8-2011

EXTRACTING DEPTH INFORMATION FROM STEREO VISION SYSTEM, USING A CORRELATION AND A FEATURE BASED METHODS

Mahmoud Abdelhamid
Clemson University, mabdelh@clemson.edu

EXTRACTING DEPTH INFORMATION FROM STEREO VISION SYSTEM, USING A CORRELATION AND A FEATURE BASED METHODS

A Thesis
Presented to
The Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

by
Mahmoud Abdelhamid
August 2011

Accepted by:
Dr. Imtiaz Haque, Committee Chair
Dr. Mohammed Omar
Dr. Ardalan Vahidi

ABSTRACT

This thesis presents a new method to extract depth information from stereo-vision acquisitions using a feature and a correlation based approaches. The main implementation of the proposed method is in the area of Autonomous Pick & Place, using a robotic manipulator. Current vision-guided robotics are still based on a priori training and teaching steps, and still suffer from long response time.

The study uses a stereo triangulation setup where two Charged Coupled Devices CCDs are arranged to acquire the scene from two different perspectives. The study discusses the details of two methods to calculate the depth; firstly a correlation matching routine is programmed using a Square Sum Difference SSD algorithm to search for the corresponding points from the left and the right images. The SSD is further modified using an adjustable Region Of Interest ROI along with a center of gravity based calculations. Furthermore, the two perspective images are rectified to reduce the required processing time. Secondly, a feature based approach is proposed to match the objects from the two perspectives. The proposed method implements a search kernel based on the 8-connected neighbor principle. The reported error in depth using the feature method is found to be around 1.2 mm.

DEDICATION

This thesis is dedicated to my parents, my beloved wife Ala and all my family.

ACKNOWLEDGMENTS

I would like to extend my gratitude to my research advisor Dr. Mohammed Omar. His support and guidance throughout these years in graduate school have made it a very productive experience.

I would like to thank my committee chair, Dr. Imtiaz Haque and Dr. Ardalan Vahidi for all their inputs and suggestions to help see this work to completion.

Special thanks to my beloved wife Ala Qattawi who has provided me with unconditional love, patience and supported me to complete this work. I would also like to thank my sister Rania, her husband Mohammad and my nephew Yanal who have been my small family in US.

Finally, I would like to thank all my professors in Clemson University for their quality teaching and all my colleagues especially my old roommates Amin and Ali. Also I would like to thank my lab colleagues for all their thoughts, suggestions and discussions that kept the research work going.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table

Page

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 Pick and Place Robots

An industrial robot is defined as an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes. The field of robotics may be more practically defined as the study, design and use of robot systems for manufacturing.

Typical applications of robots include welding, painting, assembly, pick and place, packaging and palletizing, product inspection, and testing, all accomplished with high endurance, speed, and precision.

Pick and place robot application are among the most popular material handling systems. They provide dependable solutions for production lines. Pick and place robot performs the following advantages:

- Speed - Pick and place robots allow for faster cycle times.
- Accuracy - Robotic systems are more accurate and consistent than their human counterparts.
- Production - Work cells create more because they perform applications with more accuracy,

- Reliability - Robots can work 24 hours a day, seven days a week without stopping or tiring.

Computer vision and pattern recognition techniques have been widely used for industrial applications and especially for robot vision. In many fields of industry, indeed, there is the need to automate the pick-and-place process of picking up objects, possibly performing some tasks, and then placing down them on a different location. Most of the pick-and-place systems are basically composed of robotic systems and sensors. These sensors are in charge of driving the robot arms to the right 3D location (and possibly orientation) of the next object to be picked up, according to the robot’s degrees of freedom. The placing points are usually predetermined and sensors are rarely used to guide the place phase. Conversely, object picking can be very complicated if the scene is not well structured and constrained Flexibility , So computer vision is the way to extract information from the image used to guide the robot for pick and place application. Image parameters are discussed in section 1.2.

1.2 Image Parameters

An image is defined as the projection of a set of points from the real plane (object space) into the image plane [1]. For an image *E*, it assumes a unique determined value $E(P) = E(x, y)$ for each image point or location $P = (x, y)$, so that the image value can be a single value of a certain measure of size or a numerical gray value i , which represents the intensity value or the gray tone value. So that the image E can be represented as: $E(x, y) = i$.

A digital image is distinguished by its discrete image points values, such that the coordinates x and y of an image point (x, y) are assumed to be integers with their values ranging from $1 \leq x \leq M$ and $1 \leq y \leq N$, where M and N define the image spatial resolution, whereas the value of a gray image pixel is between $0 \leq i \leq 255$ for 8 bit images. On the other hand, a binary image pixel value can only be 0 (black) or 1 (white), where the white represents the objects and the black describes the background contribution. The binary image is also called a bi-level or a two-level image because each pixel is stored as a single bit 0 or 1.

To represent color images, all the colors are can be represented as a weighted sum of the three primary colors Red, Green and Blue or RGB, whereas each primary color is identified based on their wavelength; a wavelength of 700 nm represents the Red, 546 nm the Green, and the 435 nm the Blue. So all the color values are defined as specific discrete integer value 0, 1,….., G_{max} where G_{max} = 255 for 8 bit images. The primary colors will be defined as; the Red (G_{max}, 0, 0), the Green (0, G_{max}, 0), and the Blue (0, 0, G_{max}), while other colors can be represented as the white being (G_{max}, G_{max}, G_{max}) and the black as (0, 0, 0), Yellow (G_{max}, G_{max}, 0), and the Magenta as (G_{max}, 0, G_{max}).

1.3 Pin Hole Camera Model

The pinhole camera model describes the mathematical relationship between the coordinates of a 3D point and its projection onto the image plane through an ideal pinhole camera, where the camera aperture is described as a point without a lens to focus the incoming light.

This ideal model is a simplistic model used as a first step in camera calibration, because it does not include any geometric distortions, optical aberrations, or blurring caused by the geometry of a pin hole camera; this model is depicted in Figure1.1.

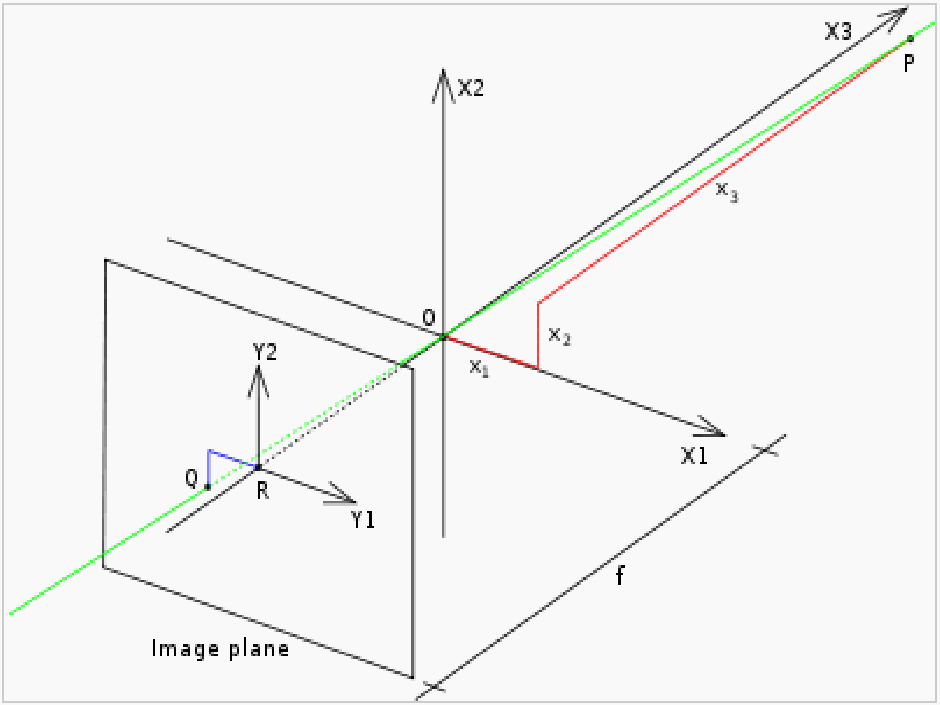


Figure 1. 1 Pin Hole Camera Model (Source Kletter [1])

The camera aperture is assumed to be located at the origin O, whereas the three axes X1, X2 and X3 are referred to as the coordinate system with the Axis X3 pointing in the viewing direction of the camera. The image plane is where the image is projected and located at a distance *f* where, *f* is the focal length of the camera. The point R is where the optical axes and the image plane are intersected, typically called the image center or the principal point.

The pinhole model is typically used to establish the relationship between any point in the real world coordinate (object space) as *P* with coordinates of (*x1, x2, x3*) and its projection in the image plane *Q* with coordinate (*y1*, *y2*); knowing that this projection will pass through the aperture center, which is assumed to be considered the point *O*.

The relationships between these two coordinate systems (object and image planes) can be found through triangles' similarity as described in equations 1.1 and 1.2:

$$\frac{-y_1}{f} = \frac{x_1}{x_3} \text{ or } y_1 = \frac{-f.x_1}{x_3} \quad (1.1)$$

$$\frac{-y_2}{f} = \frac{x_2}{x_3} \text{ or } y_2 = \frac{-f.x_2}{x_3} \quad (1.2)$$

The final relationship written in matrix form is in equation (1.3)

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{-f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (1.3)$$

1.4 Image Transformation

For images obtained from a 3D scene, their exact position and their camera orientation is unknown, if a global reference frame is not defined. The process of image transformation describes the relationship between the coordinate systems. The image transformation includes both translation and rotation [2]. So, for a general point in space (X, Y, Z), its translation will affect it through displacing its coordinates in a certain direction in a specified value, where its new coordinates are changed into (X',Y',Z') where:

$$X' = X + T \tag{1.4}$$

$$Y' = Y + T \tag{1.5}$$

$$Z' = Z + T \tag{1.6}$$

On the other hand, the rotation can affect any or all of the axes; Z-axis, X-axis or Y-axis with the new point coordinates being dependant on the rotation type.

For example, the rotation of the 2D object point (X, Y) around Z-axis through an angle θ is displayed in Figure 1.2.

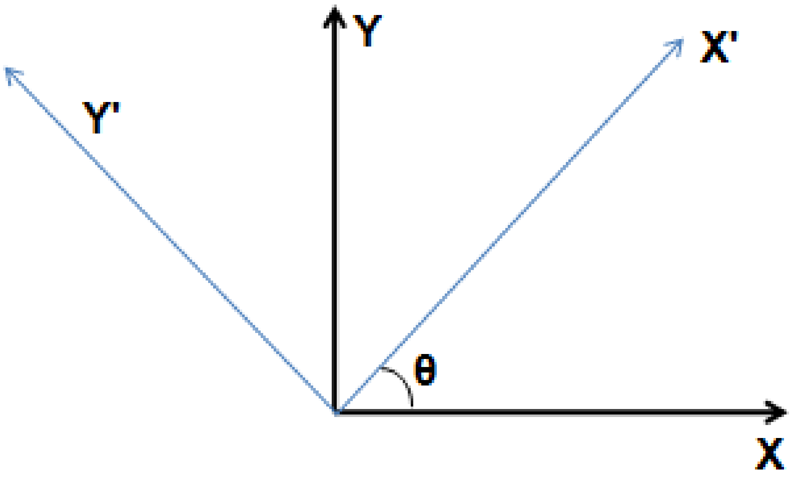


Figure 1. 2 Rotation θ Around the Origin

The coordinate of a general point (*X*, *Y*) will change through rotation into *X'* and *Y'* as computed in equations 1.7 and 1.8;

$$X' = X\cos\theta - Y\sin\theta \tag{1.7}$$

$$Y' = X\sin\theta + Y\cos\theta \tag{1.8}$$

And in matrix notation in equation (1.9);

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{1.9}$$

If, the rotation is around Z-axis for the 3D object point (X, Y, Z), then the matrix of rotation θ around the Z-axis can de described in equation (1.10)

$$Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.10}$$

However if the rotation is around X-axis for the 3D object point (X, Y, Z), then the matrix of rotation ψ around the X-axis is in equation (1.11);

$$X(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \tag{1.11}$$

Lastly, if the rotation is around the Y-axis for the 3D object point (X, Y, Z), then the matrix of rotation φ around the Y-axis is in equation (1.12);

$$Y(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \tag{1.12}$$

So, by applying sequences of rotations around the Z-axis first, then around the X-axis, then around the Y-axis, the final rotation *R* can be described mathematically in equation (1.13);

$$R = X(\psi)Y(\varphi)Z(\theta) \tag{1.13}$$

So, in matrix notation, the general results for an arbitrary *R* can be written as:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{1.14}$$

To combine a rotation matrix of 3x3 with a translation matrix of 3x1 together, a homogenous coordinate system is used, where the fourth dimension is added and the general rotation expressed as in equation (1.15)

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1.15}$$

While the general translation matrix becomes:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_1 \\ 0 & 1 & 0 & T_2 \\ 0 & 0 & 1 & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1.16}$$

And, lastly the displacement matrix is the translation plus the rotation, as in (1.17);

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1.17}$$

Having a homogenous coordinate system provides the convenience of having a single multiplicative matrix for any transformation, so that the homogenous coordinate system is used to provide a convenient 4x4 matrix representation for the 3D transformations of rigid bodies’ translations and rotations.

1.5 Camera Calibration

In general, when objects are viewed by the camera their position is known in the world coordinate system but yet still unknown in camera coordinate system or image plane. Because, the camera assumes an arbitrary direction, for example in robotics pick and place, the camera coordinate system will have to be priori calibrated to guide the end effector toward objects of interest.

A useful approach [2] is to assume general transformation as discussed in section 1.4 between the world coordinate and the image seen by the camera under prospective projection and to locate in the image various calibration points, where the calibration which have been placed in known position of scene. If enough points are available then should be possible to compute transformation parameters.

Let the general transformation matrix G in this homogenous form as state in equation 1.18:

$$\begin{bmatrix} X_H \\ Y_H \\ Z_H \\ H \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \\ G_{31} & G_{32} & G_{33} & G_{34} \\ G_{41} & G_{42} & G_{43} & G_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1.18}$$

Where, the final Cartesian coordinate appearing in the image will be (*x*, *y*, z) where it equal to (*x*, *y*, *f*) and *f*: if focal length of the camera.

This are calculated from the first three homogenous coordinate by diving by the fourth coordinate as these equations 1.19 to 1.21:

$$x = X_H / H = (G_{11}X + G_{12}Y + G_{13}Z + G_{14}) / (G_{41}X + G_{42}Y + G_{43}Z + G_{44}) \tag{1.19}$$

$$y = Y_H / H = (G_{21}X + G_{22}Y + G_{23}Z + G_{24}) / (G_{41}X + G_{42}Y + G_{43}Z + G_{44}) \tag{1.20}$$

$$z = Z_H / H = (G_{31}X + G_{32}Y + G_{33}Z + G_{34}) / (G_{41}X + G_{42}Y + G_{43}Z + G_{44}) \tag{1.21}$$

For simplification, the value of *z* is known so no need to determine the parameters G_{31}, G_{32}, G_{33} and G_{34}. Also, the value of G_{44} is unity. Then, only the ratio G_{ij} need to computed so this leave only 11 parameters need to be determined.

The first two equations can be written as indicated in equations 1.22 and 1.23:

$$x = G_{11}X + G_{12}Y + G_{13}Z + G_{14}) - x(G_{41}X + G_{42}Y + G_{43}Z) \quad (1.22)$$

$$y = G_{21}X + G_{22}Y + G_{23}Z + G_{24}) - y(G_{41}X + G_{42}Y + G_{43}Z) \quad (1.23)$$

So, single world point (X, Y, Z) which is known to correspond to image point (*x*, *y*) give just two equations. To provide all the 11 parameters it requires minimum six points. And the world points should use in calibration should lead to independent equations so they should not be coplanar.

Increase the number of points leads to increase the accuracy of the calibration and least square analysis can be used to perform the computation of the 11 parameters by using least square method, this method will find in section 1.6

The 2n equations expressed in matrix form like:

$$Ag = \xi \quad (1.24)$$

Where, A is 2n X 11 matrix confidents, which multiplies the G matrix in the equation 1.25:

$$g = \begin{pmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{21} & G_{22} & G_{23} & G_{24} & G_{41} & G_{42} & G_{43} \end{pmatrix}^{T} \quad (1.25)$$

And ξ is 2n element column vector of image coordinates. The least square solution is:

$$g = A^{\dagger}\xi \text{ Where, } A^{\dagger} = (A^{T}A)^{-1}A^{T} \quad (1.26)$$

The least square theory is discussed in section 1.6

1.5.1 Intrinsic and Extrinsic Parameters

The idea of the camera calibration is to bring the camera and world coordinate system into coincidence. The first step is to move the origin of the world coordinate to the origin of the camera coordinate system. The second step is to rotate the world coordinate system until its axes be coincident with the camera coordinate system. The third step is to move the image plane laterally until there is agreement between the two coordinate systems.

The complete transformation for the camera calibration [2] could be written as form: G=*PLRT*

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & T_1 \\ 0 & 1 & 0 & T_2 \\ 0 & 0 & 1 & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1.27)$$

Where, *P* is perspective transformation required to form the image and *P* and *L* matrices together called intrinsic camera parameters and *R* and *T* matrices together called extrinsic camera parameters and the transformation matrix G written as:

$$G = G_{\text{internals}}\, G_{\text{externals}} \qquad (1.28)$$

Where:

$$G_{\text{int}ernal} = PL = \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 1/f & t_3/f \end{bmatrix} \qquad (1.29)$$

For good calibration, the initial translation matrix *T* move the camera's centre of projection to the correct position so the value of t_3 will be zero and the G_{internal} matrix written in 2D as the equation 1.30:

$$G_{\text{internal}} = \begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1/f \end{bmatrix} \qquad (1.30)$$

The external G_{external} written as the equation 1.31, where this matrix 4X4 homogenous coordinate system and the matrix shows results of in terms of the rows *R_1*,*R_2* and *R_3* and have taken dot product with *T* (*T_1*, *T_2*, *T_3*)

$$G_{external} = RT = \begin{bmatrix} R_1 & R_1.T \\ R_2 & R_2.T \\ R_3 & R_3.T \\ 0 & 1 \end{bmatrix} \qquad (1.31)$$

1.6 Least Square Minimization

In this thesis, we need to find solution for linear equations obtained by calibration and triangulation. So the linear equitation [3] of the form:

$$Ax = b$$

If A be an m X n matrix, then there are three possible solutions for this system:

a) If m < n there are more unknown than equations, there will not be a vector space of solutions not a unique solution.

b) If m=n there will be unique solution as A is invertible.

c) If m > n then there are more equations than unknowns.

To use least square minimization for full rank case, the case will be m greater or equal n and assume A is known to be rank of n. The idea here is to seek for x such that ||Ax-b|| is minimized, where || . || represents vector norm.

Such an x is known as least square solution to the system and found using singular value decomposition (SVD) as follow. Seeking x that minimize ||Ax-b|| = ||UDV^T x-b||. Because the norm preserving property of orthogonal transforms,

||UDV^T x-b|| = ||DV^T x- U^T b||. Consider y= V^T x and b'= U^T b, the problem becomes one of minimizing ||Dy- b'|| where D is an m x n matrix with vanishing off-diagonal entries and take this form:

$$\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & . & \\ & & & d_n \\ & & & \\ & 0 & & \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \\ .. \\ y_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ . \\ b'_n \\ b'_{n+1} \\ . \\ . \\ b'_m \end{pmatrix} \qquad (1.32)$$

So, the nearest Dy can approach to b' is the vector (b$_1$', b$_2$', b$_3$',0,…,0)^T

And this is achieved by setting $y_i = b'_i / d_i$ for i = 1,…….,n. So the assumption full rank where rank A= n ensure that do not equal zero.

Finally, the solution of $x = Vy$

1.6.1 Least Square Minimization Using Normal Equation

The linear least squares problem also can be solved by using a method called normal equations. To solve linear set of equations: $Ax = b$ Where, the solution is to find the vector x that minimize norm ||Ax-b||. As the vector x varies over all values, the product Ax varies over the complete column space of A. that is the subspace of IR^m spanned by the columns of A. So the idea here is to find the closest vector of b that lies in the column space of A. if X is the solution then Ax is the closed point to b. The difference Ax-b must be a vector orthogonal to the column of A. This mean explicitly, that Ax-b is perpendicular to each of the column of A, and hence

$A^T(Ax - b) = 0$

By rearrange the above equation: $(A^T A)x = A^T b$

This is a square n x n set of linear equations, called normal equation [2]

The solution will be: $x = A^{\dagger} b$, where $A^{\dagger} = (A^T A)^{-1} A^T$ (1.33)

1.7 Triangulation Theory

The triangulation is the problem of finding the position of a point in space given its position in two images taken with cameras with known calibration and pose. This process requires the intersection of two known rays in space. In the absence of noise, this problem is solved as use the equation below. but in real case where the noise is present, the two rays will not generally pre meet, in which case it is necessary to find the best point of intersection the details of the real case will be topic define in this thesis. Now the simple triangulation where the noise is absent will depend on the geometry of the two cameras [4]:

1.7.1 Geometry for parallel cameras

This is the standard model of the triangulation as shown in the Figure 1.3, where left camera L and the right camera R are two pinhole cameras with parallel optical axes and f is the focal length of both cameras. The line b connecting the two lenses is called baseline b. and this line perpendicular to the optical axes.

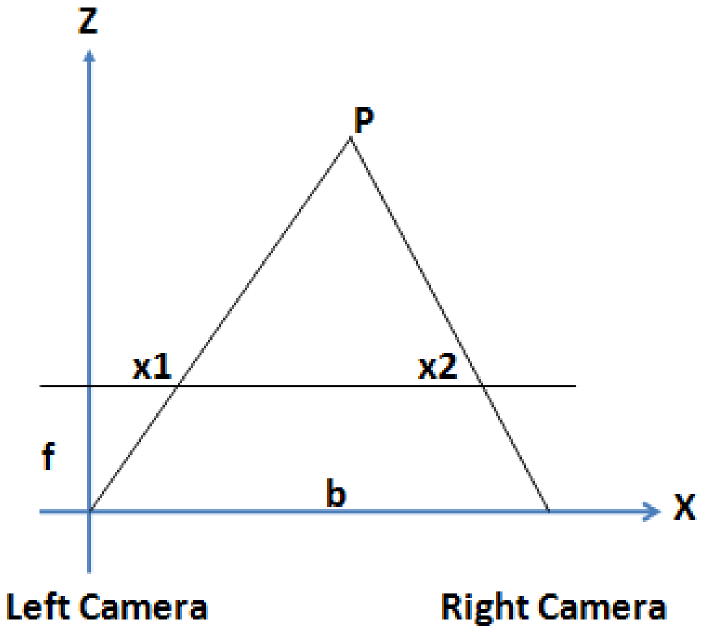


Figure 1. 3 Geometry for Parallel Cameras

XZ is the plane where the optical axes lie, *XY* plane is parallel to the image plane of both cameras, *X* axis equals the baseline and the origin O of (*X*,*Y*,*Z*) world reference system is the lens center of the left camera.

In this setting the equations of stereo triangulation are found using equations 1.34 to 1.36:

$$Z = \frac{b.f}{x_1 - x_2} \qquad (1.34)$$

$$X = \frac{x_1.Z}{f} \qquad (1.35)$$

$$Y = \frac{y_1.Z}{f} \qquad (1.36)$$

1.7.2 Non-parallel cameras

The left camera is considered as reference camera and the right camera can be rotated with respect to the left camera in three directions.

- Rotation around Y axis (θ): In this case the optical axes are not parallel, but they both lie on the XZ plane, so they intersect in a point (0,0,Z_0), that is called fixation point and could also be behind the cameras (Z_0<0).If theta is the rotation angle, then: $Z_0 = \frac{b}{\tan\theta}$

Under small angle approximation [4], the assumption still valid and the right image plane to be parallel to the left image plane and hence to XY plane.

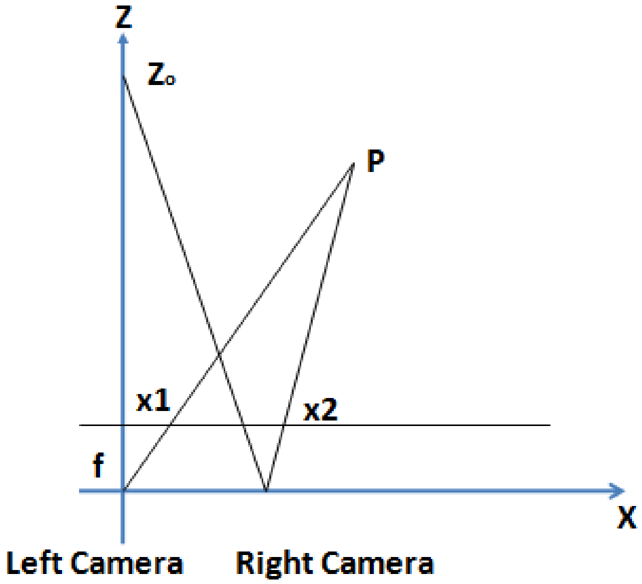


Figure 1. 4 Geometry for Non Parallel Cameras

In this case the *X*, *Y* and *Z* found using equations 1.37, 1.38 and 1.39:

$$Z = \frac{b.f}{x_1 - x_2 + \frac{f.b}{Z_0}} \tag{1.37}$$

$$X = \frac{x_1.Z}{f} \tag{1.38}$$

$$Y = \frac{y_1.Z}{f} \tag{1.39}$$

- Rotation around X axis (ϕ): Rotation around X axis only affects the Y coordinate in reconstruction. Let ϕ be the rotation angle, then stereo triangulation is

$$Z = \frac{b.f}{x_1 - x_2} \qquad (1.40)$$

$$X = \frac{x_1.Z}{f} \qquad (1.41)$$

$$Y = \frac{y_1.Z}{f} + \tan\phi.Z \qquad (1.42)$$

- Rotation around Z axis (Ψ): Rotation around optical axis is usually dealt with by rotating the image before applying matching and triangulation. In the following the rotation angle of the right camera around its optical axis will be called psi.

The general case, given the translation vector T and rotation matrix R describing the transformation from left camera to right camera coordinates, the equation to solve for stereo triangulation is:

$$P^{'} = RT(P - T) \qquad (1.43)$$

Where *p* and *p*' are the coordinates of P in the left and right camera coordinates respectively.

CHAPTER 2

2. LITERATURE REVIEW

2.1 Introduction

Stereo vision system is similar to the concept how Human beings have the ability to see an object in 3D and how they can have sense of the depth between objects. The left eye gives us left perspective of the object which differs from the right perspective created by right eye and when a person stares at an object, the two eyes converge so that the object appears at the center of the nerve layer that lines the back of the eye, senses light, and creates impulses that travel through the optic nerve to the brain which is called retina. The object appears on the center of retina in both eyes.

The brain receives the two perspective images from the two eyes and combines them to make one single mental image of a scene called cyclopean image [5]. The cyclopean view is a view constructed by the coherence-network; it was never taken by any camera.

This cyclopean representation of direction can be viewed as the mean of two vectors emanating from corresponding points relative to the fovea on the left and right retinas. The third dimension, depth (that is, the egocentric distance of the object from the cyclopean eye), has to be extracted from retinal disparity: the difference between the images obtained from the right and left eyes. However, it is still controversial how the brain computes depth. [6]

Each individual eye can only produce two dimensional images provided to the brain. However, with the help of two eyes combined, the brain is able to measure distance of objects, and give the human sense of the three dimensional world.

The process of combining images from two or more sources, either biological or manmade, resulting in a 3-D image is called stereovision which define [7] as a process directed at understanding and analyzing three dimensional objects based on image data.

2.2 Stereo vision

Different techniques and methods have been proposed and employed to approach the solution of stereovision problem. The sequence of Stereovision techniques found in literature review almost similar for biological view for the problem and have commonly the following sequence of processing steps [7].

- Image acquisition – The cameras type will discuss later in chapter three.
- Camera Modeling - known as camera calibration.
- Correspondence analysis - an automatic computer process of determining image points correspondents.
- Triangulation - a geometric technique of depth measurement given a point in an image and its correspondent in the other image.
- Interpolation- representation of the depth data in 3-D space.

2.3 Camera Calibration

Camera calibration is define as the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3-D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters) [8].

Calibration consider as the first step in stereo vision procedure and in many cases, the overall performance of the machine vision system depend on the calibration accuracy.

Very large literature review on this subject found to solve this problem where the estimation for the distortion which occurs between points on the object and the location of these points should made and the approximation model try to be close to the real model and the way the calibration beside the parameters includes to make this model is differ from one calibration method to another. A listing of 91 articles on aspects of camera calibration for the period 1889 until 1951 is provided by Roelofs [9]. According the literature review we can divide the calibration techniques to four categories:

1- Reference object based 3D calibration.

Camera calibration is performed by observing a calibration object whose geometry in 3D space is known with very good precision. Calibration can be done very efficiently [10]

Tsai [8] using this method by observing a calibration object whose has very known geometry in 3D space. The calibration object usually consists of two or three planes orthogonal to each other e.g. calibration cube. Sometimes, a plane undergoing a precisely known translation is alsowhich equivalently provides 3D reference points. The advantage for this method it is simple theory and most accurate calibration and the disadvantage it is more expensive and need more elaborate setup.

2- Plane based 2D calibration.

Zhang used calibration based on 2D and required to observe a planar pattern shown at a few different orientations different from Tsai's technique, the knowledge of the plane motion is not necessary. Because almost anyone can make such a calibration pattern, the setup is easier for camera calibration [11]. This method will discuss more in chapter three.

3- Plane calibration 1D

This technique is relatively new technique proposed by Zhengyo [12]. Calibration object is a set of collinear points, e.g., two points with known distance, three collinear points with known distances, four or more. The Camera can be calibrated by observing a moving line around a fixed point, e.g. a string of balls hanging from the ceiling and can be used to calibrate multiple cameras at once. This method is good for network of cameras mounted apart from each other, where the calibration objects are required to be visible simultaneously.

4- Self-calibration technique

This method of calibration consider as zero Dimensional technique because it makes calibration without the need of any object and only image point correspondences are required. Just by moving a camera in a static scene, in this method there is need to estimate large number of parameters because there is no calibration Objects. This method consider as a much harder mathematical problem. Sometimes pre-calibration is impossible (e.g., a scene reconstruction from an old movie), self-calibration is the only choice. A recent overview of this area can be found in [13].

In term of accuracy the four calibration techniques arrange from using (3D calibration object then 2D planer pattern then using 1D object then Self calibration) and in term of difficulty the totally opposite direction, self calibration is the most difficult one and 3D calibration is the easiest one.

2.4 Correspondence analysis

The next step in stereo vision after make calibration is to try to find related points between each point capture for both cameras. As biological principle discuss above for retinal disparity there is a need to find the disparity map between the two images captures from both camera in stereo vision scenario. In literature review this problem divide to two types that are discussed in the following two sections.

2.4.1 Correlation based matching

This technique use correlation to compare the brightness (intensity) between a pixel in one image to intensity of the pixel in another image. A point of interest is chosen in one image called window then a cross-correlation measure is then used to search for a same window size with a matching neighborhood in the other image. The disadvantage of this technique is depend on using intensity values at each pixel directly, and always the images are sensitive to distortions as a result of changes in viewing position (perspective) as well as changes in absolute intensity, contrast, and illumination. Also, the presence of occluding boundaries in the correlation window tends to confuse the correlation-based matcher, often giving an erroneous depth estimate.

Barnard and Fischler point out the relation between correlation and the window size [14]. The window size must be large enough to include enough variations of the intensities for matching and should be small enough to avoid the effect of projection distortion.

Correlation based matching give poor disparity estimate if the window size is too small and does not cover enough intensity variations and the signal to noise ratio in this case will be low in other word the intensity variation to noise will be low. On the other hand if the window size is too large and includes many intensity variations the position of the correct correlation could not be correct and the accuracy for this method decrease.

Takeo solved this problem and proposed matching algorithm with an adaptive window [15]. The idea here is make the window size adaptive and not fixed size to avoid the problems appears to choose the size of the window. Takeo method is to evaluate the local variation of the intensity and the disparity. Then employ statistical model of the disparity distribution within the window. The method assess how the disparity variation in additional to intensity variation. Finally search for window size that produces the estimate of disparity with least uncertainly for each pixel of an image. The advantages for this method are to solve the difficulty to choose window size also the method controls not only the size of the window but also the shape (rectangle) of the window. The disadvantage of this method is the difficulty in evaluation the disparity variances.

One example using correlation based matching is the method proposed by Moravec [16]. His method based on the operator that computed the local maxima of a directional variance measure over a 4 x 4 (or 8 X 8) window around a point. The sums of squares of differences of adjacent pixels were computed along all four directions (horizontal, vertical, and two diagonal), and the minimum sum was chosen as the value returned by the operator. The site of the local maximum of the values returned by the interest operator was chosen as a feature point. Another similar algorithm found in literature review called Sum of squared differences (SSD) will discuss later on chapter three.

2.4.2 Feature-based matching

Feature-based stereo techniques use symbolic features derived from intensity images rather than image intensities themselves. Hence, these systems are more stable towards changes in contrast and ambient lighting.

The features used most commonly are either edge points or edge segments (derived from connected edge points) that may be located with sub pixel precision. Feature based methods faster than correlation-based area matching methods but more complicated.

Many method found in literature review used the feature based method based on the edge detection techniques especially Marr-Hildreth edge operator [17] and the edge detectors proposed by Canny [18]. Many methods using edge detection technique found on Umesh's review paper [19].

Prashan and Farzad [20] made corresponding algorithm based on Moment invariant. This proposed method using Harris corner detection to produce reliable feature points where Moment invariants are invariant to rotation, scale and shift and the rotation invariant property is especially beneficial to the stereo correspondence problem as any misalignment or non-flat ground conditions can create slightly rotated versions of any scene in any one of the cameras. An image can be separated to a collection of blocks and they can be marked as candidates or not depending on whether they occupy corners. These blocks can be matched with the help of moment invariants and disparity of the identified features can be simply calculated. This technique consider very accurate for achieving simultaneously the high quality range obtained from global optimization with the fast run-times of local schemes.

Maurizio [21] used a method algorithm proposed by Scott and Longuet for finding corresponding features in planar point patterns direct method for stereo correspondence based on Singular Value Decomposition SVD [22]. This method started by building a proximity matrix G of the two sets of the feature where each element G_{ij} for this function is Gaussian weighted distance between the two feature I_i and I_j then perform the singular value decomposition to get diagonal matrix D contain the positive singular value. then try to calculate the new matrix called P, if P_{ij} is both the greatest element in row and column then the two feature I_i and I_j is consider corresponding between each other. The disadvantages for this method it is not too accuracy and also the time cost to run this algorithm consider high if you looking for many features in image.

2.5 Triangulation on Literature Review

The output of calibration step and the output of corresponding algorithm needed to solve the triangulation problem of the stereo vision technique, where the triangulation algorithm used to find the third dimension i.e the depth between two objects. The triangulation techniques found in literature review mainly divide to two main categories: Active and passive methods. The two methods use the same principle to make triangulation but the nature of the problem is different.

2.5.1 Active triangulation

The triangle to find the depth divide to three things: The light source, the illuminated spot in the scene and its image points. The principle here is single spot of light is projected (i.e. laser) onto the scene then using sensor usually CCD camera to view the scene then the depth is calculated [23] using equation 2.1

$$Z = \frac{B}{X/f + \tan\alpha} \qquad (2.1)$$

Where, *Z*: The depth, *B*= The distance between camera optical center and the laser (base line)

f: Focal length of the camera, *X*: The image location within the row where the laser stripe is detected. *α*: Projection angle of the laser in respect to *Z* axis.

The main two problems for active triangulation addressed on the same reference are: first, the resolution of triangulation is related to the distance between camera optical center and the laser and the resolution relative to depth of field. Second problem related to the frame rate of the scanner. i.e. CCD this mean the images must be acquired and proposed for each position. one solution for this problem addressed by Sato [24] his method is to projected a structured pattern over the entire scene and recover depth through triangulation by processing only a few or even one image.

The using for this type of triangulation is primary depend on the application, for example Mohammed [25] presented a combined scheme based active triangulation method , This scheme use Laser and CCD camera and by using a morphological edge detection principle the width and the depth calculated. This algorithm used for inspection for automotive polyethylene fender, the method showed good resolution for that application because the base line can be set large as need to let the laser sheet within the FOV of the camera but if using the same technique on robot application and increase the distance between laser source and CCD (baseline) this will increase the occurrence of occlusions and missing data and make the size of the system unpractical for robot embarked systems and also the proposed technique by Sato to calculate the depth through triangulation by processing only a few or even one image does not work in robotics application because it is sensitivity to ambient light and it is limited range which stem from the fact that it is difficult to project enough power in the scene at long range.

The advantages for active triangulation method are easy to implement and also there are no need to take care of camera geometry, calibration and corresponding algorithm because the correspondence problem has already been solved by using an artificial source of illumination.

2.5.2 Passive triangulation

The passive triangulation technique needs to achieve with the help with only the ambient of the existing ambient illumination. There is no illumination source. i.e. laser source , Hence to do this type of triangulation there is needed to provide the position of centers of projection, the effective focal length, the orientation of the optical axis typically all theses parameters should be known after calibration step and also there is need to make correspondence algorithm to establish the relation between features from two images that correspond to some physical feature in space.

The main difference between active and passive triangulation is that: active triangulation calculates the depth by known source and observed pixel and the passive triangulation by matching pixels in images. By using passive triangulation the problem can solve using the equations below and depend on the geometry of the both camera in relation to each other but this problem become more complicated in real world during to the present of the noise and need to make more approximation for the parameters which make the two projection line from the camera to meet. In the case of projective intersection the ideal case of the projection the equation found in literature to solve these problems are: [26]

Case One: The stereo geometry for parallel camera is shown in Figure 2.1.

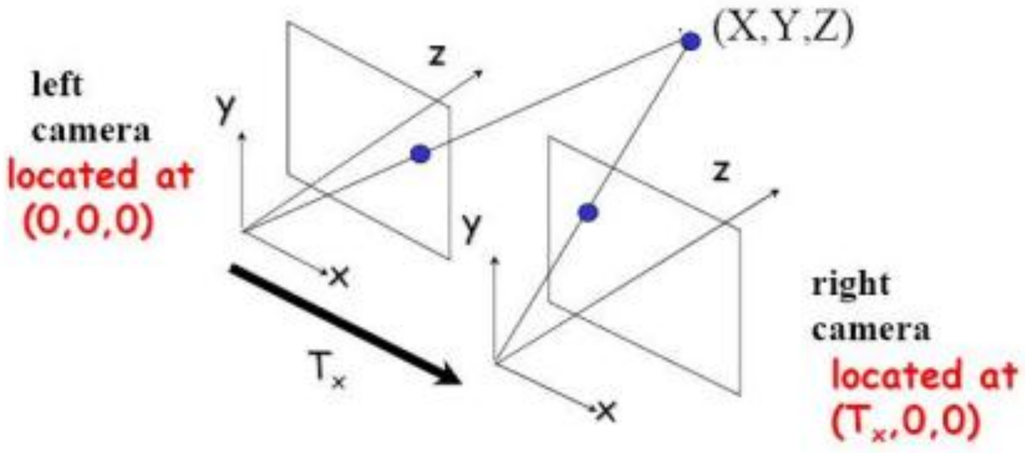


Figure 2. 1 Parallel Stereo Geometry (Marten orkman [26])

The dot point appear on image plane of left camera with coordinate $P_L(X_L,Y_L)$ and the dot point appear on the right image plane with coordinate $P_R(X_R,Y_R)$. The distance between left camera centre (optical centre) and the right camera centre (optical centre) called baseline (T_x). The distance between X_L and X_R called disparity distance (d) .The distance between optical centre of the left camera to P_L called the effective focal length (f) which equal the distance between optical centre of the right camera to P_R (we assume the two cameras are identical) .

$d = X_L - X_R$ (found from the corresponding matching algorithm). The coordinate for the 3D point in real world (*X*,*Y*,*Z*) and the relation between this coordinate and imaginary coordinate (X_L, Y_L, X_R *and* Y_R) are found by using triangle similarity as:

$$Z = \frac{T_X x f}{d} \quad , \; X = \frac{T_X x X_L}{d} \; , \; Y = \frac{T_X x Y_L}{d} \qquad (2.2)$$

The Depth *Z* is inversely proportional to disparity and proportional to the baseline.

Case Two: Stereo Vision in Non parallel Cameras

Ayache and Lustman [27] addressed that the 3-D reconstruction process of nonparallel stereo systems requires a more general approach, since closed form solutions may not exist for many cases. The lines joining the center of projection and the image point in each of the stereo images are projected backwards into space. Then the point in space that minimizes the sum of its distance from each of the back-projected lines is chosen as the estimated 3-D position of the matched point.

In practical where the noise exist the triangulation problem become more complex and many methods found in literature review proposed to approximate the parameters which make the two projection line from the both cameras to meet.

Beardsley [28] proposed to choose the midpoint of the common perpendicular to the two rays. Divide the common perpendicular in proportion to the distance from the two camera centers. This method depend on many approximation and sometimes the angle will not be precisely equal in two camera, So this make angular error and make this method not optimal.

Another method proposed by Beardsley [29] based on Quasi Euclidean reconstruction and by making an approximation to the correct Euclidean frame and then make the midpoint for this frame. This method better than the previous method but the disadvantage is that approximate calibration of the camera is needed.

The new algorithm based on epipolar correspondence and fundamental matrix proposed by Hartley and Sturm [30] this algorithm is simple and no iterative. The concept here based on minimize the chosen cost function, this algorithm does not need camera calibration. This method called polynomial method since it requires solution of sex order equation. The method model the noise as Gaussian noise so it is consider optimal under this assumption.

Hartley and Chang [31] proposed most common triangulation algorithm called liner algorithm. This method based on find the solution for four linear equations derived from the two views. The method need to find the best solution based on Linear Eigen method which consider simple and speed algorithm. There is another method linear method based on Linear Least Square solution will discuss later in chapter three.

2.6 Interpolation

Finally, the representation of the depth an application is needed. Many applications are found today using this technique for example: Surveying or land surveying [32] is the technique and science of accurately determining the terrestrial or three-dimensional position of points and the distances and angles between them.

Navigation, Navigation is the process of monitoring and controlling the movement of a craft or vehicle from one place to another [33].

The Pick and Place for robotics: Pick and place robots take a product from one spot in the manufacturing process and drop it into another location. A good example is a robot picking items off a conveyor belt and placing them in packaging boxes.

In pick and place application there are 2D, 2.5D or 3D methods of robot vision where in 2D vision systems there is a measure of x, y positions only and 2.5D measure x, y positions and rotation of products. In case the distance between camera and work-pieces varies arbitrary, 2D vision system is inadequate. So there are strong demands from manufacturers for a robust and accurate 3D robot guidance system in handling applications. The different vision technique is shown in Figure 2.2.

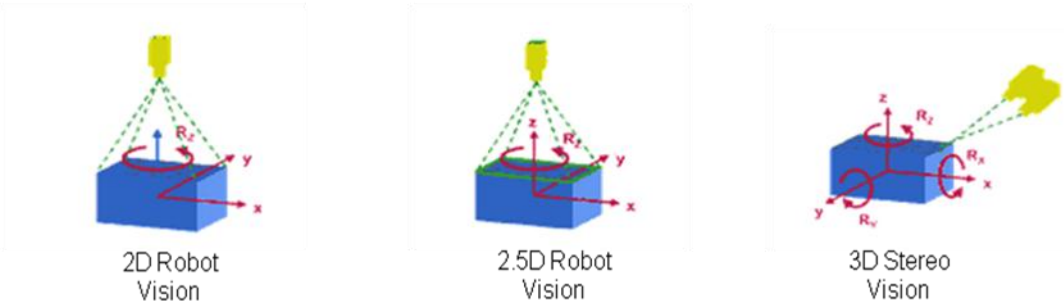


Figure 2. 2 Method of robot vision (Source: ISRA Vision [34])

Many techniques used to find the 3D vision for pick and place robot, for example Wes [35] proposed using of active triangulation to make 3D vision for robotics. This system has 2D gray-scale camera and laser vision sensor. Firstly it calculates the rough position of object for picking by using a camera attached on ceiling. According to the rough position, the robot approaches primary grasping position. The accurate pick-up position and orientation of object is computed by laser vision sensor, which projects cross type pattern on surface of object to measure the slant and depth of object. The 3D pose in this method is only calculated on the surface where the laser slit is projected.

Another method proposed by Kazunori [36] is using single camera based bin picking system. This system can measure 3D pose of object by using only one camera. It registers patterns which are captured diverse view point to object in initial stage. 3D pose of object is estimated by comparing position and orientation between pre-registered patterns and captured image. The disadvantage for this system it takes long times to register various patterns in diverse position and orientation.

Method proposed by Parril [37] is to use passive triangulation with features based corresponding algorithm to find 3D vision and provide these information UMI robot arm for pick and place application. The corresponding algorithm here based on finding feature in each images related to straight lines and circular arcs. The author referred as this method just give the initial starting point for the development of other visual and geometrical reasoning competences. The main contribution for this thesis will be on this area where 3D pick and place will depend on finding depth between objects by using passive triangulation.

CHAPTER 3

3. 3D VISIONS PICK AND PLACE ROBOT APPLICATION

3.1 Introduction

Currently, Autonomous Pick and place robotics are being used in several applications within the manufacturing processes. For example, in one application a robot picks a product from one spot (e.g. a conveyor belt) and places it into another location (e.g. packaging boxes). The application complexity will decide on the pick and place system set up and on the vision guidance needed for the picking process; a vision-guidance system that provides the 2D location of the objects might be enough, however with unstructured objects within a structured or unstructured environment, the 3D location is required.

Nowadays, manufactures rely on 3D camera systems for the 3D pick and place, where single camera provide the robot with information about the world coordinate in the three directions X, Y and Z where the X and Y are the coordinates of the objects and the third dimension Z is the actual depth. However, the 3D camera system is limited in terms of its accuracy and still requires extensive calibration steps.

This thesis proposes an autonomous pick and place vision guidance system based a stereo vision setup to provide the robot arm the coordinates of the depth dimension. This chapter will start by introducing the steps and the techniques that are used to build the stereo-vision setup, followed by the experimental results will be discussed and analyzed.

3.2 The Stereo System

As discussed in the literature review (chapter two), the stereo-vision scenarios rely on following mains steps; the image acquisition, the camera modeling, the correspondence analysis and lastly the triangulation. The following sub-sections will discuss the details of these steps.

3.3 Image Acquisition

The first step is setup the imaging devices because the quality of the images acquired affects the final outcome of the stereo system in terms of the predicted depth accuracy. So based on the application design and intent, one can determine the minimum specifications needed in terms of the spatial resolution of the imagers (number of pixels), the lenses, and data format (number of bits).

Figure 3. 1 Imaging Device

In this thesis the imaging device is based on an open architecture Charged Coupled Devices CCDs (product of National Instrument NI) shown in Figure 3.1.

Two identical cameras are used to build the stereo-vision setup; the cameras' specifications are in Table 3.1.

Table 3.1 Imaging Device Specification

Image Sensor	
Sensor	Sony CCD
Resolution	VGA (640 x 480)
Sensor Size	1/3 in. (VGA)
Pixel Size	7.4 x 7.4 µm (VGA)
Bits per pixel	8 bits, 256 gray levels
Lenses	
Focal Length	8.5 mm
Working Distance	Above 200 mm
Field of View FOV	116.1 mm, 33°
Processor Characteristics	
Processor	533 MHz Freescale PowerPC and 720 MHz Texas Instruments DSP
Physical Specifications	
Dimensions	11.765 by 8.85 by 5.06 cm
Weight	525 g

These CCD cameras can capture monochrome images with a spatial resolution of 640 pixels across its width and 480 pixels across its length. In stereo vision, it is assumed that the images are captured simultaneously without any delay between; meaning that the object is assumed stationary per image captured. In presented application, the targeted objects are static so the assumption above is valid [7].

The two cameras are securely mounted on two stable tripods as displayed in Figure 3.2. The two cameras are labeled as the right and the left camera and placed at the same height. Both cameras are also configured in a way to have their IP addresses with the subnet mask to be in the same network so that they can be connected to a single computer via a switch.

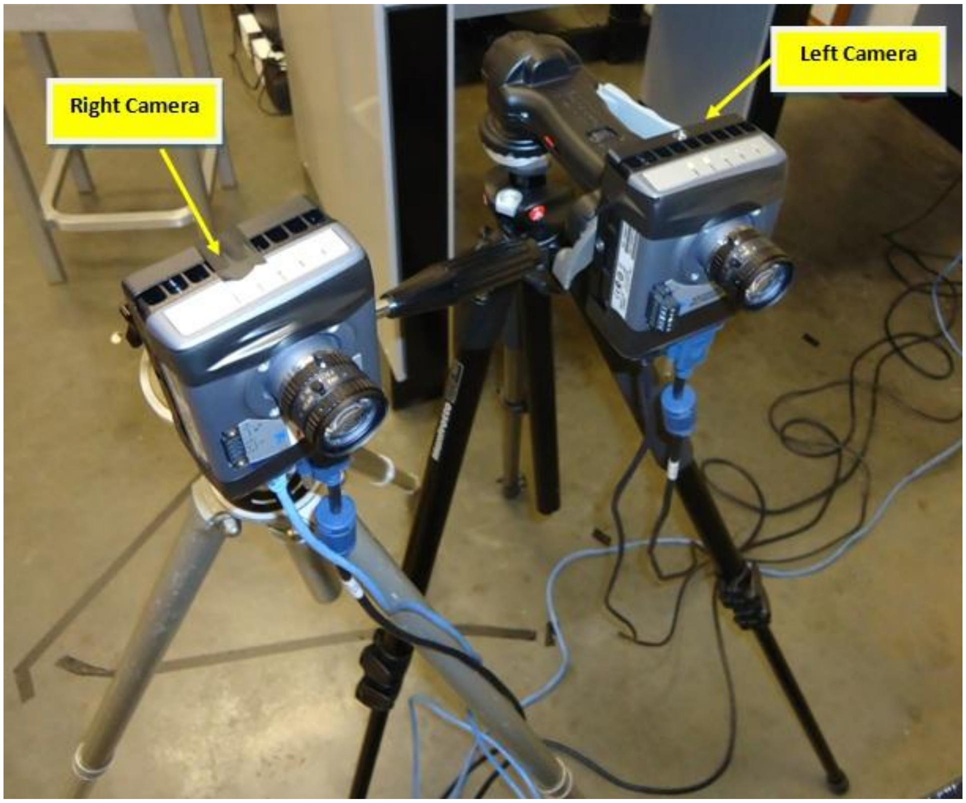


Figure 3. 2 Setup of Left and Right Cameras

The selected setup for the current case study rely on placing two cylinders in front of the two cameras while positioning them on the table, where the two cylinders are clear in both field views from the left and the right cameras' perspectives with the target being to decide on the distance between these two cylinders.

3.4 Camera Modeling

This step is also known as the camera calibration. The objective of the camera modeling step is to mathematically represent the way the cameras map the world coordinate points into the pixel coordinate system. In camera modeling, one can employ a set of mathematical equations with the camera parameters being the main variables. The camera modeling or calibration is an essential part of the stereo-vision process; also it affects the overall system accuracy. This process helps in determining and estimating the intrinsic and the extrinsic camera parameters [10].

The internal camera geometric and optical characteristics are called the intrinsic parameters while the 3-D position and camera frame orientation relative to a certain world coordinate system is called the extrinsic parameters. In multi-camera systems, the extrinsic parameters describe the relationship between the different cameras. [8]

As discussed in chapter two, there are several ways to calibrate the stereo-vision cameras. The 2D plane-based calibration technique is used in this thesis. The first step is to generate and print a 30X30 mm check-board pattern, which is then pasted on a flat panel. After that the actual size of the squares are measured. The difference in the generated square size and the printed one can be due to distortions induced through the printing process. Figure 3.3 shows the calibration pattern, where each square has an actual dimension of 29mm x 29mm.

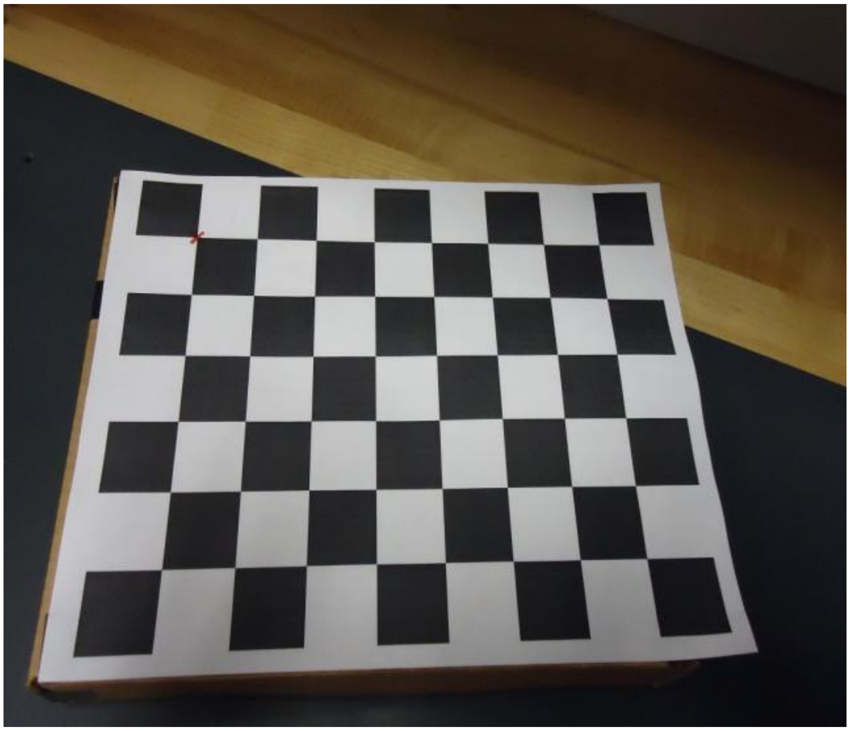


Figure 3. 3 Calibration Pattern

The calibration is done using the Camera Calibration Toolbox for in a Matlab environment, developed by Jean-Yves Bouguet [38]. The calibration can be done through one of two implementations, either moving the imagers while fixing the pattern or moving the pattern while fixing the imagers.

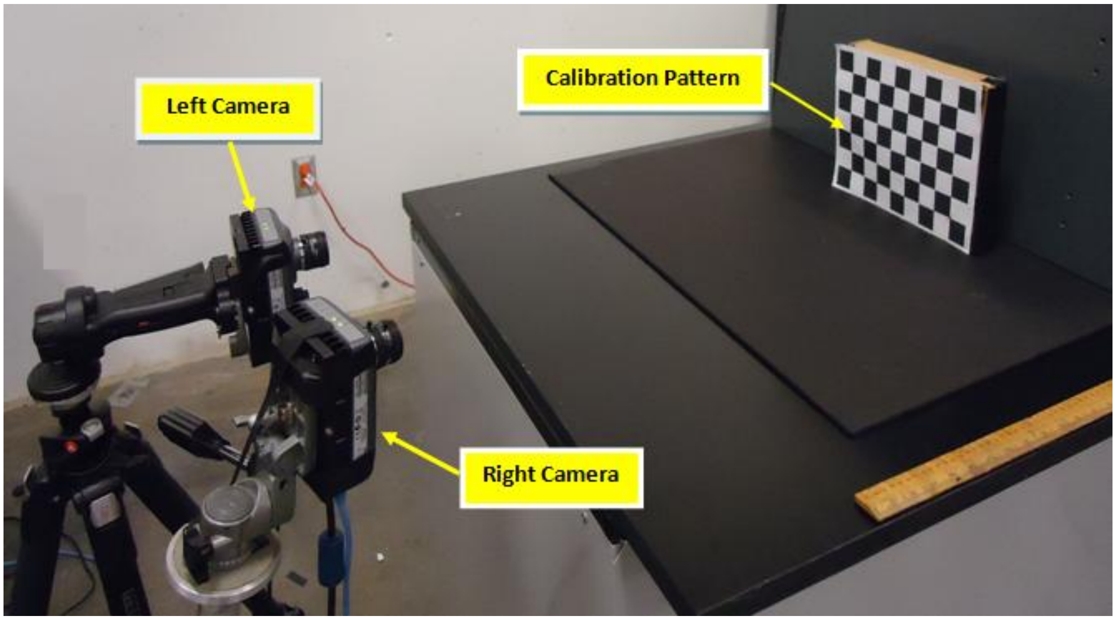


Figure 3. 4 Calibration Pattern Placed on Front of Two Cameras

The second option is used; figure 3.4 displays how the calibration pattern is placed. The higher the number of images that are taken for the different orientations, the more the accurate the calibration is; the images that showed distortion effects could be deleted. In this study, 32 images are captured for the calibration step; 16 images from the right camera and 16 images from the left camera; shown in Figure 3.5 and Figure 3.6.

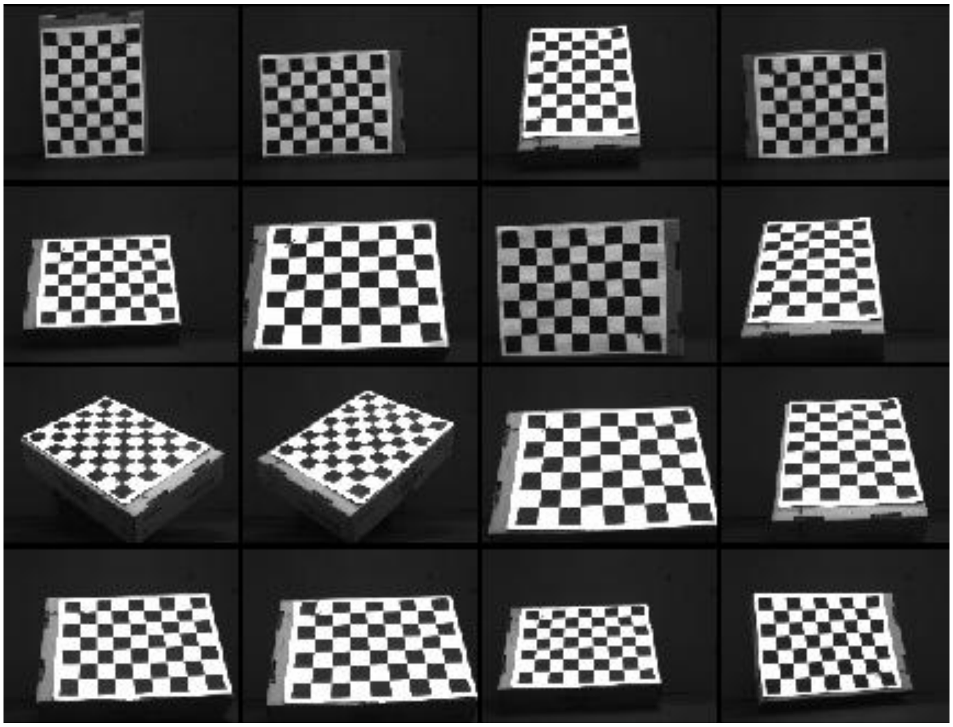


Figure 3. 5 Left Calibration Images

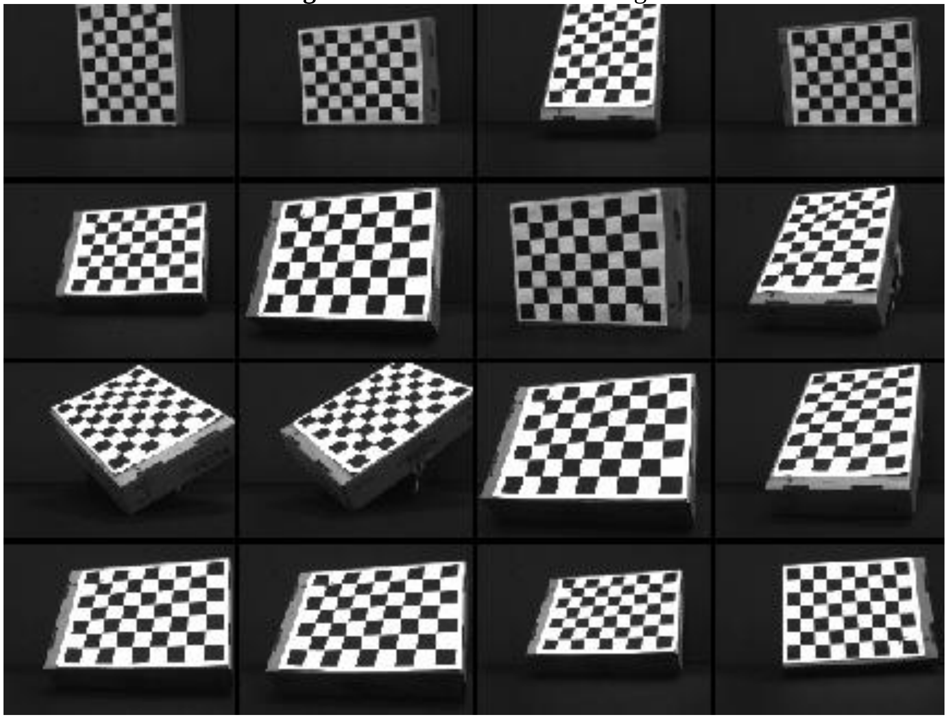


Figure 3. 6 Right Calibration Images

3.4.1 Intrinsic camera parameters

The intrinsic parameters define the internal geometric and optical characteristics of the camera based on following metrics; the Focal length (f_c), which is the effective horizontal and vertical focal length in pixels. The horizontal focal length is $f_c(1)$ and the vertical focal length is $f_c(2)$. The $f_c(2)/f_c(1)$ is called aspect ratio . In the proposed camera system, the aspect ratio is set to about one because the pixels used are squares; if the pixel is not square then the ratio should be different from one.

Principal point (*cc*): This is center of the image. The resolution of the used cameras is; nx =640 pixels and ny =480 pixels. The upper left corner has the coordinate (0,0) and the coordinate of upper right upper is (nx-1,0). The down left coordinate is (0,nx-1) and the right down coordinate (nx-1,ny-1). In some cases where it is difficult to know the exact location of the principal point or when there are not enough images for calibration good estimation for the principle point will be the center of the image (nx-1/2,ny-1/2).

Skew coefficient (*alpha_c*): This coefficient defines the angle between the x and y axes. In this case, this coefficient will be zero because of the square pixels used. The skew coefficient is shown in Figure 3.7: where P_x and P_y are the width and the height of the pixels.

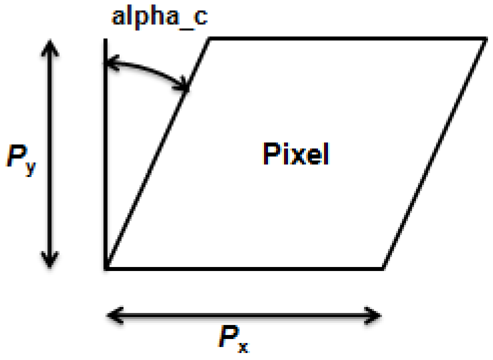


Figure 3.7 Skew Coefficient

Distortions (k_c) describe the image distortion coefficients which include the radial distortion when using a standard glass lenses. Lines that are straight in the 3-D world will sometimes appear to be curved in the image. The effect is more pronounced near the borders of the image. Another distortion type to be considered is the tangential distortion, which is due to imperfect centering of the lens components and other manufacturing defects in compounding the optical lenses.

By knowing these parameters, the projection on the image plane for any point in space is known based on the following transformation; suppose that a point P in world space with the coordinate vector [*Xc; Yc; Zc*] in the camera reference frame, to calculate this point's projection onto the image plane using the intrinsic parameters mainly f_c, *cc*, *alpha_c*, and k_c, one can apply the following calculations;

Let X_n be the normalized (pinhole) image projection, defined in equation (3.1);

$$X_n = \begin{bmatrix} X_c / Z_c \\ Y_c / Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad (3.1)$$

And $r^2 = x^2 + y^2$, after including the lens distortion effect, the new normalized point coordinate x_d can be defined in equation (3.2):

$$x_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = \left(1 + k_c(1)r^2 + k_c(2)r^4 + k_c(5)r^6\right) x_n + d_x \qquad (3.2)$$

Where, d_x is the tangential distortion vector:

$$d_x = \begin{bmatrix} 2k_c(3)xy + k_c(4)(r^2 + 2x^2) \\ k_c(3)(r^2 + 2y^2) + 2k_c(4)xy \end{bmatrix} \qquad (3.3)$$

Jean in [38] estimated that both the radial distortion to be of a 6th order, so once the final distortions are applied, the pixel coordinates namely; x_pixel = [x_p;y_p] of the projection of point P onto the image plane can be described in equation (3.4) as;

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_c(1)(x_d(1) + alpha\_c.x_d(2)) + cc(1) \\ f_c(2)x_d(2) + cc(2) \end{bmatrix} \qquad (3.4)$$

Therefore, the pixel coordinates vector x_pixel and the normalized (distorted) coordinate vector x_d are related to each other through the linear equation (3.5);

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = KK \begin{bmatrix} x_d(1) \\ x_d(2) \\ 1 \end{bmatrix} \qquad (3.5)$$

Where, KK is known as the camera matrix, and can be defined as in (3.6)

$$KK = \begin{bmatrix} f_c(1) & alpha\_c.f_c(1) & cc(1) \\ 0 & f_c(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.6)$$

The list of the output intrinsic parameters from the left camera from the calibration is summarized below;

Focal Length: *fc_left* = [1126.16240 1129.79384] ± [8.30235 8.36960]

Principal point: *cc_left* = [298.95439 260.18405] ± [15.96042 13.74085]

Skew: *alpha_c_left* = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees

Distortion: *kc_left* = [-0.26468 2.07282 0.00890 0.00173 0.00000] ± [0.09587 1.81005 0.00285 0.00331 0.00000]

And the list of the output intrinsic parameters from the right camera is shown below:

Focal Length: *fc_right* = [1121.12572 1125.23248] ± [7.91944 8.71945]

Principal point: *cc_right* = [328.41312 266.00955] ± [18.47137 14.66643]

Skew: *alpha_c_right* = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees

Distortion: *kc_right* = [-0.18696 0.75452 0.00292 0.00774 0.00000] ± [0.08217 1.06827 0.00417 0.00445 0.00000]

3.4.2 Extrinsic parameters

The extrinsic parameters describe the camera position and the orientation with respect to the calibration pattern. They are represented through two matrices; a rotation matrix R and a translational matrix T. These two matrices are used to uniquely identify the transformation between the unknown camera reference frame and the known world reference frame [39]. The extrinsic parameters depend on the camera pose/orientation and unlike the intrinsic parameters it changes once the camera pose is changed. So, to define the extrinsic parameters, one can consider the calibration grid #i (attached to the ith calibration image), and concentrate on the camera reference frame attached to that grid as schematically shown in Figure 3.8. Without loss of generality, take i = 1, and let P be a point space of coordinate vector *XX* = [*X;Y;Z*] in the grid reference frame (reference frame shown on the previous figure 3.8) while having *XXc* = [*Xc;Yc;Zc*] be the coordinate vector of the point P in the camera reference frame. Then *XX* and XX_c are related to each other through the following rigid motion equation (3.7):

$$XX_c = R_{c\_1}.XX + T_{c\_1} \qquad (3.7)$$

In particular, the translation vector T_{c_1} is the coordinate vector of the origin of the grid pattern (O) in the camera reference frame, and the third column of the matrix R_{c_1} is the surface normal vector of the plane containing the planar grid in the camera reference frame.

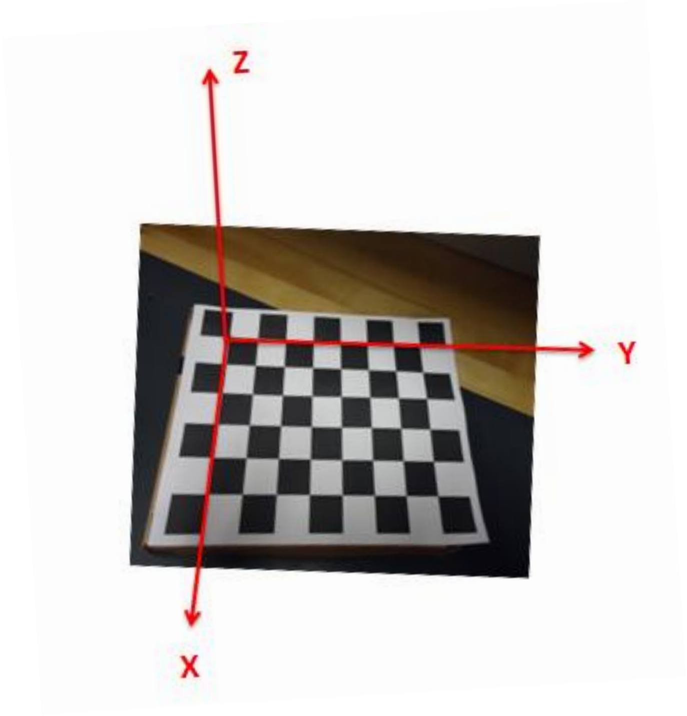


Figure 3. 8 Grid Calibration with Reference Coordinates

The same relation holds for the remaining extrinsic parameters (R_{c_2}, T_{c_2}), (R_{c_3}, T_{c_3}), etc. So, once the coordinates of a point is expressed in the camera reference frame, it may be projected on the image plane using the intrinsic camera parameters. The right and left cameras' extrinsic parameters are listed below; with respect to the spatial configuration of the two cameras and their calibration planes, it is shown in Figure 3.9.

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

Rotation vector: om = [0.08355 0.29692 -0.00108] ± [0.01618 0.02051 0.00275]
Translation vector: T = [-205.77721 10.81310 42.23043] ± [1.07962 0.71668 4.86009]

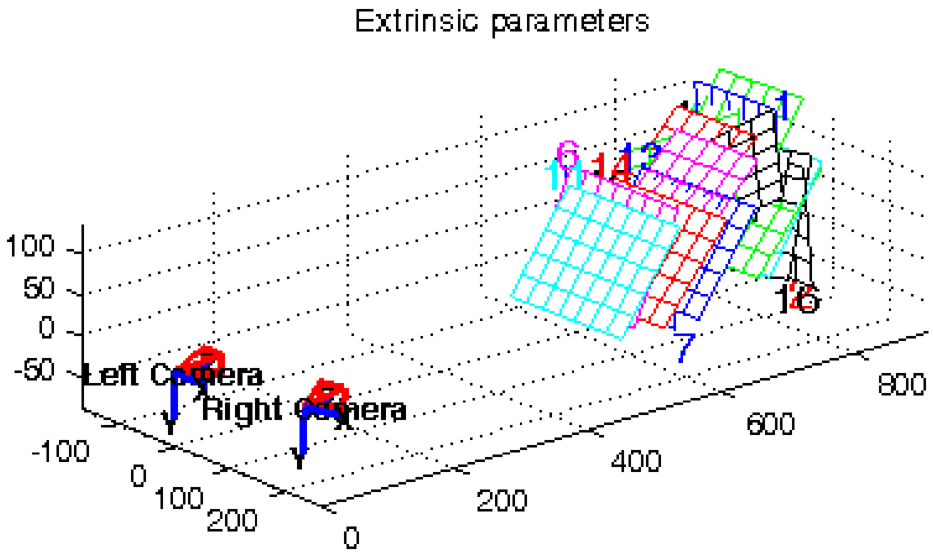


Figure 3. 9 Spatial Configuration of the Two Cameras and the Calibration Planes

The calibration resulted in a pixel error around 0.21298 pixels in the x-direction and around 0.44828 pixels in the y-direction. The calibration error in mm units is in Table 3.2.

Table 3.2 Error in Calibration in mm Units

	Image Size in mm units	**Image size in pixels**	**Error in pixels**	**Error in mm units**
X-direction (Horizontal)	150 mm	640 pixels	0.21298 pixels	0.05 mm
Y-direction (Vertical)	110 mm	480 pixels	0.44828 pixels	0.1 mm

3.5 Correspondence Analysis

In the previous section, the calibration of each camera is done following the stereo-vision system calibration, which defines the intrinsic and the extrinsic parameters. These parameters are needed to implement the triangulation code, but before this the corresponding points between the left and right images should be identified. This is done to find a set of points in one image which can be identified as the same points in another image. There are different techniques to achieve this as mentioned in Chapter 2.

3.5 SSD Algorithm

The image matching techniques are implemented here are based on the Sum of Square Difference or SSD algorithm. In this algorithm, a window with a size of mxm is chosen around each pixel, and then the algorithm finds the same size window in the second image that closely resembles the one from the first image. The comparison of the blocks is done using the following equation, equation (3.8);

$$SSD = \sum_{xy} [I_T(x,y) - I(x,y)]^2 \qquad (3.8)$$

Where *I(x,y)* is pixel intensity in the first image (left image), and I_T *(x,y)* is from the second image (right image). The block from the second window that has the smallest value of SSD as compared with others, is chosen as the window that closely resembles the block from the first image. The center of this block is then chosen as the corresponding point of the pixel from the first image.

The SSD algorithm based on the intensity of each pixel, where image intensity value is compared from both cameras to know which pixel is a candidate to be a corresponding pixel for the left camera pixels.

Two images are shown in Figure 3.10 and 3.11 captured using the left and the right cameras, to test the SSD algorithm while using a window size of 3x3.

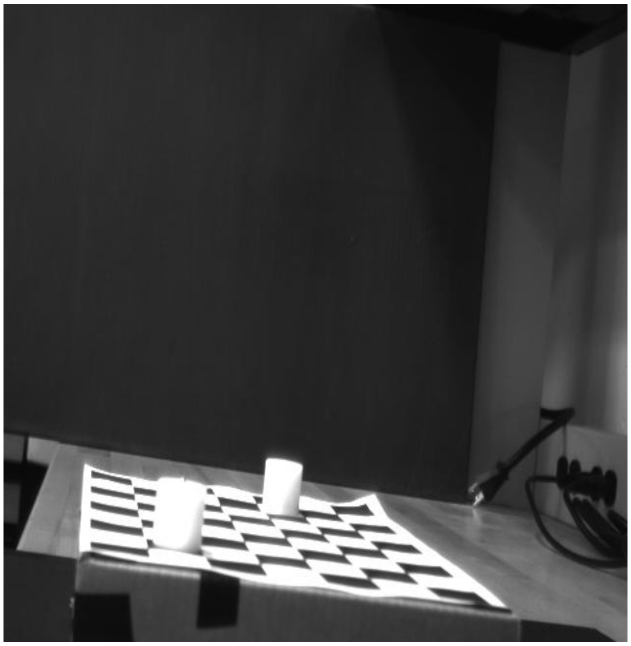


Figure 3. 10 Left Image to Test SSD Algorithm

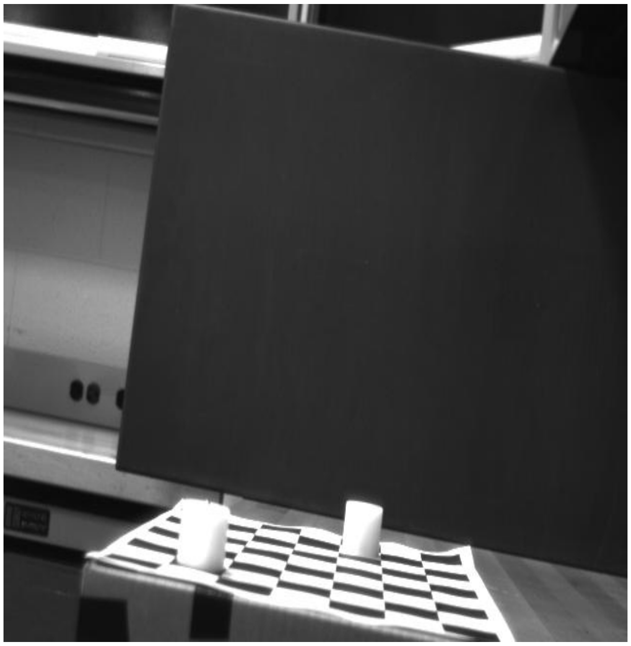


Figure 3. 11 Right Image to Test SSD Algorithm

Six points are selected in the left image as shown in the Figure 3.12, where the coordinates in the left image are known then the algorithm is implemented to find the candidate corresponding point from the right image. The candidate corresponding points are then compared with the actual corresponding pixels from the right image to test the algorithm accuracy.

The comparison is done using a manual estimation for these corresponding points, with the results included in Table 3.3 below.

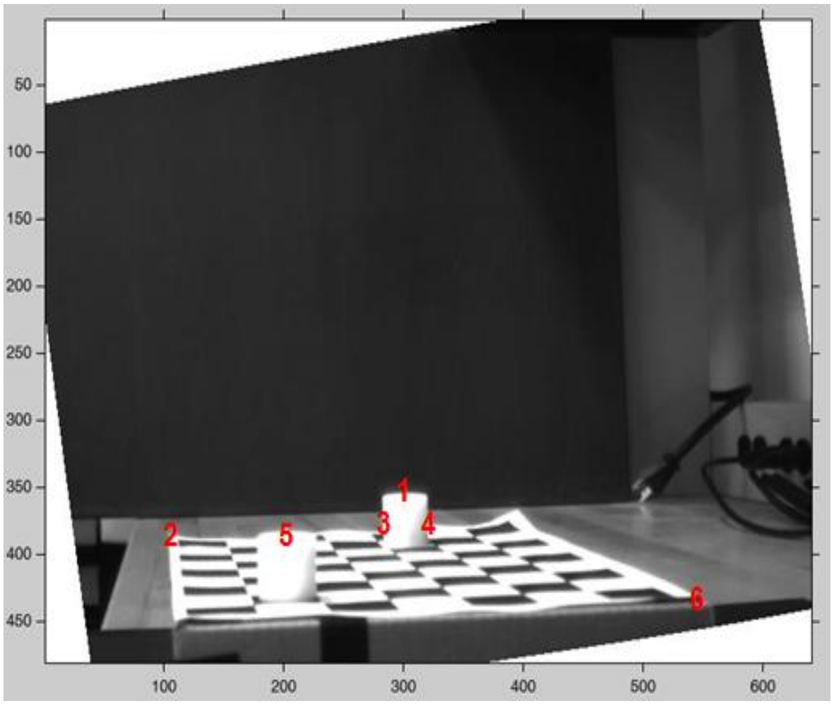


Figure 3. 12 Selection Points in left Image

The algorithm shows some good results, while identifying three main challenges, which are; Firstly, the search for all the correspondences pixels is not ideal because not all the same pixels exist in both images [40], this can lead to a large number of pixel correspondences that are not desired and are seen as errors as in point tagged as number 6. Secondly, many points in the right image are found to be candidates to be corresponding points because the algorithm will search all the pixels in the right image, this will give false matching if the algorithm finds the minimum SSD in wrong place as the case with point 5.

Table 3.3 SSD Test Results

Points Tag	Points coordinate in first image		Manually estimated position second image		Calculated coordinate using SSD algorithm		comments
	X	Y	X	Y	X	y	
1	357	300	357	**408**	357	**411**	Good Correlation
2	387	108	387	**220**	387	**223**	Good Correlation
3	375	285	375	**393**	375	**391**	Good Correlation
4	375	315	375	**423**	375	**417**	Good Correlation
5	382	203	382	**246**	382	**368**	False Correlation
6	430	542	Does not exist in right image		430	**265**	False Correlation

Thirdly, the SSD searches all the pixels in the right image 640x480 pixels, leading to long processing time for each corresponding pixel, which is found to be around 2.5509 seconds for a machine that runs a core i7 CPU 2.67 GHZ, RAM 4 GB. The figures below show two outputs from the algorithm; one considered as the correct matching as in Figure 3.13 while the other is a false matching case shown in Figure 3.14. Where the x- axis is the y coordinate of right image and y-axis is the minimum SSD. To minimize the high number of incorrect matching, a modified SSD algorithm is developed and further discussed in the section below.

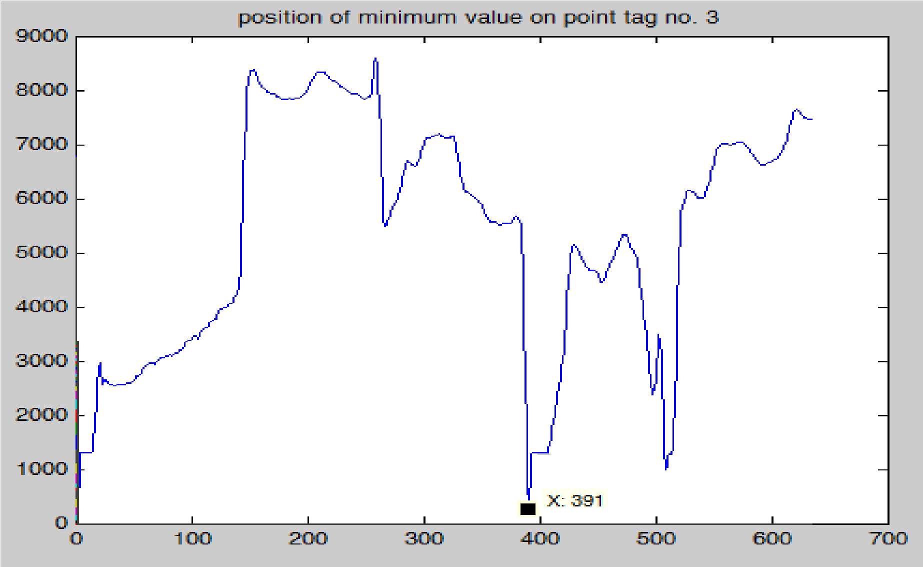


Figure 3.13 Good Correlation Related to Point Tag No.3

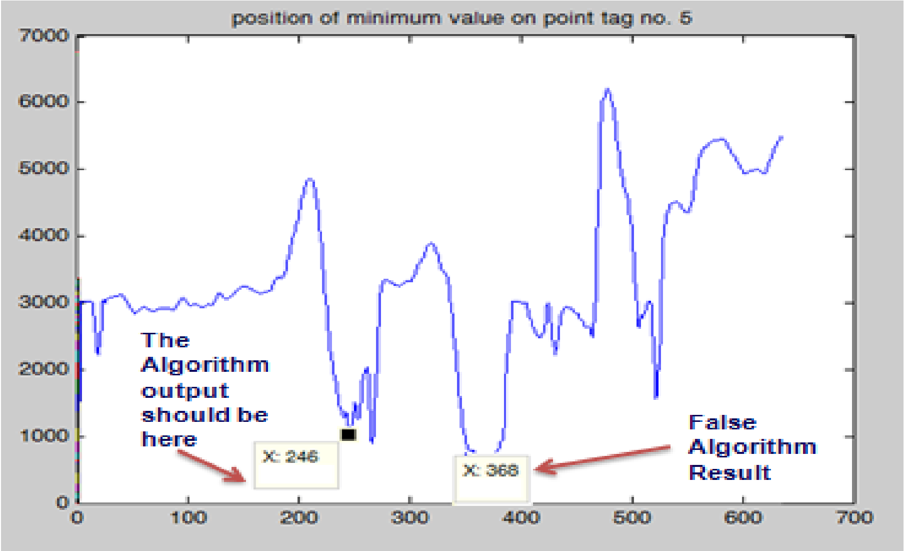


Figure 3.14 False Correlation Related to Point Tag No.5

3.6 Modified SSD Algorithm

The original SSD algorithm does not select a set of points; instead it searches for all the pixels in the left image (640x480) then it searches again for each pixel in the right image. This make the algorithm time consuming around 2.5509 seconds per pixel. In this research, several modifications are implemented to the original SSD algorithm to enhance the search routine and to reduce the time needed.

3.7.1 Automatic Field of View

Searching 640x480 pixels in each image is not practical in addition in most cases not all image pixels contain useful information about the objects of interest. For example, referring to Figure 3.12, the only useful information is contained in the regions where the two cylinders exist, so the first step is to establish a region of interest within each image so that the program can automatically restrict the search to these ROIs from the first image and the second images. In Figure 3.15, the new ROI is highlighted with a red border; using this approach the SSD can finish the search routine for the entire image in 85 seconds only.

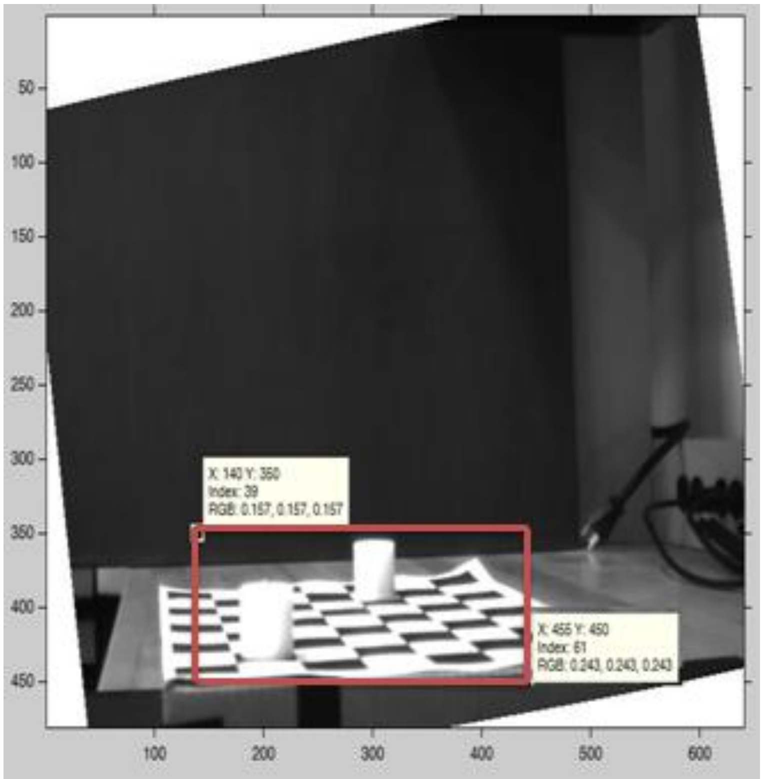


Figure 3.15 Dynamic ROI

3.7.2 SSD algorithm based on center of gravity

The Automatic ROI improved the search routine, however this study will also discuss a new method based on a new criteria for the search that is the feature based, where the search targets objects not individual pixels. This will also be useful in identifying the pick location for each found object that is the objects' center of gravity or Centroids.

The new search algorithm can be done using following steps;

- Step 1: Read or inputting the image, the image is a gray-scale image with 256 levels as mentioned in Table 3.1.
- Step 2: The image is converted into a binary scale by choosing a suitable threshold value in which each pixel is restricted to a value of either 0 or 1. This process aims at reducing the data content while classifying the pixels to be either value-added (or 1) or non-value added or background pixels (that is 0). Each pixel in the background is displayed as black, while each pixel in within the object is displayed as white as shown in Figure 3.16. In this step a prior knowledge of the object shape is helpful.
- Step 3: The noise in the image is removed, where all the objects that contain less than 60 pixels can be considered noise and then removed. Also, other filtering routines can be applied such the salt and pepper noise filters.
- Step 4: Filling the objects using a combination of two morphological operations; the erosion and the dilation. In erosion, every object pixel that is touching a background pixel is changed into a background pixel. In dilation, every background pixel that is touching an object pixel is changed into an object pixel. Erosion makes the objects smaller, and can break a single object into multiple objects. Dilation makes the objects larger, and can merge multiple objects into one. To fill the region inside the object, the closing technique is applied using dilation followed by erosion; the result will prevent any object to contain a hole and it helps in smoothing the objects' boundaries.

- Step 5: The boundary of the object is detected and the Centroids is drown as shown in Figure 3.17.

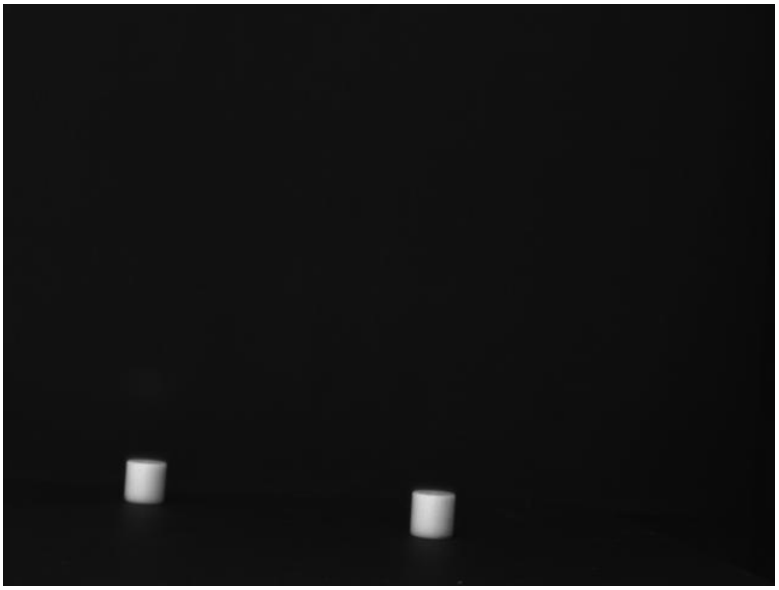


Figure 3.16 Binary Image

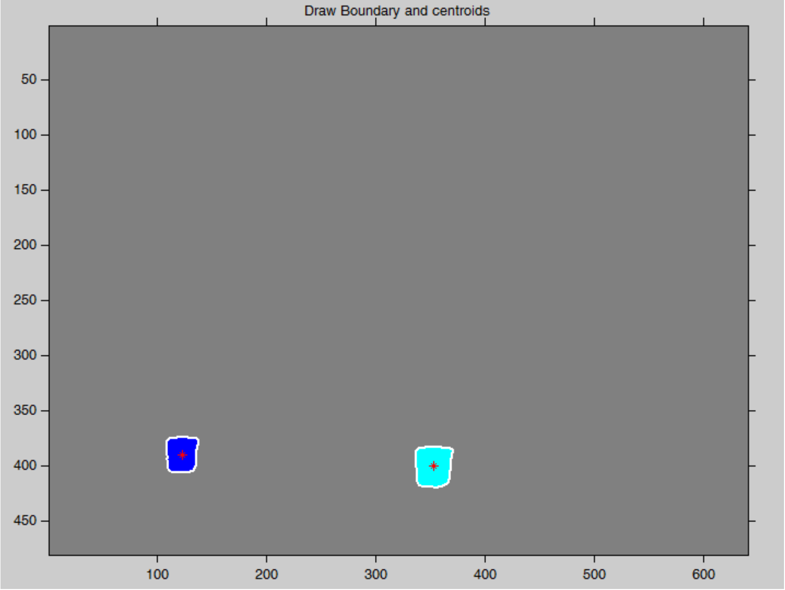


Figure 3. 17 Centroids of the Two Objects

3.7.3 Rectification

In the previous section, the SSD algorithm is modified to make the selection of the points automatically by detection the Centroids in the first image then the algorithm run to find the corresponding points in the right image but the problem still exist in this algorithm in terms of the accuracy and the slowness so the rectification principle is used in this thesis.

The rectification [41] determines a transformation of each image plane such that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes (usually the horizontal one). The relation of the epipolar plane and image plane is shown in Figure 3.18.

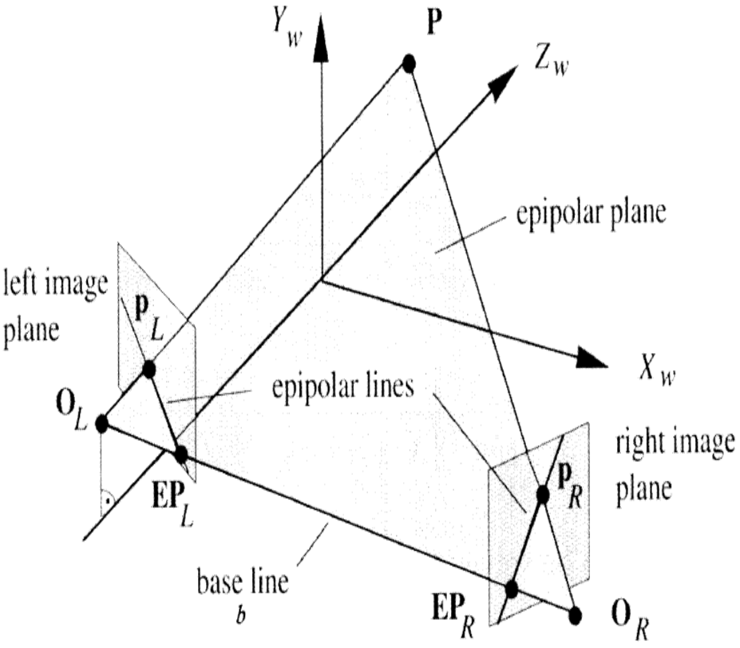


Figure 3.18 Epipolar Geometry [Source: Faugeras [10]]

Where, P is the point in space in 3D, Epipolar plane: is a plane that includes point P and the two optical centers O_L and O_R of both cameras, Epipolar line: The intersection between an epipolar plane and an image plane and baseline b: is the distance between two optical centers.

The rectified images can be thought of as acquired by a new stereo rig, obtained by rotating the original cameras. The important advantage of rectification is that computing stereo correspondences [42] is made simpler, because search is only need to done along the horizontal lines of the rectified images this make corresponding algorithm faster and more accurate and reduce the searching from 2D to 1D.

The calibration step should be done before the rectification because the internal parameters should be known in this stage. The rectification [41] is define new (Perspective Projection Matrix PPM) by rotating the old ones around their optical centers until focal planes becomes coplanar, thereby containing the baseline. This ensures that epipoles are at infinity; hence, epipolar lines are parallel. To have horizontal epipolar lines, the baseline must be parallel to the new X axis of both cameras. In addition, to have a proper rectification, conjugate points must have the same vertical coordinate.

The Perspective Projection Matrix PPM is found through the equation 3.9 below.

$$PPM = KK.[R\,|\,T] \tag{3.9}$$

Where, R is the rotational matrix, T is the translational matrix which found as extrinsic parameters in calibration step.

The camera matrix *KK* contains intrinsic parameters which found in calibration step.

$$KK = \begin{bmatrix} f_c(1) & alpha\_c.f_c(1) & cc(1) \\ 0 & f_c(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.10)$$

The result of rectification step is found in Figure 3.19 and Figure 3.20.

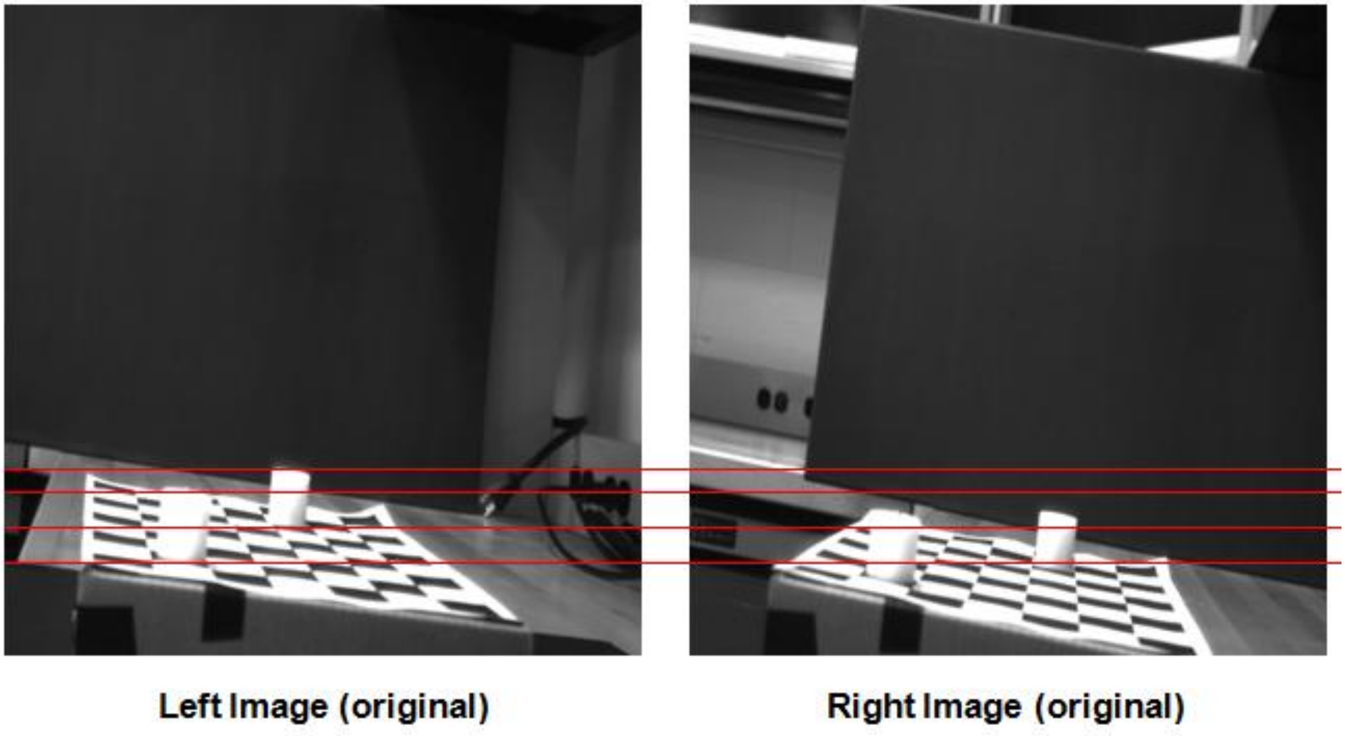


Figure 3.19 Right and Left Images before Rectification Step

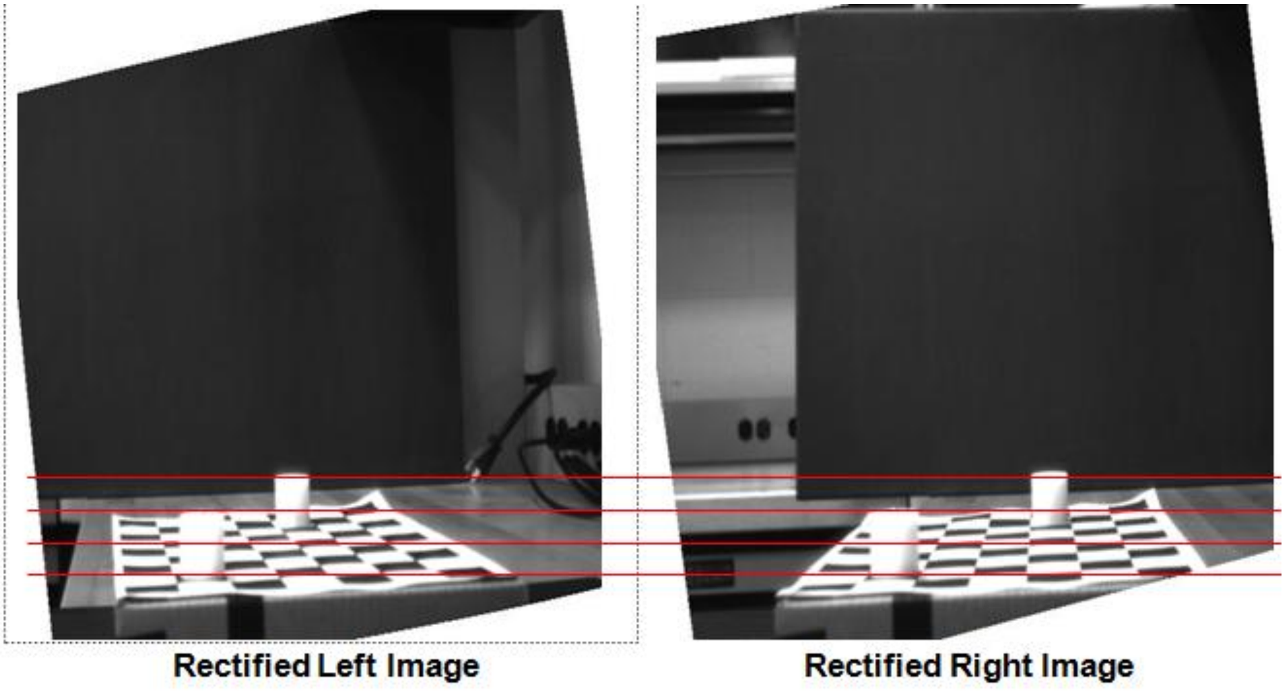


Figure 3.20 Right and Left Images after Rectification Step

After the rectification , the length of the process to find one corresponding point is minimized from 2.5509 seconds to 0.1740 seconds because now to find corresponding point the algorithm just search though horizontal line (1 x 637 pixels) instead of the whole pixels as before rectification (480 x 640 pixels). Also, the accuracy of searching is increase because the candidate corresponding points is decreased.

The modified SSD algorithm proposed in this thesis is the combination of the three previous steps to carry out the stereo vision in pick and place application which is the scope of this thesis. The case study here is to find the useful corresponding points between in left and right images which will be useful in pick and place application.

The modified SSD is started by enhance the ROI then the rectification is done for right and left images then the Centroids in the left image is calculated using and finally the corresponding points for this Centroids is found through equation 3-8. The output results in Table 3.4 for modified SSD showed good results in term of accuracy and the efficiency.

The error results showed in pixel equal 1.4854 pixels in x-direction and 2.7246 pixels in y-direction. The modified SSD error in length units are found in Table 3.5.

Table 3.4 Results of modified SSD

	Actual Centroids for Two Cylinders	**Centroids using Modified SSD Algorithm**	**The Maximum Error in Pixels**
Left Image	(389.8832,123.7254) (400.7207, 354.0469)	(390,124) (401,354)	0.2793
Right Image	(388.0638, 231.5842) (399.5146, 442.7246)	(390,231) (401,440)	1.4854 in x-direction and 2.7246 in y-direction
Elapsed time	0.969299 seconds.	0.022325 seconds	

Table 3.5 Error in modified SSD in mm Units

	Image Size in mm units	Image size in pixels	Error in pixels	Error in mm units
X-direction (Horizontal)	150 mm	640 pixels	1.4854 pixels	0.1 mm
Y-direction (Vertical)	110 mm	480 pixels	2.7246 pixels	0.62 mm

The modified SSD algorithm is correlation based method where it attempts to establish a correspondence by matching image intensities. It showed good results but because it based on matching intensity it fails when viewpoints are very different due to change in illumination direction. Another algorithm is proposed in this thesis to find corresponding points based on match features between two images.

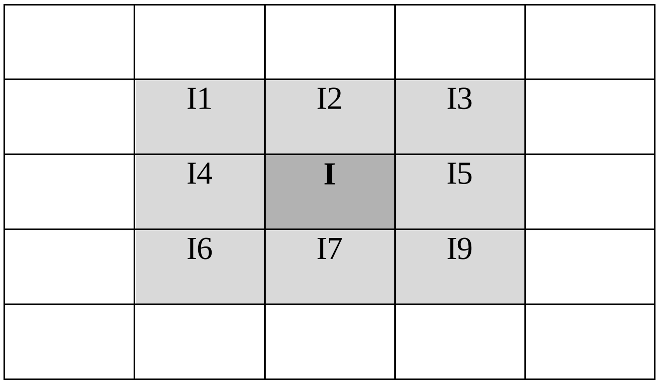
3.8 Corresponding points by matching features

Finding corresponding point by matching a sparse set of image features between the two images is solved the disadvantage found in modified SSD algorithm. The feature here could be lines, area or any other feature like edges.

The proposed algorithm is start by detecting how many important objects in each rectified image by using concept of pixel connectivity where pixel (x, y) is neighbors to every pixel that touches one of their edges or corners. This pixel is connected horizontally, vertically, and diagonally this called 8-connected pixels. In terms of pixel coordinates, every pixel that has the coordinates *(x±1, y)* or *(x, y±1)* or *(x±1, y±1)* is connected to the pixel at *(x, y).*

For example a pixel, Q, is a 8-neighbor of a given pixel, I, if Q and I share an edge. The 8-neighbors of pixel I (namely pixels I1, I2, I3, …. I8) are shown in Figure 3.21. Where a set of white pixel I, is a 8-connected component if for every pair of

pixels Ii and Ij in I, there exists a sequence of pixels Ii,..Ij such that: a) all pixels in the sequence are in the set I ,are white, and b) every 2 pixels that are adjacent in the sequence are 8-neighbors and using 8-connected pattern, two objects is detect on left image as shown in Figure 3.22.



	I1	I2	I3	
	I4	**I**	I5	
	I6	I7	I9	

Figure 3.21 8-Connected Patterns

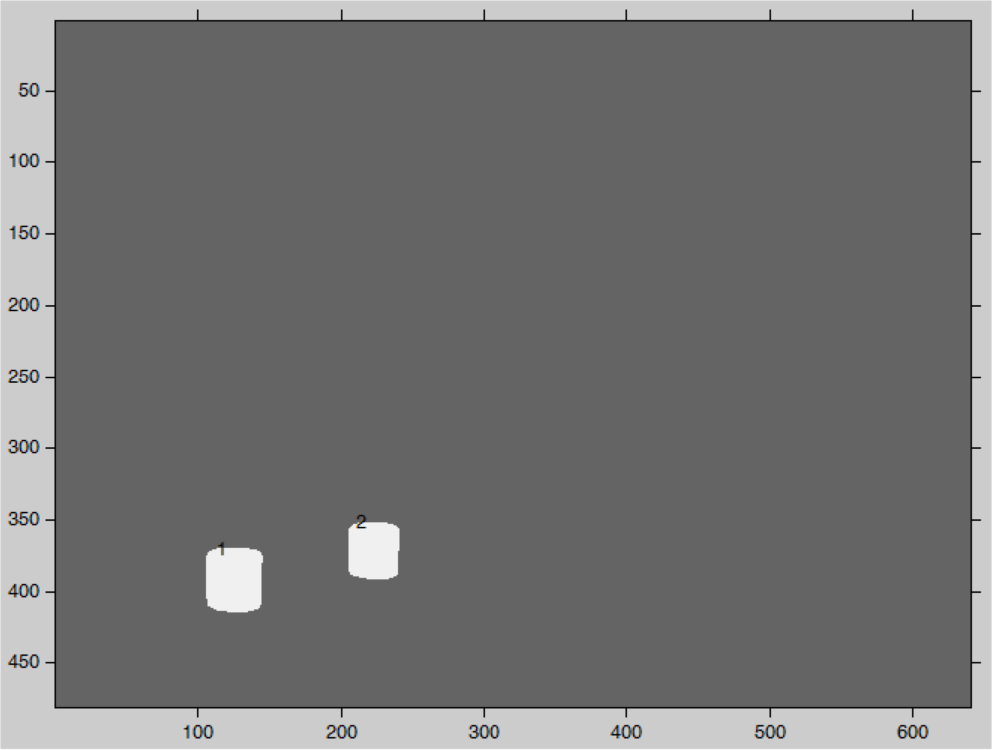


Figure 3.22 Two Objects in Left Image

Then there is need to find matching criteria to relate objects in left image with the corresponding object in right image. The proposed matching criteria will based on area calculation for each object in the left image then compare them with the objects area in the right image, the minimum difference will decide which objects in right image related to which objects in the left image. For example, to find object 2 in left image is related to which objects in right image, the criteria 3-11 is used to find first the error in area (δ).

$$\min(\delta) = \min(\begin{cases} A_{Obj2} - A_{N1} \\ A_{Obj2} - A_{N2} \\ \quad . \\ \quad . \\ A_{Obj2} - A_{Nx} \end{cases}) \qquad (3.11)$$

Where, x: all objects in the right image. And Object $_{Nx}$ = Object $_{2}$, if it has the related $\min(\delta)$ error. Table 3.6 shows the area calculation for each object in left and right image and table 3.7 shows the difference area calculation.

Table 3.6 Area of each Object in Right and Left Image in pixel^2 unit

	Left Image	**Right Image**

Areas of object	Object1=1677
	Object2=1319
		Object1=1830
		Object2=1399

Table 3.7 Area Error Difference (δ) between each Object in Left and Right Image in pixel^2 unit

	Object1 in Right Image	Object2 in Right Image
Object1 in Left Image	**135**	278
Object2 in Left Image	511	**80**

Figure 3.23 and 3.24 show the left and right images and every object detected in each image, while Figure 3.25 shows the related objects in each image.

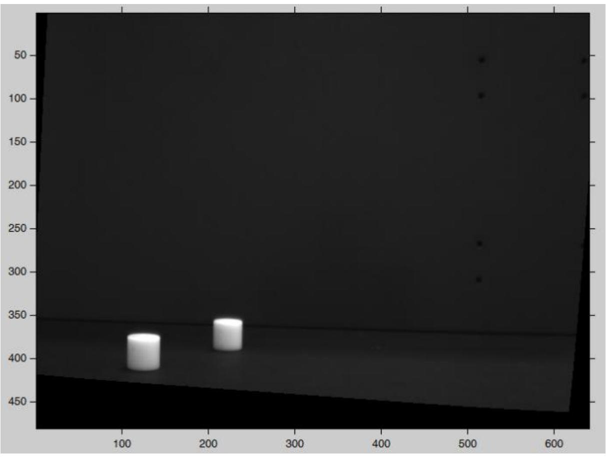
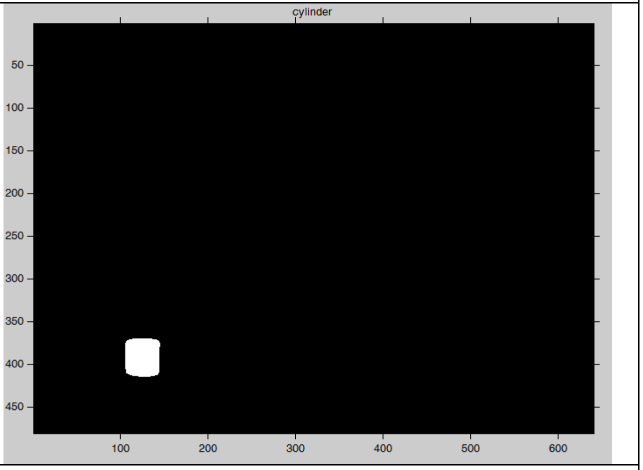
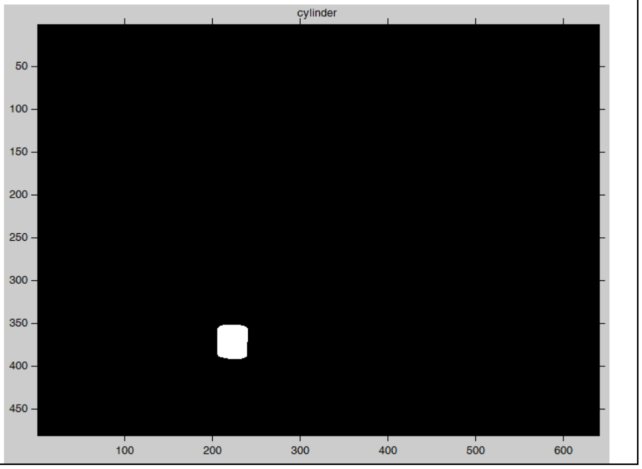
Left Image	 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600
Object1	 cylinder 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600
Object2	 cylinder 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600

Figure 3. 23 Object1 and Object2 in Left Image

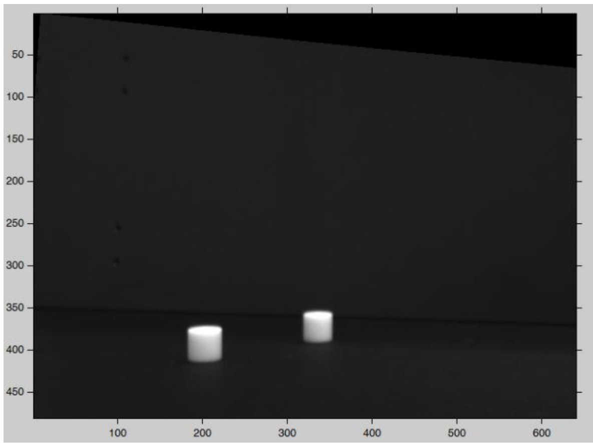
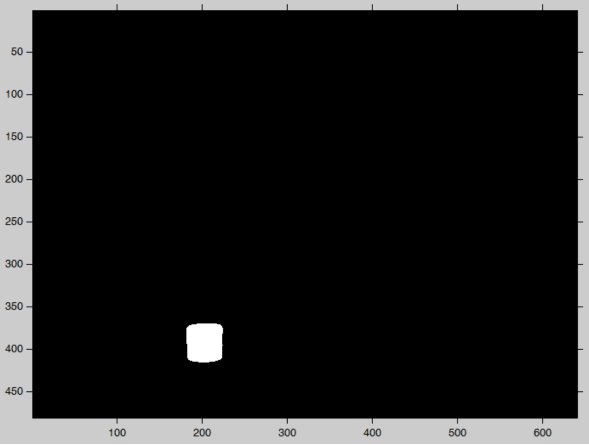
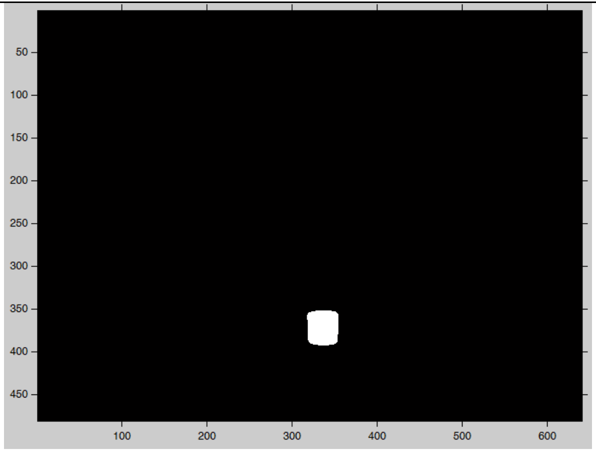
Right Image	 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600
Object1	 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600
Object2	 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600

Figure 3. 24 Object1 and Object2 in Right Image

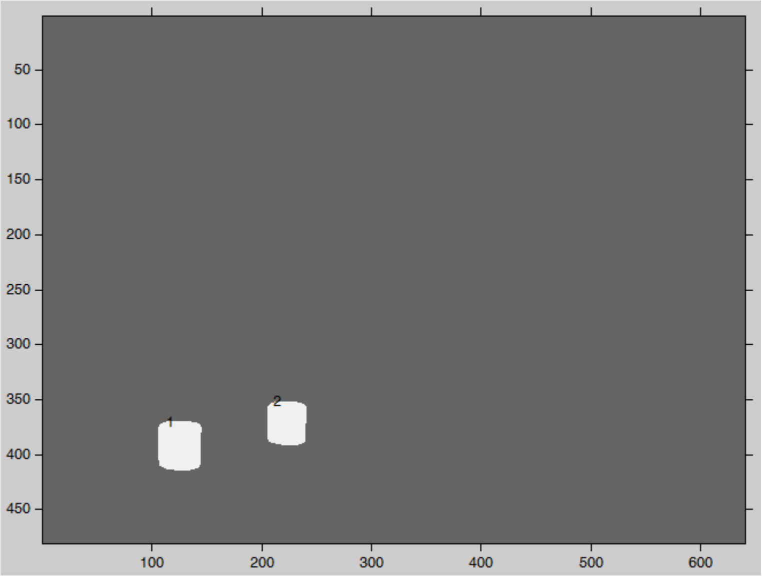
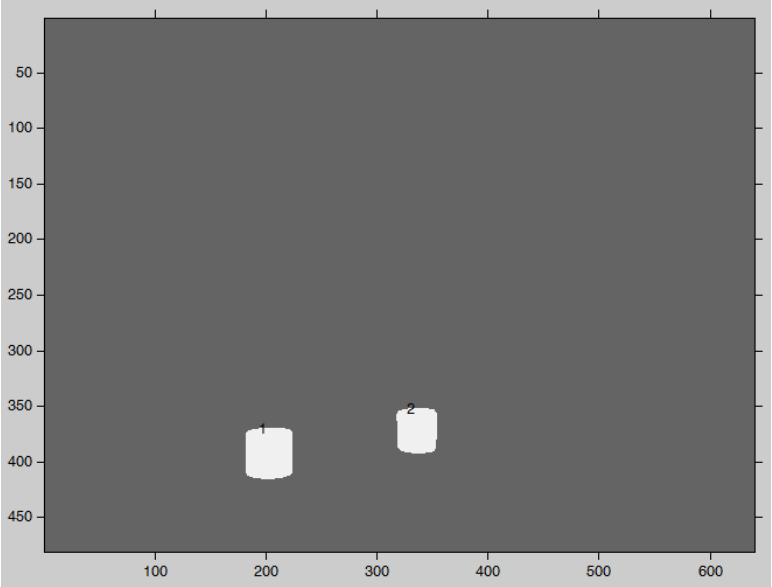
Left Image	
Right Image	

Figure 3. 25 Related Object in Right and Left Image

The next step after the objects is related in both image is to calculate the Centroids for each object in left and right image and now the corresponding points between the two objects will be the Centroids as shown in Table 3.8.

Table 3.8 The Centroids of each Object in Right and Left Image

	Left Image	**Right Image**
Centroids Calculation	Centroids Object1=(126,392) Centroids Object2= (223,371)	Centroids Object1=(203,337) Centroids Object2= (391,372)
Elapsed time	1.476514 seconds.	

The two Centroids are consider as the two corresponding points needed for making triangulation as will discuss in the section below. This algorithm differ from the modified SSD algorithm where the modified SSD algorithm start by detecting the Centroids in the left image then build window 3X3 around this Centroids and then match this block with every same size block in right image by seeking the minimum intensity difference between each block because the modified SSD algorithm does not know which Centroids in left image related to which Centroids in the right image where these things is known in this proposed matching algorithm.

To test the limitation for this new algorithm the number of objects is increase to three objects as shown in Figure 3.26, and the area calculation for the three objects is shown in Table 3-9.

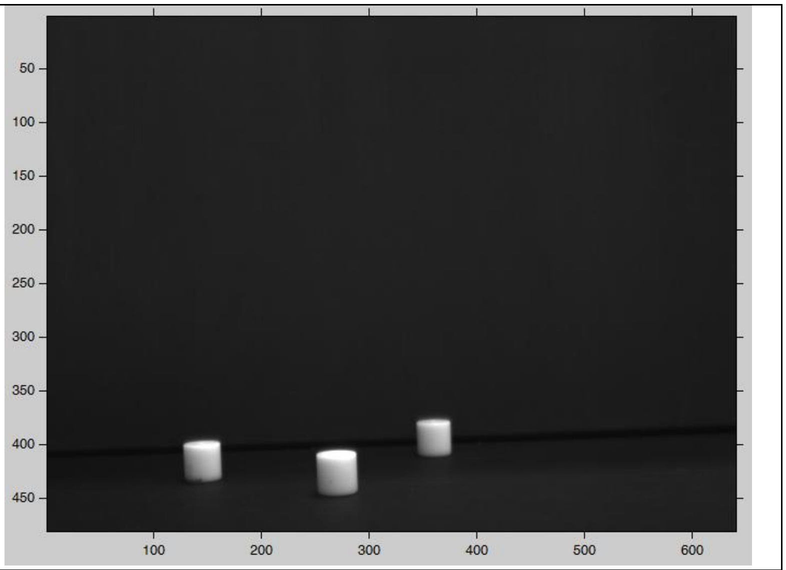
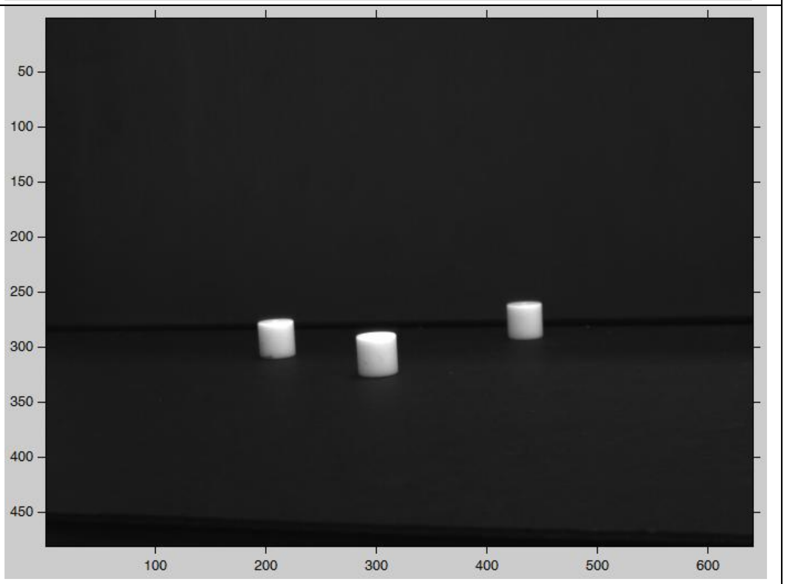
Left Image	 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600
Right Image	 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600

Figure 3. 26 Left and Right Image for Three Objects

Table 3.9 Area of each Object in Right and Left Image in pixel2 unit

	Left Image	**Right Image**

Areas of object	Object1=1078
	Object2=1341
	Object3=898
		Object1=1023
		Object2=1334
		Object3=948

The area error difference (δ) between each objects are found in Table 3.9.

Table 3.10 Area Error Difference (δ) between each Object in Left and Right Image in pixel^2 unit

	Object1 in Right Image	Object2 in Right Image	Object3 in Right Image
Object1 in Left Image	**55**	256	130
Object2 in Left Image	318	**7**	393
Object3 in Left Image	125	436	**50**

The results shows in Table 3.10 that objects1 in left image related to object1 in right image and object2 in left image related to object2 in right image and also object3 in left image related to object3 in right image. So the algorithm works well for this test and the results shown below in Figure 3.27.

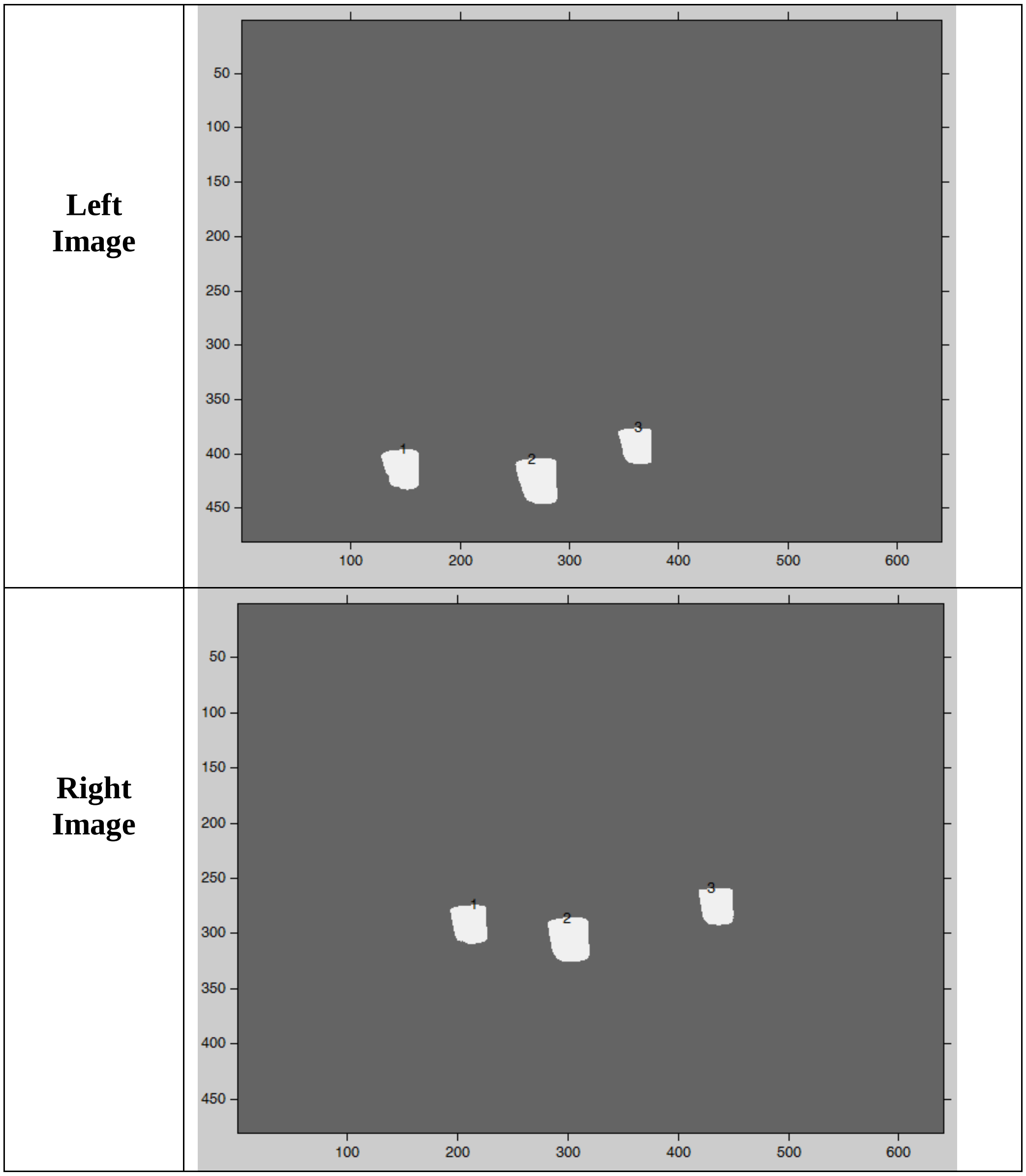


Figure 3. 27 Related Objects in Left and Right Image for Three Objects

Another test is done by increasing the number of objects to four objects. The left and right image is shown below in Figure 3.28.

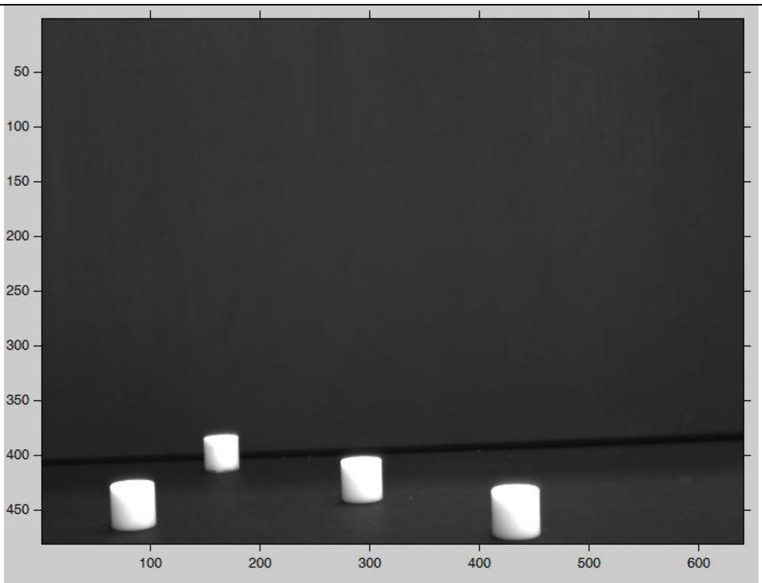
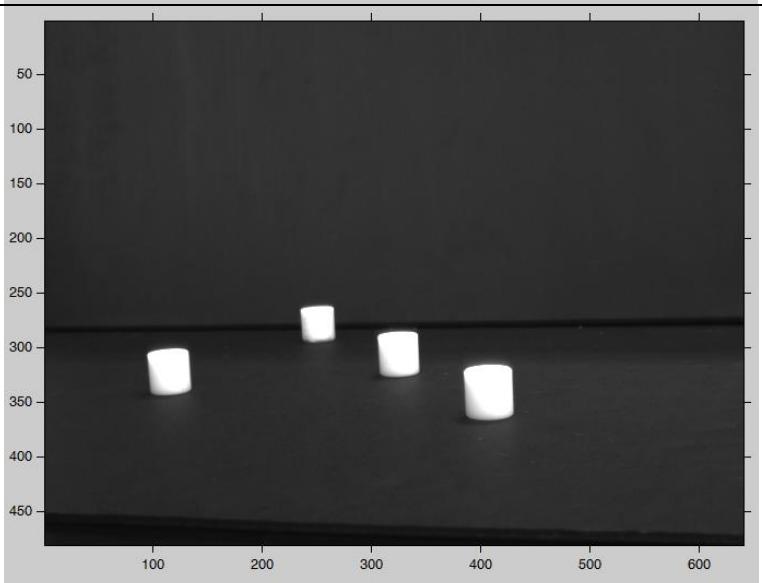
Left Image	 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600
Right Image	 50 100 150 200 250 300 350 400 450 100 200 300 400 500 600

Figure 3. 28 Left and Right Image for Four Objects

The area calculation for the four objects is shown in Table 3.11.

Table 3.11 Area of each Object in Right and Left Image pixel^2 unit

	Left Image	**Right Image**

Areas of object	Object1=1688
	Object2=953
	Object3=1453
	Object4=2004
		Object1=1479
		Object2=906
		Object3=1423
		Object3=2068

The area error difference (δ) between each objects are found in Table 3.12.

Table 3.12 Area Error Difference (δ) between each Object in Left and Right Image in pixel^2 unit

	Object1 in Right Image	Object2 in Right Image	Object3 in Right Image	Object4 in Right Image
Object1 in Left Image	**199**	782	255	380
Object2 in Left Image	536	**47**	480	1115
Object3 in Left Image	36	547	**20**	615
Object4 in Left Image	515	1098	571	**64**

The results shows in Table 3.12 that objects1 in left image related to object1 in right image, object2 in left image related to object2 in right image , object3 in left image related to object3 in right image and also object4 in left image related to object4 in right image. So the algorithm works well for this example and the results shown below in Figure 3.28.

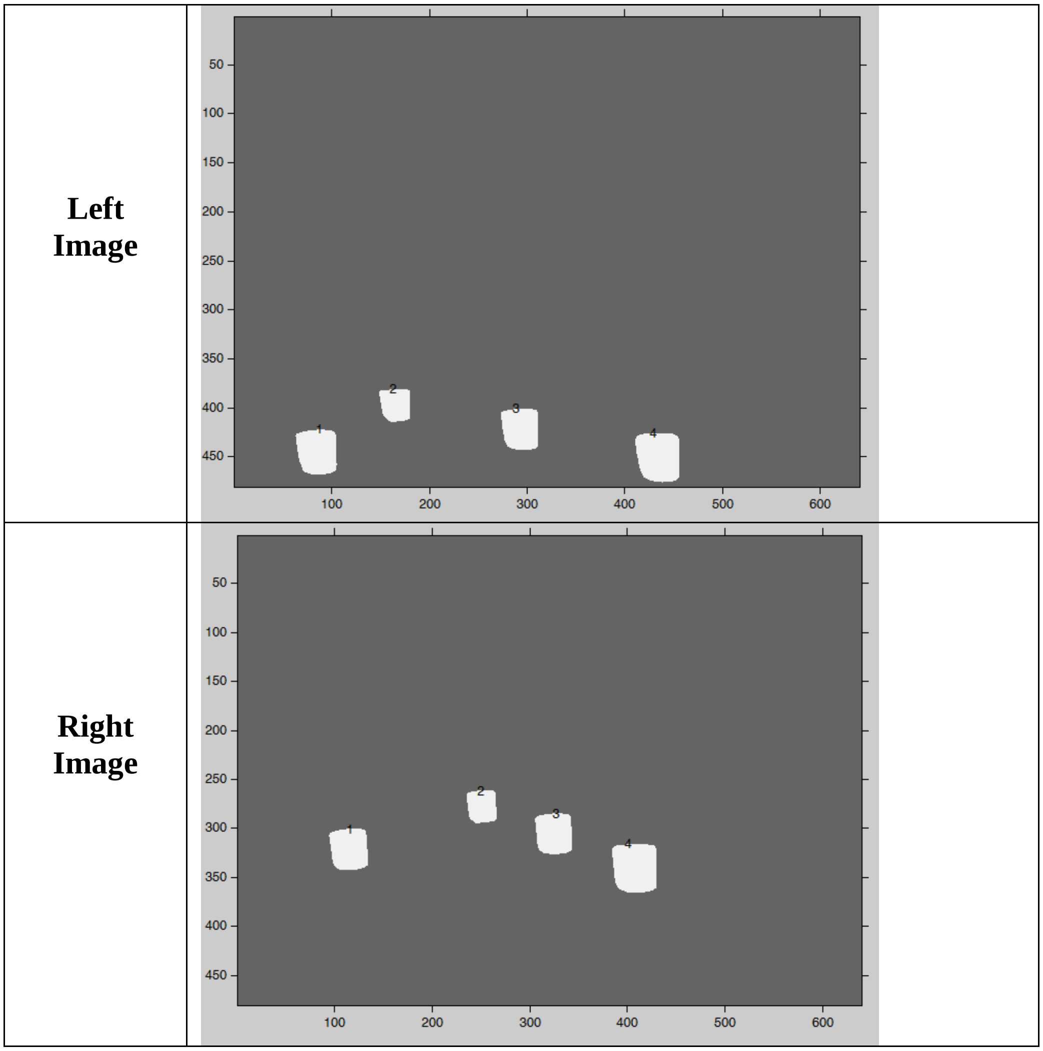


Figure 3. 29 Related Objects in Left and Right Image for Four Objects

The algorithm is successes to detect the four objects in left and right image. the limitation in this algorithm is found, during some location when the objects start to be too close to each other the algorithm will fail to detect the correct number of objects and the shared boundary is noticed. In Figure 3.30 shows the case where the objects is too close and Figure 3.31 shows the false detection to number of objects.

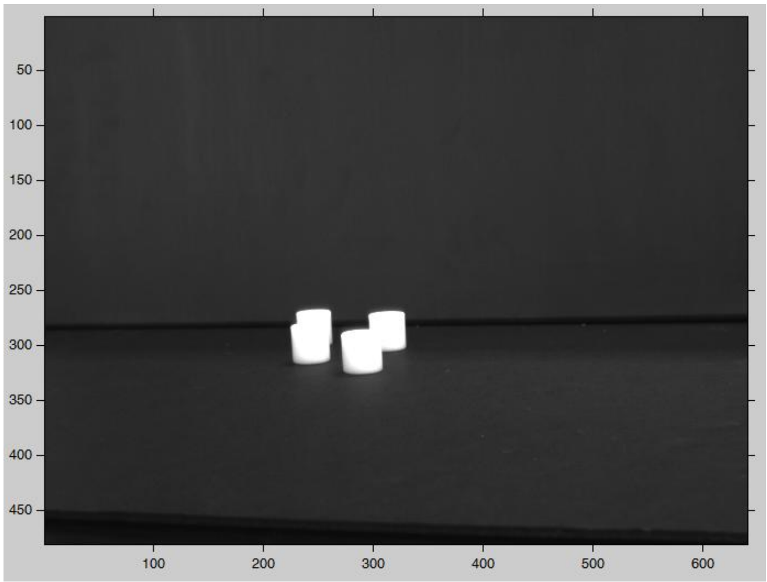


Figure 3. 30 Four Objects in Right Image

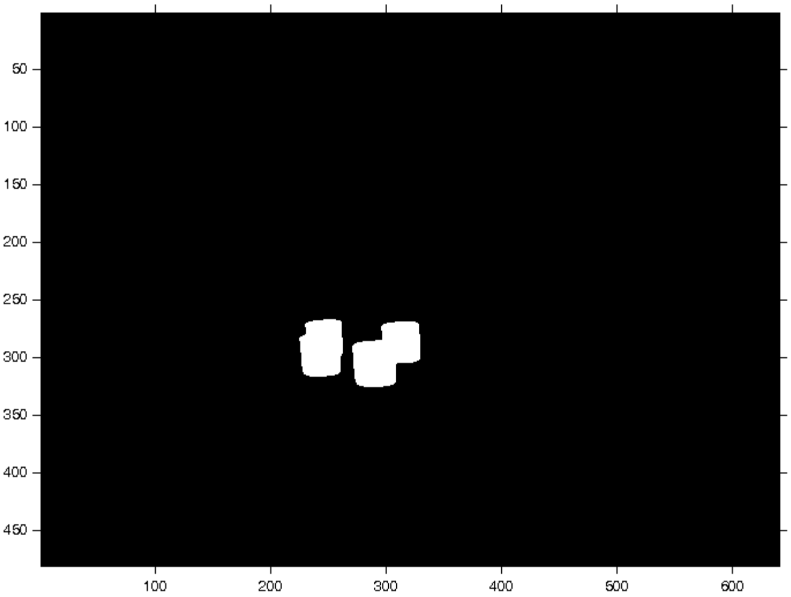


Figure 3. 31 Connected Boundary Problem

3.9 Triangulation

The little bit of theory about triangulation is discussed in chapter one. The final step is to implement triangulation in stereo vision system which is done by taking two views of the object from right and left camera and calculate the depth between the two cylinders.

The triangulation is done based on these derivations [38]. Suppose the first cylindrical has coordinate in right camera reference frame as:

$$P_R = \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} \qquad (3.12)$$

And the first cylindrical has coordinate in left camera reference frame as:

$$P_L = \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} \qquad (3.13)$$

After the calibration, the extrinsic parameters like rotation matrix and translational matrix are known. Then the relation between P_R and P_L related as:

$$P_L = R.P_R + T \qquad (3.14)$$

Then, the projective of the two points p_l and p_r on the image plane is called perspective projection. The idea of the triangulation is to retrieving the $\mathrm{P_L}$ and $\mathrm{P_R}$ from the coordinate of the p_l and p_r are:

$$p_r = \frac{P_R}{Z_R} = \begin{bmatrix} x_R \\ y_R \\ 1 \end{bmatrix} \qquad (3.15)$$

$$p_l = \frac{P_L}{Z_L} = \begin{bmatrix} x_L \\ y_L \\ 1 \end{bmatrix} \qquad (3.16)$$

By institute these relations to the following equation:

$$P_L = R.P_R + T \qquad (3.17)$$

The new equation will be

$$Z_L.p_l = Z_R.R.p_r + T \qquad (3.18)$$

Rearrange the previous equation to:

$$\begin{bmatrix} -R.p_r & p_l \end{bmatrix} \begin{bmatrix} Z_R \\ Z_L \end{bmatrix} = T$$

Where, $A = \left[-R.p_r \quad p_l \right]$ is matrix 3x2 by using the least square solution the previous equation will be:

$$\begin{bmatrix} Z_R \\ Z_L \end{bmatrix} = \left(A^T A \right)^{-1} A^T T \tag{3.19}$$

Where,

$$\alpha = -R.p_r \tag{3.20}$$

The final equation can be written as: where < > is standard scalar product operator

$$Z_R = \frac{\| p_l \|^2 \langle \alpha, T \rangle - \langle \alpha, p_l \rangle \langle p_l, T \rangle}{\| \alpha \|^2 \| p_l \|^2 - \langle \alpha \ , p_l \rangle^2} \tag{3.21}$$

Where, $p_l = \begin{bmatrix} x_1 & x_2 & .. & .. \\ y_1 & y_2 & .. & .. \end{bmatrix}$, this is the x and y coordinates of all objects Centroids in the left image and $p_r = \begin{bmatrix} x_1 & x_2 & .. & .. \\ y_1 & y_2 & .. & .. \end{bmatrix}$, this is the corresponding points found through modified SSD algorithm or by area matching algorithm.

The following flow chart illustrated in Figure 3.32 shows all steps needed to calculate the depth between two objects using modified SSD algorithm.

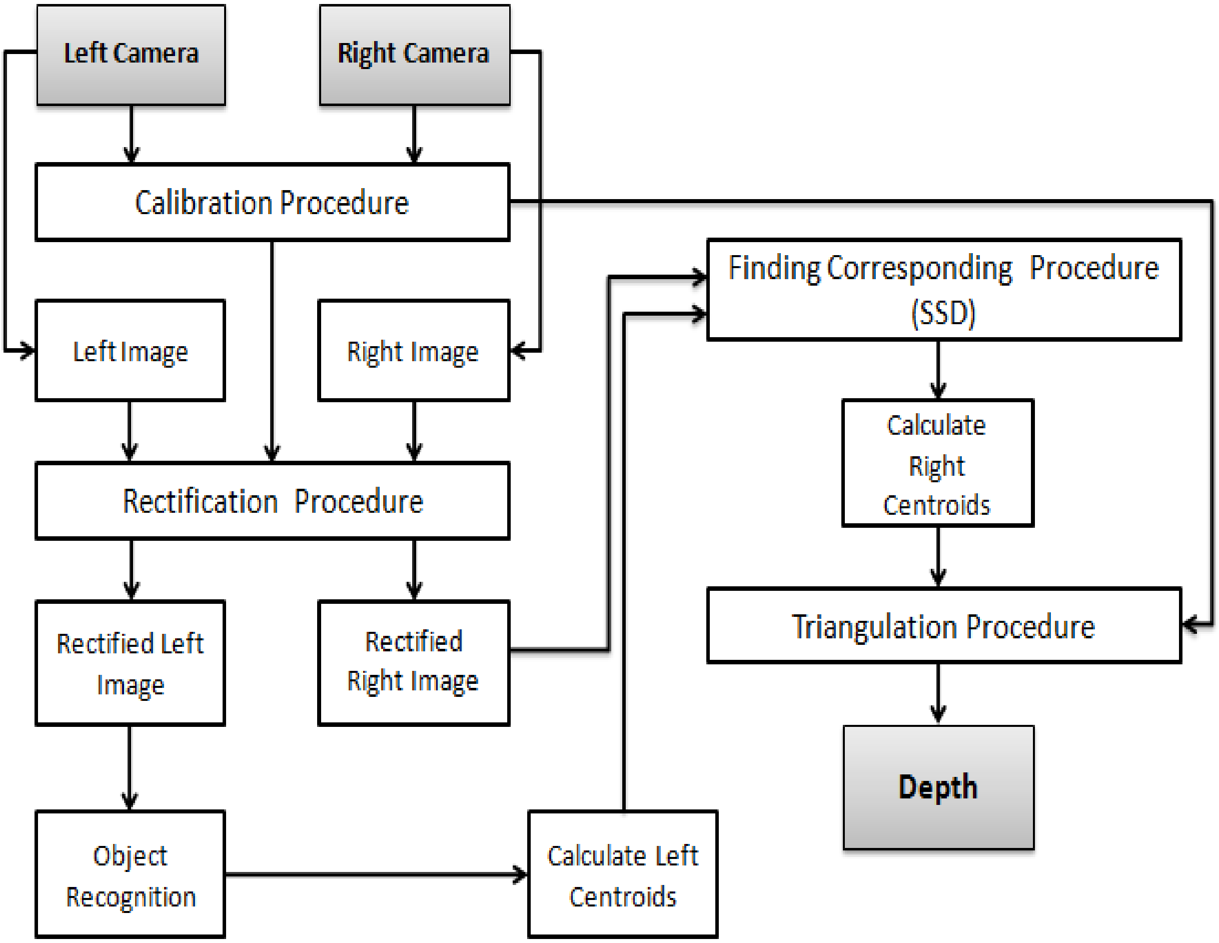


Figure 3. 32 Flow chart to find triangulation using Modified SSD algorithm

Where the following flow chart in Figure 3.33 shows all steps needed to calculate the depth between two objects using proposed feature based algorithm.

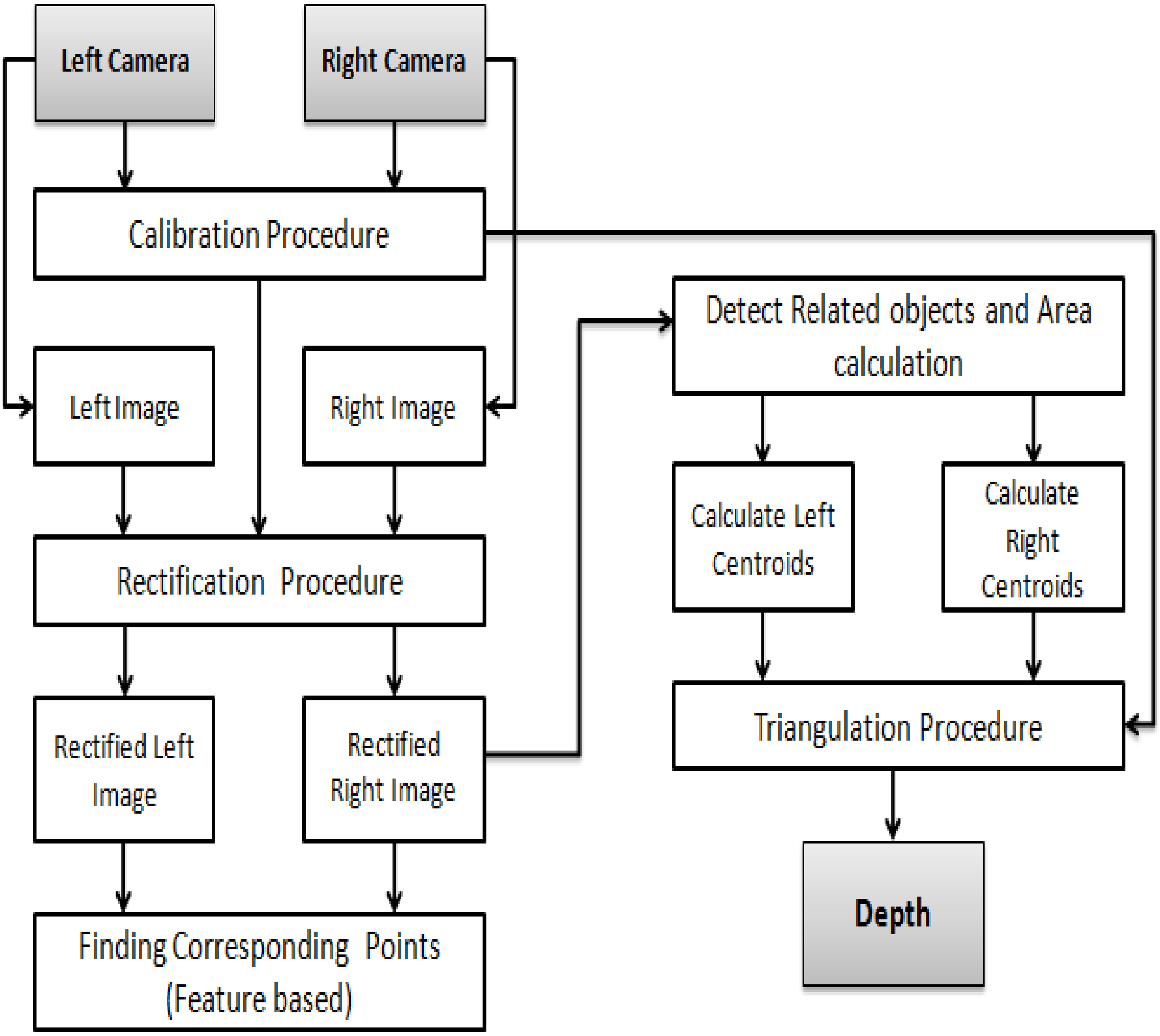


Figure 3. 33 Flow chart to find triangulation using feature based algorithm

The setup of this experiment is shown in Figure 3.34 where two cylinders are placed in front of the left and right camera and the depth is calculated.

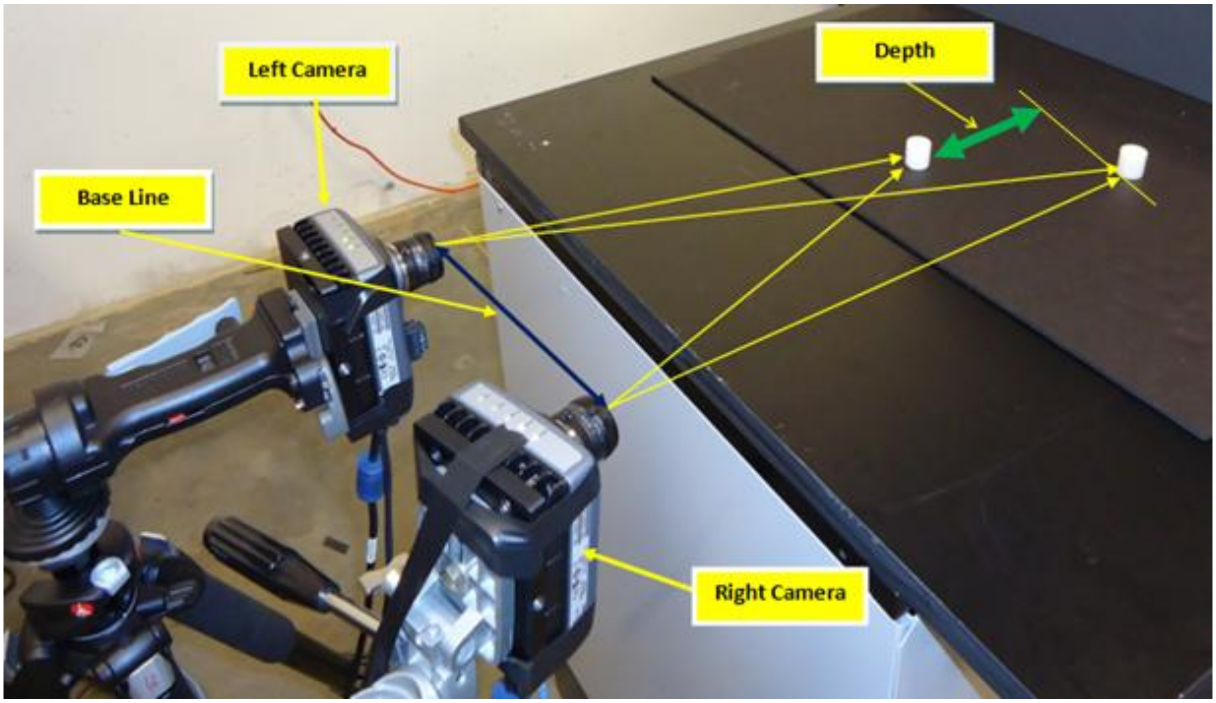


Figure 3. 34 Triangulation based on stereo vision

The first time, the two cameras are separated by 66 cm and the first object is placed at distance around 110 cm from the cameras and then the depth is measured. The experiment is repeated many times and the results are shown in Table 3.13.

Table 3.13 Triangulation Results with baseline equal 66 cm.

Case No.	**Actual Depth in cm unit**	**Calculated Depth in cm unit**	**Depth error in cm**	**Percentage Error (%)**
case1	19.00	16.00	3.00	15.79
case2	11.00	10.10	0.90	8.18
case3	13.00	15.96	2.96	22.77
case5	14.00	16.20	2.20	15.71
case6	13.00	14.60	1.60	12.30
case7	6.50	7.40	0.90	13.85
case8	1.00	1.70	0.70	70.00
Maximum Depth Error in cm			3.00	
Minimum Depth Error in cm			0.70	
Average Depth Error in cm			1.75	

The experiment repeated 8 times where the location of both cylindrical is changed and the new depth between the two cylinders is calculated and compared with the actual depth. The results showed average depth error 1.75 cm and maximum depth error 3 cm which consider large error related to our application pick and place. So new experiment was set and the baseline is changed to get the optimal baseline.

The optimal baseline for this case was around 20 cm, the first object is placed around 90 cm far a away the cameras and again the experiment is repeated many times. The results are shown in Table 3.14.

Table 3.14 Triangulation Results with baseline equal 20 cm.

Case No.	Actual Depth in cm unit	Calculated Depth in cm unit	Depth error in cm	Percentage Error (%)
case1	10.00	10.10	0.1	1.00
case2	6.50	6.43	0.07	1.08
case3	19.00	18.83	0.17	0.89
case4	15.50	15.43	0.07	0.45
case5	12.50	12.37	0.13	1.04
case6	3.50	3.21	0.29	8.29
case7	24.00	23.96	0.04	0.17
Maximum Depth Error in cm			0.29	
Minimum Depth Error in cm			0.04	
Average Depth Error in cm			0.12	

The results showed very good results compare to the previous results and the average depth error found through 7 cases are about 1.2 mm compare to the previous one 17.5 mm.

Another test for triangulation technique was with the place four objects in front of the two cameras and calculates the depth between objects using the optimum baseline distance 20 cm. Figure 3.35 shows the left and right image and Table 3.15 shows the

Centroids location detect using feature based algorithm and Table 3.16 shows comparison between the actual depth and the measured depth in mm unit.

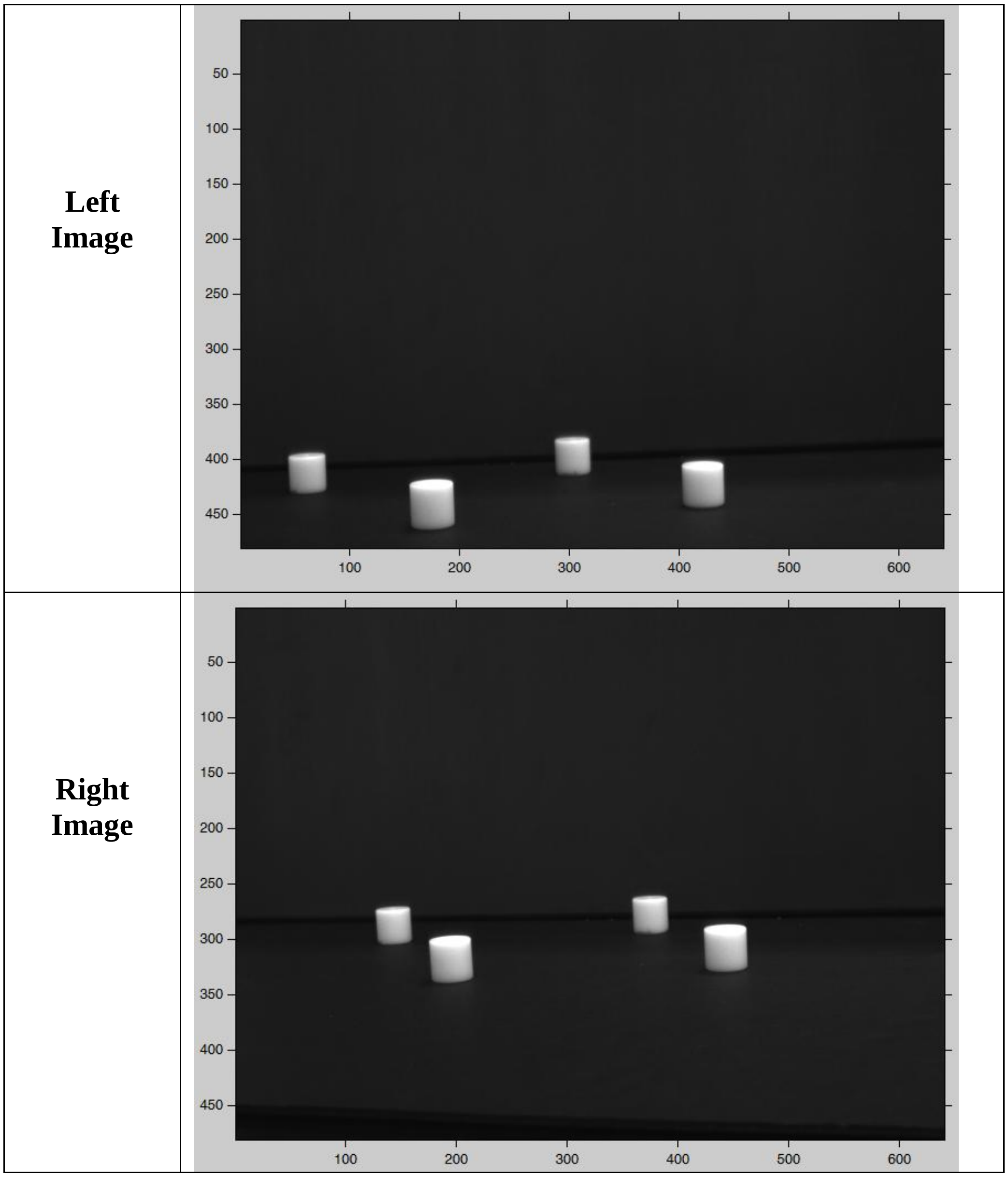


Figure 3. 35 Triangulation based on stereo vision for Four Objects

Table 3.15 Centroids Calculation.

	Left Image	Right Image
Centroids Calculation	Centroids Object1=(62.37, 410.94) Centroids Object2= (176.12, 438.88) Centroids Object3= (303.98, 395.26) Centroids Object4= (422.89, 420.57)	Centroids Object1=(143.80, 286.43) Centroids Object2= (195.9, 316.56) Centroids Object3= (375.615, 276.477) Centroids Object4= (443.94, 306.36)

Table 3.16 Depth Calculation between Objects

Depth between objects	Calculated Depth in cm unit	Actual Depth in cm unit	Depth error in cm
Object 1 and object 2	12.18	12.00	0.18
Object1 and object 3	4.76	5.00	0.24
Object 2 and object 3	16.88	17.00	0.12
Object 1 and object 4	7.57	7.50	0.0 7
Maximum Depth Error in cm			0.24

The maximum depth error was 0.24 cm which is still less than the maximum error found in the first experiment.

CHAPTER 4

4. CONTRIBUTION AND FUTURE STEPS

This Thesis proposed a solution for 3D pick and place for robotics application by used the stereo vision system to calculate the depth between objects and provided this information to the robot.

First, the study based on traditional algorithm used correlation matching called square sum difference SSD then modified version for this SSD algorithm is proposed using automatic ROI and rectification principle to let the new modified SSD algorithm is capable to pick and place in real time application, then the corresponding points was based on center of gravity of the two objects this enhance the elapsed time to 0.022325 seconds with reported error in depth maximum equal 1.2 mm.

Another contribution is proposed another method based on feature matching between two objects on left and right image based on finding match criteria between the two objects based on area calculation. The new algorithm shows good results when two, three and four objects appears in the image and solved the problem found in modified SSD about light change conditions. The limitation found in this method is when the objects start to be too close to each other so the connected boundary condition is found and make the algorithm fails to detect correctly the number of objects in left or right image. The elapsed time is little bit long 1.476514 seconds comparing to the modified SSD algorithm.

The average error in depth between objects in this proposed algorithm is 1.2 mm with average percentage error is about 1.85%.

In Future works, the elapsed time in feature based algorithm should enhance and solve the connected boundary problem.

LIST OF REFERENCES

[1] Kletter, Reinhard, Schluns, Karsten, and Koschan, Andreas “Three-Dimensional Data from Images” 1st Edition. (1996).

[2] Davies, E.R. “Machine Vision: Theory, Algorithm, Practicalities” 2nd Edition. (1997).

[3] Hartley, Richard and Zisserman, Andrew “Multiple View Geometry in Computer Vision” 2nd Edition. (2003).

[4] Iocchi, Luca “http://www.dis.uniroma1.it”

[5] Tyler,C. " Binocular Vision, Foundations of Clinical Ophthalmology” (1982).

[6] DeAngelis, G.C., Cumming, B.G., Newsome, W.T. “Cortical Area MT and the Perception of Stereoscopic Depth” (1998).

[7] Kletter, Reinhard, Schluns,Karsten, and Koschan, Andreas “ Three-Dimensional Data from Images”. 2nd Edition. (1998).

[8] Tsai, R.Y. “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using of the Shelf TV Cameras and Lenses”. IEEE Journal of Robotics and Automation , 3(4): 323-344, (1987).

[9] Roelofs, R., “Distortion, Principal Point, Point of Symmetry and Calibrated Principal Point” (1951).

[10] Faugeras ,O., “Three-Dimensional Computer Vision: a Geometric Viewpoint” (1993).

[11] Zhang, Z., “A Flexible New Technique for Camera Calibration”. IEEE Trans. Pattern Analysis and Machine Intelligence , 22(11):1330-1334, (2000).

[12] Zhang, Z., “Camera Calibration with One-Dimensional Objects”. IEEE Trans. Pattern Analysis and Machine Intelligence, 26 (7), (2004).

[13] Heyden, A. and Pollefeys, M., “Multiple View Geometry,” Emerging Topics in Computer Vision, G. Medioni and S.B. Kang (Ed.), chapter 3, pp. 45-108, Prentice Hall, (2003).

[14] Barnard, S.T., and Fischler, M.A.,“Stereo Vision” in Encyclopedia of Artificial Intelligence , pp. 1083-1090, John Wiley, (1987).

[15] Kanade, Takeo “A Stereo Matching Algorithm with Adaptive Window: Theory and Experiment” fellow, IEE and Masatoshi Okutomi.

[16] Moravec, H.P., “Towards Automatic Visual Obstacle Avoidance” Proceedings of 5th Int. Joint Conf. Artificial Intell., p. 584 , (1977).

[17] Marr, D., and Hildreth, E., “Theory of Edge Detection” Proceedings of Roval Soc. London, B207: pp. 187-217, (1980).

[18] Canny, J.F., “A Computational Approach to Edge Detection”, IEEE Trans. Pattern Anal. Machine Intell., 8 (6): 679-698 , (1985).

[19] Dhond, R.U., Aggarwal, J.K., “Structure from Stereo-A Review”, IEEE Trans. On System Man. And Cypernetics, 19(6): 1489- 1510, (1989).

[20] Premaratne, P., Safaei, F., “Stereo Correspondence Using Moment Invariants” Proceedings of ICIC , 3: pp.447-454 , (2008).

[21] Scott, G., Longuet–Higgins, H., “An Algorithm for Associating the Features for Two Patterns”. Proc. Royal Society London, 244: pp. 21-26, (1991).

[22] Pilu, M., “A Direct Method for Stereo Correspondence Based on Singular Value Decomposition” Proceedings of Computer Vision and Pattern Recognition Conference, pp. 261-266, (1997).

[23] Hebert, M. “Active and Passive Range Sensing for Robotics”, Proceedings of the 2000 IEEE international Conference on Robotics & Automation, pp. 102-110, (2000)

[24] Sato,K., Inokuchi, S., “Range Picture Input System Based on Space – Encoding”. Transaction of the Institute of Electronics and Communication Engineers of Japan, Part D, J68D (3).(1985).

[25] Omar, M., Zhou, Y., Planting, E., et.al. “Combined Active Triangulation, Morphology Scheme for Active Shape Retrieval”, Sensor Review, 29(3): 233 – 239, (2009).

[26] Orkman, M., “Stereo Geometry and Matching”, lecture notes, Computational Vision and Active Perception School of Computer Science and Communication, (2010)

[27] Ayache, N., and Lustman, F., “Fast and Reliable Trinocular Stereovision” Proceedings of 1st Int. Conf. Computer Vision, pp.422-427, (1987).

[28] Beardsley, P.A., Zisserman, A., Murray, D.W., “Navigation Using Affine Structure from Motion”, Proceedings of the Third European Conference on Computer Vision, 2:85-96, (1994).

[29] Beardsley, P.A., Zisserman, A., Murray, D.W., “Sequential Update of Projective and Affine Structure from Motion”, Int. J. Computer Vision, 23: 235-259, (1997).

[30] Hartley, P.I. and Sturm, P., “Triangulation” Proceedings of the ARPA Image Understanding Workshop , 68: pp. 146-157, (1997).

[31] Hartley,R., Gupta,R., and Chang,T., “Stereo From Uncalibrated Cameras”, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 761–764. (1992).

[32] Natural Resources Canada Surveying Good overview of surveying with references to construction surveys, cadastral surveys.

[33] Bowditch, Nathaniel “The American Practical Navigator”. Bethesda, MD: National Imagery and Mapping Agency. ISBN 0939837544. (2002).

[34] Website: http://www.isravision.com/

[35] Iversen,W., “Vision-Guided Robotics: In Search of the Holy Grail” Automation World, (2006).

[36] Ban, Kazunori, Warashina, Fumikazu, Kannoand, Ichiro, Kumiya, Hidetoshi “Industrial Intelligent Robot,” FANUC Tech. Rev., 16(2): pp 29-34, (2003).

[37] Porrill, J., Pollard, S.B., Pridmore, T.P., Bowen, J.B., et al. “TINA: A 3D Vision System for Pick and Place” J. Image and Vision Computing, 6(2): 65-72, (1988).

[38] Bouguet, J.Y, Intel Corp, "Camera Calibration Toolbox for Matlab" website: www.vision.caltech.edu/bouguetj/calib_doc.

[39] Trucco, E., and Cerri, A., “Introductory Techniques for 3-D Computer Vision” Prentice-Hall, (1998).

[40] Ayache, N., translated by Sanders, P.T., “Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception”. The MIT Press,(1991).

[41] Fusiello, A., Trucco, E., Verri, A., “A Compact Algorithm for Rectification of Stereo Pairs”, Machine Vision and Applications, 12(1) : 16-22, (2000).

[42] Dhond ,U.R., Aggarwal, J.K., “Structure from Stereo – A Review”. IEEE Trans. Syst. Man. And Cybern. 19(6):1489–1510, (1989).