrinsic and intrinsic parameters.

The technique proposed by Zhang [10] in which the radial distortion is modeled next to external camera parameters, only requires a camera to observe a planar pattern shown at a few (at least two) different orientations. Either the camera or the planar pattern can be moved by hand. The motion need not be known. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion.

Prescott and McLean [36] determine the radial distortion parameters with a technique based on the analysis of distorted images of straight lines. Lines are extracted by grouping the individual pixels as belonging to a particular line support region, and the distortion parameters are optimized, so that distortions can be obtained in linear features.

Heikkila and Silven [4,37] develop a four-step calibration procedure that is an extension of two-step methods, in which initial parameter values are computed linearly and final values are obtained with a nonlinear minimization. The third step is the correction of asymmetric projection and finally, the fourth step is the correction of image by avoiding tangential and radial lens distortions by back-projection.

5.3 Lens Distortion

As a result of imperfections in the production and assembly of lenses, the image of a planar object lies in general on a slightly curved field [8], wherein objects at the edge of the field of view appear somewhat smaller or larger than they should be.

Lens distortions include two components: radial distortion that bends the rays by more or less than the correct amount [32]. Two typical radial distortions are pincushion and barrel distortions. The pin cushion distortion results, for example when a lens is used as a magnifying glass, whereas barrel distortion results when an object is viewed through a lens at some distance from the eye. On the other hand, tangential distortions are usually caused by a decentering in the lens (decentering distortion) due to imperfections in lens manufacturing or a tilt in the camera sensor or lens (thin-prism distortion)

Formulation of the Lens Distortion

The radial distortion can be formulated as:

$$\begin{bmatrix} \delta u_j^{(r)} \\ \delta v_j^{(r)} \end{bmatrix} = \begin{bmatrix} u_i^{ud}(k_1 r_i^2 + k_2 r_i^4) \\ v_i^{ud}(k_1 r_i^2 + k_2 r_i^4) \end{bmatrix} \tag{5.1}$$

and the expression for the tangential distortion is often written in the following form:

$$\begin{bmatrix} \delta u_j^{(t)} \\ \delta v_j^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1 u_i^{ud} v_i^{ud} + p_2(r_i^2 + 2u_i^{ud^2}) \\ p_1(r_i^2 + 2v_i^{ud^2}) + 2p_2 u_i^{ud} v_i^{ud} \end{bmatrix} \tag{5.2}$$

where (u_i^{ud}, v_i^{ud}) are undistorted image coordinates, r_i is the distance between the observed point and the origin, $(\delta u_j^{(r)}, \delta v_j^{(r)})$ and $(\delta u_j^{(t)}, \delta v_j^{(t)})$ are respective radial and tangential distortions on these points. Tangential distortion coefficients are p_1 and p_2, while the radial distortion coefficients are k_1 and k_2.

Combining Equations (5.1) and (5.2) for N number of points and arranging known coefficients and unknowns ones, the following equation can be obtained.

$$\begin{bmatrix} \delta u_0 \\ \delta v_0 \\ \delta u_1 \\ \delta v_1 \\ \cdots \\ \cdots \\ \delta u_{N-2} \\ \delta v_{N-2} \\ \delta u_{N-1} \\ \delta v_{N-1} \end{bmatrix} = \begin{bmatrix} u_0^{ud} r_0^2 & u_0^{ud} r_0^4 & 2u_0^{ud} v_0^{ud} & (r_0^2 + 2{u_0^{ud}}^2) \\ v_0^{ud} r_0^2 & v_0^{ud} r_0^4 & (r_0^2 + 2{v_0^{ud}}^2) & 2u_0^{ud} v_0^{ud} \\ u_1^{ud} r_1^2 & u_1^{ud} r_1^4 & 2u_1^{ud} v_1^{ud} & (r_1^2 + 2{u_1^{ud}}^2) \\ v_1^{ud} r_1^2 & v_1^{ud} r_1^4 & (r_1^2 + 2{v_1^{ud}}^2) & 2u_1^{ud} v_1^{ud} \\ \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot \\ \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot \\ u_{N-2}^{ud} r_{N-2}^2 & u_{N-2}^{ud} r_{N-2}^4 & 2u_{N-2}^{ud} v_{N-2}^{ud} & (r_{N-2}^2 + 2{u_{N-2}^{ud}}^2) \\ v_{N-2}^{ud} r_{N-2}^2 & v_{N-2}^{ud} r_{N-2}^4 & (r_{N-2}^2 + 2{v_{N-2}^{ud}}^2) & 2u_{N-2}^{ud} v_{N-2}^{ud} \\ u_{N-1}^{ud} r_{N-1}^2 & u_{N-1}^{ud} r_{N-1}^4 & 2u_{N-1}^{ud} v_{N-1}^{ud} & (r_{N-1}^2 + 2{u_{N-1}^{ud}}^2) \\ v_{N-1}^{ud} r_{N-1}^2 & u_{N-1}^{ud} r_{N-1}^4 & (r_{N-1}^2 + 2{v_{N-1}^{ud}}^2) & 2u_{N-1}^{ud} v_{N-1}^{ud} \end{bmatrix} . \begin{bmatrix} k_1 \\ k_2 \\ p_1 \\ p_2 \end{bmatrix} \qquad (5.3)$$

Using the following definitions for A and C vectors and B matrix

$$A = \begin{bmatrix} \delta u_0 \\ \delta v_0 \\ \delta u_1 \\ \delta v_1 \\ \cdots \\ \cdots \\ \delta u_{N-2} \\ \delta v_{N-2} \\ \delta u_{N-1} \\ \delta v_{N-1} \end{bmatrix} \quad B = \begin{bmatrix} u_0^{ud} r_0^2 & u_0^{ud} r_0^4 & 2u_0^{ud} v_0^{ud} & (r_0^2 + 2{u_0^{ud}}^2) \\ v_0^{ud} r_0^2 & v_0^{ud} r_0^4 & (r_0^2 + 2{v_0^{ud}}^2) & 2u_0^{ud} v_0^{ud} \\ u_1^{ud} r_1^2 & u_1^{ud} r_1^4 & 2u_1^{ud} v_1^{ud} & (r_1^2 + 2{u_1^{ud}}^2) \\ v_1^{ud} r_1^2 & v_1^{ud} r_1^4 & (r_1^2 + 2{v_1^{ud}}^2) & 2u_1^{ud} v_1^{ud} \\ \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot \\ \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot \\ u_{N-2}^{ud} r_{N-2}^2 & u_{N-2}^{ud} r_{N-2}^4 & 2u_{N-2}^{ud} v_{N-2}^{ud} & (r_{N-2}^2 + 2{u_{N-2}^{ud}}^2) \\ v_{N-2}^{ud} r_{N-2}^2 & v_{N-2}^{ud} r_{N-2}^4 & (r_{N-2}^2 + 2{v_{N-2}^{ud}}^2) & 2u_{N-2}^{ud} v_{N-2}^{ud} \\ u_{N-1}^{ud} r_{N-1}^2 & u_{N-1}^{ud} r_{N-1}^4 & 2u_{N-1}^{ud} v_{N-1}^{ud} & (r_{N-1}^2 + 2{u_{N-1}^{ud}}^2) \\ v_{N-1}^{ud} r_{N-1}^2 & u_{N-1}^{ud} r_{N-1}^4 & (r_{N-1}^2 + 2{v_{N-1}^{ud}}^2) & 2u_{N-1}^{ud} v_{N-1}^{ud} \end{bmatrix} \quad C = \begin{bmatrix} k_1 \\ k_2 \\ p_1 \\ p_2 \end{bmatrix}$$

Equation (3) can be written as

$$A_{2Nx1} = B_{2Nx4}\, C_{4x1} \qquad (5.4)$$

When elements of matrices A and B are known, the following pseudo inversion technique could be used in order to solve the equation for the C.

$$C = (B^T B)^{-1} B^T A \qquad (5.5)$$

After determining the distortion coefficients, the distorted points are back-transformed to their undistorted coordinates. The formulation for the back-transformation of distorted points is as follows:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} u_i^{ud} + \delta u_i^{(r)} + \delta u_i^{(t)} \\ v_i^{ud} + \delta v_i^{(r)} + \delta v_i^{(t)} \end{bmatrix} \tag{5.6}$$

5.4 Proposed Method

In this study, the extrinsic camera parameters are solved by using the 3-D configuration reconstruction algorithm, so the internal camera parameters could be solved independent of these external parameters. It would be better to use a back-projection algorithm rather than an optimization based method. In the back-projection step, a calibration object is needed.

It is aimed to calibrate the camera in an unstructured environment, without the measured data for the external parameters. The information of control points used will be extracted from the calibration pattern itself. Since, as a matter of definition of lens distortion, distortions at the camera center (so the image center) are zero, one can assume that distortions very close to the center is also zero. Referring to some points near the center, distortions on the whole image can be obtained.

In order to measure distortions, a pattern having white circular spots on a black background is proposed. Control points are chosen as centroids of these circular spots. In the middle of the pattern, there is a circle with a center spot in it. Since the image of the pattern is grabbed in an unstructured environment, in the absence of a positioning set-up, this circle is utilized for the determination of the external parameters of the pattern by using the 3-D configuration reconstruction algorithm. Four spots around this circle are used for the sample pattern and they are assumed to be free of distortion. The calibration pattern can be seen in Figure 5.1.

Another important property of the image of the pattern is the image center, which is marked for the easy selection. In the image grabbing step, the image center is adjusted such that image center is projected on the circle. The image should all be full of grids, so that a homogeneous distribution of sample points can be obtained, which will effect the correctness of the distortion parameters, so the calibration procedure.

After determining the external parameters of the pattern, a look-up table is developed by transforming the original grid to the image coordinate frame using these external parameters and compared with the image of the grid pattern. The error for each control point is determined and an error matrix is developed. By substituting the parameters in the equation (5.3) and solving Equation (5.5), the tangential and radial distortion parameters are determined.

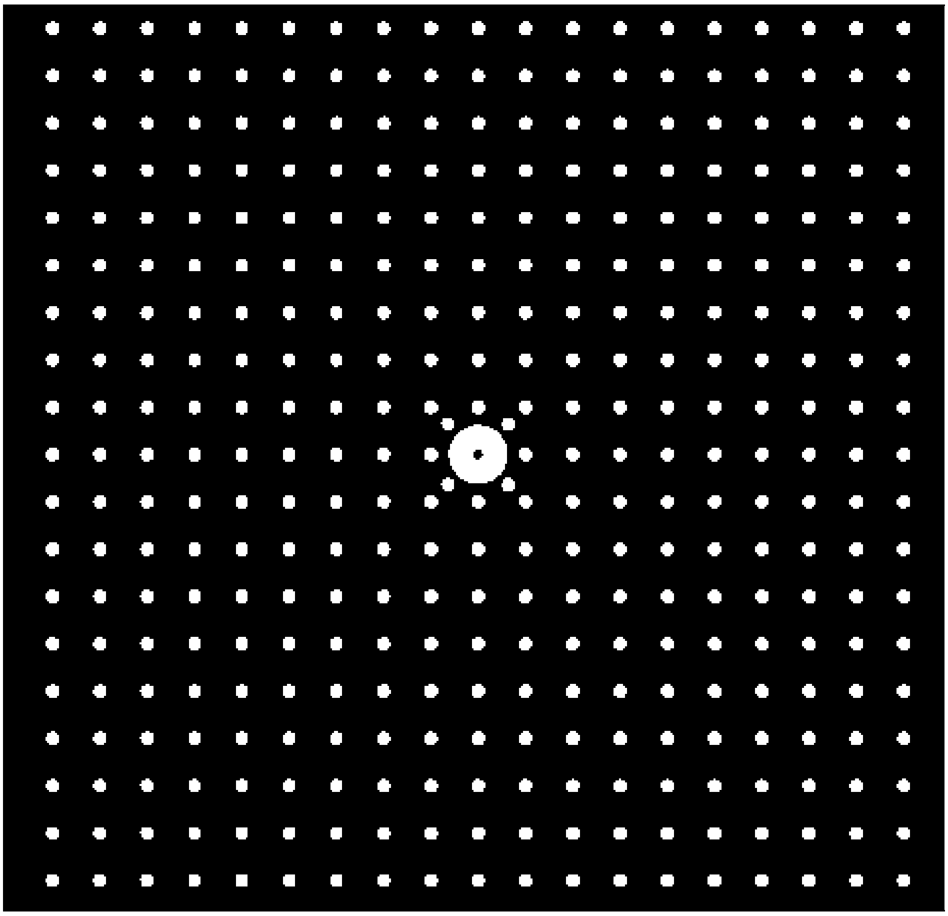


Figure 5.1 Calibration pattern and a sample pattern in the middle

5.4 Calibration Procedure

The calibration steps are as follows:

- The calibration pattern is viewed, by taking as many grids as possible in order to have a homogeneous image full of grids. The image center should be as near as the pattern center.

- The sample calibration grid, which involves the circle in the middle and the four spots around it, is taken in a window with the cursor.

- The software program determines the centroids of these four spots, and these points are assumed to be undistorted (Figure 5.1).

- A greater area, which will be used to calculate the distortion parameters on the image, is selected. The number of vertical and horizontal spots is calculated, and using the sample pattern and external camera parameters, correct places of intersection points are calculated and a look-up table is generated.

- Calculated distortion parameters will not used on the whole image each time, instead after determining the ellipse area, the pixels on that area are back-projected to their undistorted coordinates, and the 3-D configuration reconstruction algorithm will continue with these undistorted coordinates.

CHAPTER 6

SUMMARY AND CONCLUSIONS

In this chapter, a summary of the study and conclusions reached will be presented, and some recommendations for future research will be given.

6.1 Summary

A computer based target recognition algorithm is developed in order to make an existing 3-D configuration reconstruction algorithm, autonomous. The reconstruction algorithm assumes that a monocular image of a scene is available, which contains a circular white secondary target with two inner dots over a predominantly dark background.

The target recognition algorithm developed in this study processes a raw image of size 640Hx480V pixels to obtain pixel coordinates of the boundary of an ellipse, which is supposed to be the image of the circular secondary target. In the preprocessing steps of the algorithm, first a binarization is performed in order to separate object-like white regions from dark background regions of the scene. The binarization is performed by an automatic selection of a threshold gray-scale value based on the method “Minimum Error Thresholding”. This method is applied by using both global and local thresholding approaches in order to see their relative merits. Comparative results have led the use of the global thresholding in this study.

In order to fit an ellipse to a given region in a full image, pixel coordinates and local slopes of the edge of such a region are needed. So, an edge detection process is performed on the binarized image by using a Sobel operator and edge gradients in horizontal and vertical directions are obtained. Using these gradients the edge information in the whole image are generated.

After the edge detection, it is observed that, the thickness of edges is larger than one pixel size, usually about three pixels for the images used in this study. In order to reduce the computational load for the rest of the recognition algorithm, a further process called as the edge thinning is performed on the edge detected image.

In order to fit ellipses to edge thinned images, a Hough Transform based algorithm available in the literature is used. However, it is seen that its results cannot be used directly, by barely using the algorithm. Therefore, a set of modifications and enhancements are developed and implemented to make the algorithm suitable to the purpose of this study.

The algorithm utilizes edge points on the image as pairs and their slopes, which are obtained as gradients in the edge detection step. After performing the ellipse fitting, an accumulation space is build up for center coordinates of possible ellipses fitted to all possible pairs of edge points on the image. In the original algorithm pairs of edge points were selected in a restricted region defined in the software. In this study, first the image is partitioned into its connected regions and then edge pairs are selected from these connected regions for use in the ellipse fitting algorithm.

Another enhancement developed in study is in the interpretation of the accumulation space. In the original algorithm, only scores of center candidates were used for classification. In this study, the accumulation space is observed and some features of elliptical regions in this space are extracted. In addition to the score values in the accumulation space, features like quality factors of the maxima and distances of symmetry around these maxima are used in the classification step.

The determination of the center candidate is followed by the determination of the ellipse contour by a contour following algorithm. Then the region containing this elliptical contour in the image is segmented. Using only this segmented area, a second binarization is performed on the original image. The final contour is determined by repeating all intermediate steps of the target recognition process limited to the segmented area. This contour information is then used in the 3-D configuration reconstruction algorithm.

For the detection of distortion parameters, a method with a special pattern is proposed. The method assumes no special structuring for the environment nor any information about the external parameters of the pattern used. The pattern needs not to be configured in a special orientation; only the image of the center of the pattern should coincide with the image center. While determining the pattern configuration, the 3-D configuration reconstruction algorithm is used. Positions of control points on the image and their undistorted positions are compared and an error matrix is developed. By using the pseudo inversion technique, the distortion parameters are calculated.

In order to implement all algorithms summarized above, a user friendly software is developed in the Windows environment using Borland C++ Builder Professional compiler.

6.2 Conclusions

In the binarization part of the study, both the local and global binarizations are studied. In addition to the eye inspection of resulting binary images, these methods are compared analytically according to the homogeneity of their distributions. It is concluded that, the global binarization is more efficient to be used in this study. In order to avoid inner spots to vanish in some bright image cases, a second binarization is needed on a restricted area after recognition of the ellipse area.

In the ellipse fitting part of the study, an edge thinning and partitioning the image into its connected components bring computational advantage to the algorithm. As a result of selecting edge pixels from connected components, regions in the accumulation space become more separated and properties of their regions more perceivable.

The target recognition algorithm developed in this study is able to detect a full ellipse in an image without any difficulty. But, it can also be used to identify partially captured ellipses of symmetrical type up to a ratio of partition. In the target recognition process, a set of decision boundaries are developed in terms of the quality factor and distance of the symmetry for choosing the center of an elliptical shape among multiple candidates. It is shown that the recognition of partial ellipses may become critical if there are other curved patterns next to the partial ellipse in the same image. In such cases, lowering the distance of symmetry limit appears to be the best solution to catch any possible partial ellipse especially when no ellipse is detected within decision boundaries specified for the full ellipse case.

In previous studies [6,7], after detecting the ellipse contour, eight contour pixels were selected for use in the 3-D configuration reconstruction algorithm. This selection was achieved with a method called “MaxMinCross Pixel Selection” method [6,7]. Since the 3-D configuration reconstruction software is integrated into this new study, this pixel selection method is not modified. However, this method is not suitable for partial ellipse cases. This part of the algorithm should be changed with another pixel selection method, so that the configuration reconstruction of the partial target can be achieved.

A Windows based software environment is developed for the implementation of the target recognition algorithm. The process time for the whole target recognition algorithm for an IBM compatible AMD Athlon 1800+ PC is between 10-35 seconds depending on the complexity of the scene.

The target recognition algorithm is integrated to the 3-D configuration reconstruction algorithm in a user friendly computer platform so that a complete vision system to detect the configuration parameters of a rigid body in an automated environment is obtained.

A method compatible with the target recognition algorithm developed in this study is also proposed for the internal camera calibration, which can be used in an unstructured environment.

6.3 Guidelines for Future Work

The following list contains some practical points that may be focused on in future studies for further improvements.

- The experimental implementation and performance tests of the internal camera calibration method suggested in this study may be investigated for exploring its possible use in practical applications.

- Optimization on the whole external camera and internal camera parameters may be performed in order to avoid some other unbiased errors on these parameters.

- The pixel selection method that is used for the 3-D configuration reconstruction should be changed with the one that is suitable to the partial ellipse cases.

- The resolution of the algorithm may be improved by embedding sub-pixel techniques in the algorithm. A finer digital representation is important in the determination of distortion parameters, as well as in the external camera calibration step.

- The target recognition algorithm developed in this study may be used in motion tracking. However, the solution speed should be optimized for this application by using features of previously detected frames.

REFERENCES

[1] **Schalkoff, R., (1992)**, “Pattern Recognition: Statistical, Structural and Neural Approaches”, John Wiley & Sons, New York.

[2] **Duda, R. and Hart, P., (1973)**, “Pattern Classification and Scene Analysis”, John Wiley & Sons, New York.

[3] **Tsai, R. Y., (1986)**, “A Versatile Camera Calibration Technique for High – Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses”, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, pp. 323-344.

[4] **Heikkila, J., (2000)**, “Geometric Camera Calibration Using Circular Control Points”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, pp. 1066-1077.

[5] **Horn, B., (1986)**, “Robot Vision”, The MIT Press, Cambridge, Massachusetts.

[6] **Kılınç, İ. A., (1994)**, “Determination of the Position and Orientation of Rigid Bodies by Using Single Camera Images”, M.S. Thesis, Mechanical Engineering Department, Middle East Technical University, Ankara, Turkey.

[7] **Acar, E. U., (1995)**, “Experimental Investigation and Implementation of a 3D Configuration Reconstruction Algorithm for an Object Using a Single Camera Image”, M.S. Thesis, Mechanical Engineering Department, Middle East Technical University, Ankara, Turkey.

[8] **Chatterjee, C. and Roychowdhury, V. P., (2000)**, “Algorithms for Coplanar Camera Calibration”, Machine Vision and Applications, Vol. 12, pp. 84-97.

[9] **Han, M. and Rhee, S., (1992)**, “Camera Calibration for Three Dimensional Measurement”, Pattern Recognition, Vol. 25, No. 2, pp. 155-164.

[10] **Zhang, Z., (1998)**, “A Flexible New Technique for Camera Calibration”, Technical Report MSR-TR-98-71, Microsoft Research.

[11] **Ahn, S., Rauh, W., and Kim, S., (2001)**, “Circular Coded Target for Automation of Optical 3-D Measurement and Camera Calibration”, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, No. 6, pp. 905-919.

[12] **Özdemir, K., (1996)**, ‘’Estimation of Mobile Robot Location by Using a Single CCD Camera”, M.S. Thesis, Mechanical Engineering Department, Middle East Technical University, Ankara, Turkey.

[13] **Çetin, O. L., (2000)**, “Improvement of an Image Processing System for Autonomous Mobile Robot Navigation”, M.S. Thesis, Mechanical Engineering Department, Middle East Technical University, Ankara, Turkey.

[14] **Olgac, N., Gan, Z., and Platin, B. E., (1989)**, “3D Reconstruction of Object Configurations by Hybrid Projection Analysis Using a Single Camera Image”, Proceedings of the 1st National Conference on Applied Mechanisms and Robotics, Vol.1, Paper No. 89AMR-4C-6, November, Cincinnati, USA.

[15] **Platin, B. E., Gan, Z., and Olgac, N., (1990)**, “3D Object Configuration Sensor Utilizing Single Camera Images”, ASME Paper No. 90-WA/DSC-22, presented at the Winter Annual Meeting, November 25-30, Dallas, USA.

[16] **Olgac, N., Craig, P. D., and Platin, B. E., (1991)**, “Advancements in Utilizing Monocular Object Configuration Sensor, Part II”, Advances in Instrumentation, DSC-Vol. 30, ASME Book No. H00691, pp. 21-24, also presented at the Winter Annual Meeting, December 1-6, Atlanta, USA.

[17] Data Translation Catalogue, DT3851 User Manual by Data Translation Company.

[18] **Özkılıç, S. and Platin, B. E., (2002)**, "Elliptical Target Recognition for Automation of a 3-D Configuration Reconstruction Algorithm of an Object Using a Single Camera Image", Proceedings of the 6th International Conference on Mechatronic Design and Modeling, September 2002, Cappadocia, Turkey.

[19] **Shapiro, L. G. and Haralick, R. M., (1993)**, Computer and Robot Vision, Vol. 1, Addison -Wesley Publications, Boston.

[20] **Shaoo, P., Wilkins C., and Yeager, J., (1997)**, "Threshold Selection Using Reyni's Entropy", Pattern Recognition, Vol. 30, pp. 71-84.

[21] **Cheng, J. C. and Don, H. S., (1992)**, "Segmentation of Bilevel Images Using Mathematical Morphology", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 6, pp. 595-628.

[22] **Kittler, J. and Illingworth, J., (1986)**, "Minimum Error Thresholding", Pattern Recognition, Vol. 19, pp. 41- 47.

[23] **Lei, Y. and Wong, K. C., (1998)**, "Ellipse Detection Based on Symmetry", Pattern Recognition Letters, Vol. 20, pp. 41-47.

[24] **Sewisy, A. A. and Leberl, F., (2001)**, "Detection Ellipses by Finding the Lines of Symmetry in the Images via an Hough Transform Applied to Straight Lines", Image and Vision Computing, Vol. 19, pp. 857-866.

[25] **Yip, R. K. K., Tam, P. K. S., and Leung, D. N. K., (1992)**, "Modification of Hough Transform for Circles and Ellipses Detection Using a 2-D Array", Pattern Recognition, Vol. 25, pp. 1007-1022.

[26] **Guil, N. and Zapata, E. L., (1996)**, “Lower Order Circle and Ellipse Hough Transform”, Pattern Recognition, Vol. 30, pp. 1729-1744.

[27] **Yoo, J. H. and Sethi, I. K., (1993)**, “An Ellipse Detection Method from the Polar and Pole Definition of Conics”, Pattern Recognition, Vol. 26, pp. 307-315.

[28] **McLaughlin, R. A., (1997)**, "Randomized Hough Transform: Improved Ellipse Detection with Comparison”, Pattern Recognition Letters, Vol. 19, pp. 299-305.

[29] **Bennett, N., Burridge, R., and Saito, N., (1999)**, “A Method to Detect and Characterize Ellipses Using the Hough Transform”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, pp. 652-657.

[30] **Fitzgibbon, A. W., Pilu, M., and Fisher, B. F., (1996)**, “Direct Least Squares Fitting of Ellipses”, International Conference on Pattern Recognition, Vienna, August 1996, pp. 253-257.

[31] **Halir, R. and Flusser, J., (1998)**, “Numerically Stable Least Squares Fitting of Ellipses”, Proceedings of the International Conference in Central Europe on Computer Graphics and Visualization. WSCG '98. Plzen, Czech Republic, 1998, pp. 125-132.

[32] **Jain, R., Kastun, R**., and **Schunck, B. G., (1995**), Machine Vision, McGraw – Hill, New York.

[33] **Stein, G. P., (1993)**, “Internal Camera Calibration using Rotation and Geometric Shapes”, M.S. Thesis in Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, Cambridge, MA, USA.

[34] **Lenz, R. and Tsai, R., (1988)**, "Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machine Vision Methodology", IEEE Transactions on Pattern Analysis and Machine Vision Intelligence, Vol. 10, No. 5, September 1988.

[35] **Devernay, F. and Faugeras, O., (2001)**, “Straight Lines Have to be Straight”, Machine Vision and Applications, Vol. 13, pp. 14-24.

[36] **Prescott, B. and McLean, G. F., (1997)**, “Line-Based Correction of Radial Distortion”, Graphical Models and Image Processing, Vol. 59, January, pp. 39-47.

[37] **Heikkila, J. and Silven, O., (1997)**, “A Four Step Camera Calibration Procedure and Implicit Image Correction”, IEEE Computer Vision and Pattern Recognition (CVPR’97), San Juan, Puerto Rico, pp. 1106-1112.

[38] **Davies, E. R.**, **(1997)**, Machine Vision, Theory, Algorithms, Practicalities, 2nd Edition, Academic Press, San Diego.

APPENDIX A

SOME IMAGE PROCESSING METHODS

A.1 Edge Detection

Edge detection is a contour based segmentation method. In the first part of the thesis study, a binarization is performed before the edge detection. The important problem in both edge detection and binarization is to determine a suitable threshold, especially for autonomous cases. For the binarization part, a histogram modeling gives the necessary and sufficient information in order to determine the threshold value autonomously. Once binarized, the image has step edges, and an edge detection is easier than a gray-scale image.

Basic Theory of Edge Detection

There are two types of edge detection methods, namely, Template Matching (TM) and Differential Gradient (DG) [38]. In either case, the aim is to determine where the intensity gradient magnitude, g, is sufficiently large to be taken as a reliable indicator of an object. Both TM and DG operators estimate local intensity gradients with the aid of suitable convolution masks. In case of a DG type operator, only two such masks are required for x and y directions. For a TM case, it is usual to employ up to 12 convolution masks capable of estimating local components of the gradient in different directions. In the TM approach, the magnitude of a local edge gradient (in short "edge magnitude") is approximated by taking the maximum of the responses for the component mask as

$$g = \max (g_i : i=1 \text{ to } n) \text{ where usually n is 8 or 12.} \tag{A.1}$$

In the DG approach, the local edge magnitude may be computed vectorially using the following nonlinear transformation:

$$g = (g_x^2+g_y^2)^{1/2} \tag{A.2}$$

where g_x and g_y are gradients in x and y directions, respectively.

In order to save the computational effort, it is a common practice to approximate this formula by using one of the following simpler methods

$$g = |g_x| + |g_y| \tag{A.3}$$

or

$$g = \max (| g_x| , |g_y|) \tag{A.4}$$

both of which gives equally accurate results for Equation (A.2), on the average.

In the TM approach, the edge orientation is estimated simply as that of the mask giving rise to the largest value of gradient in Equation (A.1). In the DG approach, it is estimated vectorially by a more complex equation as

$$\theta = \arctan(g_y/g_x) \tag{A.5}$$

In general, there are two types of gradient operators [32]. First group takes the first derivative into account. If it is above a threshold value, the presence of an edge point is assumed. Second approach determines only those points that have local maxima in gradient values and considers them as edge points. This means that, at edge points, there will be a peak in the first derivative, and equivalently, there will be a zero crossing in the second derivative. Thus, edge points may be detected by finding zero crossings of the second derivative of the image intensity. The Laplacian and second directional derivative operators may not be used in machine vision since any operator involving two successive derivatives is affected by noise more than an operator involving a single derivative. Even very small local peaks in the first derivative will result in zero crossings in the second derivative. In order to avoid effects of noise, some filtering methods must be used.

Since there are step edges after the binarization, a first derivative operator is found to be sufficient in this study. There are various first derivative operator masks with changing number of elements and weights. Some of them are as follows:

Roberts Operator:

R_x=

1	0
0	-1

R_y=

0	-1
1	0

Since the operator is a 2x2 gradient operator, the differences are computed at the interpolated point [i+1/2, j+1/2]. The Roberts operator is an approximation to the continuous gradient at that point and not at the point [i,j] as might be expected.

Sobel Operator:

In order to avoid having the gradient calculated about an interpolated point between pixels, the following 3x3 mask called as the Sobel operator is used.

S_x =

-1	0	1
-2	0	2
-1	0	1

S_y =

1	2	1
0	0	0
-1	-2	-1

Note that, this operator places more emphasis on pixels that are closer to the center of the mask.

<u>Prewitt Operator:</u>

$$P_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

$$P_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Unlike Sobel operator, this operator does not place any emphasis on pixels that are closer to the center of the mask.

A.2 Edge Thinning

Thinning is an iterative neighborhood operation, which generates a skeletal representation of an object [38]. Thinning can be viewed as a logical neighborhood operation where object pixels are removed from an image. Obviously, this removal must be constrained. These constraints can be summarized as follows:

(a) The pixel must lie on the border of the object. This implies that it has at least one background pixel with 4-connected neighboring pixels.

(b) The deletion of a pixel should not destroy the object's connectedness; i.e., the number of skeletons after thinning should be the same as the number of objects in the image before thinning.

(c) The algorithm should preserve the object's length.

A.3 General Hough Transform Technique

The Hough transform algorithm requires an accumulator array whose dimension corresponds to the number of unknown parameters in the equation of family of curves being sought [19]. In case of ellipse, there are five parameters to be determined which are its center coordinates, its orientation, and the lengths of its major and minor axes.

The equation of an ellipse in polar form is given as

$$r^2 \cos(\psi - \phi)/a^2 + r^2 \sin(\psi - \phi)/b^2 = 1 \tag{A.6}$$

where the radial and angular positions of a point on the ellipse are

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{A.7}$$

$$\psi = \arctan((y - y_0)^2 /(x - x_0)^2) \tag{A.8}$$

and

a, b : lengths of the major and minor axes of the ellipse
ϕ : orientation of the major axis
x, y : coordinates of any point on the ellipse
x_0,y_0: coordinates of the ellipse center

In the standard Hough transform (STH) technique, using every point in the image the parameters (r,Ψ) are calculated for each edge point in the image. Using the specified range for the orientation θ (it may be between angles 0-π, since the ellipse has a symmetrical shape), and major and minor axes lengths (they may be at most equal to the diagonal length of the ellipse), a five-dimensional accumulator array is obtained. The content of this array A (x_0,y_0, θ, a, b) is incremented by one, when an edge pixel P(x,y) justifies the equation of the ellipse.

The number of elements in the accumulator space is given as

$$R_{\theta}.R_{x0}.R_{y0}.R_{a}.R_{b} \tag{A.9}$$

where R_k means the number of elements for the variable k in its specified range.

After completing the accumulation, peaks are selected as the parameters of the ellipses in the image.

APPENDIX B

COMPUTER PROGRAM DEVELOPED

B.1 General Features of the Software

The computer program developed in this study aims to implement the techniques and algorithms proposed in this thesis in order to recognize the image of a circular target and extract its features automatically. There should also be options available for the user to execute various steps of the process such as binarization, edge detection, edge thinning, ellipse detection, and contour detection, one by one.

In addition to the image processing menu, there is a camera calibration menu in the program. It is also capable of determining external camera calibration parameters and the lens distortion parameters. A computer program written by E. U. Acar compiled in DOS environment was already available developed during the implementation of the 3-D configuration reconstruction algorithm [7]. In this study, this program is modified and compiled in Windows environment and added to the interface.

It is also aimed to generate a user friendly program complied in the Borland C++ Builder Professional compiler. The interface has four menus; namely **[File]**, **[Image Processing]**, **[Camera Calibration]**, and **[Help]**. Since the main aim of the computer program is to perform an automatic 3-D configuration algorithm, in addition to these menus, there is an individual button for the automatic operation of the algorithm.

B.2 User's Manual

In this section, menus and submenus of the program are explained, some visual examples are presented and subroutines used in the software are given.

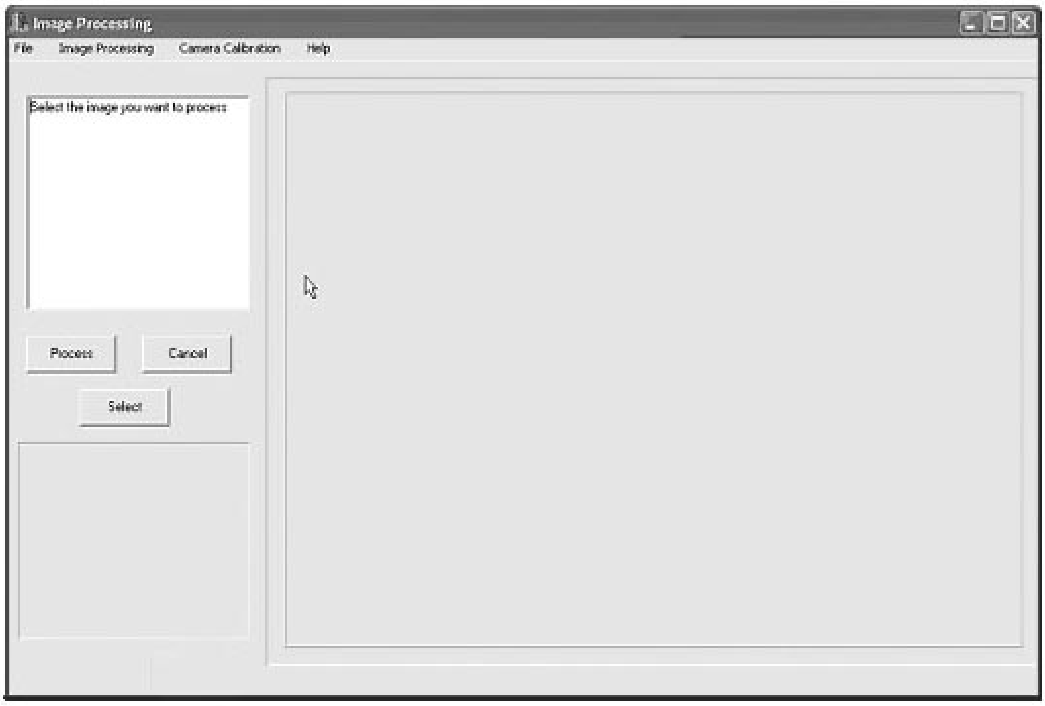


Figure B.1   Opening window of the software interface

When the program is executed, the opening window (Figure B.1) is initialized. Menus **[File]**, **[Image Processing]**, **[Camera Calibration]**, and **[Help]** can be seen on the top of the window. On the top left, a text window exists, which is used to display some text information by the software at different steps of the computation. The aim of this text window is to report the current status of the computation as well as to display some intermediate results to the user. The small image window below this text window is used to show the histogram of a selected part of an image. The large area on the right hand side is the main image window, on which images are viewed. The size of this area is 640 pixels horizontal x 480 pixels vertical.

Menus and the submenus of the software interface are as follows.

[File] This menu has three submenus, namely, **[Open]**, **[Save as]**, and **[Exit]**.

[File] [Open] When this option is selected, a dialog box opens (Figure B.2). First, from the “Files of Type” option on the dialog box, the type of the image file is selected among two options, namely, the raw data format with an extension “img” and the bitmap format with an extension “bmp”. Only those files with extensions matching the selected file type will be displayed. After selecting an image file in any directory in the computer, the “Open” button is pressed to load the content of that file to the software. By pressing the “Cancel” button, one can exit from the dialog box without making any selection.

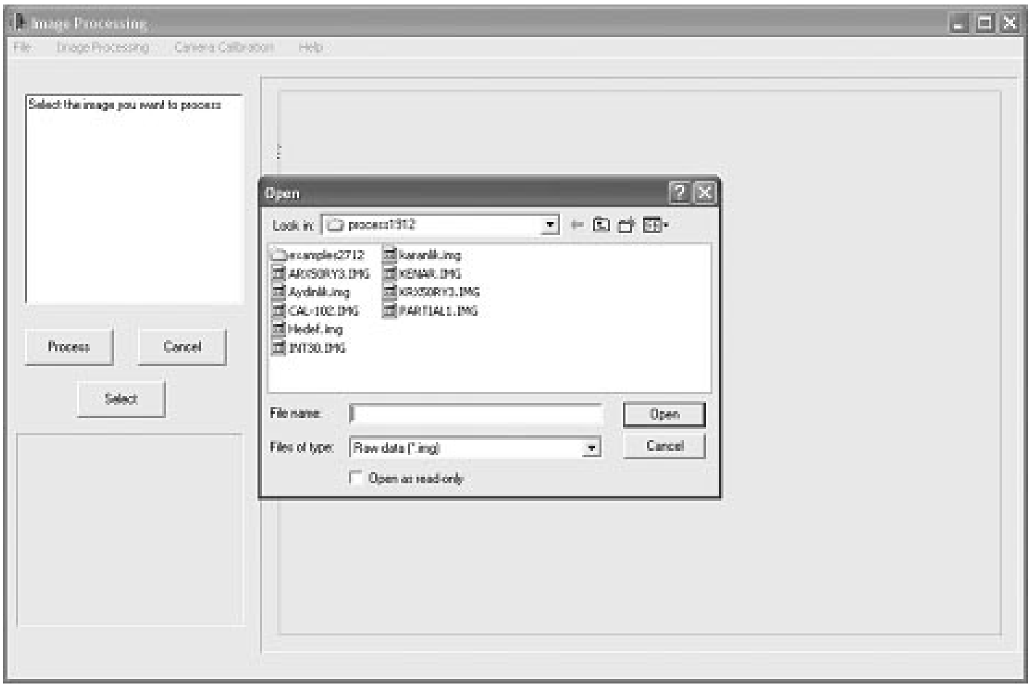


Figure B.2 **[File] [Open]** submenu of the software interface

[File] [Save as] In order to save an image seen in the main image window, this option must be used. When this option is selected, a dialog box opens (Figure B.3) by which images can be saved in “bmp” format. An existing file may be selected from any directories to replace or a new file name may be written in the "File name" box with the ".bmp" extension. After this, the "Save" button is pressed in order to save this file. If one wants to close the dialog box without saving a file, the "Cancel" button must be pressed.

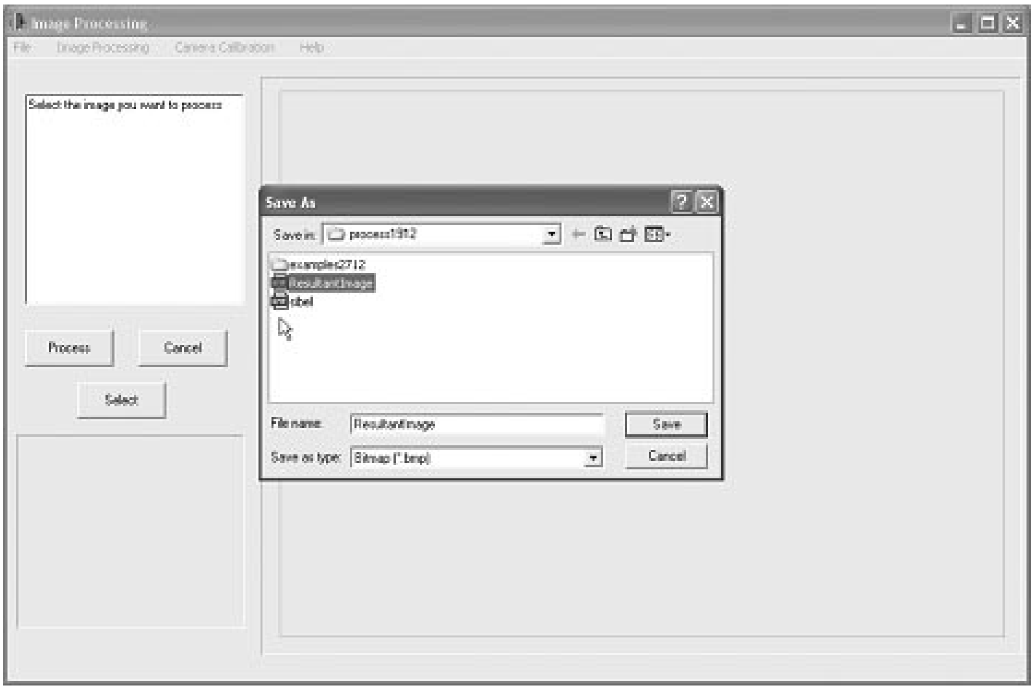


Figure B.3 **[File] [Save as]** submenu of the software interface

[File] [Exit] If the user wants to leave the program, this option must be selected. When it is selected, the interface window will be closed.

[Image Processing] This menu has six submenus, namely, **[Target Recognition]**, **[Stepwise Image Processing]**, **[Nearest Neighbor Algorithm]**, **[Quality Factor Calculation]**, **[Connectivity]**, and **[Histogram]**.

[Image Processing] [Target Recognition] When this submenu is selected, the computer program takes the raw image data, automatically recognizes the target area and segments it. First, the image file selected is opened; the raw image is displayed in the main image display box, followed by resultant images after binarization, edge detection and edge thinning procedures in sequence. Concurrently, the user is informed through the text box about the status of these procedures. At the end, the segmented binary target is shown on the left upper corner of the image area (Figure B.4a-B.4b).

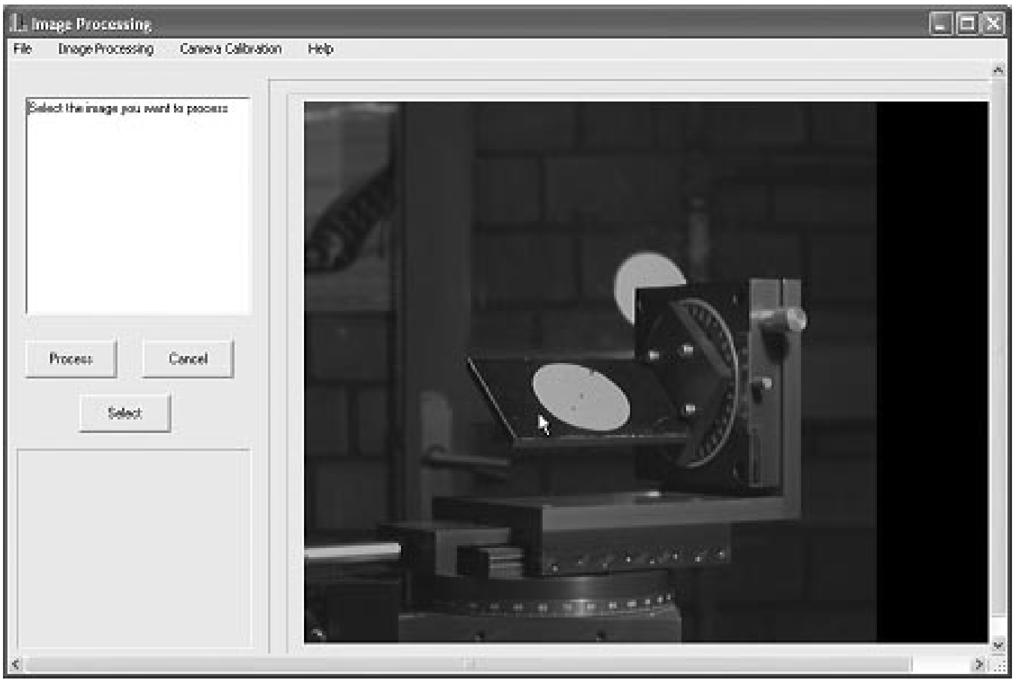


Figure B.4a **[Image Processing] [Target Recognition]** submenu of the software interface – initial screen-shot

Figure B.4b **[Image Processing] [Target Recognition]** submenu of the software interface - final screen-shot

[Image Processing] [Stepwise Image Processing] When this submenu is selected, a secondary submenu is opened, which includes four procedures, namely, **[Binarization]**, **[Edge Detection]**, **[Edge Thinning]**, and **[Contour Detection]**.

[Image Processing] [Stepwise Image Processing] [Binarization]
When one wants to binarize a raw data image, this submenu is used. First a raw data image is opened and this submenu is selected. The gray-level image is binarized with the global binarization. The resulting binarized image is viewed in the main image box and the threshold calculated for the image is written in the text box.

[Image Processing] [Stepwise Image Processing] [Edge Detection]
This submenu is to detect the edges of a gray-level image. First a gray-level image is opened and this submenu is selected. The gray-level image is first binarized and then an edge detection is performed. In the main image box,

the binarized and the edge detected images are shown consequtively.

[Image Processing] [Stepwise Image Processing] [Edge Thinning]
This option gets the gray-scale image as the input and after performing the binarization and the edge detection step, an edge thinning is carried out. In order to use this submenu, a gray-level image is opened and this submenu is selected. First the binarized image is shown and then the resulting edge detected and edge thinned images are viewed consecutively in the main image box.

[Image Processing] [Stepwise Image Processing] [Contour Detection]
This submenu is used to detect the contour of an object, which may be used in any other process. The input to this option is an edge-thinned image; so first the edge thinning is performed on the gray-level image by using the **[Edge Thinning]** submenu. Afterwards, a point inside the interested contour is assigned with mouse, then the “Contour Detection” option is selected. The algorithm detects the outer contour and paints the pixels on the contour in blue color.

[Image Processing] [Nearest Neighbor Algorithm] This submenu is used to detect nearest neighbors of an object point, which may be used in any other process. The input to this option is an edge-detected image. First a gray-level image is opened and the edge detection is performed using the **[Edge Detection]** submenu. An object point is marked with the cursor on the image. Brightness value of the selected image point is written in the text box. If it is an object point, the brightness is written as 255, or if it is a background pixel, the brightness value is written as 0. When the **[Nearest Neighbor Algorithm]** submenu is selected, this algorithm determines and paints the nearest neighbor object pixels in red color.

[Image Processing] [Quality Factor Calculation] The input to this procedure is a binary image. First, a gray-level image is binarized with the using **[Binarization]** submenu or a binary bitmap image is used. When this option is selected, the image is segmented into its connected regions and the ellipse-fitting algorithm is applied. The edge-thinned image of a gray-scale image is shown in the main image box with twenty colored points on it. These are points with highest scores in the accumulation space. The one with the yellow color is the one with a highest score. Quality factor values of points are written in the text window (Figure B.5b).

[Image Processing] [Connectivity] The input to this algorithm is an edge detected or an edge thinned image. Before selecting this submenu, a gray-level image is opened and the **[Edge Detection] or [Edge Thinning]** submenu is selected. On the resultant image, an object pixel is assigned with the mouse on the image, and **[Connectivity]** option from the menu is selected. The algorithm determines object points, which are connected to the selected object point.

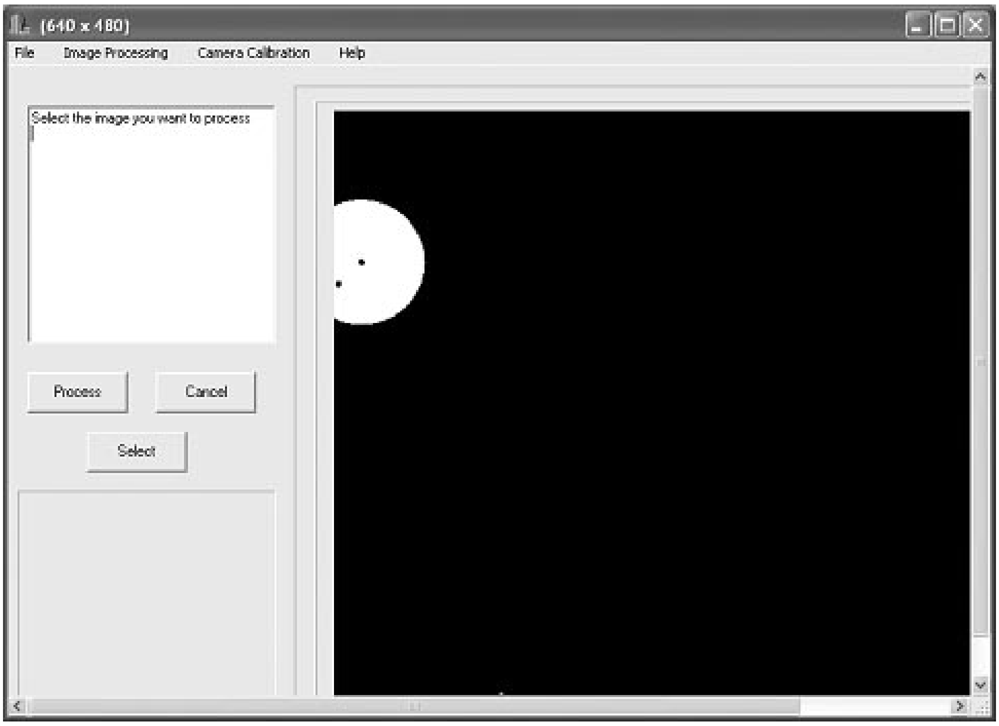


Figure B.5a **[Image Processing] [Quality Factor Calculation]** submenu of the software interface – initial screen-shot

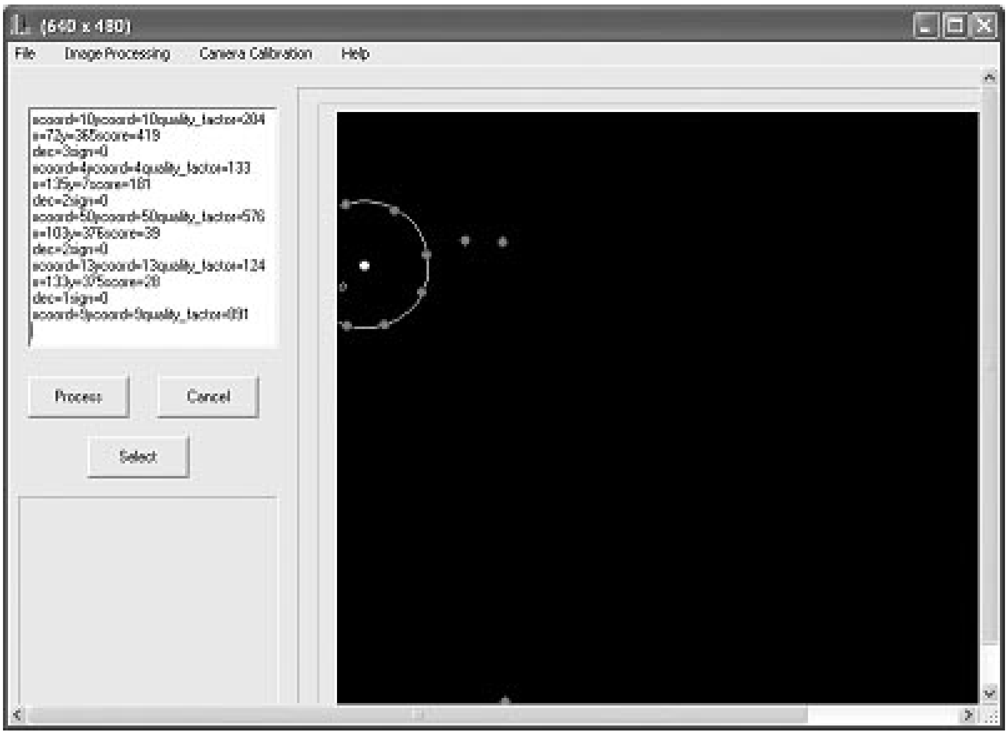


Figure B.5b **[Image Processing] [Quality Factor Calculation]** submenu of the software interface – final screen-shot

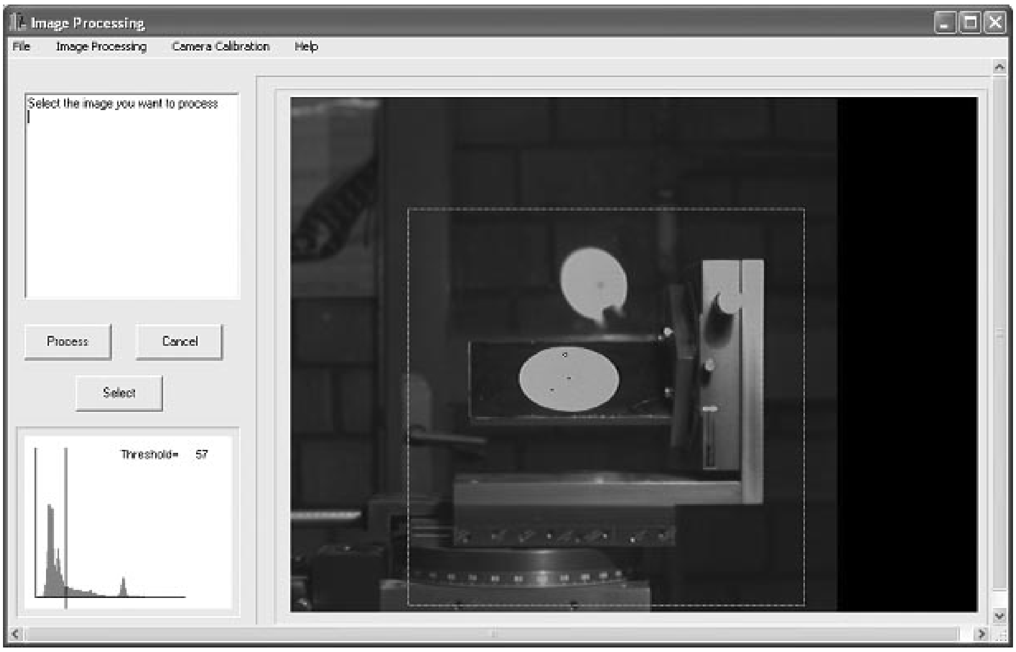


Figure B.6 **[Image Processing] [Histogram]** submenu of the software interface

[Image Processing] [Histogram] This submenu is used to see the histogram and threshold value of a portion of an image. First, a gray-level image is opened and a window is drawn around the interested image, or part of the image with the cursor. The “Select” button on the left of the image window is pressed, and then the **[Histogram]** submenu is selected from the menu. The histogram for the image is drawn on the small image window. The threshold is also calculated and shown on the histogram (Figure B.6).

[Camera Calibration] This menu has two submenus, namely, **[Distortion Parameters]** and **[External Camera Calibration]**.

[Camera Calibration] [Distortion Parameters] When this option is selected, a submenu is opened and the steps that will be used for the determination of the distortion parameters of lens are listed.

[Camera Calibration] [Distortion Parameters] [Pattern Define] After drawing a window around the sample pattern in the middle of the calibration pattern, the “Select” button is pushed and **[Pattern Define]** submenu is selected from the menu. It calculates external camera parameters of the pattern.

[Camera Calibration] [Distortion Parameters] [Area Selection] After drawing a window around the selected calibration area, the “Select” button is pushed and **[Area Selection]** submenu is selected. It calculates image coordinates of centroids of spots on the pattern image.

[Camera Calibration] [Distortion Parameters] [Parameter Detection] After performing the **[Pattern Define]** and the **[Area Selection]**, this option is selected. Distortion parameters are calculated and written in the text box.

[Camera Calibration] [External Camera Calibration] This submenu is used to determine external camera parameters of a circular target in the image.
Before using this option, first a gray-level image is opened and **[Target Recognition]** is selected. After the recognition is completed and the resultant

image with binary segmented ellipse area is shown in the main image box, **[External Camera Calibration]** submenu is selected. At the end of the process, the original gray-level image is with the calculated external parameters written in the corner is shown in the main image box.

One can perform the automatic 3-D configuration reconstruction continuously until the end of the determination of external parameters, by using the “Process” button on the left-hand side. After opening a gray-level image, the “Process” button is pressed. At the end of the process, the calculated 6 external parameters are shown on the corner of the main image box. In order to exit the process, the “Cancel” button is pressed.

[Help] When this menu is selected, a PDF document that includes explanations for methods used in the software and the user manual is viewed. By using links in the document, the user may get the necessary information about the program.

B3. Description of Software Routines

The main routine of the software is the file “imagemain1409.cpp”. Some important subroutines used in this study can be summarized as follows:

- struct coordlong2* **connectivity** (int** image, int hor, int ver, int* count): This routine is called from the main routine of the program, namely “imagemain1409”. Inputs to the routine are the image that is processed and horizontal and vertical coordinates of the selected point whose connected components are sought for. Another input to this routine is the address of an integer, on which the number of connected points will be assigned. Outputs of the routine are horizontal and vertical coordinates of connected points.

- int** **histogram** (int **image1, int r, int c, int* finalthreshold): This routine calculates the histogram array of the input “image1” and sends it to the “localthreshold” routine. The output is the binarized image, elements of which are 255 for object points and 0 for the background.

- int **localthreshold** (float *h): It is called from the "histogram" routine and calculates the threshold of the processed image using only its histogram. The "Minimum Thresholding Method" is used in this routine. It returns the threshold value of the image to the "histogram" routine.

- int** **edgedetect** (int **image): The input to this routine is a binarized image. Edge points on the image are detected by a Sobel operator. The output is a 2-D array, elements of which are 255 for edge points and 0 for the background. Also horizontal and vertical edge gradients of edge points are written to the created files 'Edgefilesx' and 'Edgefilesy'. These files are utilized in the ellipse fitting step.

- int** **thin** (int** image): The input to this routine is an edge detected image. The edge thinning is performed in this routine by deleting edge pixels according to the criteria given in (Appendix A.2). The output is a 2-D array, elements of which are 255 for edge points and 0 for the background.

- struct cell* **ellipsedetect** (int** image, struct coordlong2**, connected_pixels, struct cell* index, long edge_number): Inputs to this routine are the image of interest, connected components in the image, and the number of edge points. An ellipse fitting is performed in this routine using the "Bennett's Ellipse Fitting Approach". Elements of the accumulation space are stored in files. These files can be used in drawing 3-D graphs of the accumulation space. Scores in the accumulation space are sorted and local maxima are suppressed. The output of the routine is an array containing horizontal coordinates, vertical coordinates and scores of first ten points in the accumulation space.

- struct result_observe **quality_factor** (struct cell maxscore): This routine calculates the quality factor and the distance of symmetry for highest ten peaks in the accumulation space. It returns the output in an array of structures to the target recognition algorithm.

- struct coordinates2** **look_up_table** (double** position_orientation, coordinates2* pattern_centroid, int number_of_colomns, int number_of_rows): This routine is used in the internal camera calibration. It calculates undistorted coordinates of control points. Inputs to the routine are configuration parameters of the pattern calculated in the main routine, control points on the sample pattern, and numbers of control points in the horizontal and vertical directions in the calibration area.

- double* **distortion_parameters** (struct coordinates**actual_corners, struct coordinates **corner_points, int number_of_rows, int number_of_colomns): This routine is used in the internal camera calibration. It compares actual positions of control points and undistorted coordinates of these control points, and builds up an error matrix and some other matrices used in the solution of distortion parameters. Distorted coordinates of control points are calculated in the main routine. The subroutine for the pseudo inversion solution is called in this routine. The output is an array that contains four distortion parameters.