Object Distance Measurement Using a Single Camera for Robotic Applications

by

Peyman Alizadeh

A thesis Submitted in partial fulfillment
of the requirements for the degree of
Master of Applied Sciences (MASc) in Natural Resources
Engineering

The Faculty of Graduate Studies
Laurentian University
Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE

Laurentian Université/Université Laurentienne

Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis
Titre de la thèse — Object Distance Measurement Using a Single Camera for Robotic Applications

Name of Candidate
Nom du candidat — Alizadeh, Peyman

Degree
Diplôme — Master of Applied Science

Department/Program
Département/Programme — Natural Resources Engineering

Date of Defence
Date de la soutenance — September 2, 2014

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Meysar Zeinali
(Supervisor/Directeur de thèse)

Dr. Brahim Chebbi
(Committee member/Membre du comité)

Dr. Eduardo Galiano-Riveros
(Committee member/Membre du comité)

Dr. Mir Behrad Khamesee
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. David Lesbarrères
M. David Lesbarrères
Acting Dean, Faculty of Graduate Studies
Doyen intérimaire, Faculté des études supérieures

ABSTRACT

Visual servoing is defined as controlling robots by extracting data obtained from the vision system, such as the distance of an object with respect to a reference frame, or the length and width of the object. There are three image-based object distance measurement techniques: ***i***) using two cameras, i.e., stereovision; ***ii***) using a single camera, i.e., monovision; and ***iii***) time-of-flight camera.

The stereovision method uses two cameras to find the object’s depth and is highly accurate. However, it is costly compared to the monovision technique due to the higher computational burden and the cost of two cameras (rather than one) and related accessories. In addition, in stereovision, a larger number of images of the object need to be processed in real-time, and by increasing the distance of the object from cameras, the measurement accuracy decreases. In the time-of-flight distance measurement technique, distance information is obtained by measuring the total time for the light to transmit to and reflect from the object. The shortcoming of this technique is that it is difficult to separate the incoming signal, since it depends on many parameters such as the intensity of the reflected light, the intensity of the background light, and the dynamic range of the sensor. However, for applications such as rescue robot or object manipulation by a robot in a home and office environment, the high accuracy distance measurement provided by stereovision is not required. Instead, the monovision approach is attractive for some applications due to: ***i***) lower cost and lower computational burden; and ***ii***) lower complexity due to the use of only one camera.

Using a single camera for distance measurement, object detection and feature extraction (i.e., finding the length and width of an object) is not yet well researched and there are very few published works on the topic in the literature. Therefore, using this technique for real-world robotics applications requires more research and improvements.

This thesis mainly focuses on the development of object distance measurement and feature extraction algorithms using a single fixed camera and a single camera with variable pitch angle based on image processing techniques. As a result, two different improved and modified object distance measurement algorithms were proposed for cases where a camera is fixed at a given angle in the vertical plane and when it is rotating in a vertical plane. In the proposed algorithms, as a first step, the object distance and dimension such as length and width were obtained using existing image processing techniques. Since the results were not accurate due to lens distortion, noise, variable light intensity and other uncertainties such as deviation of the position of the object from the optical axes of camera, in the second step, the distance and dimension of the object obtained from existing techniques were modified in the X- and Y-directions and for the orientation of the object about the Z-axis in the object plane by using experimental data and identification techniques such as the least square method.

Extensive experimental results confirmed that the accuracy increased for measured distance from 9.4 mm to 2.95 mm, for length from 11.6 mm to 2.2 mm, and for width from 18.6 mm to 10.8 mm. In addition, the proposed algorithm is significantly improved with proposed corrections compared to existing methods. Furthermore, the improved distance measurement method is computationally efficient and can be used for real-time robotic application tasks such as pick and place and object manipulation in a home or office environment.

ACKNOWLEDGMENTS

I would like to take this opportunity to acknowledge and thank those who made this work possible. First of all, I would like to extend my utmost gratitude to my supervisor, Prof. Meysar Zeinali, School of Engineering (Mechanical Engineering Program) at Laurentian University, whose sincerity and encouragement I will never forget. Prof. Zeinali has been my inspiration as I hurdle all of the obstacles in the completion of this research work.

I would also like to express my sincerest appreciation to my family, my father (R.I.P) and mother, Abdolali and Leila, and to my brother and sister, Pedram and Zahra. Lastly, I thank my friends, especially H. Eivazy, O. Mahmoodi, B. Maraghechi, and M. Abolfazlzadeh, for their support during the dark and cold days.

Thanks again to all who helped me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$A \circ B$	Morphological opening
$A \bullet B$	Morphological closing
A'	The distance such that if the object is positioned there, the bottom of the object is seen in the lower part of the image.
$\boldsymbol{A_{in}}$	Matrix of the intrinsic parameters
a	Distance from the bottom of the camera on the ground to the beginning of the field of view (mm)
O	Object's size (mm)
O_x	Horizontal object dimension (mm)
O_y	Vertical object dimension (mm)
B'	The distance such that if the object is located at that position, the bottom of the object is seen in the image center.
{B}	Robot's base frame or link zero
$\boldsymbol{BA}$	Camera location with respect to the base frame
$\boldsymbol{BT}$	Distance of the tool frame with respect to the base frame
C_f	Calibration factor
C	Obtained coefficient from the distance equations
c	Skewness of two image axis
D_H	Horizontal distance from the camera to the object(s) on the ground
D_o	Oblique distance from the camera to the object(s)
D_1	The distance from image plane to the object plane (mm)
e_a	Average percentage error
$F_0(x_\mathrm{i},y_\mathrm{i})$	The center of visual field for the complex log mapping method

f	Focal length of the camera (mm)
$f(x,y)$	Light intensity function
$\{G\}$	Robot’s goal frame
$g(x,y)$	Resulting threshold image
h	Height of camera from the ground
I	Image height (mm)
I_p	Image height (pixel)
j	Distance in xy-plane from the beginning of the field of view (v = 0) to the center of the image
k	Horizontal distance in xy-plane from camera to the beginning of the field of view (v=0)
L*a*b, L*u*v	Typical color spaces used in MATLAB program
$L_{1,} L_2$	Base line lengths of the triangle
l	Horizontal distance in xy-plane and in X-direction from the image center to the beginning of the field of view (u = 0)
l_p	Length of the object (pixel)
$M_0 (m_i,n_i)$	Gray scale on the mapped pixel (m_i,n_i)
$\boldsymbol{M}\ (x,y,z)$	A point in global coordinate system
$\boldsymbol{M'}(x,y,z,1)$	Augmented vector of $\boldsymbol{M}$
$\boldsymbol{M}\ (u,v,\lambda)$	Projection of $\boldsymbol{M}$ on the image plane
$\boldsymbol{m'}(u,v,\lambda,1)$	Augmented vector of image pixel coordinate $\boldsymbol{m}$
(m_i,n_i)	Obtained mapped pixel using complex log mapping method
N_p	Sum of the pixels from the beginning of the field of view to the current position of the object in Y-direction
$\mathrm{N_{pmax}}$	The total pixels of the camera’s field of view in Y-direction
N	Duration of the impulse response of the FIR filter
N_1	Size of the mapped image using complex log mapping method
n_1	Distance between two points along the optical axis

P	Predicted distance obtained from the regression equation
P_x	Horizontal sensor size (pixel)
P_y	Vertical sensor size (pixel)
q	Image distance
$\boldsymbol{R}$	Rotation matrix
r	The mapping radius from the origin to the original image pixel
s	Scale factor
Q	Ratio of $I_{\mathrm{p}1}$ to $I_{\mathrm{p}2}$
S_x	Horizontal sensor size (mm)
S_y	Vertical sensor size (mm)
{S}	Robot’s station frame or universe frame
$\boldsymbol{T}$	Translation vector
$T\mathrm{h}$	Threshold value
t	The time taken for the signal to reflect back to the object
$\boldsymbol{TG}$	Distance of the moving object with respect to the tool frame
{T}	Robot’s tool frame
v_{s}	Velocity of the signal
v	y value of the image coordinate (u,v)
{W}	Robot’s wrist frame
w_{kl}	Filter coefficient
x	Horizontal image dimension (pixel)
$(x_{\mathrm{i}},y_{\mathrm{i}})$	Original pixel coordinate
$x\,(i, j)$	The pixel values at point (i, j) before the FIR filtering
X	True horizontal distance in object plane (mm)
X_0	Total pixels between the image bottom positions to the point that has the lowest y value in the image
x_{p}	Pixel change in X-direction

XYZ	Camera coordinate system
xyz	Global coordinate system
y	Vertical image dimension (pixel)
Y	True vertical distance in object plane (mm)
y_p	Pixel change in Y-direction
$y\ (i, j)$	The pixel values at point (i, j) after the FIR filtering
γ_r	Angle of the moving object from the beginning of the camera's field of view
z	Mapping distance from origin to the original image pixel
β_1, β_2	Base line angles of the triangle
β_O	Object orientation (degree)
δ	Vertical tilt angle of the camera (the camera pitch angle)
θ_1	Mapping direction from origin to the original image pixel
2θ	Camera's field of view in X- direction
2α	Camera's field of view in Y- direction
2D FIR	Two-dimensional finite impulse response
$\oplus$	Dilation operator
$\ominus$	Erosion operator

NOMENCLATURE

ACF	Adaptive contour feature
AGV	Automated guided vehicle
BPN	Back projection neural network
CAD	Computer-aided design
CCD	Charge-couple device
CLM	Complex log mapping
CMOS	Complementary metal-oxide-semiconductor
DAS	Driver assistance system
FOV	Field of view
IBVS	Image-based visual servo
Open GL	Open graphic library
PBVS	Position-based visual servo
PnP	Perspective-n-point
RGB	Red, green, and blue light
SIFT	Scale-invariant feature transform
SURF	Speeded up robust features

Chapter 1

INTRODUCTION

One of the main tasks in robotic vision is to find the position and orientation of the objects surrounding the robot in 3D space, relative to the reference frame. Determining the camera's tilt angle in a vertical plane and the object-to-camera distance (the distance between the camera and the objects) is essential for localizing, navigating, and performing some high-level task planning. There are two common methods to calculate the object-to-camera distance [1]: ***i*)** using the object's given size and the camera's focal length; and ***ii*)** using the height of the camera and the point of contact where the object meets the ground. Unlike in the first method, the dimension of the object in the second method is unknown.

1.1 Background

For decades, researchers have been motivated to develop efficient techniques to transfer the computer vision capabilities to applications and products such as automotive safety, manufacturing, video surveillance and Visual Servoing (VS), which is an important robotics application. Visual servoing is a technique for controlling a robot's motion using feedback information sent from a vision sensor. In VS, it is difficult to track the moving object if data on object distance are not accessible [2].

There are two different types of visual servo control: position-based visual servo control (PBVS) and image-based visual servo control (IBVS). The reference inputs for PBVS are the 3D-relative position and the orientation between the object and the robot end-effector (i.e., robot’s hand) in Cartesian space. In image-based visual servo control, the reference input is the 2D object’s position that is obtained from the camera’s image plane. The IBVS approach fails where an accurate estimation of the object’s distance and motion is not available, especially in dynamic environments [3]. Where a camera is employed in the PBVS method, a small measurement error would propagate and significantly affect the servoing accuracy [3].

To calculate object distance for a moving object, the object must first be tracked by a camera. Object tracking can be classified into four main categories: model-based, appearance-based (region-based), feature-based, and contour-based methods.

Model-based tracking techniques require previous knowledge of the object’s shapes for the matching process in order to find the object in the scene and apply the exact geometrical models of the object. These techniques, however, have two shortcomings: ***i*)** an object that is not in the database cannot be recognized by these models; and ***ii*)** implementing these models is complicated and sometimes impossible [4].

Appearance-based methods track an object using the 2D shape of the connected region. This tracking approach relies on information that is provided by the entire region’s pixels. Motion, color, and texture are some examples of the information. Overall, appearance-based techniques are not robust with complex deformation [5].

Feature-based methods track the specific features of an object, such as points and edges, and have been developed in many applications. Some advantages of feature-based methods are their simplicity and stability for tracking the objects, but these methods are not efficient in real-time object tracking applications. Other shortcomings of feature-based techniques are their high cost and the lengthy time required when a large number of features occur in one object.

Contour-based methods track the contour (boundary) of the object rather than tracking all of the pixels that make up the object. In contour-based methods, the contour of the object in the next frame captured by the camera is determined using the motion information of the object. Thus, the shape and position of the object's contour in the next frame are improved to fit into the object. Furthermore, the object's motion information is updated by any changes in the contour location [4]. One shortcoming of contour-based methods is their failure to track objects that are partly occluded. Nevertheless, contour-based techniques have the following merits: ***i*)** reduced computational complexity compared to other methods; ***ii*)** the ability to track rigid and non-rigid objects.

To measure object distance for a moving object, several methods are presented in the literature. Zhang et al. [6] developed a 3-step algorithm to compute the 3D positions of a target object in a camera's coordinate frame. This method measures the distance between the object's feature (e.g., a point on the object) and the principal point (the central point in the image plane) based on the calculated area in the image. In the algorithm proposed by Zhang et al., the intrinsic camera's parameters are first calibrated. Then, a model is set up to measure the moving object-to-camera distance along the optical axis according to the mapping relationship between the objects in the camera's coordinates frame and its projection in the pixel coordinate frame. Finally, the absolute distance is calculated. In a method proposed by Coman and Balan [7], the application of object distance measurement can be a starting point for complex applications as long as the object's geometry remains square-shaped. Yamaguti et al. [8] calculated object surface distance using the ratio of two images taken by a monocular camera at two different locations and a complex log mapping method applied to the two images. However, the Yamaguti et al. model is very time-consuming and is only applicable to objects parallel to the camera's plane. To date, the development of an efficient distance measurement using a single camera has not been satisfactorily addressed in the literature and is an open problem.

1.2 Thesis Objectives

The objective of this thesis is to design and develop two different vision-based systems using a single fixed camera and a single camera with variable pitch angle for a dynamic environment in order to accomplish the following goals:

- **Detect and track the desired object**

  Object detection is required by every tracking methods to initialize a tracker and to detect the existence of known objects in a given image frame. However, in this thesis, object detection is described as detecting the existence of any object in the field of view that can be potentially manipulated. Since the main goal of the proposed algorithm is to analyze the movement of objects, object tracking is one of the system's essential parts. In the real world, an object can move freely in a 3D space with 6 degrees of freedom (DOF). Due to the complexity of the problem, we divide the tracking problem into two sub-problems: object point (closest point of the object to the camera) tracking, which is related to the object's movement in the xy-plane; and estimation of the object's orientation, which is the object's movement about the Z-direction.

- **Find the object distance**

Case 1 (Single fixed camera method): Since the obtained object distance includes horizontal and vertical errors, an improved image processing method (explained in section 4.2) is proposed in this thesis, based on the point feature extraction.

Case 2 (Single camera with variable pitch angle method): This method is based on the Taha and Jizat [49] approach (explained in section 3.9.2), which is versatile under illumination conditions. To overcome the problem of object distance measurement under varying illumination conditions, an improved version of the Taha and Jizat [49] approach using the least square method is proposed.

- **Calculate an object's dimension (e.g., length and width)**

To find an object's dimension, the major and minor axis of the ellipse (blob analysis) is corrected using orientation, horizontal and vertical error corrections, as explained in sections 5.4-6.

Note that it can be a valid assumption that the appearance of an object does not change drastically, regardless of the illumination condition and occlusion. This thesis is mainly concerned with analyzing and investigating the two vision-based systems in order to develop an accurate, fast, and efficient algorithm to measure object-to-camera distance. The experimental results of implementing the developed vision-based systems demonstrate their applicability in real-time robotic applications.

1.3 Thesis Contributions

The main contributions of this thesis are to develop two different object distance measurement algorithms that are:

- Accurate, reliable, and consistent;
- Able to simultaneously localize the objects using point feature extraction and then measure the object distance using a single fixed camera and a single camera with variable pitch angle;
- Able to measure the distance between the camera and moving objects anywhere in the camera's field of view; and
- Efficient in measuring an object's dimension of unknown shape and size.

1.4 Thesis Outline

The remainder of this thesis is organized as follows: in Chapter 2, a literature review is presented to reflect the background of object tracking, object distance calculation, and current visual servoing methods. Chapter 3 mainly discusses background and theory information on various computer vision topics, object distance measurements, and range finding techniques. Chapter 4 presents the object distance measurement method using a single fixed camera and provides the results and discussions for the proposed object distance measurements. Chapter 5 presents the object distance measurement method using a single camera with variable pitch angle and provides the simulation and experimental results for object distance, length and width measurements. Finally, Chapter 6 presents the conclusion and the contributions of the thesis, along with some recommendations for future work.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides an overview of the state of the art with a survey of existing techniques in object tracking, object distance measurements, and visual servoing. The main goal of this chapter is to give the reader a comprehensive knowledge of previous studies and technical information about the above-mentioned techniques. This chapter also classifies the techniques and then presents the strategies to efficiently tackle the challenges of object tracking, object distance measurements, and visual servoing.

2.2 Object Tracking

Object tracking is a challenging problem in robot vision due to various factors such as camera motion, occlusions, non-rigid object structures, and unanticipated rapid changes in both the object and the scene. Object tracking is relevant in the following topics [9]:

- Surveillance systems, for monitoring a behavior or changing information in order to detect unusual activities.
- Video indexing, for the retrieval and recovery of videos in databases.
- Traffic monitoring, for simultaneous traffic inspection to direct traffic flow.
- Vehicle navigation, for real-time path planning and obstacle avoidance capabilities in robotics.

2.2.1 Object Tracking Problems

Tracking a desired object can be a complex task, since estimating the trajectory of a moving object in the image plane has to be accurate. The main problems related to object tracking are as follows [9]:

- Information loss when projecting from a 3D world to a 2D image;
- Existence of noise in the images;
- Complex motion of the objects;
- Complex characteristics of the objects, such as non-rigid/articulated objects;
- Partial or full object occlusions;
- Complex shapes of the objects;
- Illumination variations in the scene;
- Real-time processing requirements.

2.2.2 Object Tracking Classifications

In a tracking strategy, the object is defined as anything that is of interest in the scene. Therefore, the object(s) could be: ***i*)** a point or a collection of points; ***ii*)** primitive geometric shape(s); ***iii*)** object silhouette and contours; ***iv*)** articulated shape model(s); and ***v*)** skeletal model(s) [9].

The point object is usually set as the centroid point of a target or a set of points on the target. Point object representation is especially important for tracking objects in small regions of the image. Primitive geometric shapes, usually characterized by a regular shape such as a rectangle or ellipse, are suitable for representation of simple rigid and non-rigid target objects. In the object silhouette and contour representations, the contour is defined as the boundary of the object that surrounds the region inside the contour, called the silhouette of the object. The object silhouette and contour are the best models for complex shape tracking. Articulated shape models are used to handle body parts that are connected with joints. The relationship between articulated shape models is established by kinematic motion models. Skeletal models can be extracted by applying a medial axis transform to the object silhouette and are mostly used for recognizing and modeling articulated and rigid objects. Figure 2.1 demonstrates some of the object representations for object tracking purposes.

2.2.3 Object Tracking Features

Some common features that are considered for object tracking are color, edges, optical flow, and texture [9].

Color: The color of an object depends on two factors: ***i*)** the spectral power distribution of the lighting; and ***ii*)** the surface reflectance of the object. Although three color spaces RGB (red, green, blue), *L*u*v*, and *L*a*b* are commonly used in image processing to represent an object's color, there is a variety of color spaces that can be used in object tracking.

Edges: The edge detection feature is able to identify strong variations in an image's intensities. The edge detection feature is less sensitive to illumination variation than the color feature. The canny edge detector is one of the most common edge detection methods.

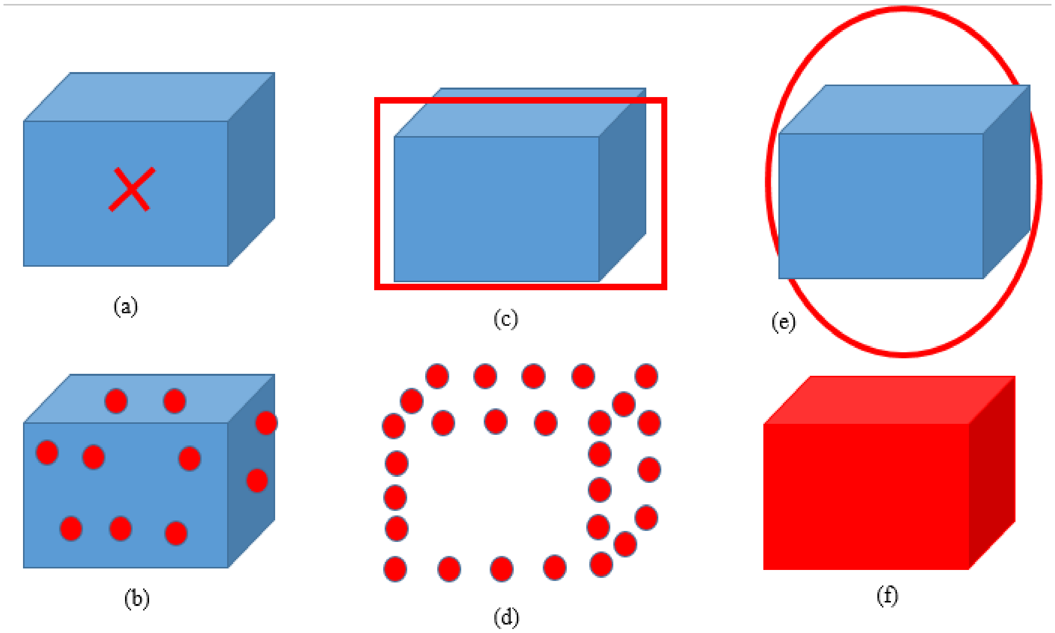


Figure 2.1: Various object representations: a) centroid b) set of points, c) rectangular, d) object contour, e) elliptical, and f) object silhouette [9].

Optical Flow: Optical flow is a field filled with vector displacements and represents the translation of each pixel in a region. Optical flow is computed by assuming the brightness constancy of the corresponding pixels in consecutive frames. Motion-based segmentation and motion-based tracking are two common applications in the optical flow feature.

Texture: The texture feature, as a measure of a surface's intensity variation, accounts for the smoothness and regularity of an object. It requires a processing step to generate descriptors. This feature is similar to the edge feature and is less sensitive to illumination variation compared to the color feature [9]. One of the most important and robust texture feature descriptors is the Gabor wavelet. Gabor filters can be viewed as orientation and scale invariance for edge and line detectors. Applying these features in a specific region is critical for characterizing underlying texture information [10].

Yang et al. [11] extended the work of Yilmaz et al. [9] in the visual tracking field by describing the characteristics of the feature descriptors for visual tracking and summarizing the most recent advances in online learning-based tracking methods.

Selecting the most appropriate feature descriptors (e.g., uniqueness) for visual tracking is a challenging task. This difficulty has motivated investigators to look for image features other than texture, edges, color, etc.

Gradient features: Gradient features are mostly practiced in human detection, where the shape or contour represents a human body (object). Another category of the gradient-based method is statistical summarization of gradients, for which there are numerous descriptor schemes, such as the Scale-invariant feature transform (SIFT) descriptor, Speeded up robust features (SURF), and Adaptive Contour Features (ACF) [11].

2.2.4 Object Tracking Methods

Traditional template-based tracking algorithms are divided into offline and online classes. Offline approaches are based on using similar visual examples or learning during the first few frames. The two main shortcomings of offline methods are:

***i*)** once the model is created, it cannot be updated; and

***ii*)** tracking may fail due to changes in the object’s shape.

The second type of tracking algorithms, online tracking, employs online learning techniques to learn about the object’s changes during the tracking period. The first online tracking technique was introduced by Jepson et al. [12], in which a combination of three components (stable, transient, and noise components) were proposed for the representation of the object. Another online tracking method was developed by Matthews et al. [13], comprised of a template update algorithm that can be updated in every frame.

The Matthews et al. [13] algorithm does not suffer from "drifts", which is a small error initiated in the location of a template whenever it gets updated in each frame. These small errors accumulate and then trigger the model to gradually drift away from the object. Figure 2.2 shows a comparison between the three update strategies [13]. In strategy 1 of Figure 2.2, the template is not updated properly and the object tracking fails. In strategy 2, the template is updated in every frame, but the template drifts away from the object. In strategy 3, the template is updated in every frame since a "drift correction" is added and the object is tracked and the template updated appropriately.

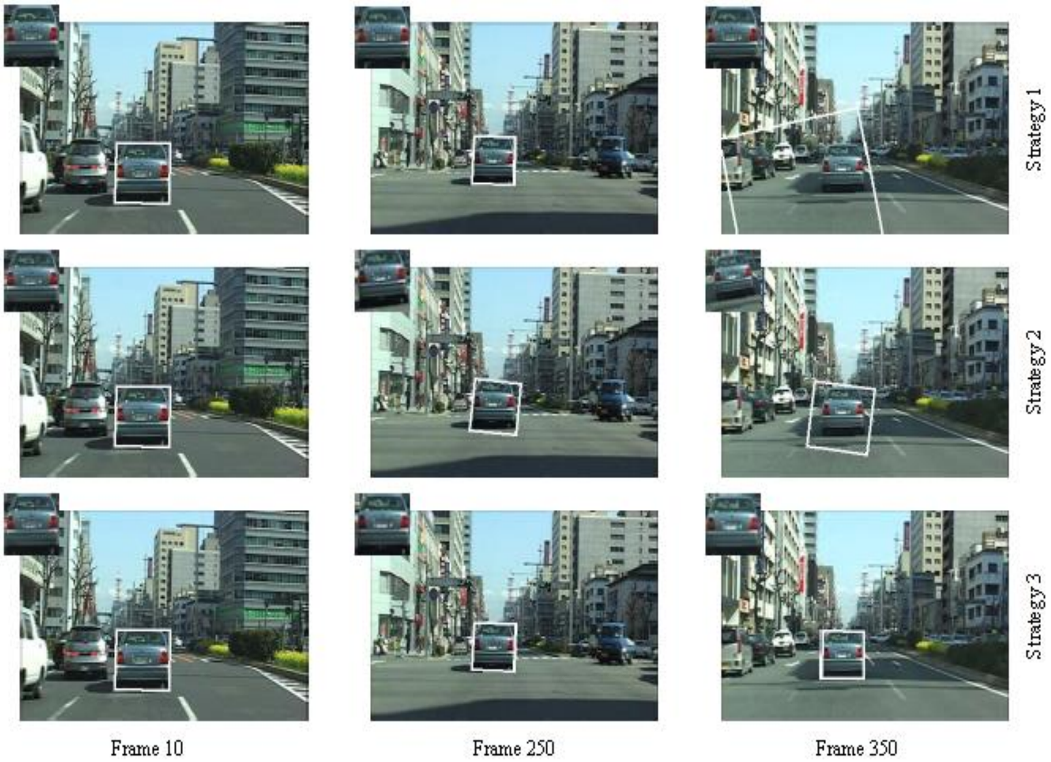


Figure 2.2: Comparison of different update tracking algorithm methods [13].

2.3 Distance Measurement

Vision-based pose estimation and camera tracking are two prominent areas in robotic applications such as localization, positioning tasks, and navigation. The main function of vision-based pose estimation is to estimate the position and orientation of the camera and object using a set of *n* feature points from the object coordinates was well as the resulting 2D projections from the camera coordinates. The vision-based pose estimation, which is a perspective-n-point (PnP) problem [14], is one of the critical problems in photogrammetry. It can be practiced in many applications, such as computer vision, robotics, augmented reality, etc. [14]. The two methods currently in use for solving the vision-based pose estimation problem are non-iterative and iterative algorithms. In non-iterative algorithms, linear models are applied to obtain algebraic solutions, whereas in iterative methods, the pose estimation is formulated as a nonlinear least-squares problem with constraints. If the rotation matrix is assumed to be orthogonal, this problem can be solved using a nonlinear optimization algorithm such as the Levenberg-Marquardt method [14].

2.3.1 Distance Measurement Using a Single Fixed Camera

When objects are being tracked by a camera, it is essential to determine their position and orientation with respect to the camera in order for the robot to navigate the object or for end-effector to do a task.

The accuracy in calculating the position and orientation of remote objects is a critical issue in robotic vision. Tao et al. [15] proposed a new monocular vision method to find the pose estimation of remote objects through translation and rotation matrices using image feature extraction and data optimization. To reduce environmental light variations and achieve a better contrast between target and background, Tao et al. [15] used near-infrared light as the light source. Tao et al. [15] also designed a new feature-circle-based calibration drone to accomplish automatic camera calibration. The results of Tao et al.'s [15] experiments demonstrated less than 8″ and less than 0.02 mm in the repeatability precision of angles and the repeatability precision for displacement, respectively. Figure 2.3 displays Tao et al.'s [15] vision system, which is reportedly already being used in a wheel alignment system.

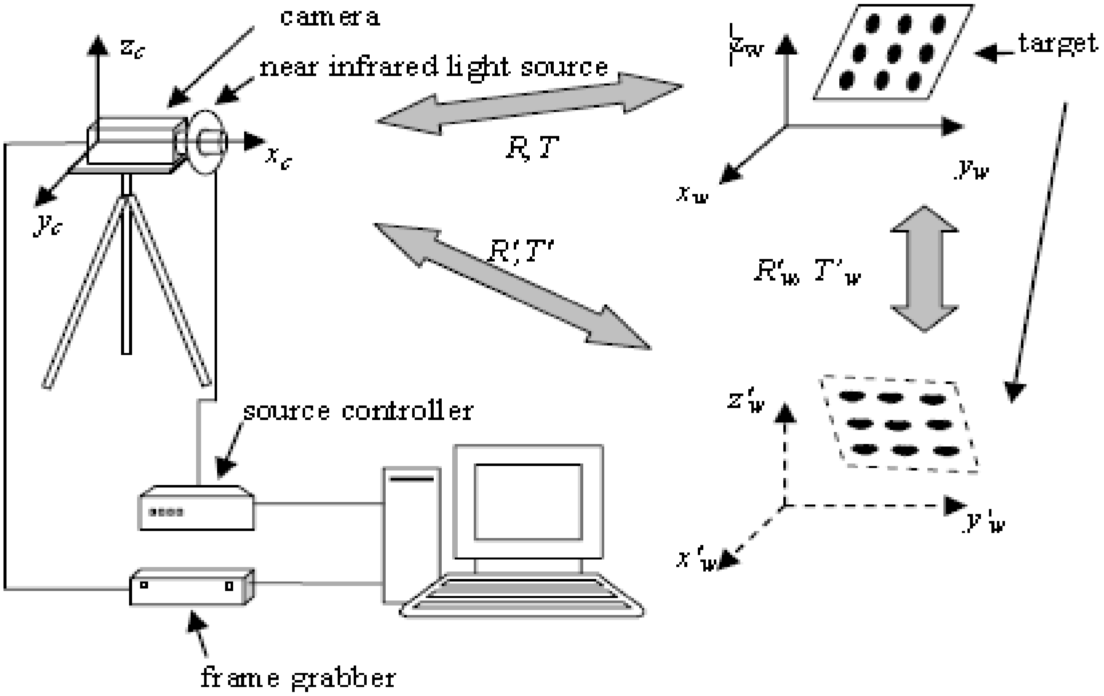


Figure 2.3: Diagram of the Tao et al. measurement system [15].

Some researchers, such as Krishnan et al. [16], proposed an object-to-camera distance based on a complex log mapping method. This method has the advantage of measuring the distance between the camera and an object's surface with an arbitrary texture pattern. The idea behind this technique is to use two images taken at two different camera positions to measure the object-to-camera distance. The object-to-camera distance, in this technique, is calculated through the ratio between the object's sizes projected on the two images that are moved on the camera's optical axis.

Calculating the distance of the object to the defocused image has recently drawn the attention of some researchers. Objects positioned at a particular distance from any optical system are called focused, while other objects are considered defocused or blurred. Subbarao[17] proposed a new method that uses the defocus or blur information in the images that are captured by an optical system such as a convex lens. The objects of interest in the Subbarao [17] method are categorized into bright points, lines, step edges, blobs, stripes, and smooth edges. This method is accurate for close isolated objects, with the presence of other objects nearby having an effect on depth estimation. The two main advantages of this method are:

***i*)** there is no restriction on the form of the point spread function of the camera system; and

***ii*)** this method can generally define the distance of a simple object by measuring the degree of image blurriness.

The majority of the known methods' procedures for pose estimation are based on image information such as intensity, edge, and absolute depth values. Barrois and Wöhler [18] proposed a 3D pose estimation scheme by comparing the input image with images generated by an OpenGL-based renderer. In Barrois and Wöhler [18], the information about the object used to do the 3D pose estimation was specified by CAD data. The error term produced by this comparison was then minimized by an iterative optimization algorithm to estimate all six degrees of freedom, using only a monocular camera. This method was evaluated on a toy example (rubber) and two objects in an industrial quality inspection. The results of the experiments for complex real-world objects at a distance of about 0.5 m to the camera showed accuracies of less than one degree for the rotation angles, 1–2 image pixels for the lateral translations, and several millimeters or about 1% for the object distance to the camera.

Kendal [19] suggested a general method of horizontal and vertical object distance calculations, where the object plane was parallel to the image plane or was tilted in the vertical plane using digital images. The size, density, and spatial distribution of a sample (shrubs) were also investigated in the Kendal [19] method. The experimental results showed a strong relationship between calculated distances and actual distances for different cameras, focal lengths, distances, and vertical tilt angles.

Tinnachote and Pimprasan [20] tried to use the combination of lens equation law and polynomial equations to find the object's distance to the camera. Tinnachote and Pimprasan [20] presented a method of regression co-efficient analysis to calculate an object's distance to the camera using data extracted from captured photos. In the Tinnachote and Pimprasan [20] study, photos of known height objects were taken at different distances using two cameras. The extracted data from those photos were applied to build relationship equations based on the lens equation. Since a proper relationship equation could not be generated using the lens law, regression co-efficient analysis was applied in the form of a 2nd-degree polynomial equation. The experimental results of Tinnachote and Pimprasan [20] showed that the object's distance to the camera calculated from a 3rd-order polynomial equation had a root mean square error as small as 0.25 m. The limitation of this method was that the polynomial equation formulation would not have good results with objects that are different in size than the one used in the polynomial equation.

2.3.2 Distance Measurement Using a Single Camera with Variable Pitch Angle

The object distance estimation in the robotic field is a key element, especially in tasks such as robot grasping, robot navigation, and general pick and place. Although achieving these tasks might appear easy at first sight, due to clearly structured and color-coded environments, an accurate estimation of the object’s pose and distance are required in practice. The contribution of this section is to provide a summary of approaches for the object distance measurement using a single camera with variable pitch angle.

Jamzad et al. [21] designed a mini-sized soccer robot called “Arvand” that followed RoboCup rules and regulations. The robot’s software, written in C^{++}, performed real-time image processing and object recognition. Arvand was the 2nd generation of robots designed by the Jamzad et al. [21] team. This mini-sized robot was comprised of a unique mechanics design that could simultaneously rotate around the ball center and find the goal’s position. Jamzad et al. [21] proposed a method for object distance calculation based on an object’s position on the image. This method did not depend on the object’s size, since the object might occasionally be unseen by the robot.

Robot soccer games are held in a dynamic, unpredictable, and challenging environment where the robot must recognize its position all the time. This is known as robot self-localization, which is one of the most important issues influencing a robot’s performance. Chang et al. [22] proposed an efficient method of self-localization based on a single landmark for the humanoid robot. Chang et al.’s [22] localization mechanism was based on three components: ***i*)** information that was given by the pan/tilt motors; ***ii*)** a single camera that was located on the robot’s head; and ***iii*)** an artificial neural network to adaptively localize the humanoid robot’s position. Their experimental results showed an accuracy rate of 88.5% at the frame rate of 15 (fps), and an average error of 6.68 cm between the actual distance and the improved distance. The flow chart of Chang et al.’s approach is demonstrated in Figure 2.4.

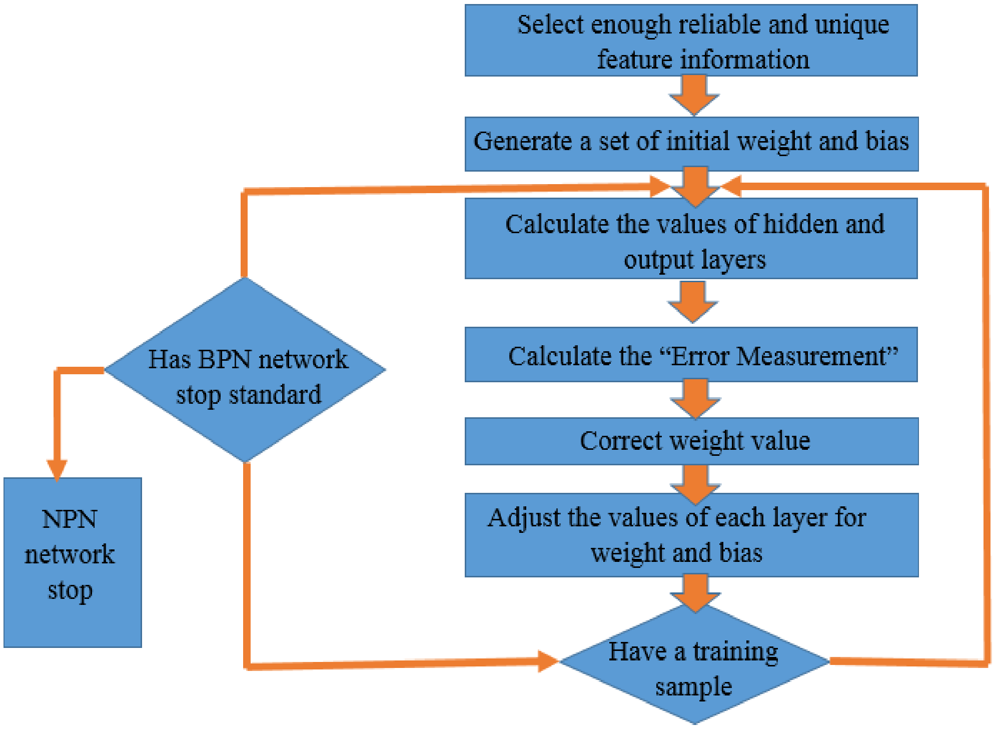

Select enough reliable and unique feature information
Generate a set of initial weight and bias
Calculate the values of hidden and output layers
Calculate the "Error Measurement"
Correct weight value
Adjust the values of each layer for weight and bias
Have a training sample
Has BPN network stop standard
NPN network stop

Figure 2.4: Flow chart for improving the precision of the Chang et al. [22] method.

Figure 2.5 shows the error related to the actual distance and improved distance for the Chang et al. [22] approach. The blue and red dashed lines chart are the unimproved distance and the improved distance, respectively.

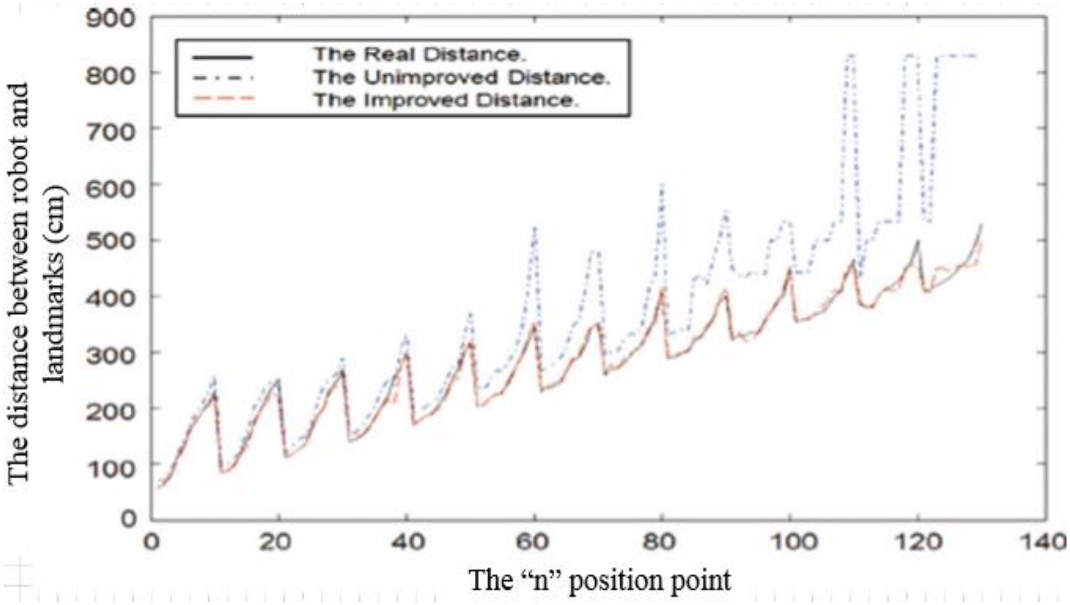


Figure 2.5: Comparison between actual and improved distances [22].

Other researchers, such as Hsu et al. [23], presented an image-based distance and inclined angle measurement method for objects located on an oblique plane. Their method was based on the variation of the pixel number in Charge-couple device (CCD) images by referencing two arbitrary points from the image frame. To minimize radial lens distortion during the experiment, objects had to be located as close to the image center as possible. The limitation of conventional image-based methods, where objects have to be positioned perfectly perpendicular to the optical axis, was removed using Hsu et al.'s [23] method.

Measuring the visibility distance for drivers can provide effective vehicle driving information ahead of time for prevention of accidents, thereby enhancing the traffic safety level. Hautiere et al. [24] proposed a framework for measuring visibility distances under foggy weather conditions using a camera mounted on a moving vehicle. Their research was mainly focused on detecting daytime fog conditions and estimating visibility distances. However, Hautiere et al.'s [24] proposed method could not function under nighttime foggy weather conditions. This framework enables estimating the "meteorological visibility distance" in each image using the region expansion technique, which is based on a physical diffusion model of the atmosphere. Hautiere et al.'s [24] proposed future work is to develop a new measuring visibility distance method that is functional in both daytime and nighttime foggy weather conditions.

Researchers such as Royer et al. [25] applied a sensing device for autonomous robot navigation outdoors using only a camera and natural landmarks. The Royer et al. [25] approach was comprised of three steps, as follows:

i) a learning step, where the robot was guided on a path and a video sequence was recorded with a camera;

***ii*)** building a 3D map from the learning sequence using a structure from the motion algorithm; and

***iii*)** a navigation step, where the robot employed the 3D map from step 2 to compute its localization in real-time and followed the learning path.

Royer et al. [25] showed approximately 2 cm and 0.1° for the robot's localization accuracy and orientation accuracy, respectively, following the same path. The main difficulty with this vision algorithm was that the updated map of the environment had to be given all the time. Thus, Royer et al. [25] were motivated to consider creating a method to update the map automatically.

Another method of object distance measurement, proposed by Shijie et al. [26], concentrated on the measurement of the relative position and attitude of spacecraft rendezvous and docking. The two steps involved in the Shijie et al. [26] approach were: ***i*)** extracting the feature point from the target spacecraft; and ***ii*)** obtaining the relative position and attitude of the target spacecraft using a feature point-based algorithm. Since finding the relative position and altitude estimation is a non-linear problem (3D to 2D), it needs a great amount of computational work. To overcome this problem, Shijie et al. [26] put forward an iterative algorithm based on the inverse projection ray approach. Shijie et al. [26] also validated the effectiveness and rapidity of convergence of their proposed algorithm by using mathematical simulation.

Orientation compensation is another method that assists in object distance estimation. Cheung et al. [27] proposed a structure of orientation compensation to avoid undesirable camera orientations and to accurately estimate the distance of a preceding vehicle in a driver assistance system (DAS). The three advantages to this system are: ***i*)** the ability to estimate a homography H for the camera pan/tilt angle compensation; ***ii*)** achieving further accurate distance estimation compared with the RADAR estimates; and ***iii*)** the ability to accurately estimate distances in a non-planar road surface by updating the vanishing point.

A rapid growth in traffic would significantly impact the number of traffic accidents. Chiang et al. [28] developed a driver assistance algorithm to determine a collision-free path by lane tracking and obstacle detection using a monocular camera. This lane-marking detection algorithm is applicable in different illumination and complex outdoor environments. Once the algorithm detects a lane departure, it will issue a signal to warn the driver of possible deviations. In the obstacle detection strategy, the gradient information is used to find the feature points of the object, after which the 3D position of the object is estimated by means of triangulation. The experimental results of Chiang et al. [28] showed the robustness of this method against curved lanes and broken lanes in captured road images. The average error for depth estimation was reported to be about 4.87 cm and the corresponding standard deviation was 1.59 cm. The Chiang et al. [28] future work is mainly concentrated on creating a complete driver assistance algorithm that can be aware of all of the vehicle’s surroundings.

2.4 Visual Servoing Methods

The first computer vision application for control loops was in indoor and structured environments, where a line or known patterns were detected and followed by a robot [29]. Photometric visual servoing is a new technique to overcome the problem of the object tracking process. Object tracking is a process to estimate an object’s position and orientation using some known object’s features, such as a corner, edge or marker, that are extracted from the captured images. In photometric visual servoing, the tracking process is no longer required, since the image intensity (the pure luminance signal) is sufficient to control the robot's motion. Image gradient and image entropy have the same approaches as photometric visual servoing [30]. The image gradient technique is based on the extraction of information of an image which is located in its high frequency areas (contours). Marchand and Collewet [31] applied a method to use the square norm of the gradient obtained from all of the pixels in an image as visual features in visual servoing. In Marchand and Collewet's [31] applied method, the final positioning task error was found to be 0.4 mm in the translation and 0.12 degrees in the rotation.

Another method of visual servoing is entropy-based visual servoing, which works on the basis of mutual information. This type of information is mostly used in medical applications, as it is not sensitive to changes in lighting conditions or to many classes of non-linear image transformations. Entropy-based visual servoing does not require a matching or tracking process and could be applied under large illumination variations [32].

Wang and Liu [33] proposed a new visual servo control technique for the robotic manipulator, whereby a back propagation neural network would make a transition from the image feature to joint angles. The advantages of this technique were:

***i*)** eliminating the complicated calibration processes; and

***ii*)** reducing the amount of computations.

To guide the two coupled robots (mini-robot and manipulator), Pomares et al. [34] described a dynamic image-based control algorithm. This new method was used to monitor the mini-robot using dynamic control for the purpose of tracking a previously generated image trajectory. This algorithm was able to correctly track the desired trajectories in cooperation with the robotic manipulator.

A new technique that requires neither the metric information of the observed object nor the accurate camera or robot calibration parameters was proposed by Silveira and Malis [35]. This technique could directly extract: ***i*)** the projective parameters that relate the current image to the reference image; and ***ii*)** the pixel intensities to obtain these parameters. The projective parameters could be attained via a photo-geometric registration method through extracting the image's information, even from areas where no image feature exists. Therefore, both the accuracy and robustness of illumination variation, even in color images, could be achieved. This type of non-metric control error can be used for path planning.

2.5 Conclusion

In this section, a detailed literature review of concepts related to this thesis was provided. Current and past research in the fields of object tracking, distance measurement, and visual servoing necessary for the design and implementation of the intended algorithm was reviewed. Object tracking methods were elaborated and classified into different categories, and new tracking methods along with problems related to object tracking were also presented.

Next, Chapter 3 describes the background requirements for implementing the intended algorithm and presents some of the main concepts and techniques of object distance measurement for a single fixed camera and single camera with variable pitch angle.

Chapter 3

BACKGROUND AND THEORY

3.1 Introduction

The main intent of this chapter is to present different methods for object distance measurement using a single fixed camera and a single camera with variable pitch angle. The value of this chapter is that it contains descriptions of some of the most common object distance measurement methods to date together with evaluations of the relative accuracy of these techniques. Also reviewed in this chapter is background information on different types of camera technologies, visual servoing, and existing range-finding techniques.

3.2 Computer Vision

Computer vision is one of the most interesting subjects for scientists, since it plays an important role in many applications such as video surveillance, robot navigation, road traffic analysis, etc. Machine vision has six parameters, as follows [36]:

- Sensing, which deals with the visual image.
- Preprocessing, which is used for noise reduction, image enhancement, etc.
- Segmentation, which is concerned with image partitioning into the desired object.
- Description, which is the computation of the object features.
- Recognition, which is used to identify the object.
- Interpretation, which is used for assigning meaning to the recognized object.

3.3 CCD and CMOS Cameras

Vision development can be divided into the two main classes of image acquisition and image processing. Image acquisition is a system that employs a camera to capture different types of image in terms of resolution, using different lenses. Currently, there are two types of digital camera technologies: charged coupled device (CCD), and complementary metal oxide semiconductor (CMOS). The CCD camera is the most basic type used in robotic vision systems nowadays. The CCD chip is designed from a group of light sensitive picture elements called pixels and normally includes between 20,000 and several million pixels. These pixels are considered to be discharging capacitors that can be as small as 5 to 25 µm. The problem with this type of cameras is that their capacitors of all pixels need to be fully charged before the process of reading can occur. The reading process is performed at one corner of the CCD chip. This means that each charge should be sustainably transferred across the chip in a row and a column to reach one specific corner. This procedure requires a precise technique to ensure the stability of the transported charge. Figure 3.1 demonstrates a typical CCD chip and some cameras [37].

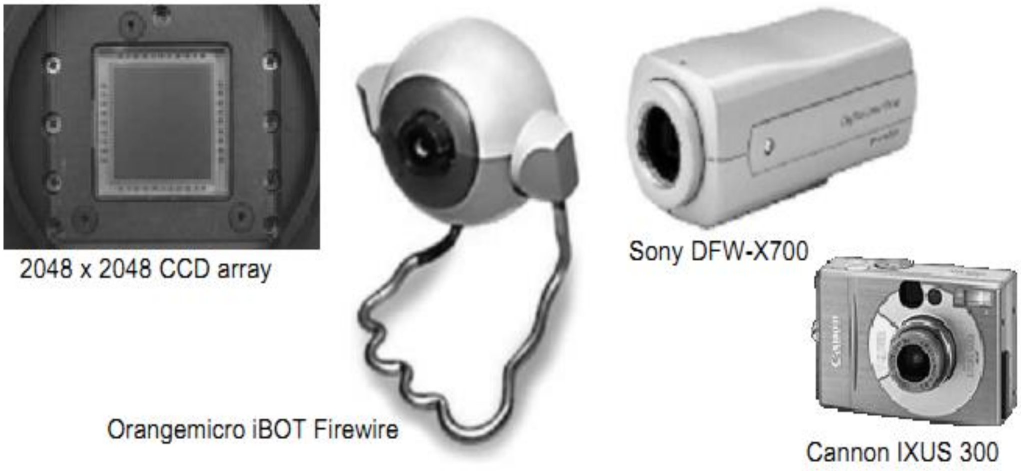


Figure 3.1: Typical CCD chip and CCD cameras [37].

The functionality of CCD cameras depends on their parameters. Some CCD cameras' parameters are pre-set (the user cannot change these parameters), while other parameters change constantly. Higher-end cameras are designed such that the user can modify the values of these parameters through software in order to capture as much of the desired image as possible. For instance, iris position and shutter speed are two parameters to regulate the amount of light measured by the camera. The iris is a mechanical opening that adjusts the incoming light, and the shutter speed is set for sustaining the integration period of the chip where the photon strikes the pixels. The shutter speed in higher-end cameras is between 1/30,000 and 2 seconds.

The CMOS chip is another important technology used in cameras. Figure 3.2 shows a typical CMOS camera that has the same array of pixels as CCD cameras, but with several transistors along with each pixel. During the data collection process in CMOS cameras, all pixels measure and amplify an individual pixel's signal in parallel for every pixel in the array. This process continues until the destination is reached and there is no need to transfer each pixel's charge down to the specific location.

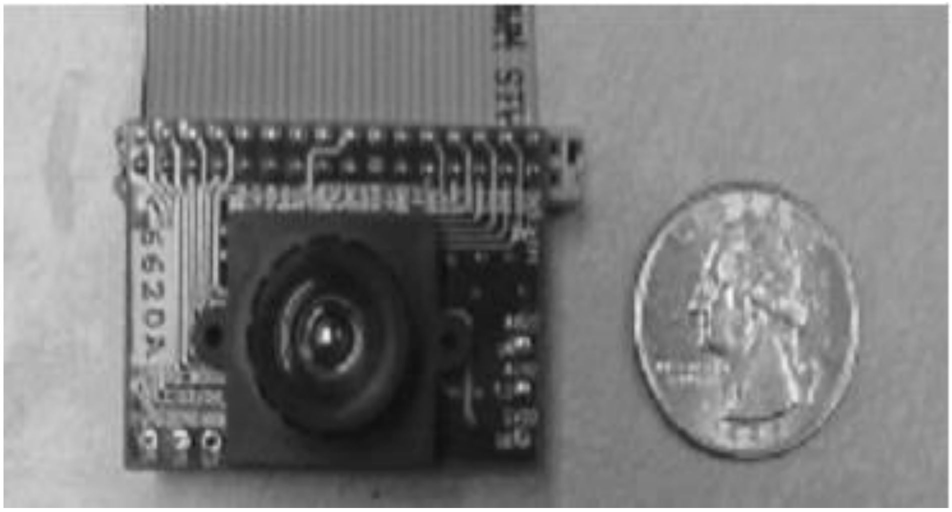


Figure 3.2: CMOS camera with lens attached [37].

CMOS technology has a much simpler configuration than a CCD chip, which enables CMOS cameras to consume significantly less power (about one-hundredth of the power consumed by CCD chips). This starkly lower power consumption makes CMOS technology suitable for use in mobile robots [37].

3.4 Visual Servoing

Visual servoing is defined as controlling robots by the data obtained from the vision system. It is classified into the two different approaches of position-based and image-based systems [38]. The position-based method uses 3D information of real workspaces, while the image-based method employs 2D information from images taken by a camera [39]. The main issue for camera-based visual servo control is to find the relationship between the Euclidean-space and the image-space. One key factor in this relationship is that the image-space is a 2D projection of a 3D Euclidean-space. To compensate for the lack of depth information in 2D image data, some researchers have focused on using alternative sensors such as laser and sound ranging technologies. Others have investigated alternative methods of a camera-based vision system using additional sensors or even utilizing two cameras (stereo vision) [40].

3.4.1 Robot Visual Servoing

The conventional robot visual servo control algorithm mainly depends on: 1) the calibration technique to determine the intrinsic parameters of the camera, 2) the hand-eye transformation, and 3) the parameters of the robot kinematics. Therefore, the control precision of the servo algorithm mainly depends on the precision of the camera calibration [40].

There are two different approaches in using cameras in robotic applications, which Flandin [41] described as eye-in-hand and eye-to-hand for controlling robots. In the eye-in-hand method (Figure 3.3), the camera is positioned on the end-effector. In the eye-to-hand method (Figure 3.4), the camera is installed outside the robot to observe the robot workspace. Of the two methods, eye-in-hand 2D visual servoing is the more important research area and has been intensely studied for the past 30 years [42]. In the eye-in-hand configuration, there is a constant relationship between the pose of the camera(s) and the pose of the end-effector. For the fixed camera case, the camera(s) is (are) related to the base coordinate system of the robot and to the object, and the camera image is independent of the robot's motion.

In both eye-in-hand and eye-to-hand camera configurations, calibrating the camera prior to the visual servoing task is critical. Calibration includes tuning the intrinsic camera's parameters such as focal length, pixel pitch, and the principal point. A fixed camera's pose with respect to the global coordinate system should be included for the extrinsic camera parameters after the camera calibration procedure is applied. Furthermore, the relative pose should be considered for the eye-in-hand case, which is known as the hand-eye calibration problem [43].

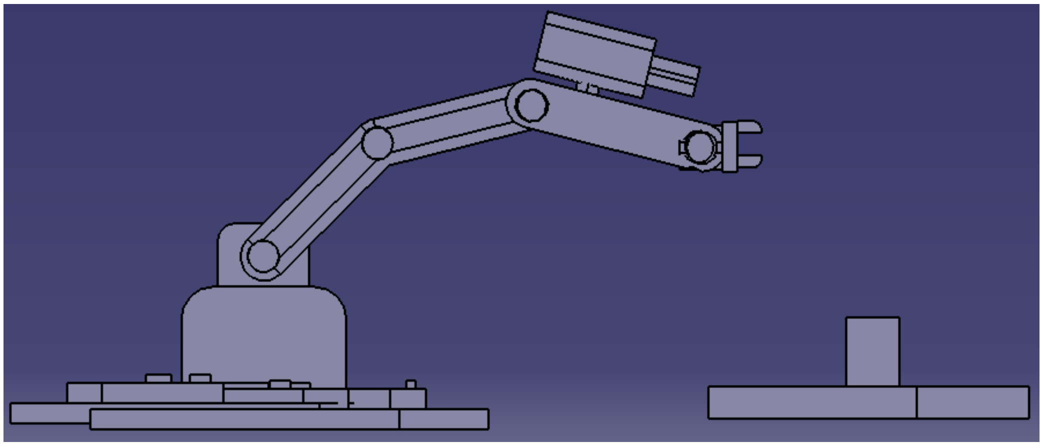


Figure 3.3: Eye-in-hand configuration.

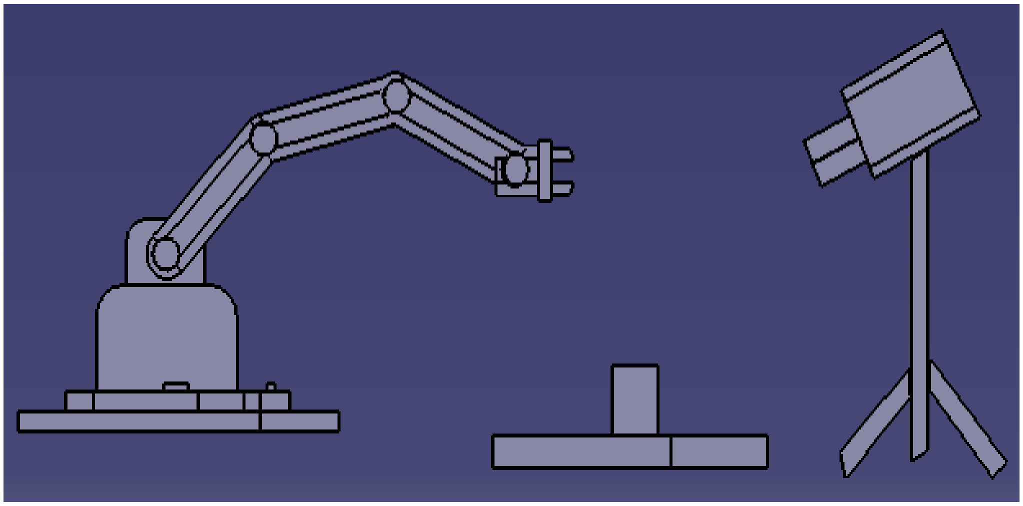


Figure 3.4: Eye-hand configuration.

3.5 Perspective Projection

The image in the camera is formed by the pinhole lens model [44]. Thus, the lens is an ideal pinhole that is located at the focal center of the camera, and light passing through this pinhole intersects the image plane. Considering ***m*** as the projection of ***M***(*x*,*y*,*z*) (a point in the global coordinate system) onto the image plane with coordinates (*u*,*v*,*λ*), the following equation is developed for some constant *s* , since the points ***M***, ***m*** and the origin of the camera frame are collinear:

$$s\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} u \\ v \\ \lambda \end{pmatrix} \qquad (3.1)$$

In this case, the corresponding projection equations are as follows:

$$\begin{aligned} u &= \lambda \frac{x}{z} \\ v &= \lambda \frac{y}{z} \end{aligned} \qquad (3.2)$$

3.6 The Complete Transformation

Every point in a 3D space with respect to the global coordinate system can be denoted by $\boldsymbol{M}$=[x,y,z] $^{\mathrm{T}}$. The augmented vector $\boldsymbol{M}$ is shown by $\boldsymbol{M'}$=[x, y, z, 1] $^{\mathrm{T}}$, which is $\boldsymbol{M}$, but adding 1 for the last element makes it a homogenous representation of $\boldsymbol{M}$. This specific point is observed by a camera and then transformed into a pixel. The pixel is shown by $\boldsymbol{m}$= [u, v] $^{\mathrm{T}}$ and corresponds to the real point $\boldsymbol{M}$. The augmented vector of $\boldsymbol{m}$ is $\boldsymbol{m'}$ and is shown as $\boldsymbol{m'}$= [u, v, 1] $^{\mathrm{T}}$, with u, and v, in this vector being the pixel coordinates of the captured image [45]. If the pinhole camera model is considered, the relationship between 3D point $\boldsymbol{M}$ and its projected 2D point $\boldsymbol{m}$ is as follows:

$$s\,\boldsymbol{m'} = \boldsymbol{A_{in}}\,[\boldsymbol{R}\boldsymbol{T}]\,\boldsymbol{M'} \tag{3.3}$$

where $\boldsymbol{A_{in}} = \begin{pmatrix} s_1 & c & u_0 \\ 0 & s_2 & v_0 \\ 0 & 0 & 1 \end{pmatrix}$ is a 3×3 matrix of the intrinsic parameters, s is the arbitrary scale factor, $\boldsymbol{R}$ is a 3×3 rotation matrix and $\boldsymbol{T}$ is the 3×1 translation vector. $\boldsymbol{R}$ and $\boldsymbol{T}$ are called the extrinsic camera's parameters that relate the global coordinate frame to the camera's coordinate frame. The intrinsic camera's parameters are composed of pixel coordinates of the principal point (u_0,v_0), s_1, and s_2 are the scale factors in the image's u and v axes, and c is the skewness of the two image axes. Although, the extrinsic parameters change once the camera moves, the intrinsic camera's parameters are fixed for a certain camera. The complete transformation is shown in the following equation:

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \boldsymbol{A_{in}} \begin{pmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{3.4}$$

3.7 Existing Range-Finding Techniques

Range-finding methods are used to determine the distance to an object or objects. The three major range finding techniques available are triangulation, structured light, and time-of-flight. These techniques are briefly described in the following sections.

3.7.1 Triangulation

The triangulation technique can be either active or passive. The active triangulation method emits a signal and then measures the reflected signals, whereas the passive triangulation method uses the background illumination [46]. Figure 3.5 demonstrates the concept behind the triangulation method.

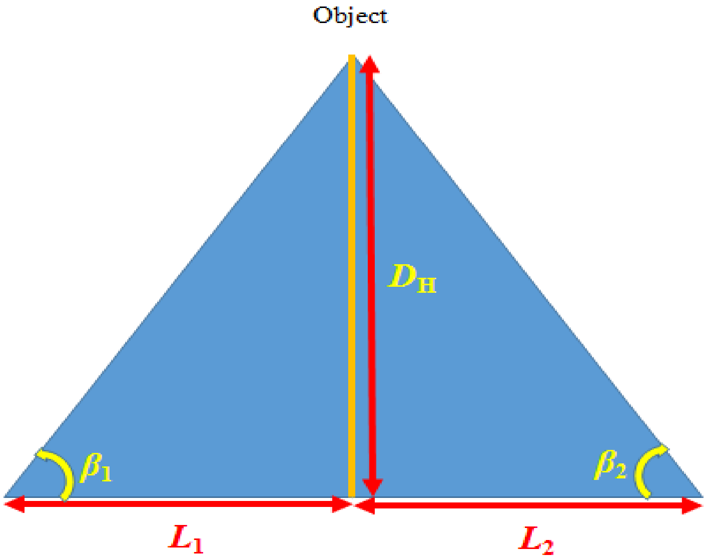


Figure 3.5: Triangulation technique (reproduced from reference [46]).

In Figure 3.5, the relationship between the base line lengths L_1, L_2, and the base line angles β_1, β_2 is summarized in the following equation:

$$L_1 + L_2 = \frac{D_H}{\tan(\beta_1)} + \frac{D_H}{\tan(\beta_2)} \tag{3.5}$$

After rearranging the above equation, the distance to the object is calculated as:

$$D_H = \frac{L_1 + L_2}{(\frac{1}{\tan(\beta_1)} + \frac{1}{\tan(\beta_2)})} \quad (3.6)$$

3.7.2 Structured Light

The structured light approach is the projection of a set of light patterns onto the scene. It analyzes the pattern deformation over the object in the scene. The basic structured light system can be implemented using a single line of the light, while the more advanced systems are able to scan the entire field of view using an infrared projector. One of the common structured light systems is the Kinect sensor used in the Xbox 360. This sensor has a 57^o horizontal field of view and a 43^o vertical field of view, with a 640×480 resolution and 30 fps-output video frames. One shortcoming of the structured light system is that the resolution highly depends on the separation between the camera and the light source [46].

3.7.3 Time-of-Flight

In the time-of-flight method, distance is measured by the time it takes for an emitting signal of known velocity to reflect back to an object. Two widely used time-of-flight systems are radar and sonar. The time-of-flight range-finding approach consists of three methods: ***i*)** direct time-of-flight measurement; ***ii*)** shuttered light pulse; and ***iii*)** indirect time-of-flight measurement. The time-of-flight distance is calculated as follows:

$$D_H = \frac{v_s t}{2} \quad (3.7)$$

where D_H is the distance to the object, v_s is the velocity of the signal, and t is the time taken for the signal to reflect back to the object [46].

3.8 Basic Lens Equation

Lenses are specifically designed to form images by refraction in optical instruments such as cameras, telescopes, and microscopes. The two most common types of images are *real* and *virtual*. A real image is one that is formed in the back of the lens, whereas a virtual image is formed in the front of the lens. In order to locate the image formed by thin lenses and also to clarify the sign conventions, ray diagrams are used. The three rays used for locating the image of converging lenses are as follows: Ray 1 is parallel to the optical axis and, after being refracted by the lens, passes through the focal point on the other side of the lens. Ray 2 passes through the focal point on the front side of the lens and continues from the lens parallel to the optical axis. Ray 3 passes through the center of the lens and continues in a straight line. Figure 3.6 demonstrates a typical real image formation using ray diagrams for thin lenses.

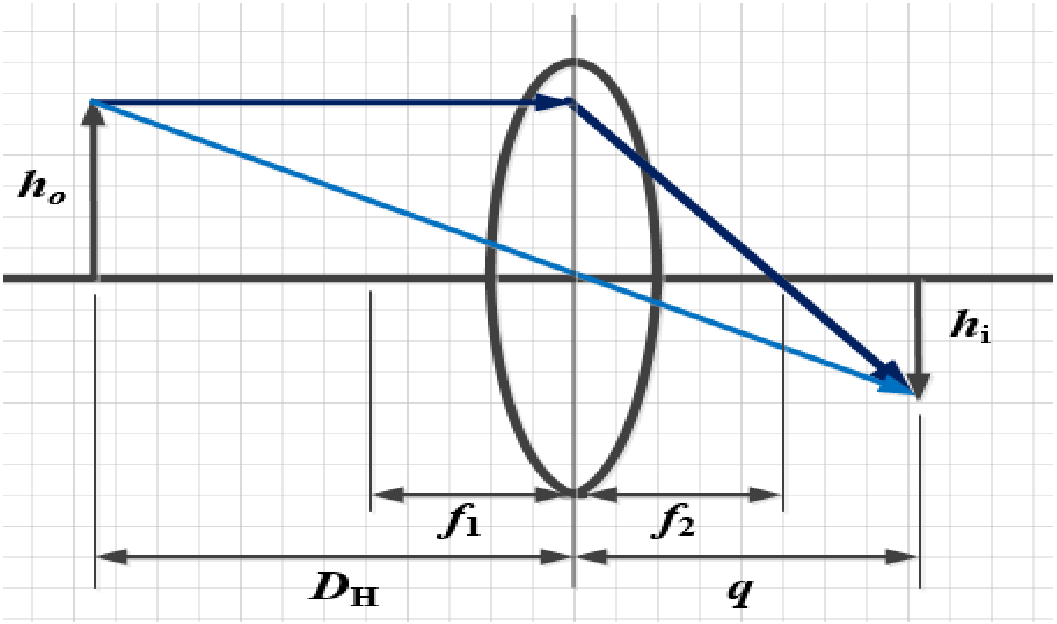


Figure 3.6: Real image formation by thin lens (reproduced from reference [47]).

Once the image is formed using thin lenses, the relationship between image distance and object distance can be calculated using the thin lens equation (Equation 3.8). In this equation, object distance is defined as the distance of the object along the optical axis to the lens, image distance is the distance of the image along the optical axis to the lens, and focal length is the distance of the focal point along the optical axis to the lens.

$$\frac{1}{f}=\frac{1}{D_{\mathrm{H}}}+\frac{1}{q} \tag{3.8}$$

where D_{H} is the object distance, q is the image distance and f is the focal length of the lens. The sign conventions for object distance, image distance, image height, and focal length of lenses are classified in Table 3.1.

Table 3.1: Sign conventions for thin lenses.

Quantity	*Positive when....*	*Negative when....*
Object location (D_{H})	Object is in front of lens (real object).	Object is in back of lens (virtual object).
Image location (q)	Image is in back of lens (real image).	Image is in front of lens (virtual image).
Image height (h_i)	Image is upright.	Image is inverted.
Focal length (f)	Converging lens.	Diverging lens.

3.8.1 Image Magnification:

The lateral magnification of an image can also calculated by Equation 3.9.

$$M=\frac{\text{Image height}}{\text{Object height}}=\frac{h_{\mathrm{i}}}{h_{\mathrm{o}}}=-\frac{q}{D_{\mathrm{H}}} \tag{3.9}$$

In Equation (3.9), if M is positive, the image is upright and located on the same side of the lens; however, if M is negative, the image is inverted and located on the opposite side of the object. As shown in Figure 3.6, since the object is located in front of the lens and outside the focal point of a converging lens, the image is real, inverted, and is formed on the back side of the lens. Another typical ray diagram is demonstrated in Figure 3.7, which shows a virtual image formation using a thin lens.

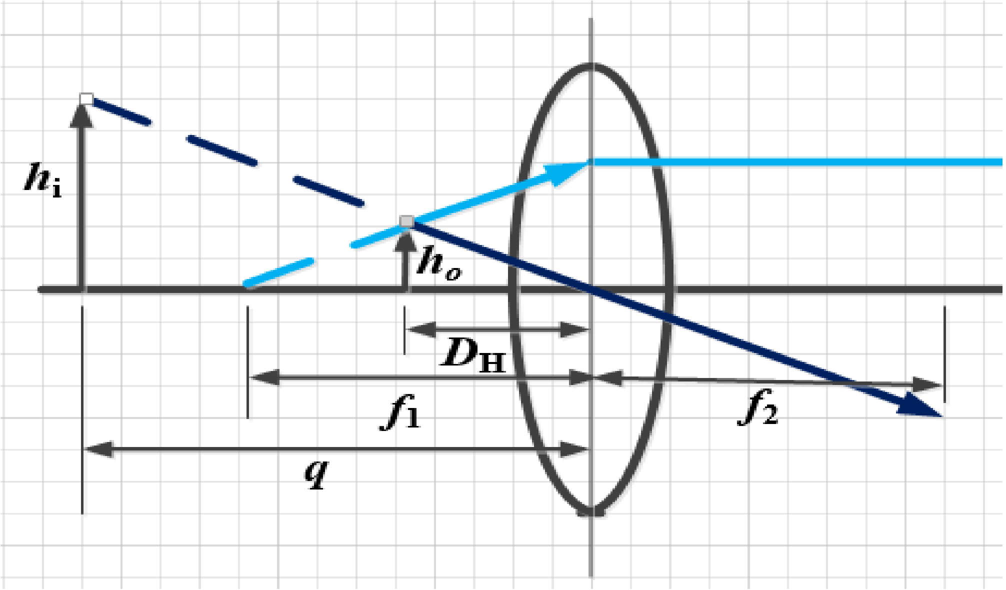


Figure 3.7: Virtual image formation by thin lens (reproduced from reference [47]).

In Figure 3.7, since the object is located between the focal point and a converging lens, the image is the virtual image, upright, larger than the object, and located on the front side of the lens.

3.9 Different Object Distance Calculation Methods

In the following sections, different methods of object distance measurement using a single fixed camera and a single camera with variable pitch angle are discussed. The two major object distance measurement methods used here are as follows: ***i)*** the method where the object's size and the camera's focal length are known; and ***ii*)** the method where the object’s size is unknown, but the object’s point of contact with the ground is known.

3.9.1 Object Distance Calculation Methods for Single Fixed Camera

There are numerous methods for calculating object distance using a single fixed camera. Krishnan et al. [16] proposed the complex log mapping (CLM) method shown in Figure 3.8. The principle of the CLM approach is that the original images are mapped from the orthogonal coordinate system to the polar coordinate system. The ratio between two images can be calculated by CLM because these two images have concentric circles features.

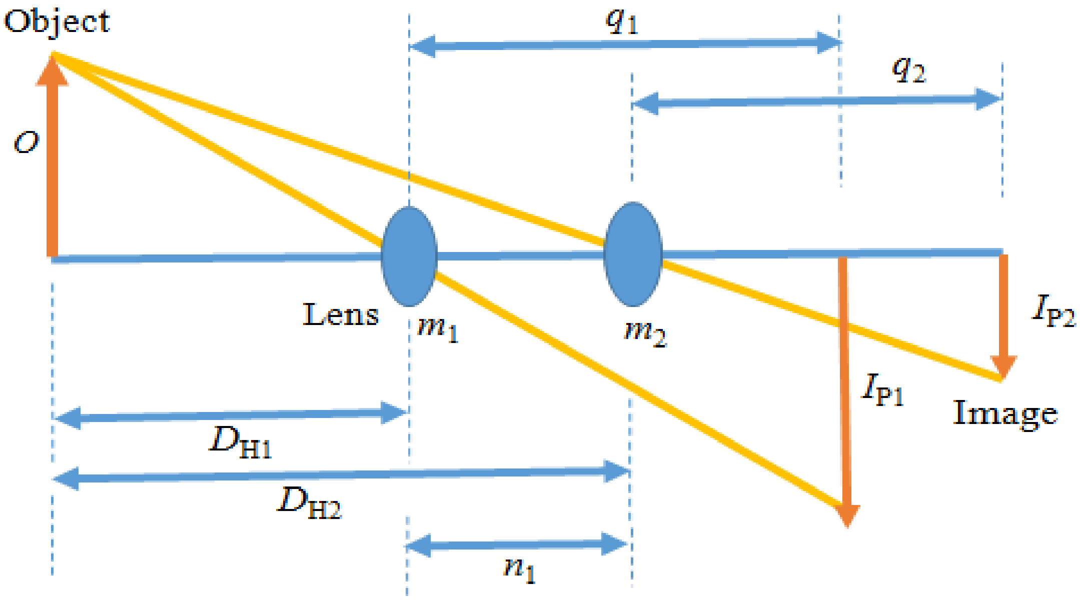


Figure 3.8: Distance measurement between camera and object (reproduced from reference [16]).

Distance measurement equations using Krishnan et al.’s [16] approach are as follows:

$$\frac{1}{D_H}+\frac{1}{q}=\frac{1}{f} \tag{3.10}$$

$$I_P D_H = O q \quad (i=1,2) \tag{3.11}$$

where O is object size, D_{H} is object distance, q is image distance, I_{p} is image height, and f is the camera's focal length. If the camera moves a known distance n_1 from m_1 to m_2 along the optical axis:

$$D_{\mathrm{H1}} - QD_{\mathrm{H2}} - f(1-Q) = 0 \tag{3.12}$$

$$n_1 = |D_{\mathrm{H1}} - D_{\mathrm{H2}}| \tag{3.13}$$

Furthermore, if Q (which is the ratio of I_{P1} to I_{P2}) is obtained, D_{H1} and D_{H2} can be calculated as follows:

$$D_{\mathrm{H1}} = f - \frac{Q}{1-Q} n_1 \tag{3.14}$$

$$D_{\mathrm{H2}} = D_{\mathrm{H1}} - n_1 \tag{3.15}$$

Now, consider that a pixel ($D_{\mathrm{H}i}$, q_i) from the original image is mapped on the pixel (m_i, n_i) by CLM of ($\mathrm{c_x}$, $\mathrm{c_y}$). The relationship between these two pixels by the CLM would then be as follows:

$$M_0(m_i, n_i) = \frac{z}{r}\ F_0(D_{\mathrm{Hi}},\ q_{\mathrm{i}}) \tag{3.16}$$

In Equation (3.16), F_0 ($D_{\mathrm{H}i}$, q_i) is called the center of the visual field, and the gray scale at (m_i,n_i) is denoted as $M_0(m_i, n_i)$. The relationship between the xy-plane and mn-plane are as follows:

$$z = \sqrt{D_{\mathrm{Hi}}{}^2 + q_i{}^2} \tag{3.17}$$

$$\theta_1 = \tan^{-1} \frac{D_{\mathrm{H}i}}{q_i} \tag{3.18}$$

$$\begin{aligned} m_i &= N_1 \frac{\theta_1}{2\pi} \\ n_i &= N_1 \log_r z \end{aligned} \tag{3.19}$$

Using Equations (3.16-19), the mapping can be described as:

$$
\begin{aligned}
m &= \log\sqrt{D_{\mathrm{H}}{}^{2}+q^{2}} \\
n &= \tan^{-1}\frac{q_i}{D_{\mathrm{Hi}}}
\end{aligned}
\tag{3.20}
$$

where r is the mapping radius, N_1 is the size of the mapped image, and m_i and n_i are calculated in terms of distance z and direction θ_1 from the mapping origin (c_x, c_y) to the original image pixel ($D_{\mathrm{H}i}$,q_i). There are, however, some disadvantages and limitations to the Krishnan et al. method:

- The camera must move along its optical axis, which is impossible in practice;
- The reference image should be as close as to the surface of the object so that all the image elements of reference image is included in those on adjusting image (final mapped image);
- Implementing this technique is computationally expensive, since many processes have to be done in order to calculate the distance between the two images;
- This method is specifically used to find the distance to the object, which is perpendicular to the optical axis. This is impractical for slanting surfaces.

The results of the Krishnan et al. [16] experiment are shown in Table 3.2.

Table 3.2: Distance measurement for various texture surfaces.

Real distance (mm)	***Grass lawn (mm)***	***Reptile skin (mm)***	***Ceramic coated brick wall (mm)***
500.0	491.4	494.2	499.6
600.0	606.6	589.9	601.3
700.0	710.8	684.1	702.2

Another method to obtain object distance measurement was proposed by Kendal [19]. This method calculates the horizontal and vertical object distances when the object plane is parallel to the image plane or is tilted in the vertical plane using digital images.

Figure 3.9 shows the projection of an object on an oblique image plane. In Figure 3.9, δ is the vertical tilt angle, D_{H} is the distance from the optical center of the image plane to the object (mm), h is the distance above the perpendicular of the optical center of the image in object plane (mm), and f represents the focal length (mm), respectively. The object distance can be calculated by Equation (3.21), as follows:

$$O = \frac{I\,D_{\mathrm{H}}}{f} \tag{3.21}$$

where O is the object's size in mm, I is the image size in mm, D_{H} is the distance from the image plane to the object plane (mm), and f is the focal length of the lens (mm). In digital images, all units are in pixels that can be easily converted to mm only if the sensor size of the camera is known. The sensor dimension is different in horizontal and vertical planes. If the object and image are positioned in parallel, Equation (3.21) will be modified to Equations (3.22-23), as follows:

$$O_x = \frac{x\,S_x D_{\mathrm{H}}}{f\,p_x} \tag{3.22}$$

$$O_y = \frac{y\,S_y D_{\mathrm{H}}}{f\,p_y} \tag{3.23}$$

where O_x is the horizontal image dimension (mm), x is the horizontal image dimension (pixel), S_x is the horizontal sensor size (mm), P_x is the horizontal sensor size (pixel), O_y is the vertical image dimension (mm), y is the vertical image dimension (pixel), S_y is the vertical sensor size (mm), and P_y is the vertical sensor size (pixel). Additionally, X represents the true horizontal distance in an object plane (mm), and Y is the true vertical distance in the object plane (mm). Since the distance d, h, and f are known, the distance

D_{H} from the focal point to the center of the object can be calculated using Equation (3.24).

$$D_{\mathrm{H}} = \sqrt{d^2 + h^2} - f \tag{3.24}$$

Moreover, the required angles δ, λ, ϕ can be calculated using the trigonometric identities:

$$\delta = \sin^{-1}\left(\frac{h}{D_{\mathrm{H}} + f}\right) \tag{3.25}$$

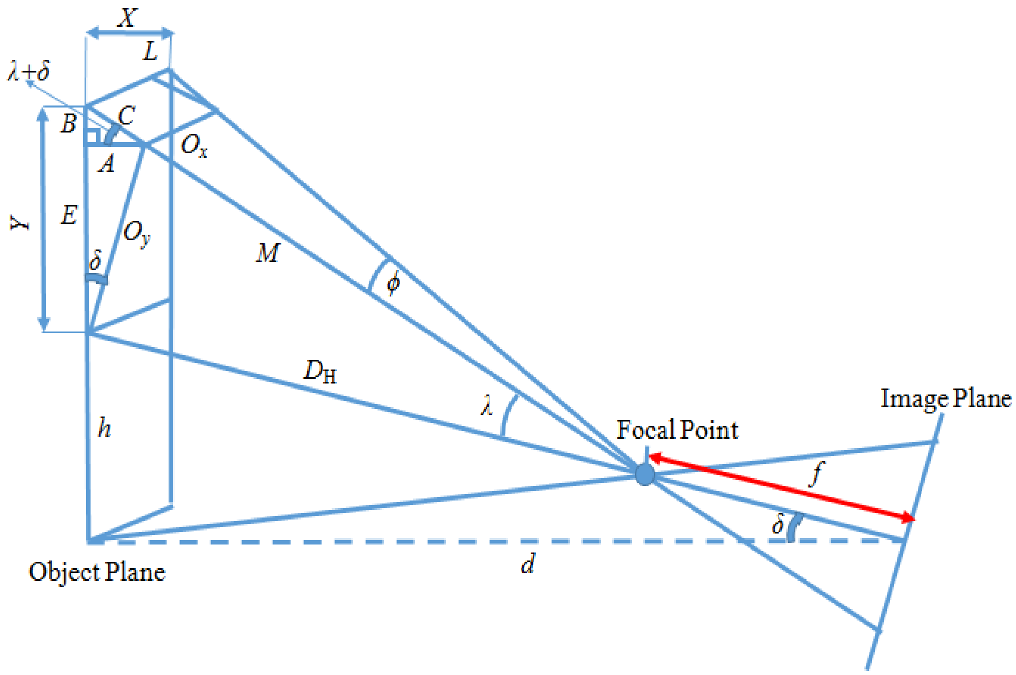


Figure 3.9: The geometry of a projected object on an oblique image plane (reproduced from reference [19]).

$$\lambda = \tan^{-1}\left(\frac{O_y}{D_{\mathrm{H}}}\right) \tag{3.26}$$

$$\phi = \tan^{-1}\left(\frac{O_x cos\,\lambda}{D_{\mathrm{H}}}\right) \tag{3.27}$$

It is also possible to calculate the real object dimension by the following equations:

$$Y = B + E = A\tan(\delta + \lambda) + O_y \cos\delta = O_y \sin\delta \tan(\delta + \lambda) + O_y \cos\delta \tag{3.28}$$

$$X = O_x + L = O_x + C\tan\phi = O_x + \frac{\mathrm{A}}{\cos(\delta+\lambda)}\tan\phi = O_x + \frac{O_y sin\,\delta}{\cos(\delta+\lambda)}\tan\phi \tag{3.29}$$

Table 3.3a shows the regression analysis results of different camera/sensor combinations for predicting distance measurements. Table 3.3b shows the regression analysis results for the calculated-versus-measured distances. In Tables 3.3a and 3.3b, P is defined as the predicted distance from the regression equation and C is obtained from the distance equations.

Table 3.3: The experimental results for object distance and dimension calculations.

Camera	*Axis*	*Sensor pixels*	*Sensor size(mm)*	*Regression equation*	*R-square*
Canon A70	Horizontal	2048	5.27	P = -0.3+0.98C	99.8%
Canon A70	Vertical	1536	3.96	P = -0.1+0.96C	99.9%
Nikon D70	Horizontal	3008	23.7	P = 0.3+0.93C	99.9%
Nikon D70	Vertical	2000	15.6	P= -0.9+0.93C	99.8%
Nikon 5400	Horizontal	2592	7.18	P= -0.9+1.00C	99.6%
Nikon 5400	Vertical	1944	5.32	P= -0.8+1.00C	99.7%

a) Results of object distance calculations

Measurements	*Regression equation*	*R-square*
Height	P= 26+1.01C	96.1%
Width	P= 10+1.02C	84.2%

b) Results of object dimension calculations

Some shortcomings of the Kendal [19] object distance method are as follows:

- The method is time-consuming, since it uses a set of 200 plant images;
- The method requires the use of a high quality camera such as a Canon or Nikon (our proposed method is implemented by a normal webcam);
- The largest error measurements in the hedge experiment using the Kendal method is due to object distance D_H, tilt angle δ, height of h, and hedge shadow;

- Cameras should be calibrated prior to use in order to check for any constant error before employing the equations proposed by Kendal [19].

Joglekar et al. [48] proposed a method to estimate depth using a monocular camera. The Joglekar et al. [48] method depends on the camera's parameters and image geometry. Figure 3.10 shows how to calculate the focal length of the camera *f* for the Joglekar et al. [48] method using the camera's field of view.

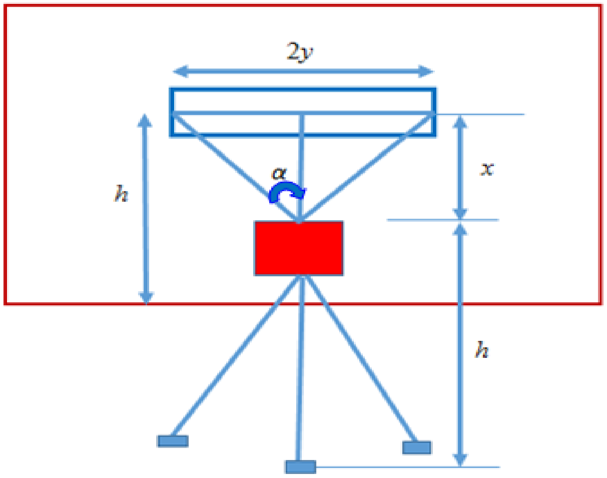


Figure 3.10: Field of view and focal length of camera (reproduced from reference [48]).

$$\alpha = \arctan(\frac{y}{x}) \tag{3.30}$$

$$\begin{aligned} FOV° &= 2(\tan^{-1}(\frac{y}{x})) \\ FOV° &= 2(\tan^{-1}(\frac{x}{2f})) \end{aligned} \tag{3.31}$$

In the experimental setup of the Joglekar et al. [48] method, the camera is mounted at height *h*, *B* is the distance D_{H1} from the back of the object to the camera's optical center, and I_{P1} is the projection point of the object and the ground onto the image plane. The point of contact for the distant object *C* onto the image plane is shown by I_{p2}, which is smaller than I_{P1} (Figure 3.11). It is important to note that, in Equation (3.31), the focal distance *f* and the image coordinates *I* are in pixel unit. The depth estimation equation (the in-path object distance) can be derived using the similarity of triangles:

$$\frac{I_p}{f} = \frac{h}{D_H}$$
$$D_H = \frac{f\,h}{I_p} \tag{3.32}$$

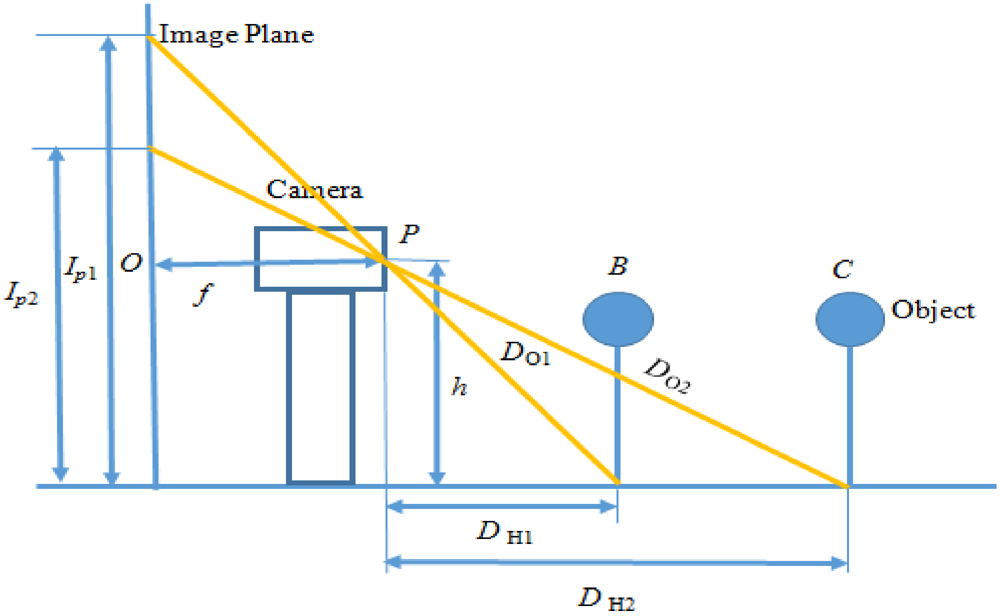


Figure 3.11: Joglekar et al.'s method (reproduced from reference [48]).

Joglekar et al. [48] defined the in-path object distance as the distance along the optical axis of the camera. The calculated in-path object distance, which is shown in Figure 3.12, is then divided by cosine of angle β to obtain the oblique distance. The oblique distance can be calculated as follows:

$$D_O = \frac{D_H}{\mathrm{Cos}(\beta)} \tag{3.33}$$

Since the calculated oblique distance includes errors that are horizontal (average percentage error of 40 cm) and vertical (average percentage error of 8 cm), Joglekar et al. [48] integrated multiple polynomial equations to obtain the real oblique distances based on the curve-fitting method. The limitations and comparisons between Joglekar et al.'s and the improved object distance methods are demonstrated in Tables 4.2, 4.4, and 4.5.

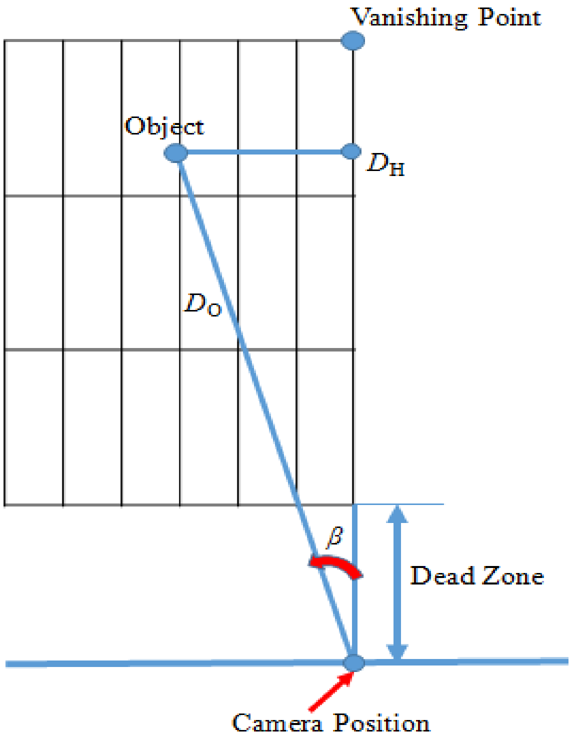


Figure 3.12: Oblique distance calculation (reproduced from reference [48]).

3.9.2 Object Distance Calculation Methods for Single Camera with Variable Pitch Angle

There are a few different object distance calculation methods for a single camera with variable pitch angle. Some of these methods are selected and discussed in this section.

Jamzad et al. [21] proposed an object distance measurement method for mid-sized robots. This method is shown in Figure 3.13. By using Equations (3.34-40), the horizontal and vertical object distances can be calculated, as follows:

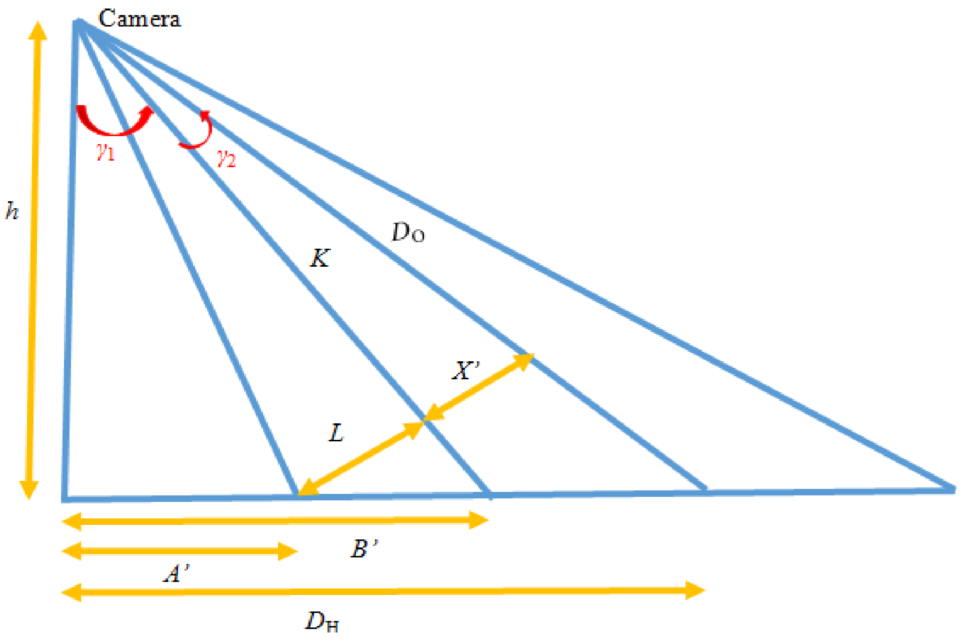


Figure 3.13: Object distance calculation for soccer robot (reproduced from reference [21]).

$$\gamma_1 = \mathrm{Arctan}\left(\frac{B'}{h}\right) \tag{3.34}$$

$$L = (B' - A')\sin\left(\frac{\pi}{2} - \gamma_1\right) \tag{3.35}$$

$$K = \sqrt{h^2 + A'^2 - L^2} \tag{3.36}$$

$$X' = L\left(1 - \frac{2X_0}{I_\mathrm{p}}\right) \tag{3.37}$$

$$\gamma_2 = \mathrm{Arc\,tan}\left(\frac{X'}{K}\right) \tag{3.38}$$

$$D_\mathrm{H} = h\tan(\gamma_1 + \gamma_2) \tag{3.39}$$

$$D_\mathrm{o} = \sqrt{D_\mathrm{H}{}^2 + h^2} \tag{3.40}$$

where X_0 is the number of pixels between the image bottom positions to the point that has the lowest y value in the object, I_p stands for the height of the object in pixels. h, A', and B' are constant parameters and can be calculated off-line, h is the height of the camera from the ground surface, A' represents the distance in the camera's field of view (such that if the object is positioned there, the bottom of the object is seen in the lower part of the image), and *B'* is the distance (such that if the object is located at that position, the bottom of the object is seen in the image center).

The Jamzad et al. [21] hardware architecture for object distance measurement is costly compared with the proposed object distance in this thesis, because Jamzad et al. [21]:

- Used a CCD camera with a 4.5 mm lens in front and two webcams for the sides' rear view (whereas our proposed method is implemented with a normal webcam);
- Used a captured card with resolution of 704×510 pixels for all of the robots;
- Used two serial ports to communicate with the control unit;
- Used two microcontrollers in the control unit, etc.

Another method of object distance measurement, proposed by Chiang et al. [28], is based on depth estimation using the triangulation approach, which is illustrated in Figure 3.14. In the triangulation method, the first step is to find the focal length of the lens, which can be calculated using Equations (3.41-44), as follows:

$$\alpha = \tan^{-1}(\frac{D_{\mathrm{H1}}}{h}) \tag{3.41}$$

$$\theta = \tan^{-1}(\frac{h}{D_{\mathrm{Hmax}}}) \tag{3.42}$$

$$\beta = \frac{\pi}{2} - \alpha - \theta \tag{3.43}$$

$$f = \frac{I_{\mathrm{pmax}}}{\tan(\beta)} \tag{3.44}$$

Finally, the distance between point *p* and the camera can be calculated using Equations (3.45 -47).

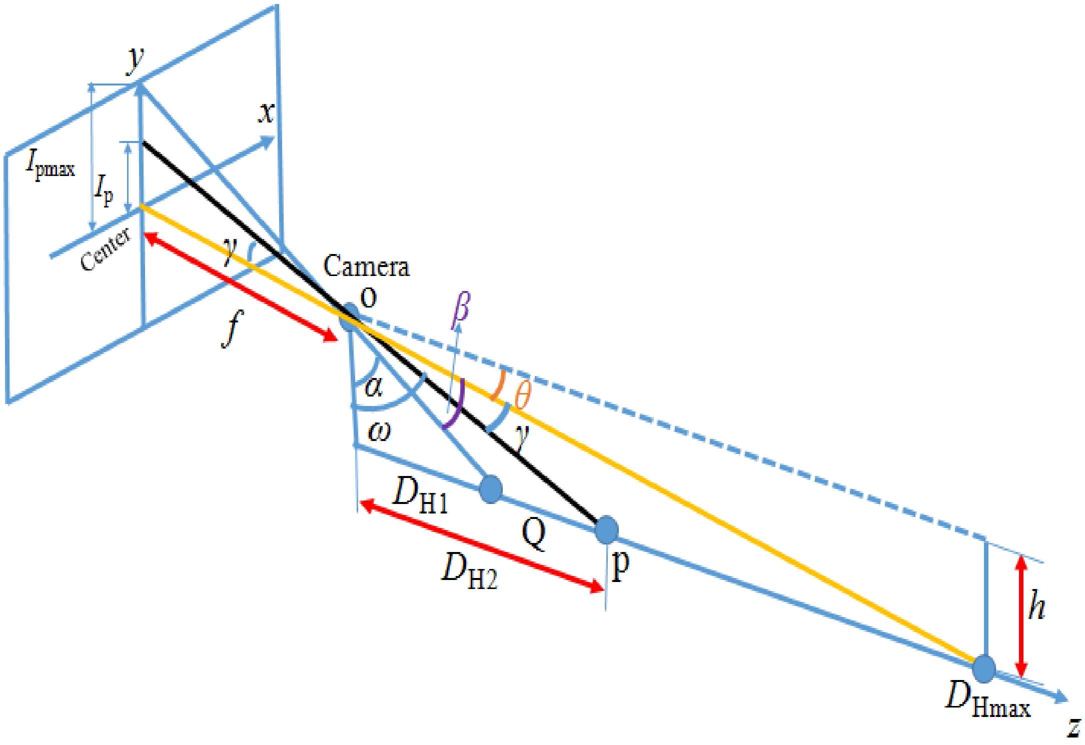


Figure 3.14: Depth estimation using triangulation method (reproduced from reference [28]).

$$\gamma = \tan^{-1}\left(\frac{I_p}{f}\right) \tag{3.45}$$

$$\omega = \alpha + \beta - \gamma \tag{3.46}$$

$$D_{H2} = h \tan(\omega) \tag{3.47}$$

The object distance proposed by Chiang et al. [28] is applicable in various illumination and complex outdoor environments. The average error for depth estimation in Chiang et al. [28] was reported to be about 48.7 mm, which is higher than the proposed method in this thesis. The results of Chiang et al.'s [28] object distance is illustrated in Figure 3.15, which shows a comparison between the real and the measured object's positions.

Taha and Jizat [49] proposed a method for the collision avoidance of an automated guided vehicle, using monocular vision. Their method is shown in Figures

3.16 and 3.17, where the camera's field of view is governed by 2θ in the X-direction and 2α in the Y-direction. If (u,v) is a point in the image plane, (x,y), which is a point in the global coordinate, can be derived.

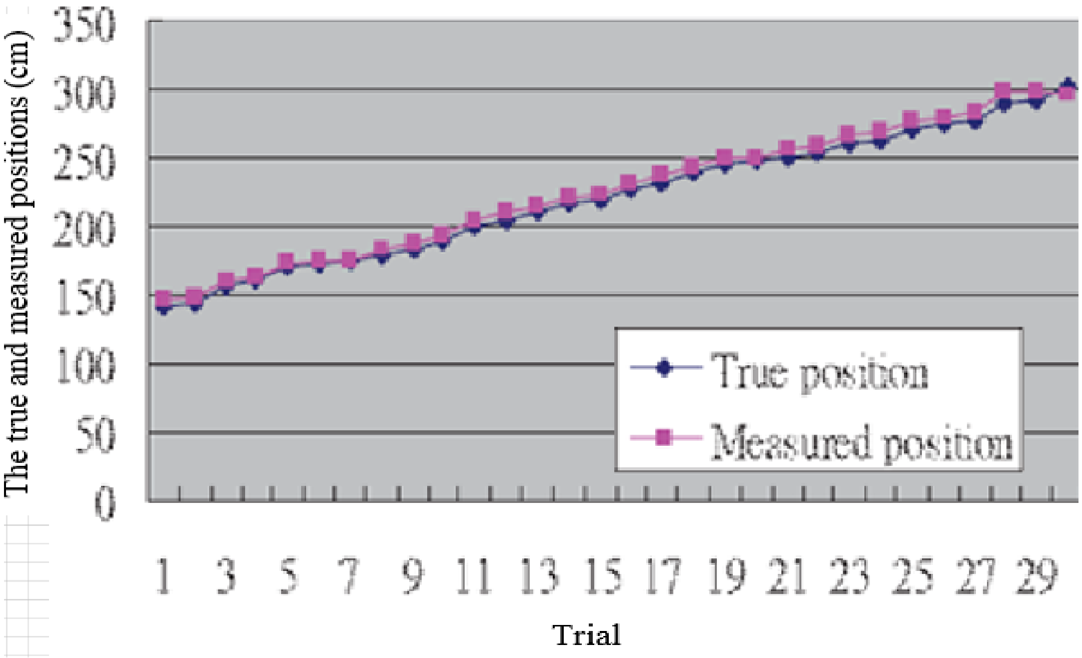


Figure 3.15: Comparison of actual and measured object distances [28].

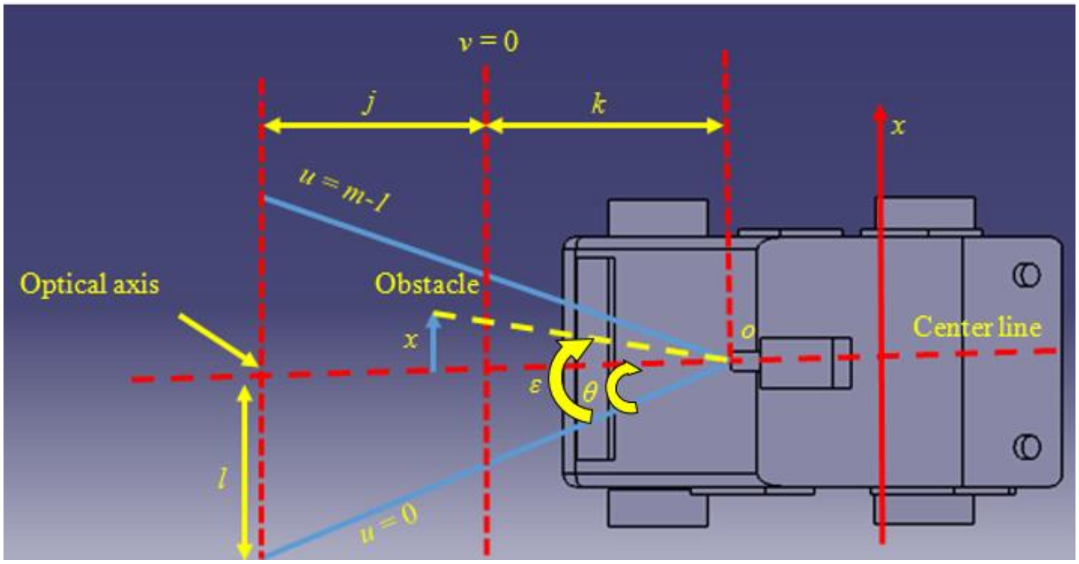


Figure 3.16: Moving car top view (reproduced from reference [49]).

Each pixel from the image corresponds to an angle (ε) of $\frac{2\theta(u)}{(m-1)}$ horizontally and an angle (ζ) of $\frac{2\alpha(v)}{(n-1)}$ vertically. α and θ are the characteristics of the camera, and the camera definition is (m×n) [49].

$$k = h\tan(\beta) \tag{3.48}$$

$$\tan(\theta) = \frac{l}{(j+k)} \tag{3.49}$$

$$\tan(\alpha+\beta) = \frac{(j+k)}{h} \tag{3.50}$$

$$\alpha+\beta+\delta = 90° \tag{3.51}$$

where h is the height of the camera from the floor, j is the distance in the xy-plane from v = 0 to the center of the image, k is the horizontal distance in the xy-plane from the camera to v = 0, and l is the horizontal distance in the xy-plane from the center image to u = 0. It is noteworthy that if the tilt angle δ increases, the values of j, k, l, and β would reduce. The object coordinates (x, y) can be expressed in terms of image coordinates (u,v) using tan (ε-θ) and tan (β+ζ). Therefore, we have the following equations:

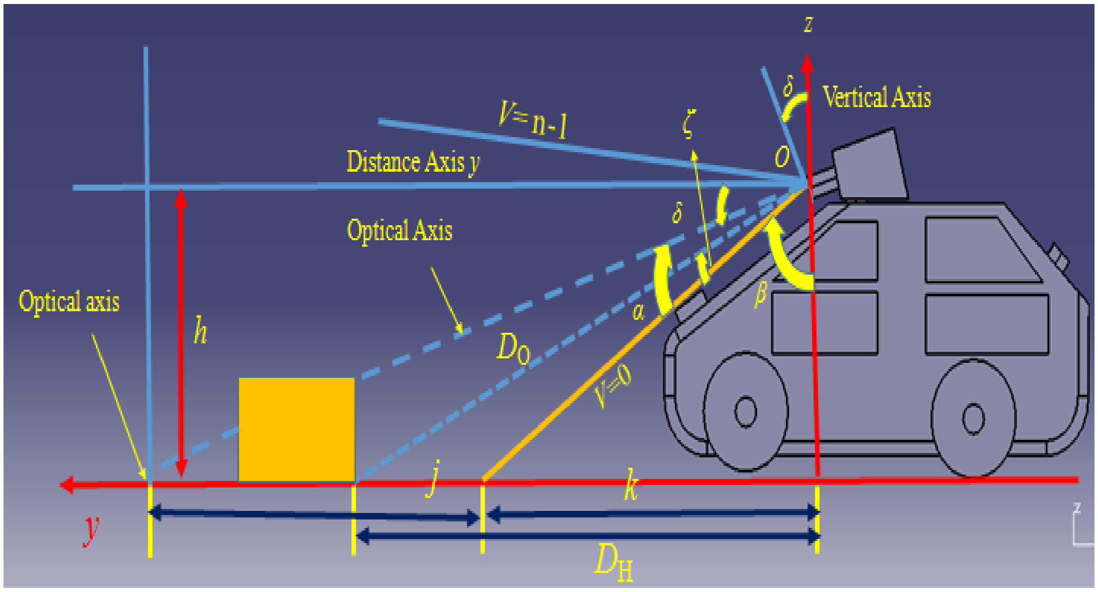


Figure 3.17: Moving car side view (reproduced from reference [49]).

$$D_{\mathrm{H}} = h \tan(\beta + (\frac{2\alpha(v)}{(n-1)}) \qquad (3.52)$$

$$D_{\mathrm{O}} = \sqrt{D_{\mathrm{H}}{}^2 + h^2} \qquad (3.53)$$

where D_{H} is the horizontal distance of the object from the camera on the ground and D_{o} is the oblique distance of the object from the camera, respectively. To improve data accuracy, we use D_{o} as our experimental data to construct the improved object distance algorithm using the least square method. The modified object distance measurement equation using this method is as follows:

$$D_{\mathrm{ol}} = (c_0 + c_1 v + c_2 v^2) h \qquad (3.54)$$

where c_0, c_1, c_2 coefficients can be obtained using least square equations, h is the camera height, and v is the pixel difference from the object to the beginning of the camera's field of view. The Taha and Jizat [49] technique is utilized in Chapter 5 of this thesis to obtain an object distance measurement using a single camera with variable pitch angle improved by least square optimization.

Another easy and accurate method of object distance measurement using a camera with variable pitch angle is defined by integrating the ratio of angle and pixels of a moving object. Figure 3.18 demonstrates the object distance estimation using the ratio method. In this method, the distance traveled on the ground is calculated and then used to calculate the oblique distance of the object from the camera, using Equations (3.55-58).

$$\frac{\gamma_{\mathrm{r}}}{2\alpha} = \frac{N_{\mathrm{p}}}{N_{\mathrm{pmax}}} \qquad (3.55)$$

$$\tan(\theta) = \frac{k}{h} \qquad (3.56)$$

$$D_{\mathrm{H}} = h \tan(\theta + \gamma_{\mathrm{r}}) \qquad (3.57)$$

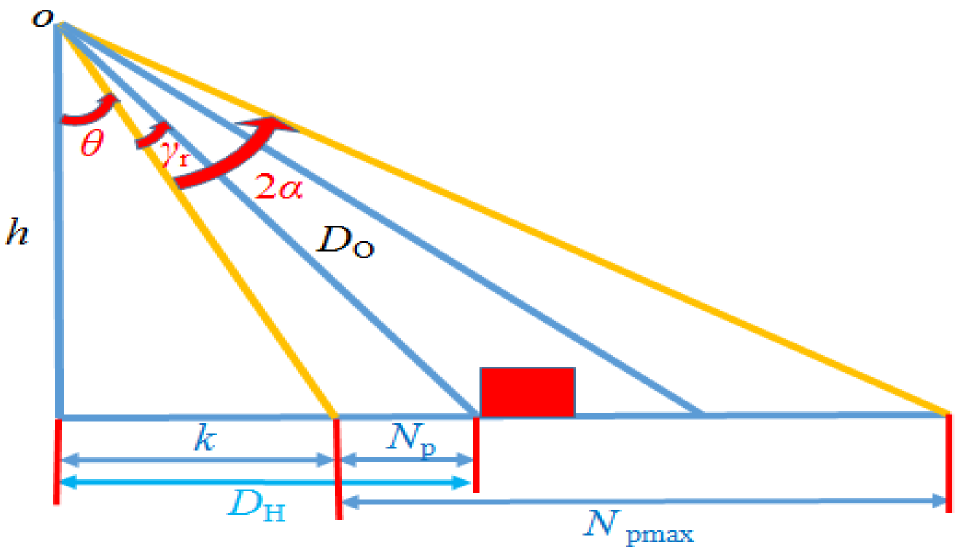


Figure 3.18: Object distance measurement using ratio method.

$$D_O = \sqrt{h^2 + D_H{}^2} \tag{3.58}$$

where γ_r is the angle of object from the beginning of the camera's field of view, 2α is the vertical angle of the camera's field of view (field of view in the y-axis direction), N_p is the sum of the pixels from the beginning of the field of view to the current position of the object, and N_{pmax} is the total pixels in the Y-direction of the camera's field of view.

3.10 Conclusion

In this chapter, the background knowledge needed for this thesis was discussed and some of the more common methods for calculating object distances for a single camera with variable pitch angle and a single fixed camera were described. The algorithms to be implemented in this thesis were chosen based on the methodologies discussed throughout this chapter. The next chapter describes object distance measurement in greater detail using a single fixed camera, which is partially selected form our conference paper.

Chapter 4

OBJECT DISTANCE MEASUREMENT USING A SINGLE FIXED CAMERA

4.1 Introduction

Image-based distance computation techniques have recently become an area of major research interest in the fields of robotic and computer vision. The three approaches for using image-based distance computation techniques are 1) stereovision-based, 2) monovision-based, and 3) time-of-flight camera.

The stereovision-based method uses two cameras to find the depth and the disparity map using a complex method. This technique is highly accurate but requires extensive computation time due to the simultaneous processing of many images of the same object. Moreover, implementing this technique is expensive (as it requires two cameras) and the accuracy of stereovision fails with increases in distance to the object compared with the baseline distance between two different views. On the other hand, the monovision-based approach is comparatively less expensive than the stereovision-based method, as it requires only one camera [50]. The time-of-flight depth estimation technique is used to find the depth information by measuring the total time required for light to transmit and reflect from an object.

Generally speaking, it is difficult to separate an incoming signal, since the signal depends on many parameters such as intensity of the reflected light, intensity of the background light, and the dynamic range of the sensor [4]. Researchers always look for inexpensive, uncomplicated, and accurate techniques. Applying such superior techniques requires researchers to tackle several other challenges like object detection, obstacle avoidance, and location finding.

Two approaches for estimating the location of any object are contact and non-contact methods. Nowadays, the non-contact distance measurement algorithm becomes useful in a wide range of applications where having actual physical contact with the object is not possible [51]. High accuracy and time saving are some advantages of using a non-contact measurement technique. Clarke and Williams [52] list the benefits of using a non-contact measurement system as lower inspection costs, better quality control, faster production, smaller tolerances, and fewer defects. Non-contact distance measurement is mainly used for quality control in manufacturing. Clarke and Williams [52] cited six different measurement systems to acquire surface information in one, two, and three dimensions. Single point optical triangulation, ultrasound, and time-of-flight techniques are utilized to get one-dimensional surface information. In addition, photogrammetry and laser tracker techniques can measure 3D surface information directly. Table 4.1 summarizes these techniques and shows the areas in which they are practiced.

Table 4.1: Six non-contact measurement systems.

Measuring system and some typical objects to be measured				
Single point optical triangulation	**Photogrammetry**	**Ultrasound**	**Time-of-flight**	**Laser trackers**
Industrial plant scanning	Aerospace metrology	Liquid level measurement	Civil engineering surveying	Robot tracking, calibration, and testing
Archaeological artifacts	Automobile manufacture	Counting objects on production line	Profiling rock faces in quarries	Aircraft manufacturing
Printed circuit boards	Gait analysis	Camera focusing	Tunnel profiling	Verification of jig design
Road surfaces	Mapping	Robotic sensing for navigation	Hydrographic surveys of buoys, barge and oil rigs	Reverse engineering
Building facades	Missile or plane tracking	Vehicle obstacle detection	Aerial surveys	Inspection and alignment
	Virtual reality	Wall-to-wall distance measurement	Range and bearing information	Surfaces

4.2 Problem Definition

The fundamental image formation technique is based on the pinhole lens model. In this model, the mapping relationship between the 3D global coordinates and the 2D image coordinates plays an important role. Let us consider ***M*** in Figure 4.1 to be a point in the global coordinate relative to the camera's frame. In this case, ***m*** would be the projection of ***M*** onto the image plane. Under the pinhole assumption, these two points and the origin of the camera's frame (the center of the projection) that is in the back of the image plane at a distance *f* are collinear [44]. Due to the perspective error, a set of straight lines on the ground plane converges to a point in the image plane, which is called the vanishing point. If the camera's axis is parallel to the optical plane, the vanishing point would be the center of the image plane. The pinhole lens model of image formation is applied in the present research, where the problem of mapping the object's location is described using measured distance and frame transformation.

For visual servoing purposes, there are five "standard" frame names associated with a robot and its workspace: the base frame, the station frame, the wrist frame, the tool frame, and the goal frame. The base frame is shown as {B} and is the fixed part of the robot, which is sometimes called the Link 0. The station frame {S} is called the universe frames, since all actions of the robot are made corresponding to this frame. The wrist frame {W} is the last link of the manipulator and is normally defined relative to the base frame. The tool frame {T} is assigned at the end of any tool that a robot is holding. Finally, the goal frame {G} is the location to which the tools need to move [53]. Figure 4.2 shows entire frame assignments to find the distance of the moving object with respect to a robot's hand. In Figure 4.2, the oblique object distance ***AG*** (equivalent to D_o) can be measured through the proposed algorithm in this thesis. The camera's location with respect to the base frame is shown with vector ***BA*** (known by measurement), and ***BT*** is obtained through forward kinematics and using the DH parameter of the robot. Using this information, it will be easy to find vector ***TG*** as the distance of the moving object with respect to the tool frame of the robot, and can be calculated using Equations (4.1-2).

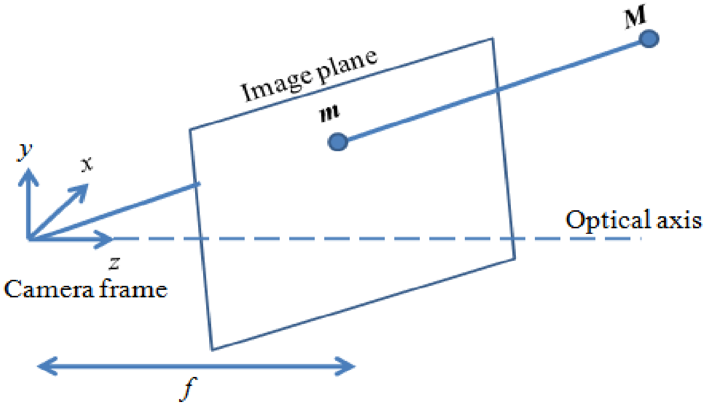


Figure 4.1: Camera coordinate frame (reproduced from reference [44]).

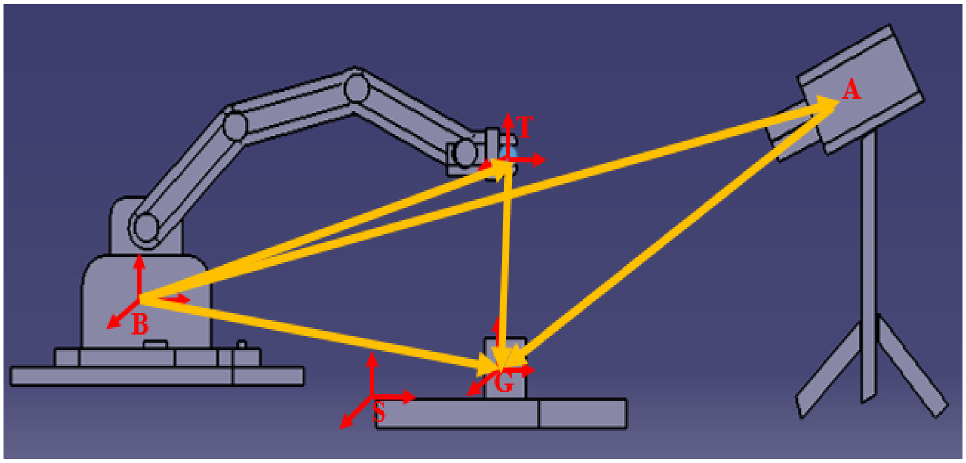


Figure 4.2: Standard frame assignment.

$$\boldsymbol{BA} + \boldsymbol{AG} = \boldsymbol{BG} \tag{4.1}$$

$$\boldsymbol{TG} = \boldsymbol{BG} - \boldsymbol{BT} \tag{4.2}$$

The object distance is defined as the distance of the desired object from the center of the lens. If the desired object is not located on the optical axes, it is called oblique object distance. The image distance is defined as the distance from the focused image to the center of the lens. The proposed object distance measurement in this thesis is based on finding the closest point from the object to the bottom-center of the camera’s field of view. Equations (4.4-12) are used to calculate the distance from the object to the camera, where *a* is a known value obtained by measurement, D_o is object distance, *h* is the height of the camera from the ground, and C_f is the calibration factor. x_3, y_3,x_4, y_4 are the converted image pixels into millimeters using the calibration factor. Figure 4.3 shows the coordinate system and the camera's field of view on the ground.

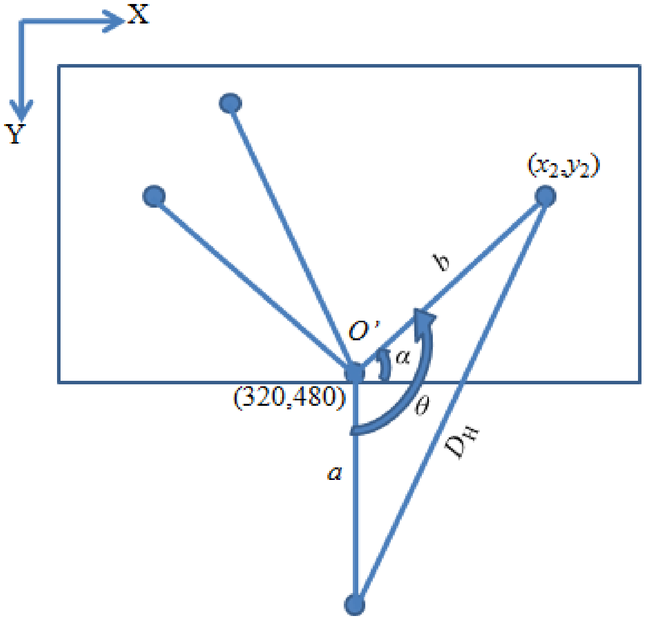


Figure 4.3: Proposed object distance calculation method.

$$\alpha = \arctan\left(\frac{480 - y_2}{x_2 - 320}\right) \tag{4.4}$$

$$\theta = \frac{\pi}{2} + \alpha \tag{4.5}$$

$$x_3\,(\mathrm{mm}) = x_2\,C_f \tag{4.6}$$

$$y_3\,(\mathrm{mm}) = y_2\,C_f \tag{4.7}$$

$$x_4\,(\mathrm{mm}) = 320\,C_f \tag{4.8}$$

$$y_4\,(\mathrm{mm}) = 480\,C_f \tag{4.9}$$

$$b = \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2} \tag{4.10}$$

$$D_H = \sqrt{a^2 + b^2 - 2\,a\,b\cos(\theta)} \tag{4.11}$$

Finally, D_o, which is the oblique object distance, can be calculated using Equation (4.12), as follows:

$$D_O = \sqrt{{D_H}^2 + h^2} \tag{4.12}$$

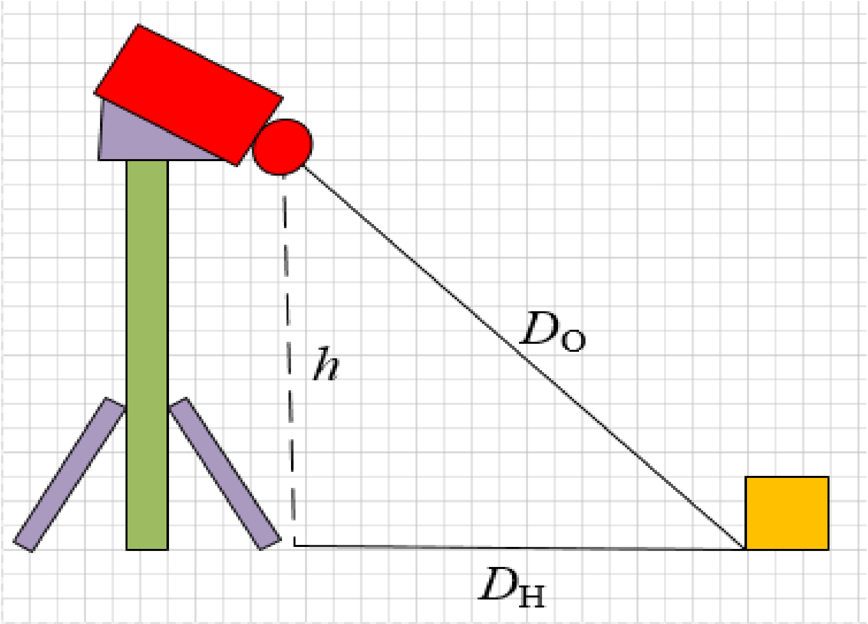


Figure 4.4: Oblique object distance calculation method.

where, in Equation (4.6-9), the values in the image plane (i.e., image coordinate) given in pixel are mapped to the coordinates attached to the physical field of view of the camera at point *O'* using the calibration factor. To find the required scale factors, the length and width of the camera's field of view are measured and used, along with the camera definition (640×480). In Equation (4.11), D_H is the horizontal distance of the object from the camera on the ground. In Equation (4.12), h is the height of camera from the ground surface, and D_o is the oblique distance of the object from the camera.

4.3 The Image Processing Algorithm

In this section, some parts of the proposed image-processing algorithms in this research are described. The major image processing algorithm in this section is classified into the following four parts:

- Background estimation
- Object tracking
- Feature extraction
- Feature analysis

This algorithm starts with an input video source and separates the background from the object using a thresholding technique. Then, the desired object is tracked frame by frame. The object's features are extracted after completion of the tracking process from the image. Finally, using the analyzed feature, the object distance, width, and length will be calculated.

The color space conversion is used to change the color information into different color spaces. Each pixel has a unique color defined by the amount of red, green, and blue. If each color has a range of 0-255, then the total different possible colors that these three major colors can produce will be 255^3=16,777,216 [54]. Figure 4.5 demonstrates a typical RGB Image.

The idea of segmentation is to simplify or divide an image into meaningful components. There are many approaches to segmentation that is concerned with finding features in an image, such as edges or partitioning the image into homogenous regions (object or background). Where each pixel classifies to either of the two regions, the resulting image is called a binary image. Pixels with a gray level greater than one threshold are considered to be objects, and pixels with a gray level less than or equal to the threshold are considered to be background. In cases where multiple objects with pixels above the threshold are presented in an image, a unique label is assigned to each connected component.

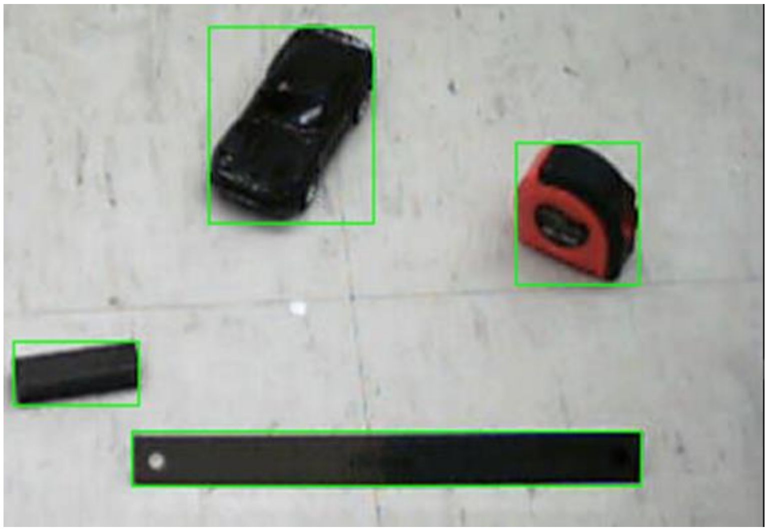


Figure 4.5: RGB model of the image.

The relationship between connected components is either 4-connectivity (blob) or 8-connectivity (blob). In other words, a 4-connectivity component has four neighbors: above, below, right, and left of the pixel. On the other hand, an 8-connectivity component has the above neighbors along with those pixels that are diagonally adjacent [44].

An auto threshold technique is applied in the algorithm in order to identify different objects in a camera's field of view. The Otsu method [55] is a widely referenced threshold technique. To establish an optimum threshold, Otsu suggested minimizing the weighted sum of within-class variances for the object and background pixels. However, the minimization of such pixels stimulates the maximization of between-class scatter. Otsu's method would be especially useful for dealing with a large number of pixels within each class with close pixel values. Figure 4.6 shows an image of a sample threshold using the auto-threshold technique.

Morphological opening can remove small objects from an image while preserving the shape and size of larger objects in the image. Disc structuring is applied as the morphological opening element to create a flat disk-shaped object with a radius of 2.

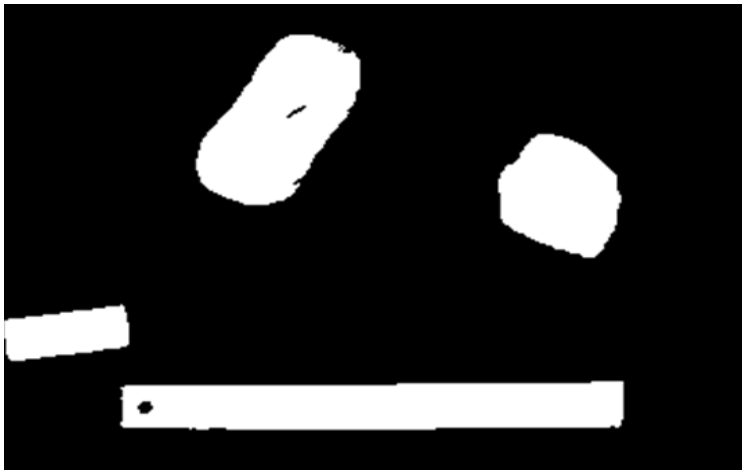


Figure 4.6: Image obtained after threshold.

4.3.1 MATLAB Function and Blob Analysis Blocks

The MATLAB Function is used in the proposed algorithm of the thesis after the connected components (blobs) are labeled. The input to the MATLAB Function block is a frame with black and white labeled objects. The output of the MATLAB Function is the object distance and dimension (e.g., width and length) calculated based on the proposed method. The blob analysis block extracts several features from each blob in the binary frame that takes them as input. The output from this block is a matrix consisting of several features for each of the blobs in the binary frame. The object's length and width calculation is based on the correction of the "Major axis", "Minor axis", "Centroid" and "Orientation" features and are shown as the output of the blob analysis block.

4.3.2 2D Finite Impulse Response (FIR) Filter

Filtering is another important technique that has to be chosen accurately to remove the noise from the image without reducing its sharpness. The finite impulse response (FIR) filter is commonly used for coding artifact removal due to the simplicity of the implementation and its stability characteristic. A digital FIR filter corresponding to pixel (*i*, *j*) of a 2D image signal can be expressed as:

$$y(i, j) = \sum_{k=-N}^{N} \sum_{l=-N}^{N} w_{kl}\, x(i+k, j+l) \tag{4.13}$$

In Equation (4.13), *y*(*i*, *j*) is denoted as the pixel values at point (*i*, *j*) after filtering, and *x*(*i*, *j*) is denoted as the pixel values at point (*i*, *j*) before filtering. Additionally, w_{kl} and *N* represent the filter coefficient and the duration of the impulse response of the FIR filter, respectively. The adaptive filter is used to generate filter coefficients w_{kl}, which are the properties of the FIR filter [56]. For all of the simulations in this research, a predefined 2D random Gaussian noise and averaging filter is applied so as to improve the image contrast.

4.3.3 Image Complement

An image complement is selected to compute the complement of a binary or intensity image. Image complements switch the values of the zero- and one-valued pixels, which are identical to an image negative. Figure 4.7 provides an example of an image complement.

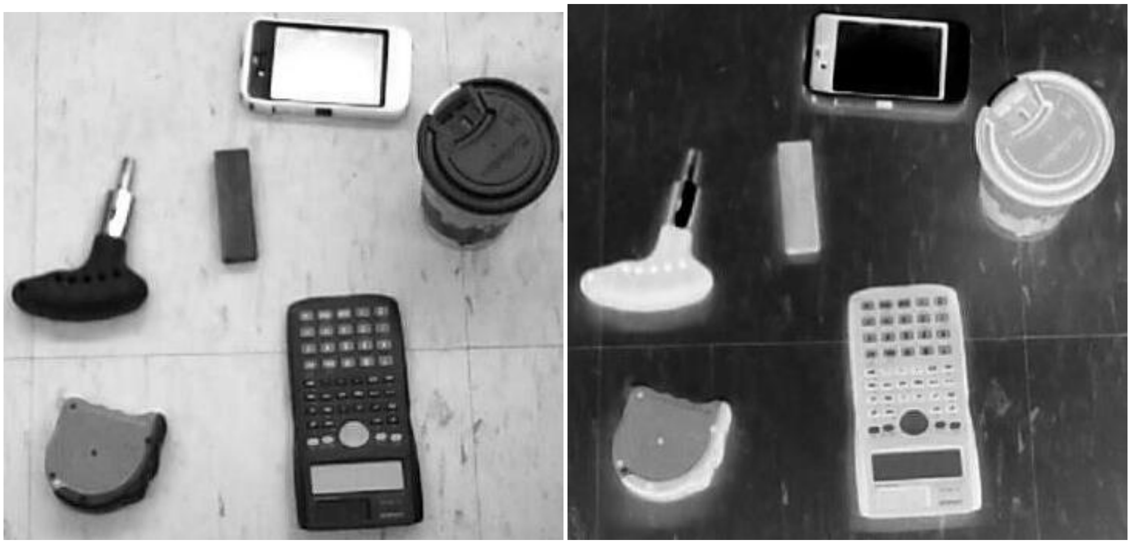


Figure 4.7: An example of a typical image complement.

4.3.4 Averaging Subsystem

The averaging subsystem method is useful for stabilizing a system by applying consecutive measurement values for the system. In the proposed algorithm, four consecutive measurement values have been assigned to mediate fluctuations arising from noise in the system [57].

4.4 Experimental Setup

In this research, Logitech Quick Cam® Communicate STX™ (Figure 4.8) is used to capture RGB images with a resolution of 640 × 480 pixels. The captured images are then converted to intensity by color space conversion, since most of the applications require the intensity of the image. In order to reduce the noise from the images of the desired object, a 2D FIR Filter is applied. Figure 4.9 shows the completed algorithm for the object distance measurement.

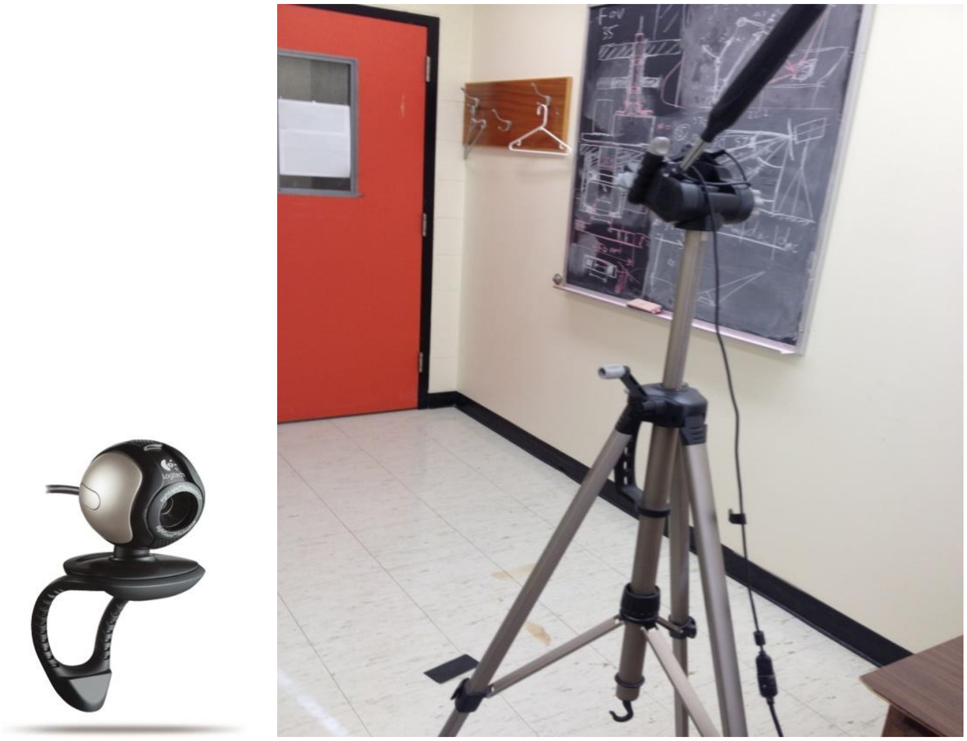


Figure 4.8: Logitech QuickCam® Communicate STX™.

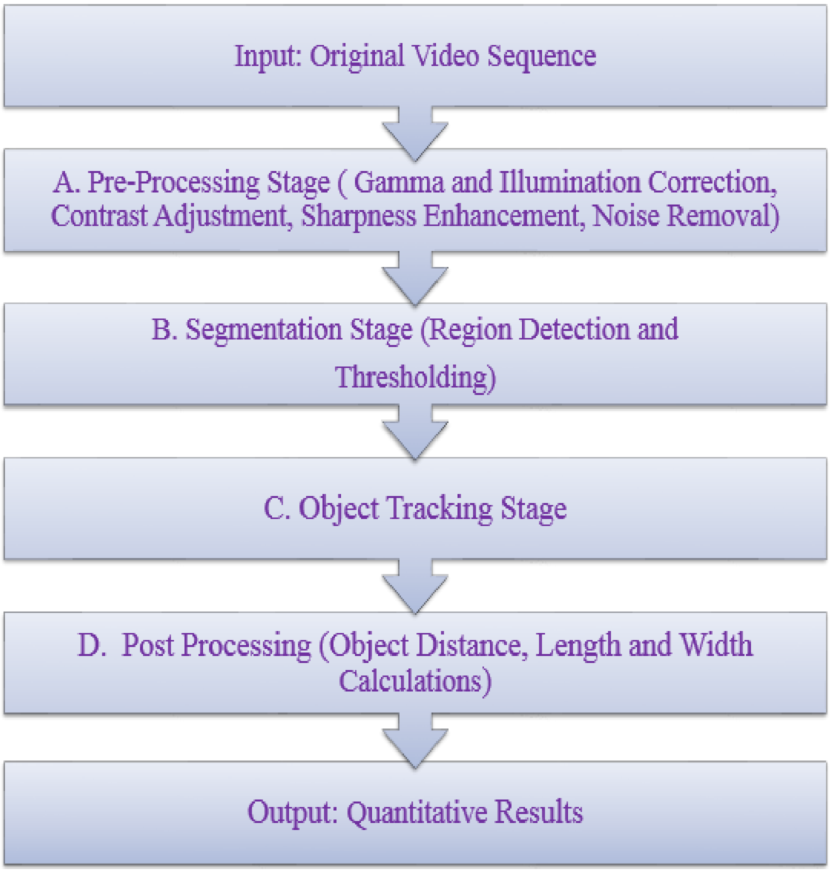

Input: Original Video Sequence
A. Pre-Processing Stage (Gamma and Illumination Correction, Contrast Adjustment, Sharpness Enhancement, Noise Removal)
B. Segmentation Stage (Region Detection and Thresholding)
C. Object Tracking Stage
D. Post Processing (Object Distance, Length and Width Calculations)
Output: Quantitative Results

Figure 4.9: The proposed algorithm for object distance measurement.

The MATLAB Function is incorporated in this research to carry out the required code in the subsystem for tracking the desired object. In the next step, implemented in the subsystem, the desired object distance is measured using the resulting image data. From implementing the object distance measurements that are shown as D_{o1} and D_{o2} for a moving object at time t_0 and t, the horizontal distance traveled by the object on the ground could be calculated using the law of cosines (Equation 4.14). The initial values are chosen as D_{o1}=1160 mm and θ_0=55°.

$$D_{O2}{}^2 - D_{O1}{}^2 - D_H{}^2 + 2\, D_{O1} D_H \cos(\theta) = 0$$

$$D_H = \frac{2D_{O1}\cos(\theta) \pm \sqrt{4\, D_{O1}{}^2 \cos^2(\theta) - 4(D_{O1}{}^2 - D_{O2}{}^2)}}{2} \qquad (4.14)$$

Figure 4.10 shows the horizontal distance of the object traveled on the ground.

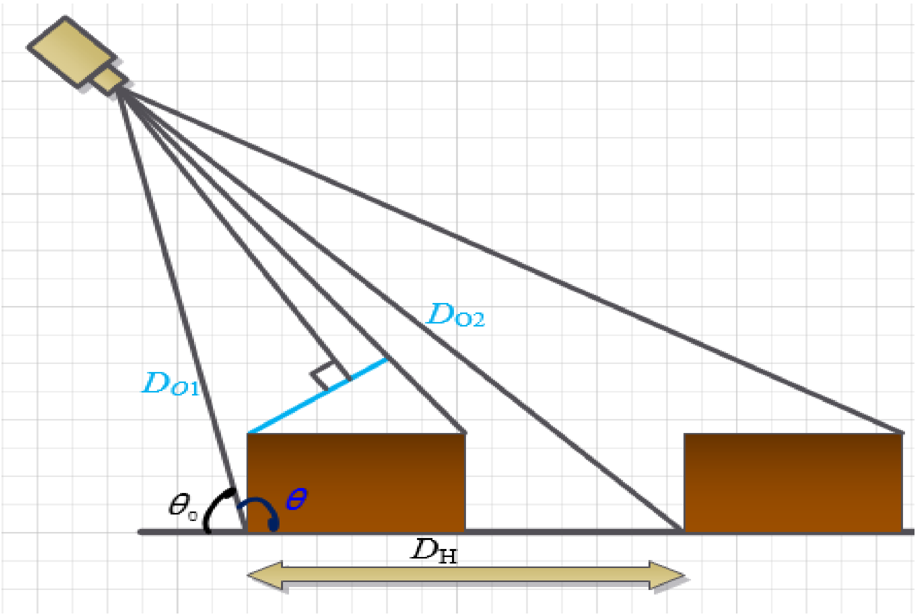


Figure 4.10: Distance traveled by the object on the ground.

4.5 Experimental Results and Discussion

In this research experiment, the distance of the desired object using the proposed method and a single fixed camera is extracted. The desired object is tracked with a green bounding box, which is shown in Figure 4.5. Tables 4.2 and 4.4 compare the experimental results of the proposed method with the results of Joglekar et al. [48] for two different objects. In addition, Table 4.3 demonstrates the average distance error for the calculator. The three tables respectively show the object distance average errors for three different samples of one rectangular block 25.43 mm×79.22 mm (W×L), one calculator 84.62 mm×156 mm (W×L) and one irregular toy car 70.21 mm×145.54 mm (W×L), which are measured at any location in the camera's field of view. In addition, Figures (4.11-13) show the tabulated results in Tables (4.2-4) in the graphical form. Two related works as well as the proposed method for the object distance measurement are compared in Table 4.5. Finally, Tables (4.6-7) show the average error for the corrected length and width of the calculator, respectively. It should be mentioned that the accuracy of the measurement using the measuring tape is ±0.5 mm. This experiment is implemented with a low resolution camera to check whether the proposed algorithm and the accuracy of the measurement could be improved by using a high-resolution camera.

Image noise is the main unavoidable reason for producing errors during the image acquisition stage. Such errors can occur when finding the exact point of contact with the object on the ground. Another potential cause of error is the variation in image illumination across a camera's field of view.

Table 4.2: Average distance error measurements for the rectangular block.

Trial	*Real distance (mm)*	*Proposed measured distance (mm)*	*Joglekar et al. [48] measured distance (mm)*	*Proposed distance error (mm)*	*Joglekar et al. [48] distance error (mm)*
1	1181	1176	1186	5	5
2	1213	1231	1131	18	82
3	1215	1190	1150	25	65
4	1272	1302	1622	30	350
5	1320	1340	1134	20	186
6	1323	1320	1399	3	76
7	1367	1414	2275	47	908
8	1432	1438	1424	6	8
9	1436	1468	1209	32	227
10	1481	1529	1433	48	48
11	1513	1521	1306	8	207
12	1548	1580	1183	32	365
13	1585	1617	1227	32	358
14	1621	1633	1156	12	465
15	1630	1615	1200	15	430
	Average absolute error			**23.43**	**269.64**

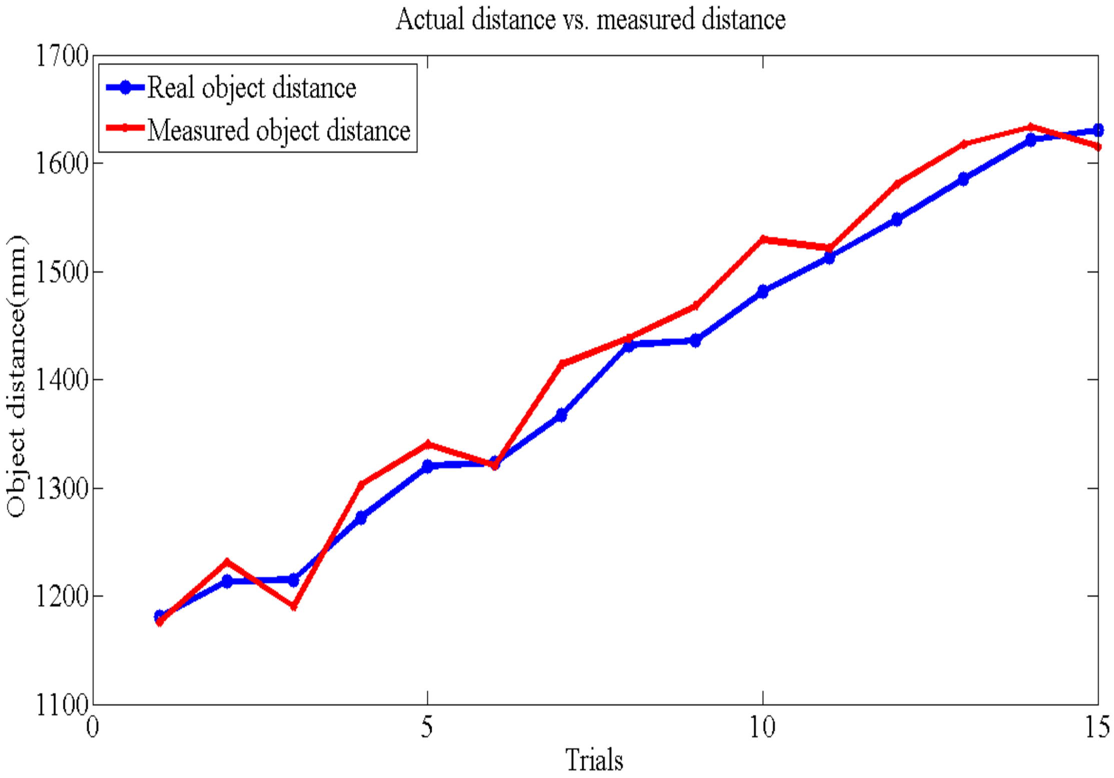


Figure 4.11: Object distance measurement for the rectangular block.

Table 4.3: Average distance error measurements for the calculator.

Trial	***Real distance (mm)***	***Proposed measured distance (mm)***	***distance error (mm)***
1	1183	1176	7
2	1181	1184	3
3	1196	1201	5
4	1219	1204	15
5	1203	1209	6
6	1274	1257	17
7	1241	1259	18
8	1253	1263	10
9	1258	1284	26
10	1302	1294	8
11	1300	1349	49
12	1319	1360	41
13	1334	1385	51
14	1349	1396	47
15	1379	1397	18
16	1413	1407	6
17	1378	1430	52
18	1430	1465	35
19	1429	1483	54
20	1450	1498	48
21	1519	1531	12
22	1496	1544	48
23	1558	1564	6
24	1526	1572	46
25	1538	1573	35
26	1609	1580	29
27	1567	1599	32
28	1571	1606	35
	Average absolute error		**27.1**

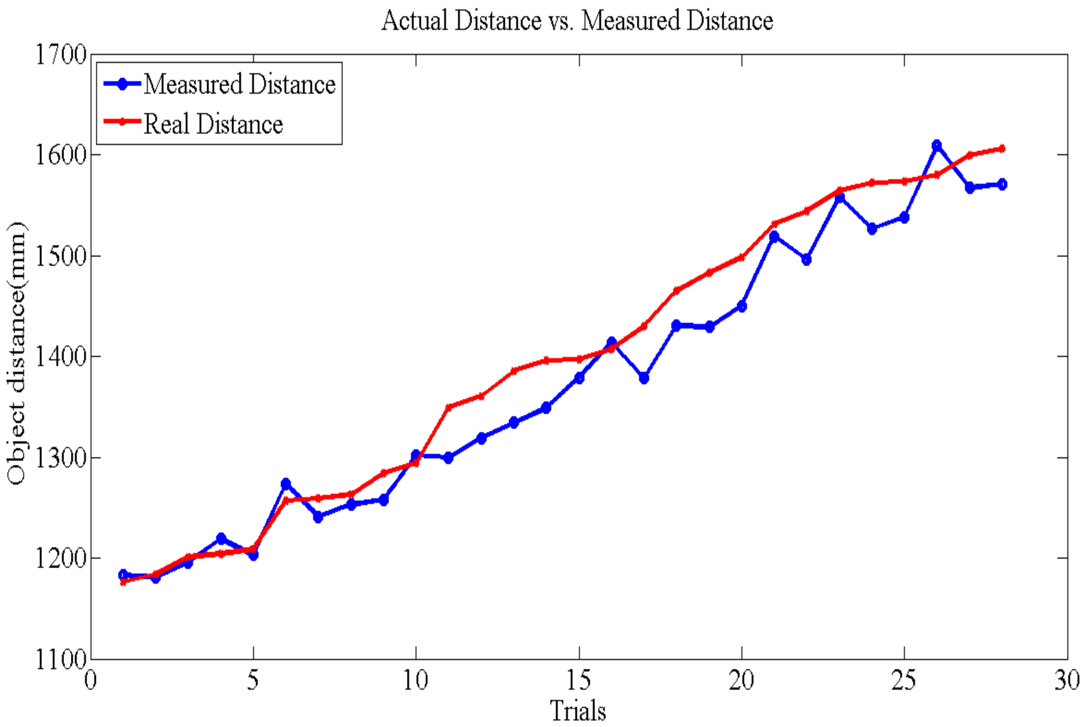


Figure 4.12: Object distance measurement for the calculator.

Table 4.4: Average distance error measurements for the toy car.

Trial	***Real distance (mm)***	***Proposed measured distance (mm)***	***Joglekar et al. [48] measured distance (mm)***	***Proposed distance error (mm)***	***Joglekar et al. [48] distance error (mm)***
1	1179	1173	1184	6	5
2	1206	1183	1160	23	46
3	1250	1262	1155	12	95
4	1253	1259	1204	6	49
5	1263	1238	1196	25	67
6	1295	1328	1953	33	658
7	1363	1352	1341	11	22
8	1373	1404	1196	31	177
9	1420	1458	2057	38	637
10	1480	1489	1344	9	136
11	1495	1542	1370	47	125
12	1503	1529	1202	26	301
13	1527	1570	1310	43	217
14	1580	1583	1236	3	344
15	1583	1613	1164	30	419
	Average absolute error			**16.50**	**381.50**

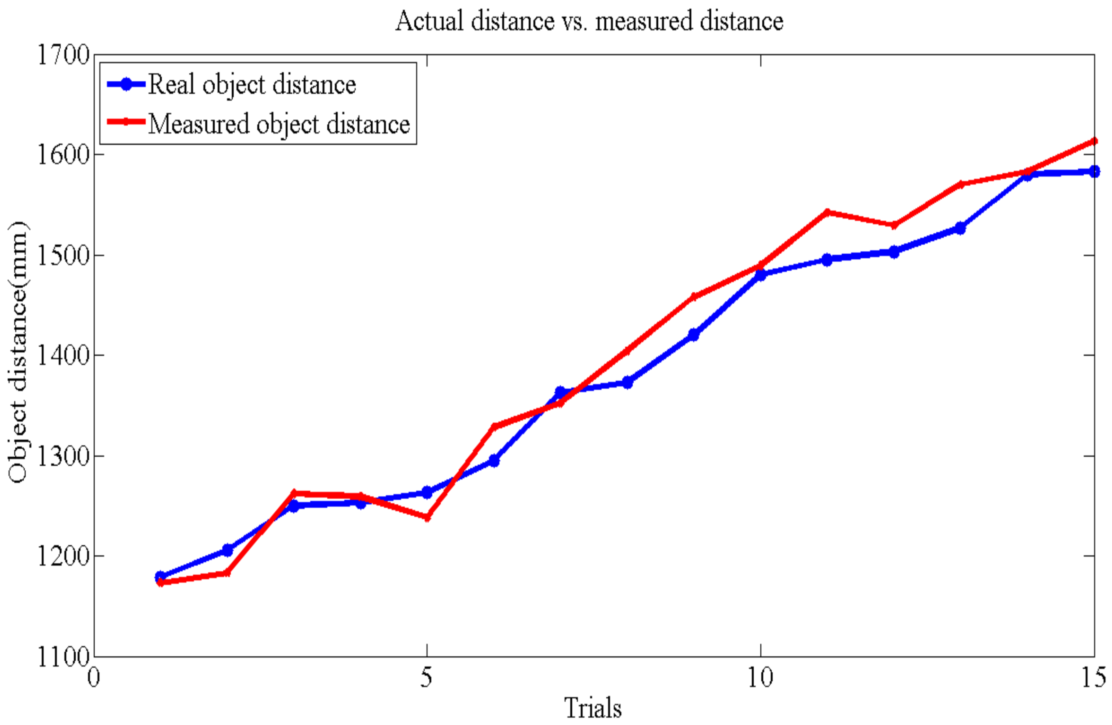


Figure 4.13: Object distance measurement for the toy car.

Table 4.5: Comparison of different methods for object distance measurement.

Method	*Characteristics* *(in terms of distance)*	*Nature of the method*
Gat et al. [58] (Optical axis is parallel to the ground)	Along the optical axis	Linear
Joglekar et al. [48] (Optical axis is parallel to the ground)	In-path and oblique	Non-Linear
Proposed Method (Optical axis is not parallel to the ground)	Anywhere on the field of view of the camera	Non-Linear

Table 4.6: Length correction for the calculator.

Trial	*Actual length (mm)*	*Measured length (mm)*	*Corrected length (mm)*	*Non-corrected length error (mm)*	*Corrected length error (mm)*
1	156	144.8	164.5	11.2	-8.5
2	156	146.2	166.7	9.8	-10.7
3	156	161.9	160.4	-5.9	-4.4
4	156	145.7	183.5	10.3	-27.5
5	156	144.4	164.6	11.6	-8.6
6	156	161.1	171.4	-5.1	-15.4
7	156	155.2	158.2	0.8	-2.2
8	156	154.8	161	1.2	-5
9	156	153.8	157.2	2.2	-1.2
10	156	157.1	170.7	-1.1	-14.7
11	156	153	152.2	3	3.8
12	156	143	151.1	13	4.9
13	156	150.7	155.8	5.3	0.2
14	156	151.5	148.4	4.5	7.6
15	156	134.3	146.3	21.7	9.7
16	156	121.4	146.1	34.6	9.9
17	156	140.7	146	15.3	10
18	156	121.9	141.4	34.1	14.6
19	156	121.6	139.1	34.4	16.9
20	156	139.5	143.4	16.5	12.6
21	156	151.3	159.6	4.7	-3.6
22	156	118.6	128.4	37.4	27.6
23	156	120.9	139.3	35.1	16.7
24	156	148.9	146.6	7.1	9.4
25	156	144.9	150.1	11.1	5.9
26	156	110.1	136.9	45.9	19.1
27	156	155.5	151.7	0.5	4.3
28	156	154	153.3	2	2.7
		Average error		**12.9**	**2.6**

The graphical representation for Table 4.6 is shown in Appendix A (Figure 1).

Table 4.7: Width correction for the calculator.

Trial	*Actual width (mm)*	*Measured width (mm)*	*Corrected width (mm)*	*Non-corrected width error (mm)*	*Corrected width error (mm)*
1	84.62	64.72	81.22	19.9	3.4
2	84.62	62.23	81.85	22.39	2.77
3	84.62	66.42	80.57	18.2	4.05
4	84.62	60.93	84.35	23.69	0.27
5	84.62	63.68	86.74	20.94	-2.12
6	84.62	63.61	75.53	21.01	9.09
7	84.62	64.4	74.48	20.22	10.14
8	84.62	63.12	72.65	21.5	11.97
9	84.62	65.29	75.48	19.33	9.14
10	84.62	62.49	73.05	22.13	11.57
11	84.62	64.02	72.02	20.6	12.6
12	84.62	66.87	74.84	17.75	9.78
13	84.62	64.03	71.11	20.59	13.51
14	84.62	67.73	75	16.89	9.62
15	84.62	70.29	78.28	14.33	6.34
16	84.62	70.97	85.27	13.65	-0.65
17	84.62	67.58	75.15	17.04	9.47
18	84.62	70.03	86.62	14.59	-2
19	84.62	68.47	80.33	16.15	4.29
20	84.62	69.09	75.28	15.53	9.34
21	84.62	63.31	66.74	21.31	17.88
22	84.62	72.66	81.5	11.96	3.12
23	84.62	71.07	79.38	13.55	5.24
24	84.62	67.21	69.41	17.41	15.21
25	84.62	65.32	67.48	19.3	17.14
26	84.62	74.93	86.29	9.69	-1.67
27	84.62	65.5	67.12	19.12	17.5
28	84.62	64.89	65.97	19.73	18.65
	Average error			**18.2**	**8.1**

Using the method presented by [58], it is possible to measure object distance when the optical axis of a camera is parallel to the ground. However, to find object distance anywhere in the field of view of a camera, the Joglekar et al. [48] method offers a possible solution. In Joglekar et al. [48] the in-path object distance and the oblique object distance are defined as the distance to the object that is on the optical axis and as the distance to the object that is not on the optical axis, respectively. We have investigated whether the Joglekar et al. [48] method can be used to estimate the object distance anywhere on the camera's field of view with the provided setup. The results from some experiments using Joglekar et al.'s [48] method show that as the object gets closer to the vanishing point, the object distance would rise and the error significantly increase. Therefore, where the camera has an angle with optical axis, methods such as [48] and [58] are no longer valid, as the object distance measurement would be a non-linear function of distance. Moreover, because the oblique object distance is calculated from the result of the in-path object distance divided by the cosine angle between the optical axis and the line joining the camera point to the object's point of contact, the error is relatively similar for the in-path object distance obtained. Therefore, the proposed vision-based object distance measurement technique is principally different from the optical techniques of [48] and [58].

4.6 Conclusion

In this chapter, an improved method was proposed to calculate the object distance using a single fixed camera, even if the object surface is not parallel to the camera and the object is not restricted to be vertically intersecting the optical axis. The proposed method is able to identify the desired object and extract the object features for moving and static objects. The experimental results show that the object distance average error for the rectangular block, the irregular toy car and the calculator are 23.43 mm, 16.50 mm and 27.1 mm, respectively.

Chapter 5

OBJECT DISTANCE MEASUREMENT USING A SINGLE CAMERA WITH VARIABLE PITCH ANGLE

5.1 Introduction

Distance measurement is the capability of a robot to understand its working environment and remotely manipulate 3D objects located in its workspace. A distance measurement system typically consists of a pair of cameras or alternative sensors such as laser and sound ranging technologies. However, in this thesis, the distance to the object is measured using a single camera with variable pitch angle (i.e., rotation about the y-axis). For the proposed method in this section, the camera is free to move up/down in the vertical plane; therefore, the only information required is the height and pitch angle of the camera with respect to the ground. In addition, the image processing algorithm is similar to the object distance measurement using a single fixed camera. However, the process and the implementation of the proposed object distance measurement are completely different than the single fixed camera method.

5.2 Problem Definition

The object distance measurement can be modeled as the transformation from a 3D Euler space to a 2D Euler space. The procedure of capturing an image by a camera involves three coordinate systems: the image coordinate system, the camera coordinate system, and the global coordinate system. The *Z* axis of the camera is vertical to the image plane so that the xy-plane will be parallel to the image plane. In Figure 5.1, the camera coordinate system and the global coordinate system are expressed by *XYZ* and *xyz*, respectively. The ***m*** is the projection of ***M*** (a point in the real world) on the mage plane [59]. The object distance measurement for this section is based on the method proposed in [49], which is improved by the least square optimization technique. In [49], the geometrical relations of the camera are practiced to calculate the world coordinate frame and the image coordinate frame. In these processes, the camera is required to be tilted to the point where the entire camera's field of view intersects with the floor.

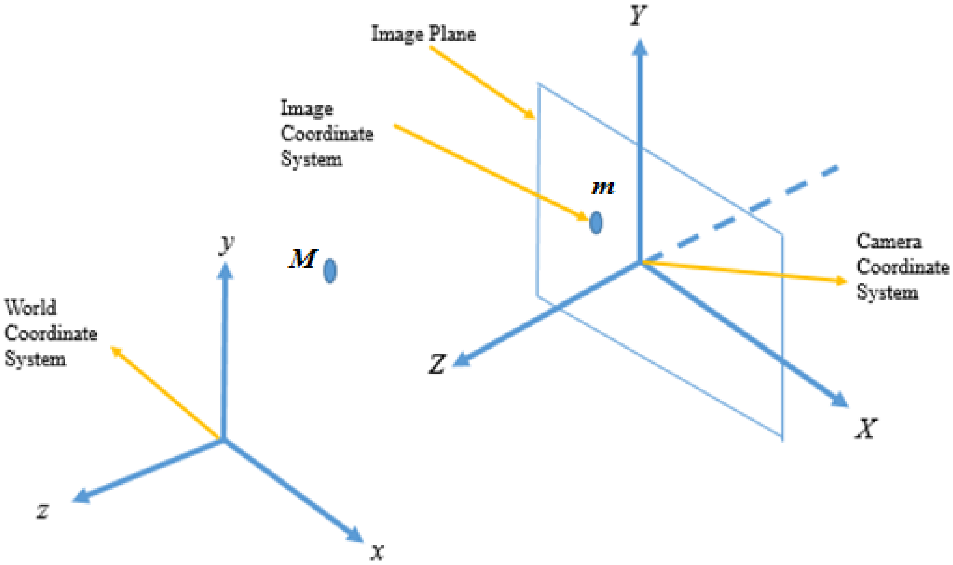


Figure 5.1: Three coordinate systems (Camera, Image, and World) (reproduced from reference [59]).

5.3 Image Processing Algorithm

5.3.1 Light Intensity

The eyes' capability to discriminate different brightness levels is essential, since digital images can be shown as a discrete set of brightness points. The range of the light intensity for a human visual system is somewhere around 10^{10} to the glare limit. The image is a 2D light-intensity function that is shown as *f*(*x*,*y*). In this function, the value of *f* at spatial coordinate (*x*, *y*) gives the brightness of the image at that specific point. The nature of the *f*(*x*,*y*) function can be characterized into two different components: 1) The amount of source light incident being viewed on the scene, and 2) the amount of light reflected from the object in the scene. Figure 5.2 demonstrates the color image before and after the brightness adaptation [60].

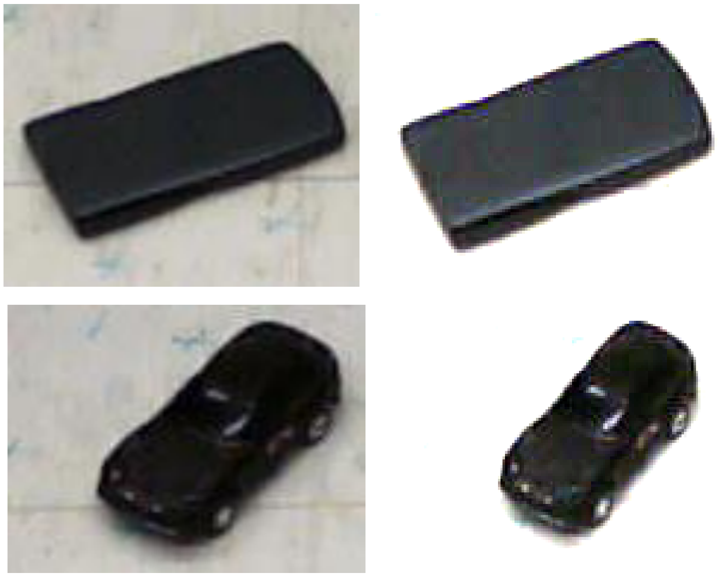


Figure 5.2: Image before and after the brightness adaptation.

5.3.2 Thresholding

One important approach in image segmentation is thresholding. The thresholding technique is used to distinguish an object from its background. Any number other than zero is attributed to the object, while zero refers to the background [60]. In other words, if the light intensity function value is more than the threshold value, the threshold image will be assigned a value of one. Otherwise, the threshold image will be assigned a zero value (Figure 5.3). A threshold image *g*(*x*,*y*) is defined as:

$$g(x,y) = \begin{cases} 1, & if\ f(x,y) > Th \\ 0, & if\ f(x,y) \leq Th \end{cases} \tag{5.1}$$

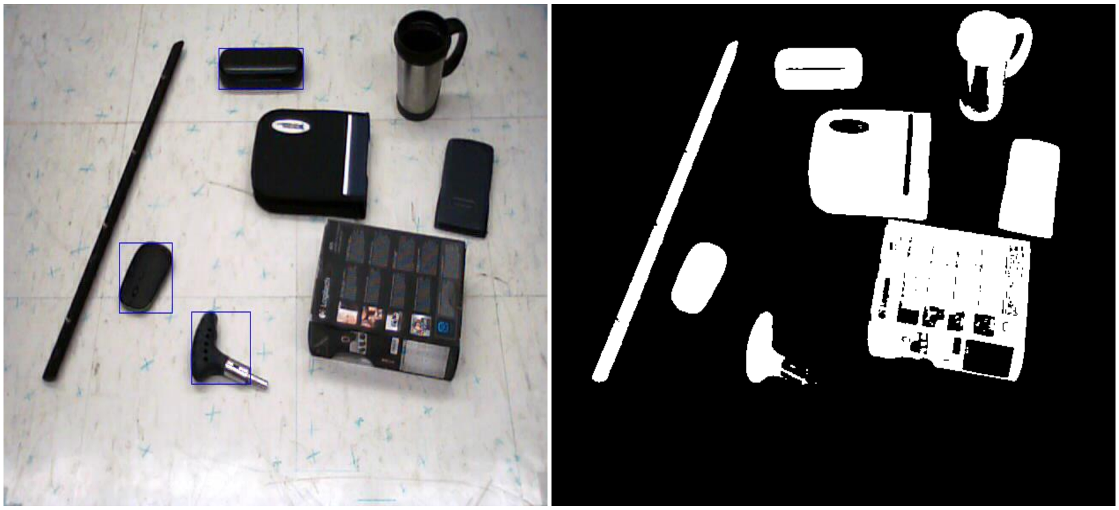


Figure 5.3: Sample threshold objects in the scene.

5.3.3 Morphological Operations

Opening and closing are two important morphological operations whose tendencies are to smooth the contour of an image. The main differences between these operations are that morphological opening will eliminate thin protrusions and break narrow isthmuses, whereas morphological closing eliminates small holes, fuses narrow breaks, and fills the gaps between contours. The opening and closing sets *A* by structuring element *B* are shown below:

$$A \circ B = (A \ominus B) \oplus B, \tag{5.2}$$

$$A \bullet B = (A \oplus B) \ominus B, \tag{5.3}$$

where $\oplus$ represents dilation, $\ominus$ represents erosion, $A \circ B$ characterizes morphological opening, and $A \bullet B$ characterizes morphological closing [60].

5.3.4 Median Filtering

Most of the smoothing filtering methods make the edges and other sharp details of objects become blurred. If the objective is to reduce noise rather than blurring objects, median filters are the best choice. The gray level of each pixel in this type of filter will be replaced by the median of the gray levels in the neighborhood of that pixel. This method is effective especially when the pattern of noise is strong and includes spike-like components. The advantage of median filters is that they can preserve edge sharpness [60].

5.4 Orientation Correction Using Object Length Variations

In this section, the orientation of the object is corrected through variations in an object's length in pixels as the object rotates 360º at the same position. It has been shown that there is a symmetric relationship between changes in length and the orientation of the object in degree. This relationship can be concluded using a curve-fitting analysis process. Table 5.1 and Figure 5.4 demonstrate the relationship between an object's length and the orientation variation. In addition, Table 5.2 and Figure 5.5 demonstrate the average percentage error calculation for object length variation when the object is rotated from 0 to 360º.

Table 5.1: Orientation correction using object length variation.

Trials	*Actual length (pixel)*	*Measured length (Pixel)*	*Measured orientation (Degree)*	*Length error (Pixel)*	*Orientation difference (Degree)*
1	63.9	63.9	0.53	0	-3.000
2	63.9	63.9	3.53	0	-3.914
3	63.9	64.1	7.44	0.2	-3.671
4	63.9	64	11.11	0.1	-4.550
5	63.9	63.7	15.66	-0.2	-3.190
6	63.9	63.5	18.85	-0.4	-7.100
7	63.9	63.2	25.95	-0.7	-10.980
8	63.9	61.7	36.93	-2.2	-5.810
9	63.9	61	42.74	-2.9	-8.910
10	63.9	59.7	51.65	-4.2	-8.450
11	63.9	58.7	60.10	-5.2	-8.590
12	63.9	57.8	68.69	-6.1	-7.110
13	63.9	57.3	75.80	-6.6	-5.330

Table 5.1 (continued)

14	63.9	56.9	81.13	-7	-7.770
15	63.9	56.7	88.90	-7.2	3.850
16	63.9	56.7	-85.05	-7.2	-6.460
17	63.9	56.9	-78.59	-7	-6.560
18	63.9	57.3	-72.03	-6.6	-6.710
19	63.9	57.9	-65.32	-6	-5.560
20	63.9	58.3	-59.76	-5.6	-7.790
21	63.9	59.2	-51.97	-4.7	-6.420
22	63.9	60	-45.55	-3.9	-4.190
23	63.9	60.7	-41.36	-3.2	-5.450
24	63.9	61.5	-35.91	-2.4	-7.080
25	63.9	62.4	-28.83	-1.5	-5.310
26	63.9	62.8	-23.52	-1.1	-7.280
27	63.9	63.3	-16.24	-0.6	-8.124
28	63.9	64.1	-8.12	0.2	-3.609
29	63.9	63.9	-4.51	0	-3.095
30	63.9	64.2	-1.41	0.3	-1.412

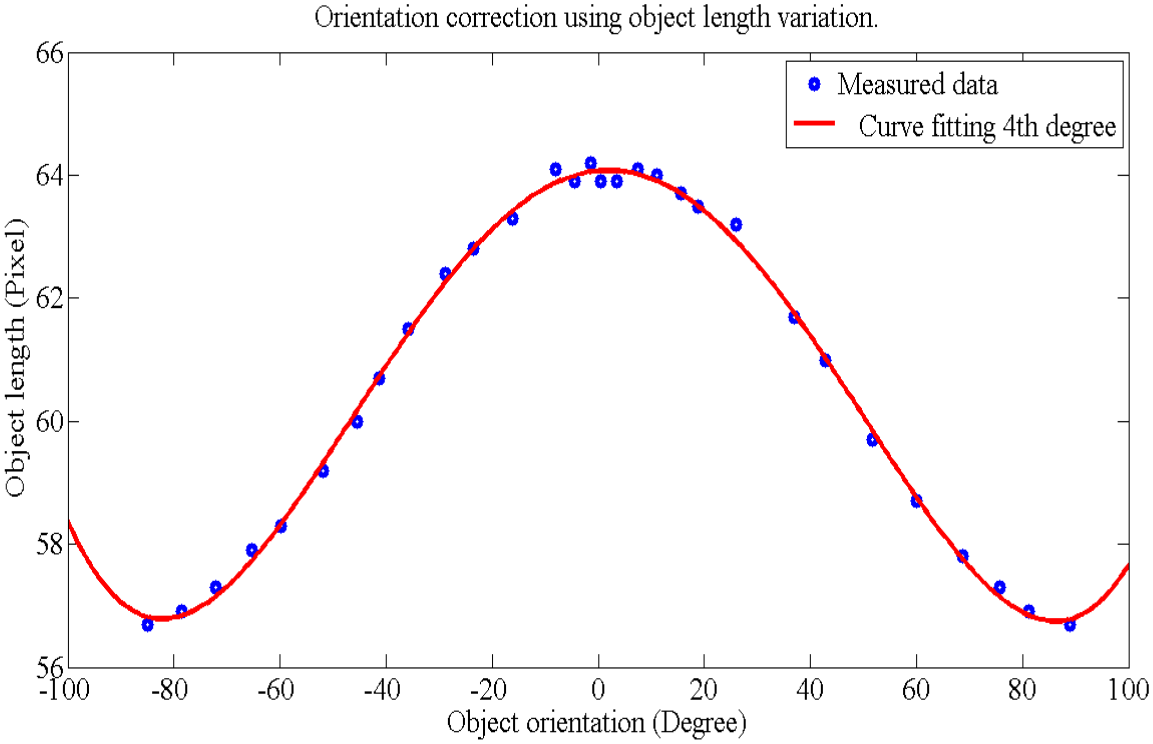


Figure 5.4: Orientation correction using object length variation.

Table 5.2: Average percentage error calculation for object length variation.

Trial	*Actual length (pixel)*	*Measured length (pixel)*	*Length error (pixel)*	*Average percentage error*
1	63.9	63.9	0	0.00
2	63.9	63.9	0	0.00
3	63.9	64.1	0.2	0.31
4	63.9	64	0.1	0.16
5	63.9	63.7	0.2	0.31
6	63.9	63.5	0.4	0.63
7	63.9	63.2	0.7	1.10
8	63.9	61.7	2.2	3.44
9	63.9	61	2.9	4.54
10	63.9	59.7	4.2	6.57
11	63.9	58.7	5.2	8.14
12	63.9	57.8	6.1	9.55
13	63.9	57.3	6.6	10.33
14	63.9	56.9	7	10.95
15	63.9	56.7	7.2	11.27
16	63.9	56.7	7.2	11.27
18	63.9	57.3	6.6	10.33
19	63.9	57.9	6	9.39
20	63.9	58.3	5.6	8.76
21	63.9	59.2	4.7	7.36
22	63.9	60	3.9	6.10
23	63.9	60.7	3.2	5.01
24	63.9	61.5	2.4	3.76
25	63.9	62.4	1.5	2.35
26	63.9	62.8	1.1	1.72
27	63.9	63.3	0.6	0.94
28	63.9	64.1	0.2	0.31
29	63.9	63.9	0	0.00
30	63.9	64.2	0.3	0.47

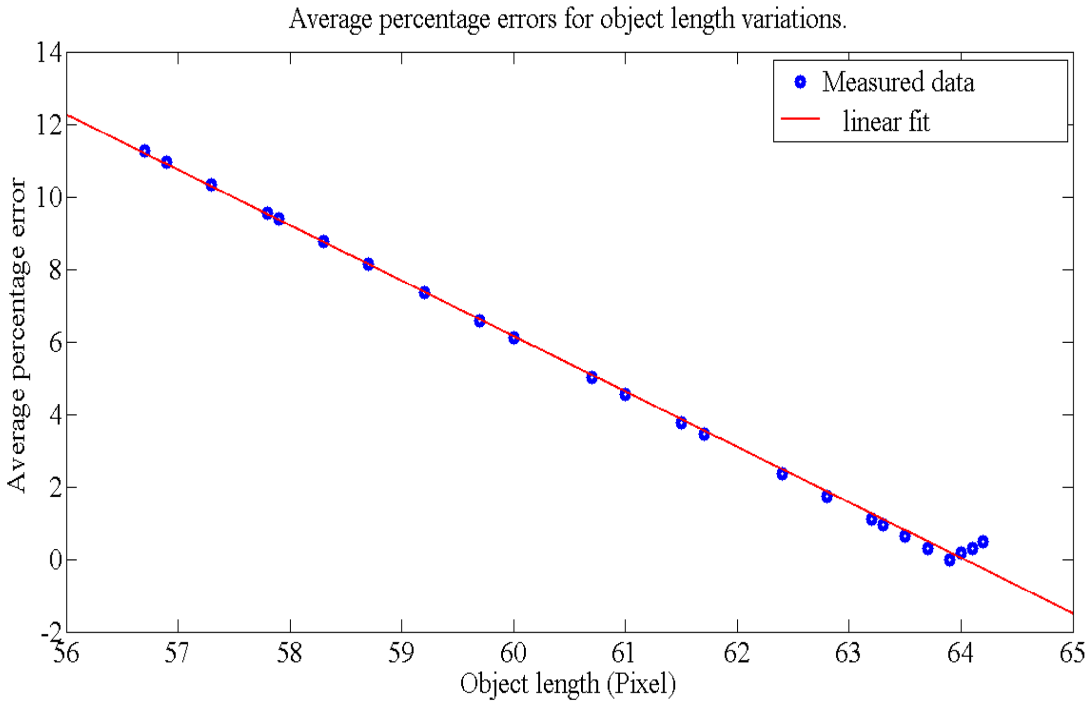


Figure 5.5: Average percentage errors for object length variation.

The discrepancy in the data for the object length in Figure 5.5 is due to the value obtained from the applied Simulink algorithm, which is inaccurate (the actual object length is 63.9 pixels). In addition, these data (any values greater than 63.9) are included in the curve-fitting method. Equation (5.4) shows the non-linear 4th-degree polynomial fitted curve equation for the graph in Figure 5.4.

$$l_p = 1.5e^{-07}\beta_O{}^4 - 1.1e^{-06}\beta_O{}^3 - 0.0021\beta_O{}^2 + 0.0079\beta_O + 64 \qquad (5.4)$$

where x_O represents the object orientation (degree) and l_p is the object's length (pixel).

To find the best polynomial curve fitting for the orientation correction using object length variation, a comparison between 3rd- and 4th-degree polynomial curve fitting is shown in Table 5.3. The comparison between these two polynomial curve fittings demonstrated the percentage error of 1.41 and 0.16 for 3rd- and 4th-degree polynomials, respectively.

Table 5.3: Comparison between 3rd- and 4th-degree polynomial curve fitting.

Measured length (pixel)	***Difference pixel (deg)***	***Third-degree polynomial curve fitting***	***Fourth-degree polynomial curve fitting***
63.9	0.0	63.000	64.000
63.9	-0.2	62.998	63.998
64.1	0.1	63.001	64.001
64.0	0.3	63.002	64.002
63.7	0.2	63.001	64.001
63.5	0.3	63.002	64.002
63.2	1.5	62.991	64.005
61.7	0.7	63.003	64.004
61.0	1.3	62.996	64.005
59.7	1.0	63.001	64.004
58.7	0.9	63.002	64.004
57.8	0.5	63.003	64.003
57.3	0.4	63.002	64.003
56.9	0.2	63.001	64.001
56.7	0.0	63.000	64.000
56.7	-0.2	62.998	63.998
56.9	-0.4	62.997	63.997
57.3	-0.6	62.996	63.995
57.9	-0.4	62.997	63.997
58.3	-0.9	62.996	63.994
59.2	-0.8	62.996	63.994
60.0	-0.7	62.996	63.995
60.7	-0.8	62.996	63.994
61.5	-0.9	62.996	63.994
62.4	-0.4	62.997	63.997
62.8	-0.5	62.997	63.996
63.3	-0.8	62.996	63.994
64.1	0.2	63.001	64.001
63.9	-0.3	62.998	63.998
64.2	0.0	63.000	64.000
Average error		**62.999**	**63.999**
Percentage error		**1.41**	**0.16**

5.5 Horizontal and Vertical Error Corrections

As an object moves away from the camera's position, the object's dimension decreases, but when an object moves closer to a camera, the object's dimension increases. In order to accurately estimate an object's dimension, correction factors should be applied to get the real size of the object. In the proposed algorithm, the object's length in the X- and Y-directions are initially not accurate. To correct the object's length in the Y-direction, the object is moved from the beginning of the field of view on the optical axis (the point 0 in the image plane for the camera definition of 640×480) towards the end of the field of view on the optical axis (the point 480 in the image plane for the camera definition of 640×480). As the object moves further away from 0 (pixel) to 480 (pixel), the object's length in pixels increases linearly. Therefore, a linear equation can be used to correct the object's length in the Y-direction. Figure 5.6 demonstrates the object's length variation in the Y-direction.

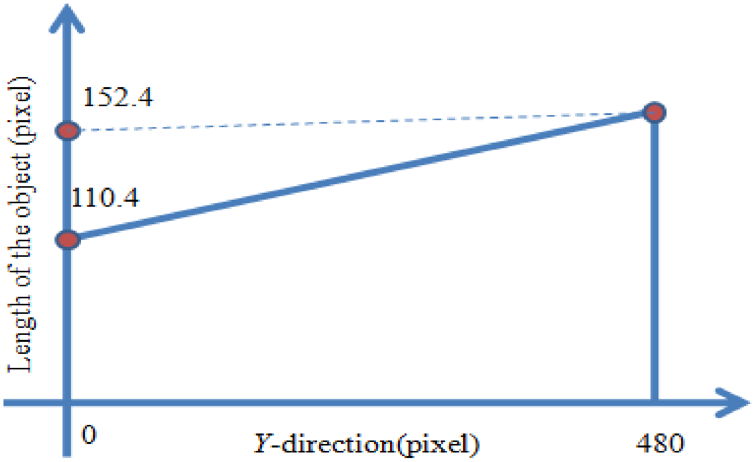


Figure 5.6: Variation of the object's length with y.

Equation (5.5) demonstrates a straight line equation for the graph in Figure 5.6. In this process, the actual length of the object is 156 (mm), which is shown as 110.4 pixels at the beginning of the camera's field of view. As the object moves further away from the camera up to the end of the camera's field of view in the Y-direction, the object's length increases to 152.4 pixels. In order to correct the object's length in the Y-direction, Equation (5.5) (which is a straight line equation) is defined as follows:

$$l_p = (0.0875\, y_p + 110.4)/152.4 \qquad (5.5)$$

where y_p is defined as the pixel's change in the Y-direction and l_p is the length of the object in pixels.

To correct the object's length in the X-direction, the procedures are the same as the Y-direction correction, but the starting point is from the left side and along the horizontal axis. Since the object's length in the X-direction is symmetrically decreasing, there is no need to find the change in the object's length for both sides. The starting point in which the object can be seen in the field of view is 30 (pixel) and the end point in the middle of the field of view is 333 (pixel). Figure 5.7 demonstrates the object length variation in the X-direction, and Table 5.4 indicates the results of the calculator's length corrections in the X- and Y-directions.

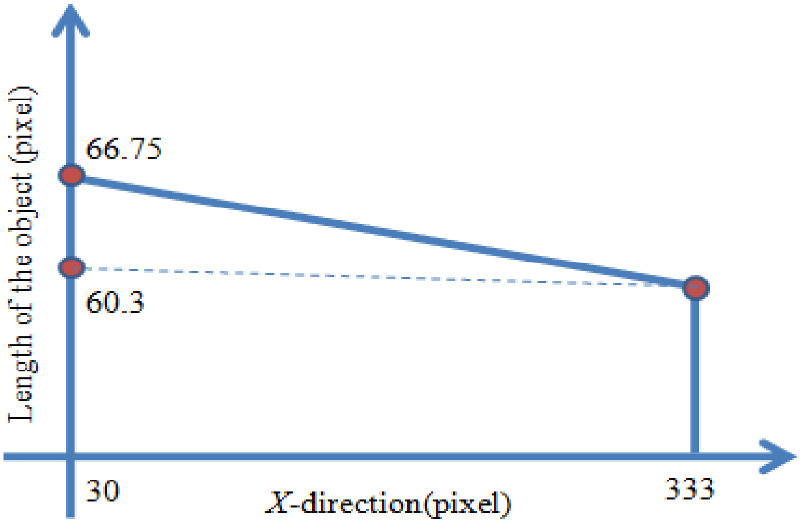


Figure 5.7: Variation of the object's length with x.

Equation (5.6) demonstrates the straight line equation for the graph in Figure 5.7.

$$l_P = (-0.0213\, x_p + 67.39)/66.75 \qquad (5.6)$$

where x_p is defined as the pixel's change in the X-direction and l_P is the length of the object in pixels.

Table 5.4: Results of the calculator's length corrections in X- and Y-directions.

Real length (mm)	*Non-corrected length (mm)*	*Y-direction correction (mm)*	*X-direction correction (mm)*	*Non-corrected length error (mm)*	*X-direction correction error (mm)*	*Y-direction correction error (mm)*
156	161.9	182.1	174.4	-5.9	-18.4	-26.1
156	148.3	157.2	158.8	7.7	-2.8	-1.2
156	143.5	159.9	153.2	12.5	2.8	-3.9
156	133.8	150.4	142.2	22.2	13.8	5.6
156	146.5	155	156.6	9.5	-0.6	1
156	136.9	144.9	145.7	19.1	10.3	11.1
156	142.8	149	152.4	13.2	3.6	7
156	117.7	136.6	124.1	38.3	31.9	19.4
156	123.3	140.8	130.4	32.7	25.6	15.2
156	147.2	155.9	157.4	8.8	-1.4	0.1
156	133	158.3	141.3	23	14.7	-2.3
156	150.7	161.7	161.5	5.3	-5.5	-5.7
156	148.9	155.7	159.4	7.1	-3.4	0.3
156	147.2	176.9	157.4	8.8	-1.4	-20.9
156	153.1	153.3	164.3	2.9	-8.3	2.7
156	136.4	147.1	145.1	19.6	10.9	8.9
156	152	150.6	163	4	-7	5.4
156	116.6	136	123	39.4	33	20
156	128.7	138.1	136.4	27.3	19.6	17.9
156	132.2	136.3	140.4	23.8	15.6	19.7
156	116.4	129.9	122.7	39.6	33.3	26.1
156	115.6	134.2	121.9	40.4	34.1	21.8
156	151.1	156	161.9	4.9	-5.9	0
		Average error		**17.57**	**8.46**	**5.31**

Graphical representation of Table 5.4 is shown in Appendix A (Figure 2).

5.6 Length and Width Correction

Although the video data can be used directly to measure the length of moving objects in real-time, the measured length will be incorrect due to uncorrected incoming pixels. In this part of the algorithm, the length of the object in any orientation or deformation can be modified by obtaining the major axis of an ellipse that encircles the object. Figure 5.8 shows a part of the Simulink block diagram for the length correction.

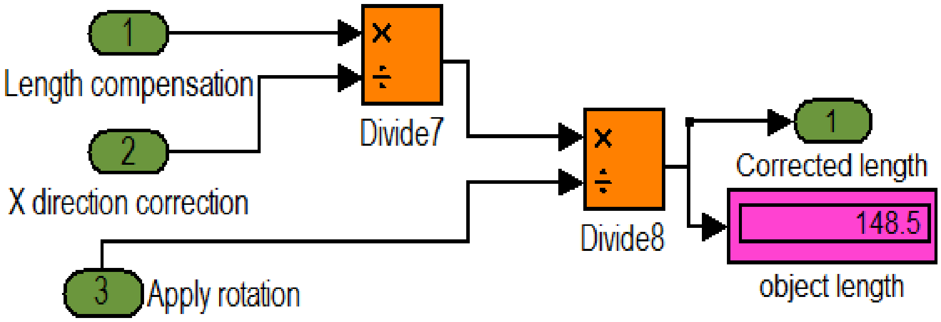


Figure 5.8: Simulink block diagram for length correction.

The object's width can also be estimated by extracting the minor axis of an ellipse. Keep in mind that the extracted width needs to be corrected in order to estimate the actual width of the object in any direction and orientation. Figure 5.9 demonstrates the correction factors that are applied for the width estimation.

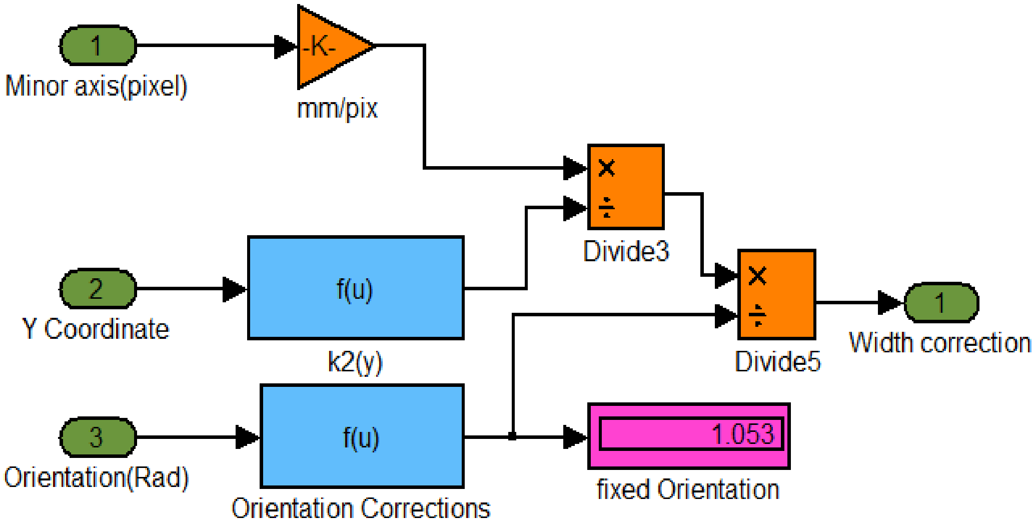
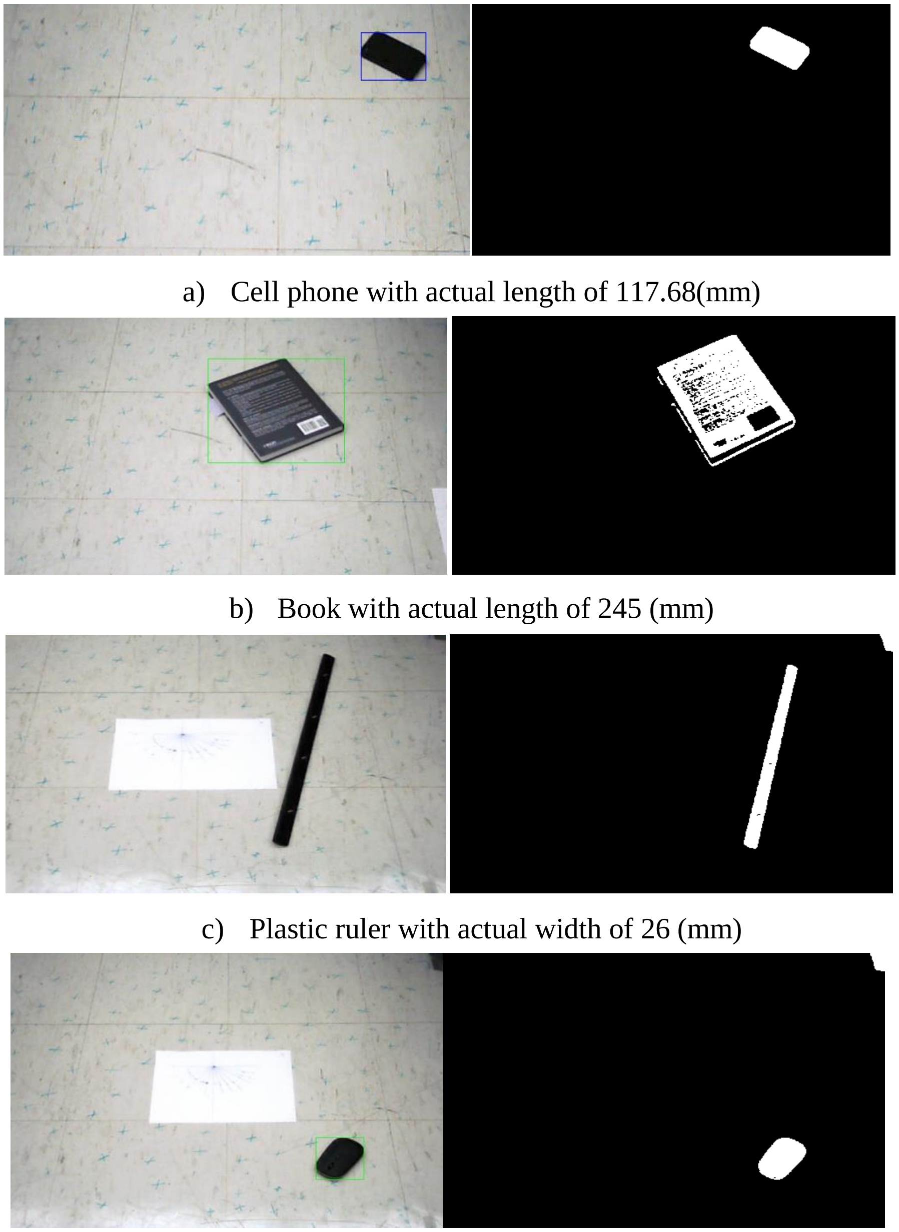


Figure 5.9: Simulink block diagram for width correction.

Using the proposed method, the length and width of the objects can be calculated with reasonably good accuracy. Figure 5.10 compares the actual and measured lengths and widths for different sized object. This method was applied to objects with different sizes in order to validate the proposed algorithm. The selected objects for investigation were a cell phone, a book, a plastic ruler, and a mouse. The results of this experiment calculated the length of a cell phone (with an actual length of 117.67 mm) and a book (actual length of 245 mm) as 114.1 mm and 223.5 mm, respectively. In addition, the width of a plastic ruler (actual width of 26 mm) and a mouse (actual width of 62.39 mm) were calculated as 23.62 mm and 60.94 mm, respectively.



a) Cell phone with actual length of 117.68(mm)

b) Book with actual length of 245 (mm)

c) Plastic ruler with actual width of 26 (mm)

d) Mouse with actual width of 62.39 (mm)

Figure 5.10: Dimension measurements (length and width) of different objects.

5.7 Simulink Block Diagram for Object Dimension Measurement

In this section, the width and length of the object are calculated using blob analysis commands of a simulink program. The minor and major axes of the ellipse that encircles the object's surface will be computed and then these extracted features will be used to find the corrected width and length of the object. The proposed algorithm is robust and reliable with respect to the orientation and position of the object and can be used for any moving objects with different sizes, and in any directions on the ground. For the moving object, the magnification decreases as the object distance increases [61]. Therefore, the measured dimension (i.e., length and width) does not show the original size.

Since the length and width of the object reduce as the object moves farther away, correction factors should be applied to determine the real size of the object. To estimate the object's length, two different subsystems are applied. The first subsystem corrects for the length reduction by using the Y-direction correction, and the other subsystem monitors for the length reduction correction based on the X-direction and the rotation correction. Finally, the original length is estimated and displayed in the output of the second subsystem. A schematic Simulink block diagram for the object's length correction is illustrated in Figure 5.11.

The width also reduces as the object goes further away from the camera. To modify this limitation, a correction subsystem is designated in which the minor axis, which is in pixel, is converted to millimeter by using a scale factor. Finally, by applying Y-direction and orientation corrections, the modified width is estimated as the output of the subsystem. Figure 5.12 shows the Simulink block diagram for the object's width correction.

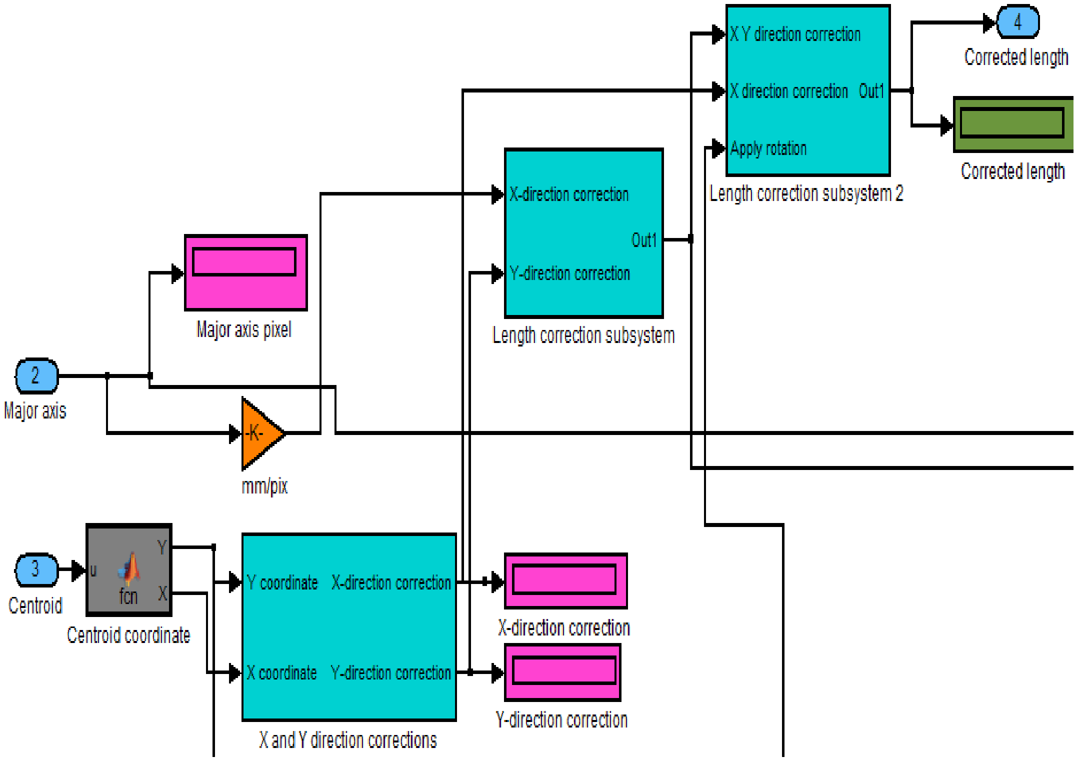


Figure 5.11: Simulink block model for the object's length correction.

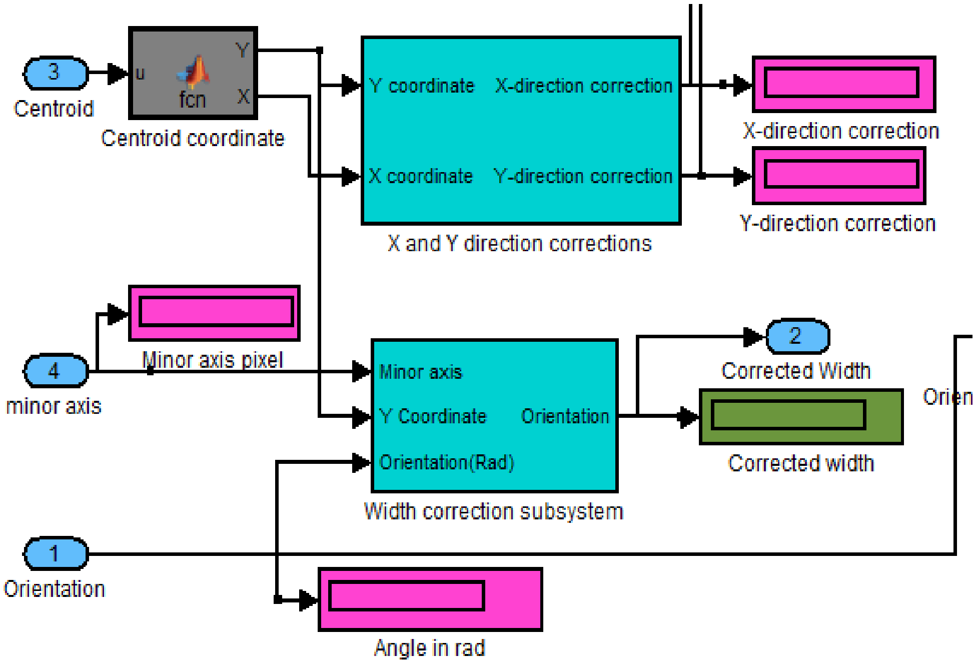


Figure 5.12: Simulink block model for the object's width correction.

5.8 Experimental Results and Discussion

In this experiment, the distance of the desired object is extracted using the point feature extraction method. The extracted point feature is usually set as the centroid point of a target, but in the proposed method it is considered as a closest point of the object to the camera. The results of Taha and Jizat's [49] object distances for a calculator measuring 84.62 mm×156 mm (W×L) and an irregular toy car measuring 70.21 mm×145.54 mm (W×L), which are optimized by the least square method, are compared in Tables 5.5-6. Experimental results using the least square method confirmed the improvement for the object distance measurement proposed by the Taha and Jizat [49] approach. The experiments are applied for objects located at any direction and orientation in the camera's field of view. In addition, the measurement is performed by measuring tape with an accuracy of ±0.5 mm. Integrating the least square approach along with the Taha and Jizat [49] method reduces the sensitivity to noise. As for the non-linear optimization process, the total projection error should be minimized after applying the least square method. Figures 5.13-14 are the graphical presentations of Table 5.5-6. Finally, Table 5.7 demonstrates the results of the object's length and width calculations after the correction was applied.

Table 5.5: Improving distance measurement methods for the calculator using least square optimization.

Trial	*Real distance (mm)*	*Taha and Jizat [49] distance (mm)*	*Least square method distance (mm)*	*Taha and Jizat [49] distance error (mm)*	*Least square method distance error (mm)*
1	1558	1477	1517	81	41
2	1481	1478	1518	3	-37
3	1581	1495	1534	86	47
4	1535	1513	1552	22	-17
5	1618	1585	1626	33	-8
6	1695	1643	1685	52	10

Table 5.5 (continued)

7	1772	1688	1729	84	43
8	1717	1729	1769	-12	-52
9	1793	1731	1773	62	20
10	1811	1810	1845	1	-34
11	1813	1811	1845	2	-32
12	1822	1854	1884	-32	-62
13	1996	1896	1921	100	75
14	1943	1982	1992	-39	-49
15	2100	2010	2014	90	86
16	2065	2131	2105	-66	-40
17	2150	2224	2169	-74	-19
18	2245	2254	2189	-9	56
19	2205	2294	2215	-89	-10
20	2360	2467	2319	-107	41
		Average error		**9.4**	**2.95**

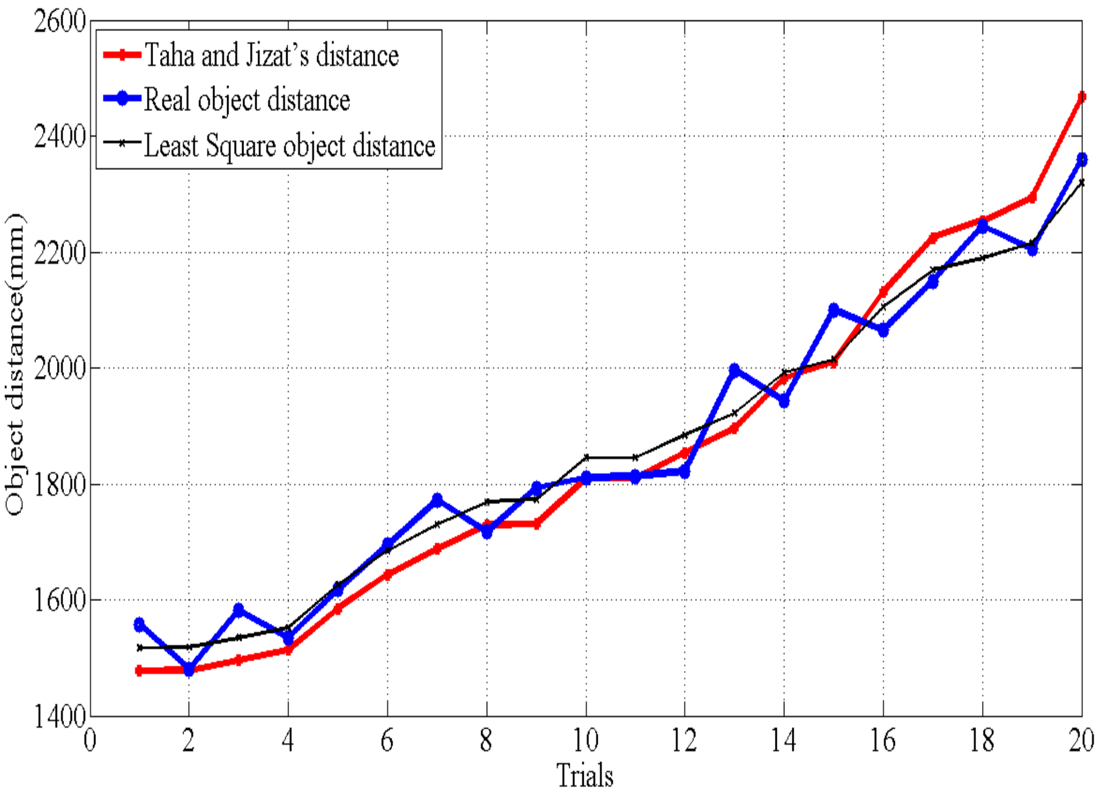


Figure 5.13: Comparison of least square, real, and measured object distance calculations for the calculator.

Table 5.6: Improving distance measurement methods for the toy car using least square optimization.

Trial	***Real distance (mm)***	***Taha and Jizat [49] distance (mm)***	***Least square method distance (mm)***	***Taha and Jizat [49] distance error (mm)***	***Least square method distance error (mm)***
1	1515	1517	1560	-2	45
2	1600	1581	1625	19	-25
3	1627	1580	1624	47	3
4	1681	1695	1710	-14	-29
5	1685	1693	1739	-8	-54
6	1721	1654	1699	67	22
7	1734	1696	1741	38	-7
8	1763	1784	1825	-21	-62
9	1766	1792	1832	-26	-66
10	1785	1753	1796	32	-11
11	1840	1742	1785	98	55
12	1867	1857	1891	10	-24
13	1896	1866	1899	30	-3
14	1980	1889	1919	91	61
15	1984	1967	1985	17	-1
16	2073	2064	2061	9	12
17	2114	2179	2144	-65	-30
18	2142	2141	2117	1	25
19	2193	2288	2217	-95	-24
20	2242	2297	2223	-55	19
	Average error			**8.65**	**-4.7**

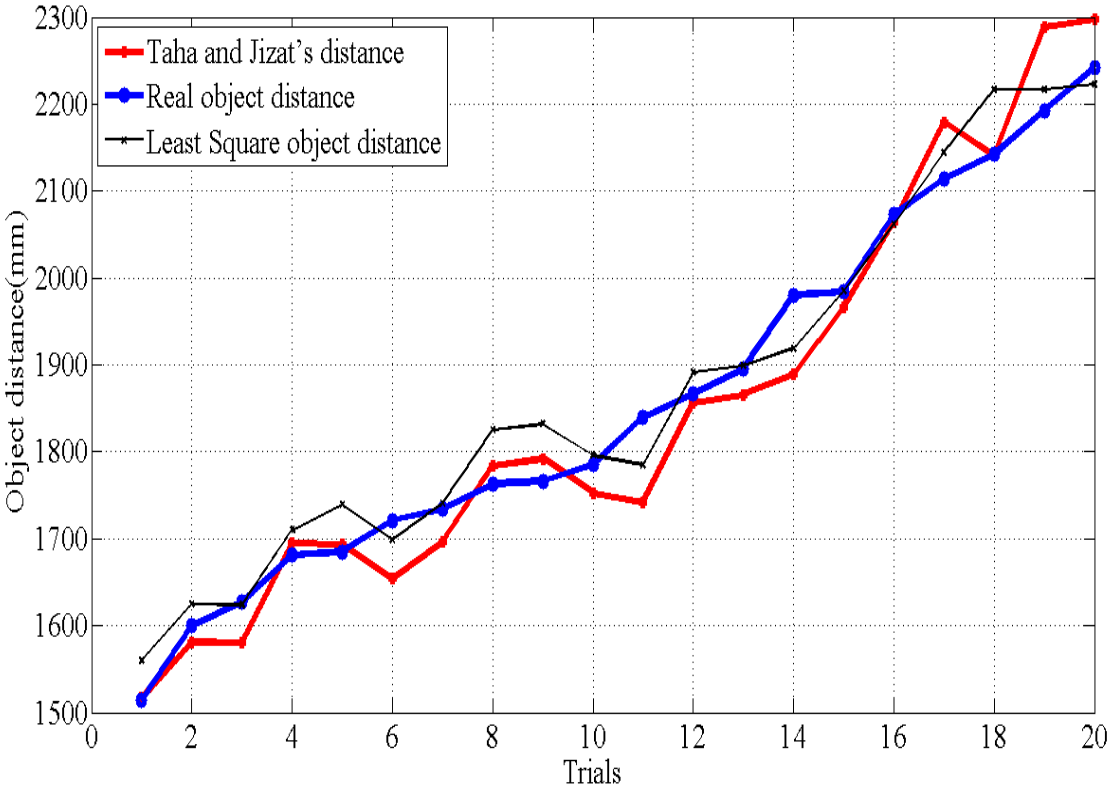


Figure 5.14: Comparison of least square, real, and measured object distance calculations for the toy car.

Table 5.7: Length and width correction for the calculator.

Trial	*Actual length (mm)*	*Actual width (mm)*	*Measured length (mm)*	*Measured width (mm)*	*Width error (mm)*	*Length error (mm)*
1	156	84.62	171.1	80.84	3.78	-15.1
2	156	84.62	169.7	81.14	3.48	-13.7
3	156	84.62	169.3	80.87	3.75	-13.3
4	156	84.62	164.5	79.67	4.95	-8.5
5	156	84.62	176.4	74.32	10.3	-20.4
6	156	84.62	153.8	78.4	6.22	2.2
7	156	84.62	157.6	83.57	1.05	-1.6
8	156	84.62	146	78.73	5.89	10
9	156	84.62	147.2	74.26	10.36	8.8
10	156	84.62	156.2	74.05	10.57	-0.2
11	156	84.62	155.8	75.35	9.27	0.2
12	156	84.62	156.7	74.38	10.24	-0.7
13	156	84.62	149.6	75.46	9.16	6.4
14	156	84.62	152.5	71.46	13.16	3.5
15	156	84.62	149.8	71.4	13.22	6.2

Table 5.7 (continued)

16	156	84.62	158.3	68.15	16.47	-2.3
17	156	84.62	151.2	68.67	15.95	4.8
18	156	84.62	153.6	76.27	8.35	2.4
19	156	84.62	138.6	74.83	9.79	17.4
20	156	84.62	149.7	69.14	15.48	6.3
21	156	84.62	139	77.71	6.91	17
22	156	84.62	138.6	80.52	4.1	17.4
23	156	84.62	153.7	66.22	18.4	2.3
24	156	84.62	147.5	66.67	17.95	8.5
25	156	84.62	151	68.21	16.41	5
26	156	84.62	140.8	69.97	14.65	15.2
27	156	84.62	156.8	63.14	21.48	-0.8
28	156	84.62	151.1	63.96	20.66	4.9
		Average error			**10.8**	**2.2**

5.9 Conclusion

The presented procedure can predict the target positions relative to the robot system using only data that are extracted from a single camera with variable pitch angle and without any additional sensors. This method, which is an improved version of the Taha and Jizat [49] approach due to applying the least square method, is able to measure object distance under varying illumination conditions. The experimental results accomplished the object distance average error of 2.95 mm for the calculator and 4.7 mm for the toy car. Moreover, the discrete sample time to run the proposed algorithm is specified by the Simulink program to be 0.033 seconds. This inexpensive distance measurement method could be applied to robotic applications such as obstacle avoidance, soccer robot, and sorting system, as well as in automated guided vehicle (AGV) applications such as collision avoidance, etc.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Introduction

Attempts to design two vision-based object distance measurement algorithms using a single fixed camera and a single camera with variable pitch angle were illustrated in this thesis. In the development of these algorithms, basic assumptions about the operating environment of the objects were made; however, no restrictions were placed on the object's size, direction or orientation. That is, we attempted to develop an algorithm that can be used to calculate object distance for objects of any size or geometry.

As is well known, image formation is fundamentally modeled by the pinhole lens model, with the lens considered to be an ideal pinhole located at the focal center of the lens. In reality, however, a vast amount of information is lost when the 3D world is projected onto a 2D image plane. Even when multiple views are available, depth information may not always be recoverable. To overcome this loss of information, restrictions such as nonlinearity are often introduced, such as the non-linearity correction suggested in this thesis.

In the proposed approaches, the object distance measurements using a single fixed camera with variable pitch angle were obtained by a procedure composed of several image processing steps. These steps can be summarized as: ***i*)** background estimation; ***ii*)** object tracking; ***iii*)** feature extraction; and ***iv*)** feature analysis. This algorithm starts by receiving an input video source and pre-processing the image to determine the background from the scene. The desired object is then tracked from the initial frame to the end frame. In the third step, the the object features are extracted from the desired tracked object in the image, and fourth and finally, the object distance, width and length calculation are determined.

As stated in this thesis, robots need to identify the orientation and the distance to the objects in order to localize, navigate, and perform certain high-level planning tasks. This work was inspired by introducing examples of the two major duties of the object distance measurement: ***i*)** finding the distance using the size of the object and the focal length of the camera; and ***ii*)** finding the distance of the object of unknown size by integrating the height of the camera and the point of contact of the object with the ground.

Although the problem of depth perception when using a single camera is of great importance in the mobile robotics community, the depth perception is a basic robotic capability which, when it is solved, immediately allows for higher-level capabilities such as localization, mapping, and path planning.

6.2 Contributions

The main contributions of this thesis are developing object distance measurement algorithms that are:

- Accurate, reliable, and consistent;
- Able to simultaneously localize the objects and then measure the object distance (through using a single fixed camera and a single camera with variable pitch angle using the feature extraction method);

- Able to measure the distance between the camera and moving objects anywhere in the camera’s field of view;
- Efficient in measuring objects of unknown shape and size.

Chapter 2 discussed object tracking and object distance calculation and also investigated some external factors that need to be considered to successfully overcome the difficulties in object tracking and distance measurements. In Chapter 2, full lists of object tracking problems and classifications were presented to provide readers with the necessary background knowledge about previous studies and technical information.

Good mathematical modeling of object distance calculations using a single fixed camera and a single camera with variable pitch angle were presented in Chapter 3. The models presented in that chapter covered some of the most common object distance measurement methods along with an evaluation of the relative accuracy of those techniques. In addition, a review of the applications of computer vision, the selection criteria for choosing different types of cameras, and visual servoing were reviewed in Chapter 3.

Two independent full robot vision implementations of the object distance measurement using a single fixed camera and a single camera with variable pitch angle techniques for robot navigation were presented in Chapters 4 and 5. In these two chapters, the capabilities such as continuous object tracking, feature extraction and object distance measurements for single cameras with variable pitch angle and single fixed cameras in structured environments were also provided.

6.3 Future Work and Recommendations

Although the use of a single camera for measuring object distance has been proven feasible and economical in robot vision, certain aspects of the algorithm may be improved to enhance its performance. This section provides suggestions for future work, along with recommendations and comments on performance implications.

A mobile robot can navigate both indoor and outdoor environments, but indoor and outdoor application strategies are vastly different. On the whole, outdoor applications are more difficult than indoor applications, since it is not possible to predict major problems such as rough terrain, weather conditions, and lighting variations in real-world environments. In this study, our object tracking and distance measurement algorithm was implemented for indoor applications.

As further study, prepared algorithms can be modified to perform in outdoor environments. The current algorithm is susceptible to light variation and would have difficulties being implementing in unstructured environments. However, it probably could be improved through the use of an adaptive filter and changes in hardware. Moreover, further algorithmic developments would increase robustness, and additional work on the distance estimation processes would satisfy the above-mentioned issues while improving accuracy.

In addition to solving outdoor robot navigation problem, future endeavors in this field should consider using a high-resolution camera to improve system robustness. As well, integrating robot vision cameras would provide better vision results and prevent radial and tangential distortions that are prevalent in cheap cameras and webcams.

REFERENCES

[1] Jüngel, M. Mellmann, H. and Spranger, M., "Improving vision-based distance measurements using reference objects, "*Robocup 2007: robot soccer world cup*, pp. 89-100, 2007.

[2] Goto, A. and Fujimoto, H., "Proposal of 6 DOF Visual Servoing for Moving Object Based on Real-Time Distance Identification," *SICE. Annual Conference,* Japan, pp. 3208-3213, 20-22 August, 2008.

[3] Firouzi, H. and Najjaran, H., "Real-time monocular vision-based object tracking with object distance and motion estimation," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 987-992, 6-9 July, 2010.

[4] Shaaban, K. M. and Omar, N. M., "3D information extraction using Region-based Deformable Net for monocular robot navigation," *Journal of Visual Communication and Image Representation*, vol. 23, pp. 397-408, 2012.

[5] Cavallaro, A. Steiger, O. and Ebrahimi, T., "Tracking video objects in cluttered background," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 4, pp. 575– 584, 2005.

[6] Zhang, Z. Han, Y. Zhou, Y. and Dai, M., "A novel absolute localization estimation of a target with monocular vision," *Optik – International Journal for Light and Electron Optics*, vol. 124, no. 12, pp. 1218-1223, 2013.

[7] Coman, M. and Balan, R., "Video Camera Measuring Application Using Matlab," *Solid State Phenomena*, vol. 166-167, pp. 139-144, 2010.

[8] Yamaguti, N. Oe, S. and Terada, K., "A Method of Distance Measurement by Using Monocular Camera," *Proceedings of the 36th SICE Annual Conference*, Tokushima, pp. 1255-1260, 29-31 July, 1997.

[9] Yilmaz, A. Javed, O. and Shah, M., "Object Tracking: A Survey," *ACM Journal of computing Surveys*, vol. 38, no. 4, pp. 1-45, 2006.

[10] Manjunath, B. S, and Ma, W., “Texture Features for browsing and retrieval of image data,” *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 837-842, 1996.

[11] Yang, H. Shao, L. Zheng, F. Wang, L. and Song, Z., “Recent Advances and Trends in Visual Tracking: A Review,” *Neurocomputing*, vol. 74, no. 18, pp. 3823-3831, 2011.

[12] Jepson, A. D. Fleet, D. J. and El-Maraghi, T. F., “Robust online appearance models for visual tracking,” *IEEE Transaction on Pattern Recognition and Machine Intelligence* , vol. 25, no. 10, pp. 1296–1311, 2003.

[13] Matthews, L. Ishikawa, T. and Baker, S., “The template update problem,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, 2004.

[14] Li, L. Deng, Z-Q. Li, B. and Wu, X., “Fast vision-based pose estimation iterative algorithm, ”*Optik – International Journal for Light and Electron Optics,* vol. 124, no. 12, pp. 1116-1121, 2013.

[15] Tao, Z. Changku, S. and Shan, C., “Monocular vision measurement system for the position and orientation of remote object, ”*International Symposium on Photo electronic Detection and Imaging*, vol. 6623, 2007.

[16] Krishnan, J. V. G. Manoharan, N. and Rani, B. S., “ESTIMATION OF DISTANCE TO TEXTURE SURFACE USING COMPLEX LOG MAPPING,” *Journal of Computer Application*, vol. 3, no. 3, 2010.

[17] Subbarao, M., “Determining distance from defocused images of simple objects,” *Tech. Rep. Computer vision laboratory,* Dept. of Electrical Engineering, State University of New-York, Stony Brook, NY 11794-2350, USA, 1989.

[18] Barrois, B. and Wöhler, C., “3D pose estimation based on multiple monocular cues”, *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8, 2007.

[19] Kendal, D., “Measuring distances using digital cameras, ”*Journal of Australian Senior Mathematics*, vol. 21, no. 2, pp. 24-28, 2007.

[20] Tinnachote, C. and Pimprasan, K., “DISTANCE MEASURMENT FROM DIGITAL PHOTOGRAPH USING 3rd ORDER POLYNOMIAL EQUATION,” *The 33RD Asian Conference on Remote Sensing*.

[21] Jamzad, M. Foroughnassiraei, A. Chiniforooshan, E. Ghorbani, R. Kazemi, M. Chitsaz, H. R. Mobasser, F. and Sadjad, S. B., “Middle sized Soccer Robots: ARVAND,” *Proceeding of RoboCup-99: Robot Soccer world Cup III, Springer*, pp. 61-73, 2000.

[22] Chang, S. H, Hsia, C. H. Chang, W. H. and Chiang, J. S., “Self-Localization Based on Monocular Vision for Humanoid Robot,” *Tamkang Journal of Science and Engineering*, vol. 14, no. 4, pp. 323-332, 2011.

[23] Hsu, C. C. J. Lu, M. C. and Lu, Y. Y., “Distance and Angle Measurement of Objects on an Oblique Plane Based on Pixel Number Variation of CCD Images,” *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 5, pp. 1779-1794, 2011.

[24] Hautiere, N. Tarel, J-P. Lavenant, J.and Aubert, D., “Automatic fog detection and estimation of visibility distance through use of an onboard camera” *Machine Vision and Applications Journal*, vo. 17, no. 1, pp. 8-20, 2006.

[25] Royer, E. Lhuillier, M. Dhome, M. and Lavest, J. M., "Monocular Vision for Mobile Robot Localization and Autonomous Navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237-260, 2007.

[26] Shijie, Z. Fenghua, L. Xibin, C. and Liang, H., “Monocular vision-based two-stage iterative algorithm for relative position and attitude estimation of docking spacecraft*,” Chinese Journal of Aeronautics*, vol. 23, no. 2, pp. 204-210, 2010.

[27] Cheung, H-K. Siu, W-C. Ng, C-S. Lee, S and Poon, L., “Accurate Distance Estimation Using Camera Orientation Compensation Technique for Vehicle Driver Assistance System”, *IEEE International Conference on Consumer Electronics (ICCE’2011),* pp. 231-232, 2012.

[28] Chiang, Y-M. Hsu, N-Z. and Lin, K-L., “Driver Assistance System Based on Monocular Vision,” *Lecture Notes in Computer Science*, vol. 5027, pp. 1-10, 2008.

[29] Campoy, P. Mondragón, I. F. Olivares-Mendez, M. A. and Martinez, C., “Visual Servoing for UAVs,” *Visual Servoing*, pp. 181-216, Croatia, 2010.

[30] Tamadazte, B. Le-Forte Piat, N. and Marchand, E., “A Direct Visual Servoing Scheme for Automatic Nanopositioning,” *IEEE transaction on mechatronic*, vol. 17, no. 4, 2012.

[31] Marchand, E. and Collewet, C., “Using image gradient as a visual feature for visual servoing,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5687-5692, 2010.

[32] Dame, A. and Marchand, E., “Entropy Based Visual Servoing,” *IEEE International Conference on Robotics and Automation,* pp. 707-713, 2009.

[33] Wang, H. B. and Liu, M., “Design of Robotic Visual Servo Control Based on Neural Network and Genetic Algorithm,” *International Journal of Automation and Computing*, vol. 9, no. 1, pp. 24-29, 2012.

[34] Pomares, J. Corrales, J. A. García, G.J. and Torres, F.,” Direct Visual Servoing to Track Trajectories in Human-Robot Cooperation, ”*International Journal of Advance Robotic System*, vol. 8, no. 4, pp. 129-138, 2011.

[35] Silveira, G. and Malis, E., “Direct Visual Servoing: Vision-Based Estimation and Control Using Only Nonmetric Information,” *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 974-980, 2012.

[36] Longoria, R.G. Basic Vision with Lab VIEW [PowerPoint slides], 2011. Retrieved from

http://www.me.utexas.edu/~longoria/me344/lab5/Basic_Vision_with_LabVIEW.pdf

[37] Siegwart,R. and Nourbakhsh,I. R., Introduction to Autonomous Mobile Robots, MIT Press, 2004.

[38] Chaumette, F. and Hutchinson, S., “Visual Servo Control, Part I: Basic Approaches”, *IEEE Robotics and Automation Magazine,* vol. 13, no. 4, pp. 82-90, 2006.

[39] Jian Chen, D. M. Dawson, W. M. Dixon, W. E. and Behal, A., “Adaptive Homography-Based Visual Servo Tracking for Fixed and Camera-in Hand Configurations, ” *IEEE Transactions on Control Systems Technology*, vol. 13, no. 5, pp. 814-825, 2005.

[40] Huang, X. H. Zeng, X. J. and Wang, M., “SVM-based identification and un-calibrated visual servoing for micromanipulation”, *International Journal of Automation and Computing*, vol. 7, no. 1, pp. 47-54, 2010.

[41] Flandin, G. Chaumette, F. and Marchand, E., “Eye-in-hand/Eye-to-hand cooperation for Visual Servoing, ”*IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2741-2746, 2000.

[42] Nourbakhsh, A., and Korayem, M. H., “6R Robots; How to Guide and Test them by Vision?”*Communications in Computer and Information Science CCIS Journal*, vol. 6, pp. 892-896, 2008.

[43] Hutchinson, S. Hager, G. D. and Corke. P. I., “A tutorial on Visual Servo Control,” *IEEE Transaction on Robotics and Automation*, vol. 12, no. 5, pp. 651-670, 1996.

[44] Spong, M. W. Hutchinson, S. and Vidyasagar, M., Robot Modeling and Control, John Wiley and Sons, Inc., 2006.

[45] Aristos, D. Pachidis, T. Lygouras, J., “Robot Path Generation by Viewing a Static Scene from a Single Camera,” *in Proc. IEEE Int. Symposium on Robotics and Automation*, 2002.

[46] Drayton, B., Algorithm and design improvements for indirect time of flight range imaging cameras. PhD thesis, Victoria University of Wellington, NZ, 2013.

[47] Serway, R. A. and Jewett, J. W., Jr. Physics for Scientists and Engineers (8th edition), Brooks/Cole, 2009.

[48] Joglekar, A. Joshi, D. Khemani, R. Nair, S. and Sahare, S., “Depth Estimation Using Monocular Camera,” *International Journal of Computer Science and Information Technologies*, vol. 2, no. 4, pp. 1758-1763, 2011.

[49] Taha, Z. and Jizat, J. A. M., “A comparison of Two Approaches for collision Avoidance of an Automated Guided Vehicle Using Monocular Vision,” *Applied Mechanics and Materials*, vol. 145. pp. 547-551, 2012.

[50] Rahman, A. Salam, A. Islam, M. and Sarker, P., “An Image Based Approach to Compute Object Distance, ”*International Journal of Computational Intelligence Systems,* vol. 1, no. 4, pp. 304-312, 2008.

[51] Lu, M. Hsu, C. and Lu, Y., *“Image*-Based System for Measuring Objects on an Oblique Plane and Its Applications in 2D Localization, ”*IEEE Sensors Journal,* vol. 12, no. 6, pp. 2249-2261, 2012.

[52] Clarke, T. A. and Williams, M. R., “Buyers guide to six non-contact distances measuring techniques,” *Quality Today, Buyers Guide,* pp. 145-149, 1999.

[53] Craig, J. J., Introduction to Robotics: Mechanics and Control, 3rd ed, NJ: Prentice-Hall, 2004.

[54] McAndrew, A., “An Introduction to Digital Image Processing with Matlab,” Victoria University of Technology, Melbourne, Australia, 2004.

[55] Sezgin, M. and Sankur, B., “Survey over Image thresholding techniques and Quantitative performance evaluation,” *Journal of Electronic Imaging,* vol.13, no. 1, pp. 146-168, 2004.

[56] Zhang, S. and Salari, E., “Reducing artifacts in coded images using neural network aided adaptive FIR filter,” *Journal of Neurocomputing*, vol. 50C, pp. 249-269, 2003.

[57] Coman, M. Stan, S. Manic, M. and Balan, R., “Application of Distance Measuring with Matlab/Simulink,” *Third Conference on human System Interaction,* Rzeszow, pp. 113-118, 13-15 May, 2010.

[58] Gat, I. Benady, M. and Shashua, A., “A Monocular Vision Advance Warning System for the Automotive Aftermarket,” *SAE World Congress & Exhibition,* Detroit, USA, 8 pages, 2005.

[59] Deshmukh, P. D. and Dhok, G. P., “Analysis Of Distance Measurement System Of Leading Vehicle,” *International Journal of Instrumentation and Control Systems*, vol. 2, no. 1, 2012.

[60] Gonzalez, R. C. Woods R. E. Digital Image Processing (Second Edition). Beijing: Publishing House of Electronics Industry, 2007.

[61] Corke, P.I. Visual control of robots High performance visual servoing (Robotics and Mechatronics Series 2), 1996.

APPENDIX A: FIGURES

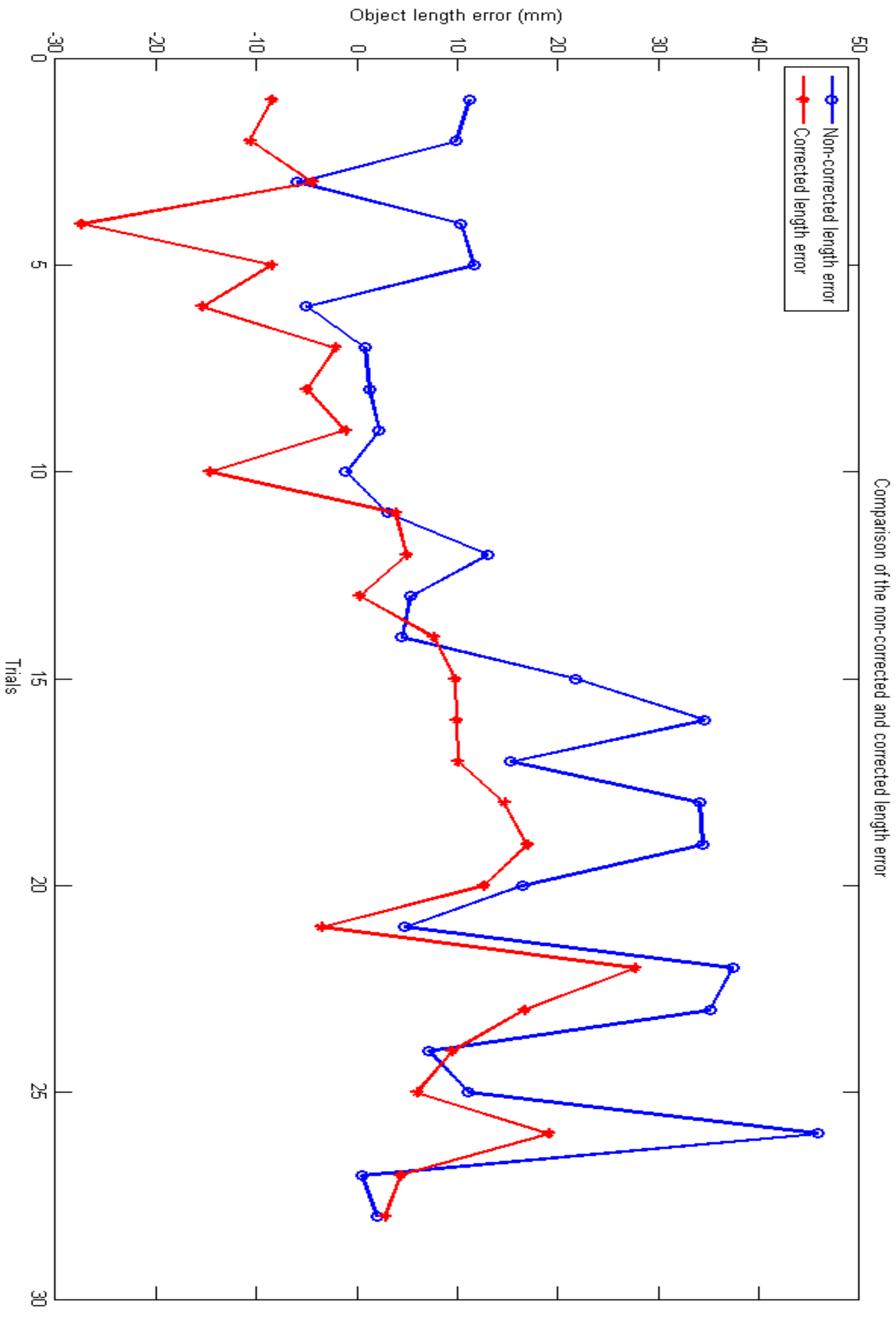


Figure 1: Comparison of the non-corrected and corrected length

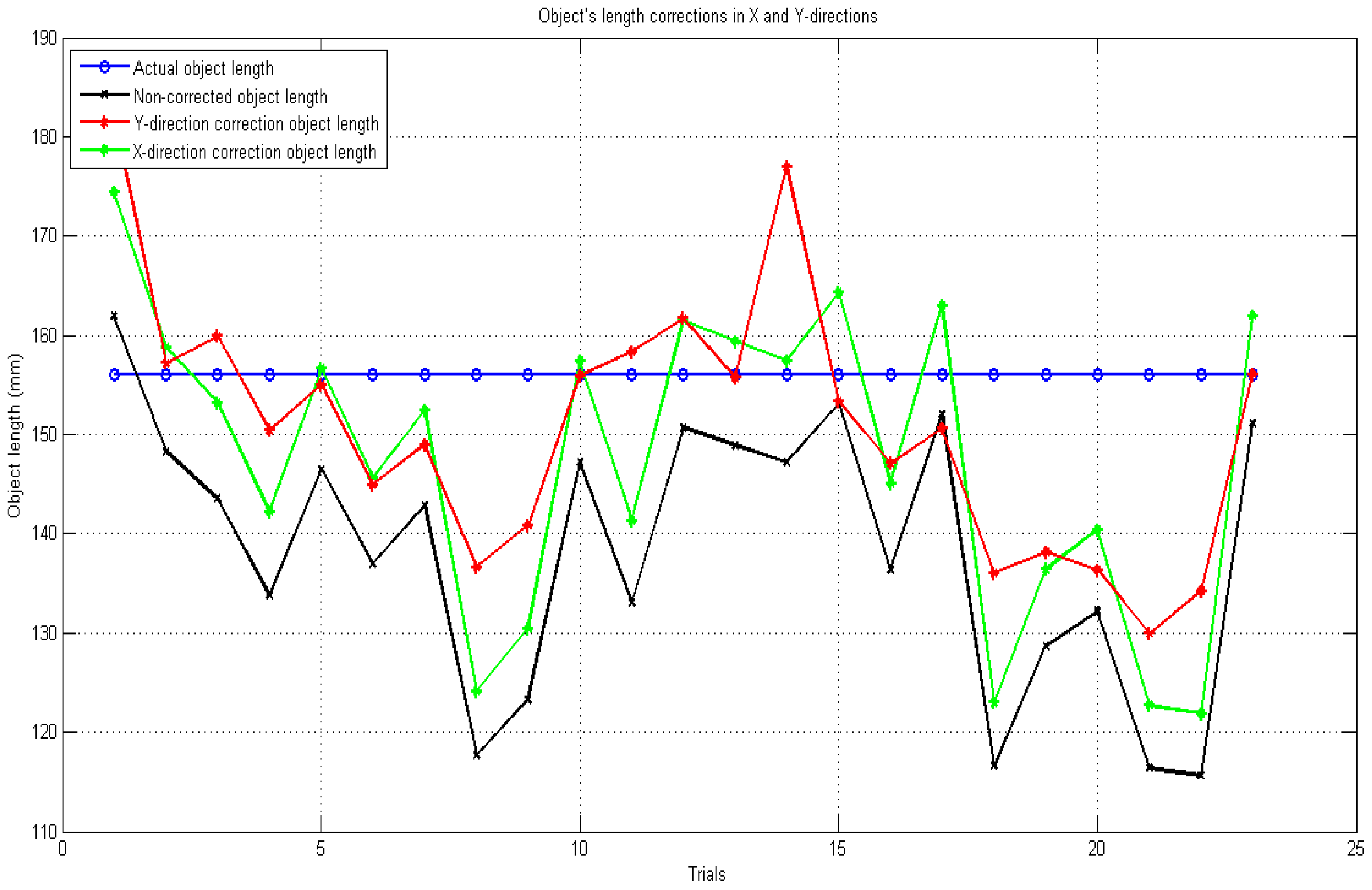
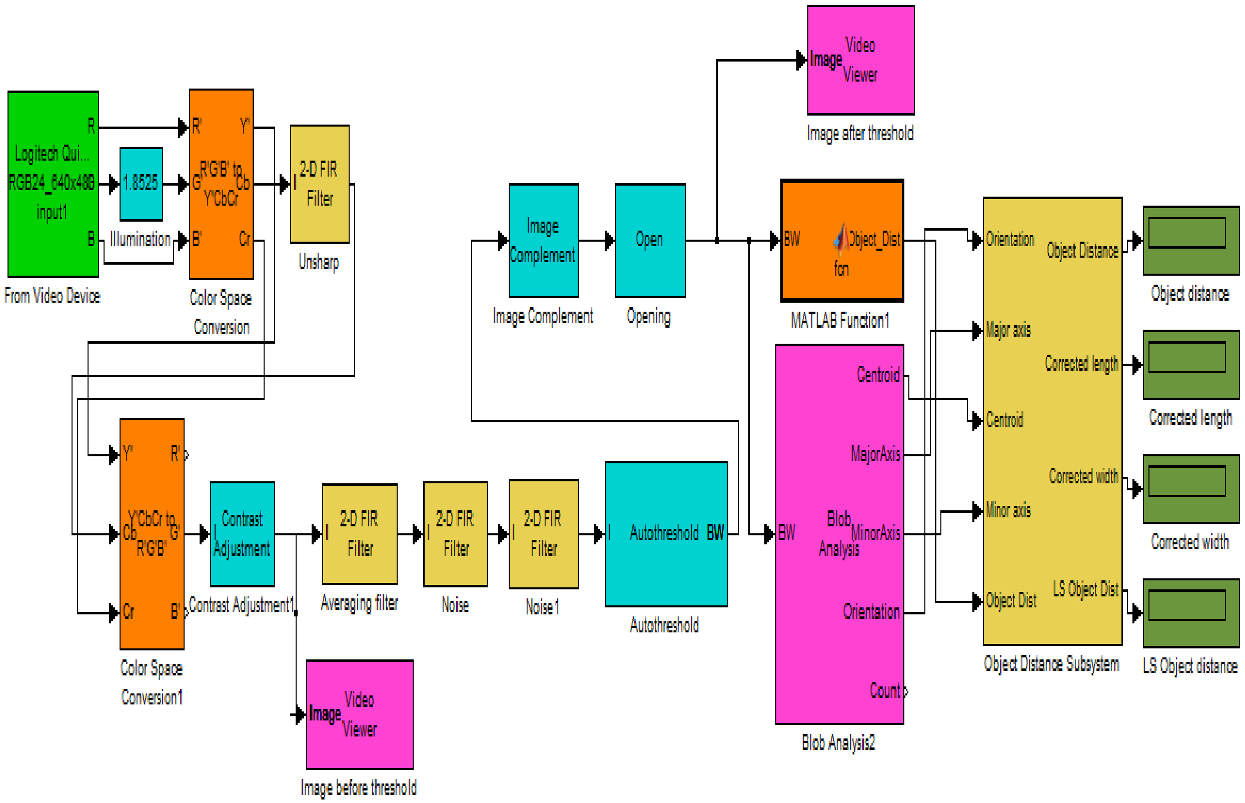
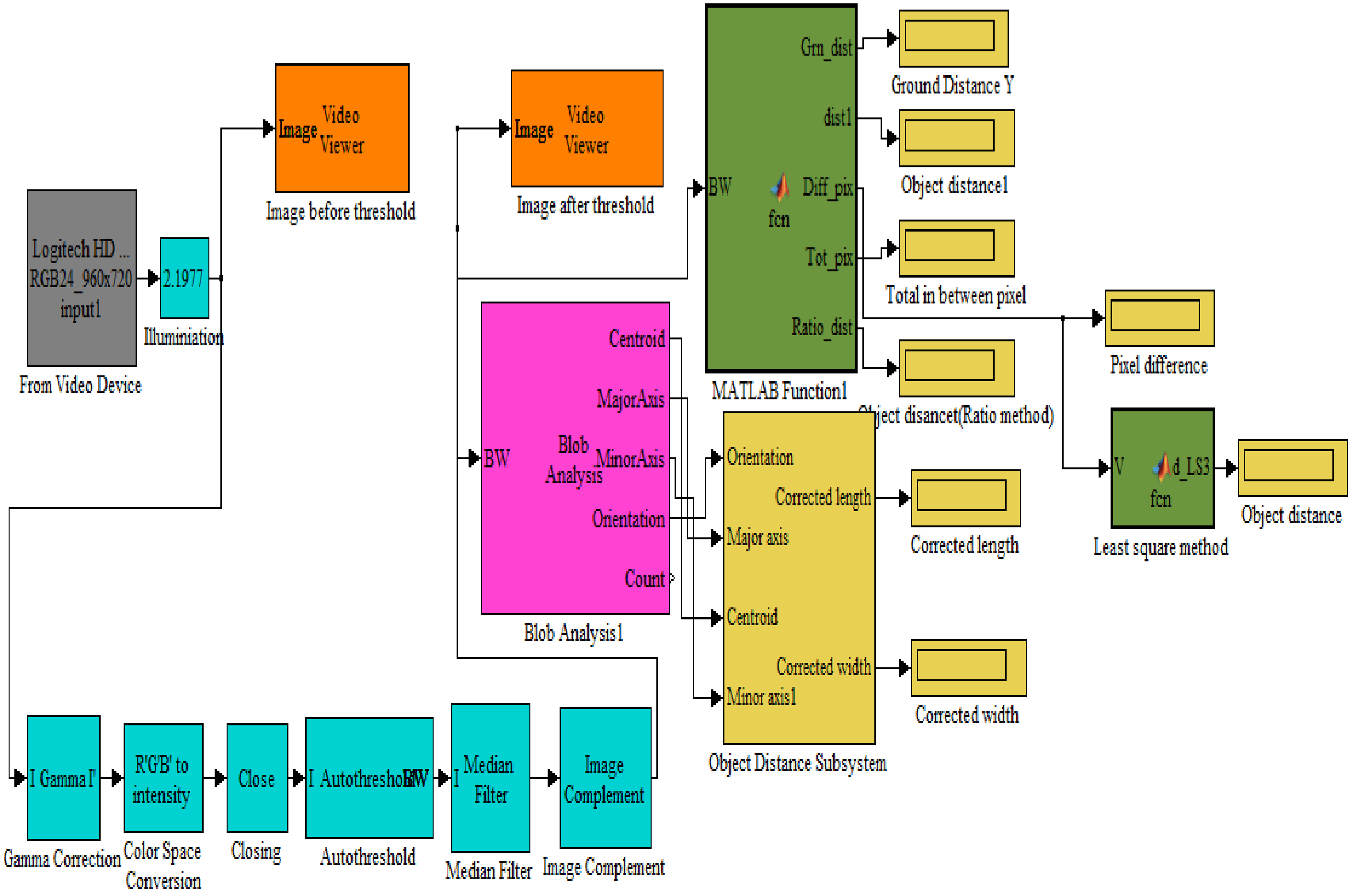


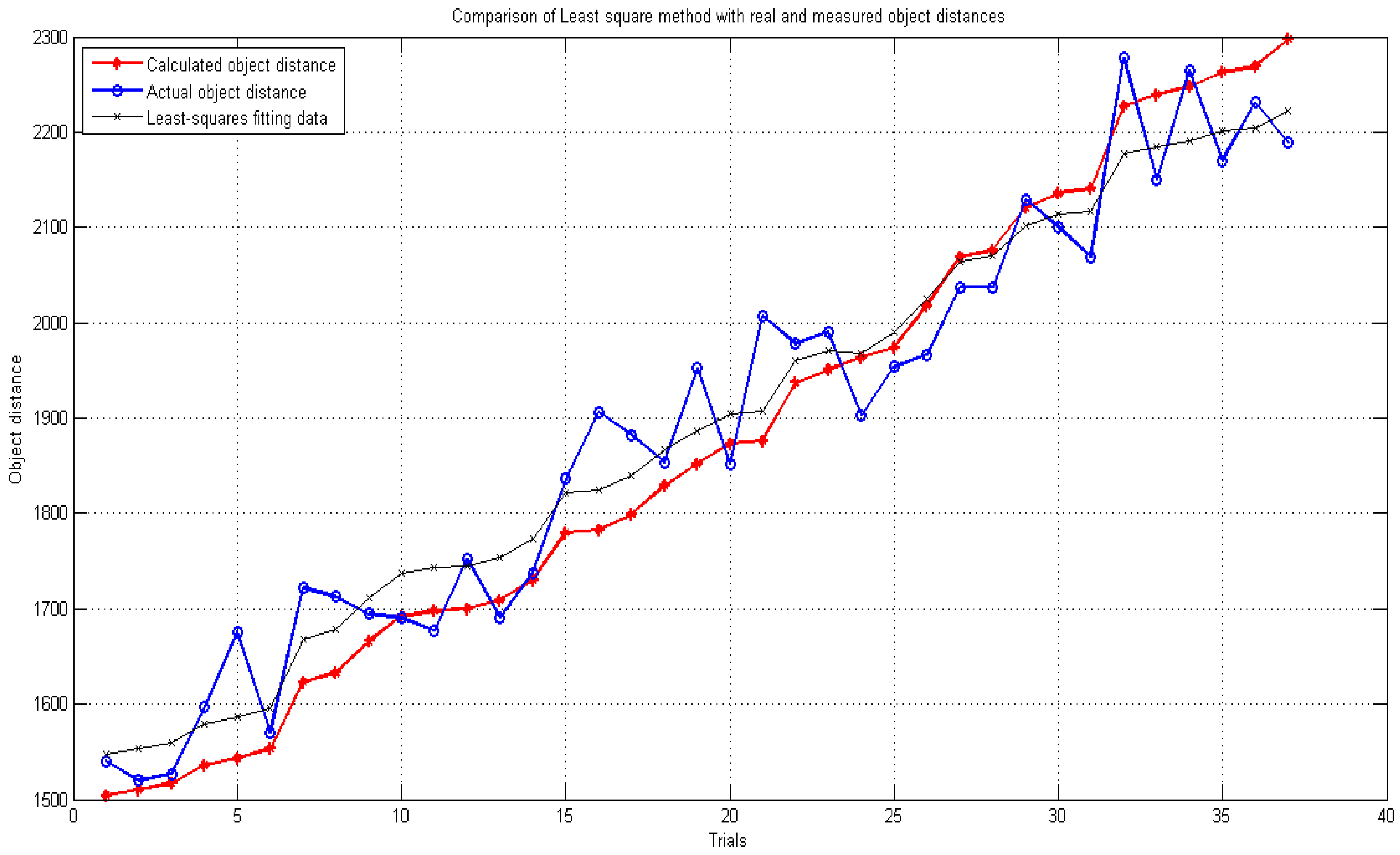
Figure 2: Object's length corrections in X and Y directions.



Object distance measurement using single fixed camera



Object distance measurement using single camera with variable pitch angle



Comparison of the actual, calculated and least square method for object distance