ABSTRACT

STEREO MATCHING ALGORITHM BY PROPAGATION OF CORRESPONDENCES AND STEREO VISION INSTRUMENTATION

by

Peerajak Witoonchart

A new image processing method is described for measuring the 3-D coordinates of a complex, biological surface. One of the problems in stereo vision is known as the accuracy-precision trade-off problem. This thesis proposes a new method that promises to solve this problem. To do so, two issues are addressed. First, stereo vision instrumentation methods are described. This instrumentation includes a camera system as well as camera calibration, rectification, matching and triangulation. Second, the approach employs an array of cameras that allow accurate computation of the depth map of a surface by propagation of correspondences through pair-wise camera views.

The new method proposed in this thesis employs an array of cameras, and preserves the small baseline advantage by finding accurate correspondences in pairs of adjacent cameras. These correspondences are then propagated along the consecutive pairs of cameras in the array until a large baseline is accomplished. The resulting large baseline disparities are then used for triangulation to achieve advantage of precision in depth measurement.

The matching is done by an area-based intensity correlation function called Sum of Squared Differences (SSD). In this thesis, the feasibility of using these data for further processing to achieve surface or volume measurements in the future is discussed.

STEREO MATCHING ALGORITHM BY PROPAGATION OF CORRESPONDENCES AND STEREO VISION INSTRUMENTATION

by

Peerajak Witoonchart

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

May 2002

APPROVAL PAGE

STEREO MATCHING ALGORITHM BY PROPAGATION OF CORRESPONDENCES AND STEREO VISION INSTRUMENTATION

Peerajak Witoonchart

Dr. Stanley Reisman, Thesis Advisor — Date
Professor of Electrical and Computer Engineering, NJIT
Professor of Biomedical Engineering, NJIT

Dr. Yun-Qing Shi, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Richard Foulds, Committee Member — Date
Associate Professor of Biomedical Engineering, NJIT

BIOGRAPHIC SKETCH

Author: Peerajak Witoonchart

Degree: Master of Science

Date: May 2002

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2002

- Bachelor of Engineering, Major Electrical Engineering,
  Sirindhron International Institute of Technology, Thammasat University, 1996

Major: Electrical Engineering

Publications:

P. Witoonchart, and R. Foulds. "Three-Dimensional Surface and Volume Measurements Using a Camera Array." *IEEE 28th Annual Northeast Bioengineering Conference* (2002): 145-146.

To my beloved family and relatives

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to Dr. Stanley Reisman, who not only served as my research supervisor, providing valuable and countless resources, insight and intuition, but also constantly gave me support, encouragement and reassurance. Special thanks are given to Dr. Richard Foulds and Dr. Yun-Qing Shi for being my technical consultant as well as actively participating in my committee. A research assistantship has been provided from Dr. Richard Foulds' start-up research funding from NJIT.

Thank you to Darnell Simon, Bruno Mantilla, Gene Moore, Don Helt, Robert DeMarco, Chairat Pinthong for support during my thesis work. I wish to thank Suravut Yaipairoj, Sutkate Yaipairoj, Noppon Pitsuthanon, Pongchit Eakarapanish, Suwat Sirinuwat for general support during this two years. I also wish to thank Hataitip Phitthaya-Apiphol for her moral support. I wish to thank my family, especially my parents, Wongwai and Gasinee Witoonchart, my aunt Somporn, my Aakung Seiang, my grand mather Somboon, for financial and moral support during my Master's study.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
1.1 Specific Aims	1
1.2 Background	3
1.2.1 Correspondence Problem	4
1.2.2 Epipolar Geometry	4
1.2.3 Camera models and parameters	5
1.2.3.1 Perspective Camera Mode	6
1.2.3.2 Camera parameters	7
1.2.4 Triangulation and Reconstruction	10
1.2.5 Previous Method	11
1.2.6 Proposed Method	14
2 MATERIAL AND METHODS	15
2.1 Apparatus	15
2.2 Camera Calibration	16
2.2.1 Real Camera and Pinhole Model	17
2.2.2 Calibration Grids	17
2.2.3 Camera Calibration Method	18
2.3 Rectification Process	22
2.3.1 Rectification Coordinates	23
2.3.2 Image Warping	24

TABLE OF CONTENTS
(Continued)

Chapter **Page**

2.3.3 Implementation of Rectification Warping 24

2.3.4 Using the Rectification Program 25

2.4 Image Matching 26

2.5 Triangulation 29

2.6 Experimental Procedure 30

3 RESULTS AND DISCUSSION 33

3.1 Calibration Results 33

3.2 Rectification Results 35

3.3 Matching Results 37

3.4 Triangulation Results 39

4 CONCLUSIONS 45

APPENDIX A PROGRAM USED IN THESIS 46

APPENDIX B RAW DATA 55

REFERENCES 69

LIST OF TABLES

Table		Page
2.1	Camera A. calibration results	21
3.1	Calibration result of camera A	34
3.2	Proposed method A-D disparity results	38
3.3	Direct matching method A-D disparity results	39
3.4	Triangulation results	40
B.1	Position and disparity of a stereo pair A-B	63
B.2	Position and disparity of a stereo pair B-C	63
B.3	Position and disparity of stereo pair A-D	64
B.4	Parameters from camera calibration	64
B.5	Triangulation results	65
B.6	Disparity of our method A-B-C	65
B.7	Proposed method B-C-D	66
B.8	Triangulation proposed method	66
B.9	Triangulation of proposed method (cont.)	67
B.10	Calibration result of camera A	67
B.11	Calibration result of camera B	67
B.12	Calibration result of camera C	68
B.13	Calibration result of camera D	68
B.14	Stereo calibration result, Translation vectors between cameras	68
B.15	Stereo calibration result, Rotation matrices between cameras	68

LIST OF FIGURES

Figure		Page
1.1	Geometry of epipolar lines	5
1.2	The perspective camera model	7
1.3	the relation between camera and world reference	8
1.4	Triangulation of SSG	10
1.5	Triangulation with rotation around Y axis	11
2.1	Equipment setup	15
2.2	Calibration grid used in experiment	18
2.3	Example of camera calibration grid shown by camera calibration toolbox for Matlab	19
2.4	Extrinsic parameters. Calibration grids and camera reference frame	20
2.5	Five Results of stereo calibration. Rotation and Translation Matrices between two cameras are defined	21
2.6	Rectification	22
2.7	Rectification results a) Original Images b) Rectified Images	25
2.8	Dense disparity map found by application of SSD through synthesis stereo pairs	28
2.9	Dense disparity map found by application of SSD through a Renault stereo pairs.	28
2.10	Correct intensity match found by searching rectified stereo pair shown in figure 2.9	29

LIST OF FIGURES
(Continued)

Figure		Page
2.1	Tracing back the original image position given a known rectified image position	31
2.2	Proposed method. Propagation of correspondences	31
3.1	Extrinsic parameter for camera A	34
3.2	Original and Rectified image pair a) Left and right images from camera A-C, b) Rectified stereo pair A-C	35
3.3	Image taken from each camera and the points of interest	36
3.4	SSD value versus candidate disparity for three stereo pairs a) SSD graph for A-B stereo pair, b) SSD graph for B-C, c) SSD graph for A-C	37
3.5	Proposed method gives superior result dealing with image blurred	41
3.6	Small baseline triangulation result	42
3.7	The precision of A-D is higher than A-B	43
3.8	Triangulation result	44
B.1	Image taken from each camera and the points of interest	55
B.2	Image taken from all views a) image taken from camera A, b) image taken from Camera B	56
	c) image taken from Camera C, d) image taken from right most camera D	57
B.3	Rectified image all pairs. a) Rectified stereo pair A-B	58
	b) Rectified stereo pair A-C	59
	c) Rectified stereo pair A-D	60
	d) Rectified stereo pair B-C	61
	e) Rectified stereo pair C-D	62

CHAPTER 1

INTRODUCTION

1.1 Specific Aims

Primates are born with two eyes side by side positioning that share a common view of the world. The locations of the eyes help the primate to be able to measure distances, hence extremely enhancing their chance of survival. One eye alone see images in monocular form, and the brain does not perceive depth. With two eyes viewing a common screen, the brain and neural systems fuse the left and right eye images together in such a way that the human gets the feeling of a three-dimensional world. The ability to see a 3-D world using two or more image devices, either biological or machine, positioning side by side is called stereovision.

A better definition of stereovision is described in [7] as "Stereo vision refers to the ability to infer information on the 3-D structure and distance of a scene from two or more images taken from different viewpoints." An example of how the brain functions to generate stereovision can be done by holding a thumb at arm's length and closing the right and left eyes alternately. The experiment shows that the relative position of the thumb and background appears to change. It is this difference that is used by the brain to reconstruct a 3-D representation of what a human sees. In doing so, the stereo systems must solve two problems, namely, "correspondence problem" which consists in determining which item in the left eye corresponds to which item in the right eye and the "reconstruction" problem to give the 3-D view. The Reconstruction problem can be solved once the correspondence problem is solved. The difference between the left and right imaging device position is called disparity. Once the disparities are known, the

reconstruction problem can be solved. The disparity map describes disparities of all image points displayed in the image. Because of positioning difference of the imaging devices, there exist some positions in the real world that can be seen only from one device but not the others. This problem is known as the occlusion problem.

Applications of stereo-based depth measurement include automated cartography, aircraft navigation, autonomous land rovers, robotics, industrial automation and stereomicroscopy. Research in stereovision lies in recovering depth information by solving corresponding problems. Reconstruction problems can be done by triangulation once all corresponding points are known.

Matching is perhaps the most important stage in stereo computation. It can be classified into area-based stereo techniques, which use correlation among brightness (intensity) patterns in the local neighborhood of a pixel in one image with the brightness pattern in a corresponding neighborhood of a pixel in the other image. First, a point of interest is chosen in one image. A correlation measure is then used to search for a point with a matching neighborhood in the other image. The area based techniques have a disadvantage in that they use intensity values at each pixel directly, and are hence sensitive to distortions as a result of changes in viewing position 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. 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.

In this thesis, a new concept for determining the geometric characteristics of surfaces found in biological systems is described. The proposed approach extends existing stereo camera methods to overcome many of the limitations presented by occlusions, lighting and camera alignment. With arrays of cameras that are used to develop accurate sparse representation of image surfaces, hypotheses relating to surface behavior and active lighting can be generated. Using multiple cameras alone does not solve the problem of macthing ambiguity that occurs with smooth untextured object surfaces in the scene. For this reason the idea of using active lighting in the form of a projected pattern on the scene may be important [13]. Combining these hypotheses with a priori knowledge of biological surfaces and less accurate data from the camera arrays allows the generation of a more complete 3-D representation of the surfaces.

Three-dimensional imaging allows the representation of geometrically complex structures that cannot be approximated by mathematical functions. With a 3-D imaging tool it would be possible to quantify dynamic surface and volumetric changes in soft tissue and other deformable materials. In addition, it will be possible represent articulated structures (e.g. hands, arms, and legs) that change the orientation of their segments over time. This can facilitate computer recognition of sign language, and assessment of human performance in ergonomics and rehabilitation.

1.2 Background

In this section, epipolar geometry, camera models, camera parameters, triangulation and some previous well known methods to solve the correspondence problem will be described.

1.2.1 Correspondence Problem

As cited in Klette et. al. [2], two major important approaches to find a pair of points that match in the two images are known to be: 1) Structured lighting, which encodes the two images so that it is easy to see pairs of corresponding points, and 2) Employing epipolar lines, which simplify the correlation calculation for corresponding point. To find a corresponding point from one camera relative to the others, one needs to find a match of that point in the other camera two dimensional image. In this approach, camera geometry can be used to show that the search of a corresponding point of a camera can be done in one dimension.

1.2.2 Epipolar Geometry

As cited in Klette et. al. [2], rectification is the term used in stereo vision that utilizes geometrical characteristics to simplify the search for 2-D correspondences into one dimension. This gives a simpler search algorithm consuming less time and giving better results because of fewer mismatches.

It can be seen from Figure1.1 that if a correspondence of a point P1 in Camera 1 whose focal point is C1 is to be found in the image plane of camera 2, the geometry of the camera dictates that the position of an object whose projection is P1 lies in the line C1P1. The corresponding point of P1 must lie on the projection of this line to the image plane of camera 2.

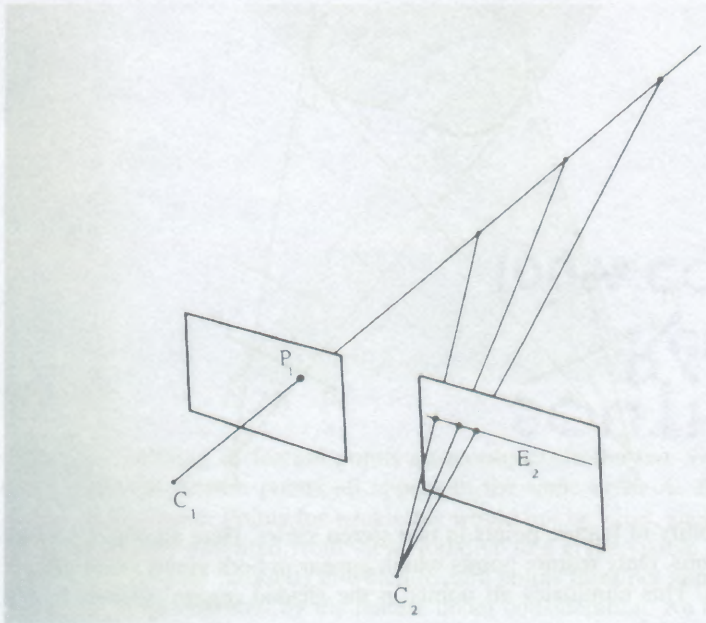


Figure 1.1 Geometry of epipolar lines.
(Note: E. R. Devies 1997 [27])

A point P1 in one image plane may have arisen from any one of a line of points in a scene, and may appear in the alternate image plane at any point on the epipolar line E2. The Epipoles are poles of epipolar lines where all epipolar lines cross each other. Epipoles can be placed at infinity, which means that all epipolar lines are parallel by aligning two cameras.

1.2.3 Camera Models and Parameters

Since cameras are the important instruments for this project, one needs an understanding of camera models, optical parameters of the lens, photometric and geometric parameters. The optical parameters of the lens include lens type, focal length, field of view, and angular apertures. The goal of camera modeling is to mathematically represent how the camera maps the world coordinate points into pixels or pixel coordinate points. The mathematical equations are called camera models and the variables are called camera

parameters. To accomplish rectification, one needs to understand the camera models and parameters. In this case, the perspective camera models, which are extensively used in stereo vision and rectification is explained in this thesis.

Photometric parameters appear in models of the light energy reaching the sensor after being reflected from the objects in the scene. They include type, intensity and direction of illumination, reflectance properties of the viewed surfaces, effect of the sensor's structure on the amount of light reaching the photoreceptors. Geometric parameters determine the image position on which a 3-D point is projected. They include types of projections, position and orientation of the camera in space, and perspective distortions introduced by the imaging process.

1.2.3.1 Perspective Camera Model. With reference to E. Trucco and A. Verri [7], the perspective camera (pin hole model) consists of the image plane, 3-D point O, the optical center or focus of the projection where all light converges to this point, as shown in Figure 1.2. The distance between camera frame and image plane is focal length. The line perpendicular to the image plane, which also passes through the optical center is called the optical axis and the point of projection of the optical axis to the image plane is the principal point or image center. The basic equations of the perspective projections in the camera frame are defined by E. Trucco and A. Verri [7] and are;

$$x=f*X/Z \tag{1.1}$$

$$y=f*Y/Z \tag{1.2}$$

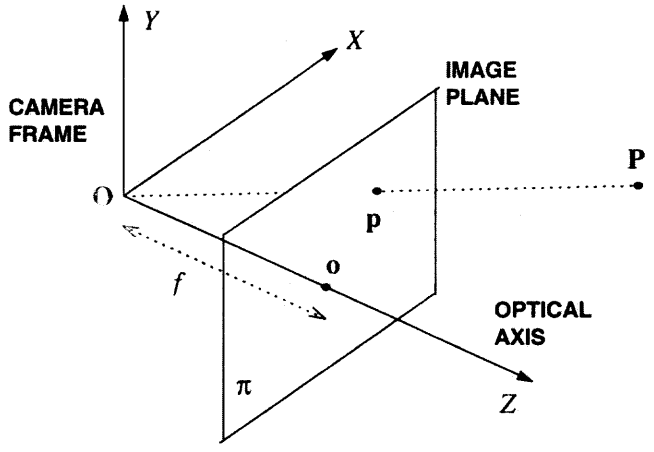


Figure 1.2 The perspective camera model.
(Note: E. Trucco and A. Verri 1998 [7])

1.2.3.2 Camera parameters. It is assumed that the camera reference frame can be located with respect to some other known reference frame, called the world reference frame, and the coordinates, of the image points in the camera reference frame can be obtained from pixel coordinates the only ones directly available from the image. Camera parameters can be categorized into two kinds. They are extrinsic parameters and intrinsic parameters. The extrinsic parameters are the parameters that define the location and orientation of the camera reference frame with respect to a known world reference frame. The intrinsic parameters are the parameters necessary to link the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame.

The extrinsic parameters are defined as any set of geometric parameters that uniquely identify the transformation between the unknown camera reference frame and a known world reference frame [7]. It involves 2 matrices called the rotation matrix and translation vector which define the rotation and translation of the camera reference frame to the world reference frame as shown in Figure 1.3. From [7], mathematically

$$\mathbf{P}_c = R(\mathbf{P}_w - \mathbf{T}) \qquad (1.3)$$

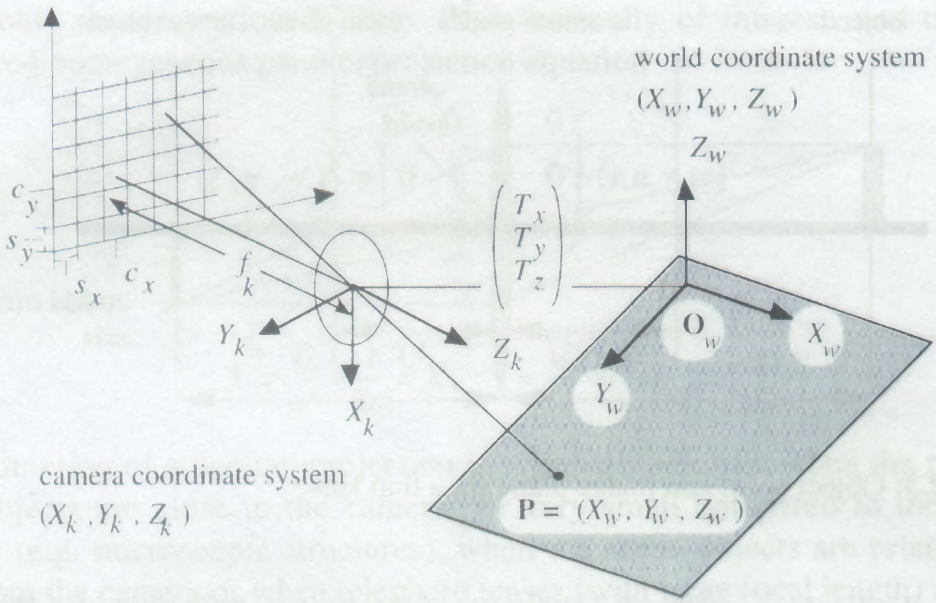


Figure 1.3 the relation between camera and world reference frame.
(Note: Reinhard Klette et. al.1998 [2])

Intrinsic parameters are defined as the set of parameters needed to characterize the optical, geometric and digital characteristics of the viewing camera[7]. The intrinsic parameters include f, focal length, Ox,Oy the principal point of the image, which is defined as the point on the image plane at which the optical ray passes the focal length through the focal point and intersects the image plane perpendicularly; and S_x , S_y, the effective size of the pixel in millimeters in the horizontal and vertical direction respectively, and finally the radial distortion coefficient k.

The relationship between the image coordinates and the world reference frame can be computed by knowing all intrinsic and extrinsic parameters, given a linear version of the perspective projection equations, as found in [7],

$$-(X_{im} - O_x)Sx = f\,\frac{\mathbf{R}_1^T(\mathbf{P}_w - \mathbf{T})}{\mathbf{R}_3^T(\mathbf{P}_w - \mathbf{T})} \tag{1.4}$$

$$-(Y_{im} - O_y)Sy = f\,\frac{\mathbf{R}_2^T(\mathbf{P}_w - \mathbf{T})}{\mathbf{R}_3^T(\mathbf{P}_w - \mathbf{T})} \tag{1.5}$$

For many cameras, there is essentially no radial distortion. The above equations can be written in matrix equations.

$$\begin{bmatrix} X1 \\ X2 \\ X3 \end{bmatrix} = \mathbf{M}_{int}\,\mathbf{M}_{ext} \begin{bmatrix} Xw \\ Yw \\ Zw \\ 1 \end{bmatrix} \tag{1.6}$$

Where

$$X1/X3 = X_{im} \tag{1.7}$$

$$X2/X3 = Y_{im} \tag{1.8}$$

$$\mathbf{M}_{int} = \begin{bmatrix} -f/Sx & 0 & Ox \\ 0 & -f/Sy & Oy \\ 0 & 0 & 1 \end{bmatrix} \tag{1.9}$$

$$\mathbf{M}_{ext} = \begin{bmatrix} r11 & r12 & r13 & \mathbf{R}_1^T\mathbf{T} \\ r21 & r22 & r23 & \mathbf{R}_2^T\mathbf{T} \\ r31 & r32 & r33 & \mathbf{R}_3^T\mathbf{T} \end{bmatrix} \tag{1.10}$$

Matrix $\mathbf{M}_{int}$ in equation (1.9) is the intrinsic parameter matrix and $\mathbf{M}_{ext}$ in equation (1.10) is the extrinsic parameter matrix.

1.2.4 Triangulation and Reconstruction

Triangulation is the recovery of depth information given the known disparity value of the 3D point. As discussed in [26],

$$Z = (b*f)/(x1-x2) \quad (1.11)$$

$$X = x1* z/f \quad (1.12)$$

$$Y= y1 * z/f \quad (1.13)$$

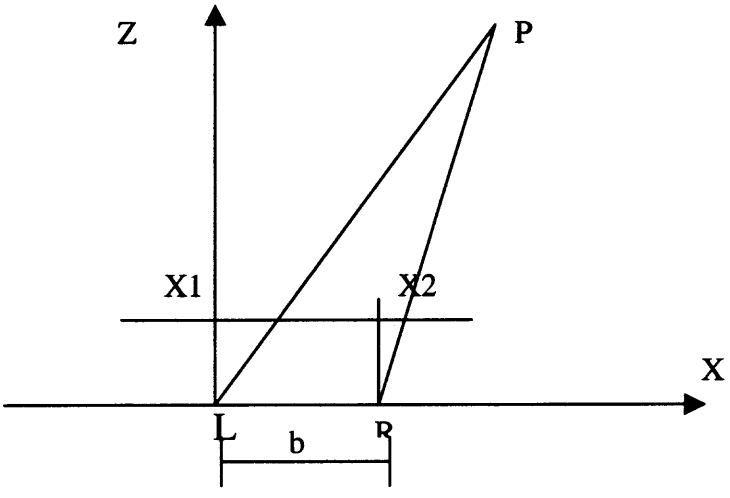


Figure 1.4 Triangulation of SSG.

Figure 1.4 is the case of "standard stereo geometry" where the optical axes of both cameras are parallel. It also has a property that epipolar lines are parallel. Given the camera position parameters such as pan angle, base lines, the depth information, Z is a function of disparity only. More over they are inversely proportional. This can be verified by experimentation. Note that if base line b is small then the uncertainty of the disparity measurement will have a greater effect on the result Z. Also, large changes in Z of the far area correspond to small change in disparity. Once the disparity of any position is known, three-dimension information can be reconstructed using triangulation.

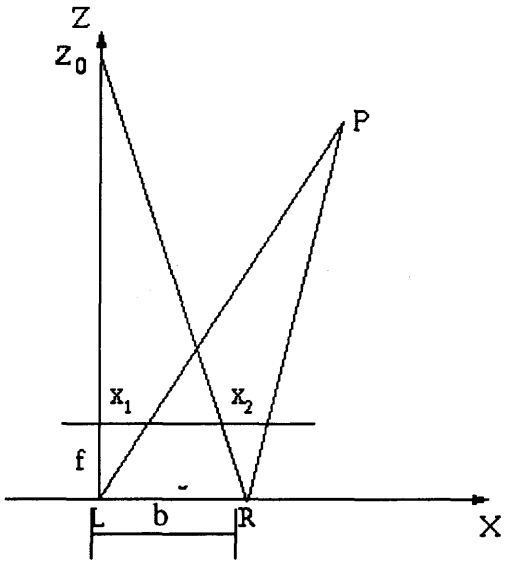


Figure 1.5 Triangulation with rotation around Y axis.

As shown in Figure 1.5, when the optical axes are not parallel, under small angle approximation, triangulation with rotation around the Y axis, the reconstruction formulas [26] are given by

$$Z= (B*f)/(x1-x2+f*B/Zo) \quad (1.14)$$

$$X=x1*Z/f \quad (1.15)$$

$$Y=Y1*Z/f \quad (1.16)$$

where Zo is the fixation point where optical axes of camera L and R intersect.

If θ is the rotation angle, $Zo=b/\tan(\theta)$

1.2.5 Previous Methods

As cited by Dhond and Aggrawal [20], two major classifications of techniques have been used for finding correspondences. These are area based stereo and feature based stereo. Area-based stereo techniques correlate the brightness (intensity) patterns in the local neighborhood of a pixel in one image with brightness patterns in a corresponding

neighborhood of a pixel in the other image. Feature-based stereo techniques use a symbolic feature such as edge points or edge segments, which may be located with sub-pixel precision, to match. However, the feature-based process may be confused by large local change in disparity, and it is very difficult to incorporate the smoothness assumption into the matching strategy since it is most likely to be violated at the edges [24].

Marr and Poggio [6] proposed a feature-point based computational model of human stereopis. Grimson [5] developed a computer implementation of their algorithm and demonstrated the effectiveness of this model on standard psychophysical test images, random dot stereograms, as well as on natural images. Many additional psychophysical predictions of the Marr-Poggio[6] model have been tested in which several modifications have been proposed [16],[17],[18],[19]. Based on their computational structure of the stereo fusion problem found in biology, Marr and Poggio [6] proposed that the human visual processor solves the stereo matching problem in five main steps. 1) Approximated by the difference of two Gaussian functions with space-constants in the ration 1:1.75, each left and right image are filtered at twelve deferent orientation-specific masks. 2) Zero-crossings are found by scanning the filtered images along lines perpendicular to the orientation of the mask. 3) For each mask size, matching takes place between the zero-crossing segments extracted from each filtered image output that are of the same sign and roughly the same orientation. Local matching ambiguities are resolved by considering the disparity sign of nearby unambiguous matches. 4) Matches obtained from wider masks control vergence movements aiding matches among output of smaller masks; 5) The correspondence results are stored in a dynamic buffer called the 2.5-D sketch. In the field of area-based stereo, much work has been done in stereo matching involving the use of

correlation measures to match neighborhoods of points in the given images. With a coarse-to-fine strategy, Moravec [21] has used area-based correlation to find corresponding match points. Initially feature points are identified in each image by the Moravec interest operator [21] that measures directional variance of image intensities in four directions surrounding a given pixel. Gennery[22] developed a high-resolution correlator. This correlator uses the matches provided by the previous correlation matcher and produces an improved estimate of the matching point based upon the statistics of noise in the image intensities. This high-resolution correlator provides improved match points, and gives an estimation of the accuracy of the match in the form of variances and covariance of the (x,y) coordinates of the match in the second image. Hannah [23] developed a correlation-based stereo system for an autonomous aerial vehicle featured with the modified Moravec Operator used to select control points. Autocorrelation in the neighborhood of a candidate match point is used to evaluate the goodness of a match. Sub-pixel matching accuracy is achieved through parabolic interpolation of correlation values.

As cited by Kanade and Okutomi [11], one of the problems in Stereo vision is a trade off between precision and accuracy (correctness) in matching. This is due to the fact that disparity and distance are related by

$$d = B*f/z \tag{1.17}$$

Where d is disparity, B is baseline and f is focal length. This equation indicates that for the same distance, the disparity is proportional to the baseline or that the baseline length B acts as a magnification factor in measuring d in order to obtain z, that is, the estimated distance is more precise if the two cameras are set farther apart from each other, which

means a longer baseline. A longer baseline, however, has its own problem. Because a longer disparity range must be searched, correspondence matching is more difficult, and thus, there is a greater possibility of a false match [11].

1.2.6 Proposed Method

To solve the accuracy precision trade-off problem, Three-Dimensional Surface and Volume Measurements Using a Camera Array [1] method is proposed in this thesis. The proposed approach employs five cameras in an array that allow each pair of adjacent cameras to have an acceptably small baseline, while the entire array provides a large equivalent baseline. The proposed method determines these correspondences in a pair-wise fashion among the cameras. Correspondences that appear with high correlations in all pairs receive the highest rank, and those that have high correlations in only one pair receive the lowest rank. The highly ranked correspondences can be used to compute a sparse depth map of the surface. These data can be used with B-spline or Besier methods to fit a smooth surface. To increase the density of the depth map, the proposed method examines the behavior of correspondence gradients from camera to camera. Since biological surfaces change shape gradually except at discontinuities, it is assumed that the correlations of pixels representing a point on the object will behave smoothly from one camera pair to another. Lower ranked correspondences that show high correlations among some camera pairs can be promoted to higher ranking if their gradient of correlations progresses smoothly through the other camera pairs. Conversely, candidate correspondences that have high correlations in one or more pairs but very low correlations in other camera pairs are viewed as errors.

CHAPTER 2

MATERIALS AND METHODS

2.1 Apparatus

In order to build the system described in this thesis, cameras with firewire (IEEE 1394) connection to the PC are used as the imaging device. There are five high quality Pixelink (Vitana Corp, Ottawa, Canada) firewire cameras attached to a firewire hub. Images of 1280 x 1024 pixels can be obtained from all cameras at frame rates 9.3 fps. The imaging devices have 7.5 μm x7.5μm pixels. Lenses are attached via C mount. Rainbow CCTV S16 1.4E-II (Rainbow CCTV, California, USA) lenses with focal length of 16mm, maximum relative aperture 1:1.4 are selected. All cameras are mounted on an optical table with high stability. The apertures of all cameras are set to receive maximum illuminations. With the Pixelink cameras and their Twain drivers, snapshot images can be taken for further processing.

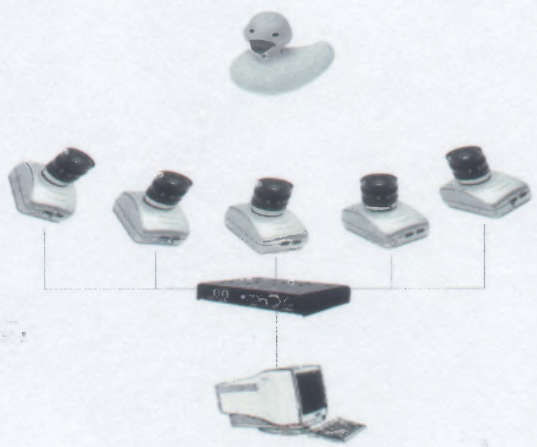


Figure 2.1 Equipment setup.

All software in this thesis is written in MATLAB version 6.0 (Mathworks, MA, USA). Section 2.2 describes the camera calibration software, which was taken from the camera calibration toolbox for Matlab[12]. Section 2.3 describes the rectification process. The rectification program (**rectify.m** and **art.m**) are taken from [25]. The warping program (**Rec_routine_interp.m**, **adjustt.m**, **Warp_interp.m**, **Checkvert.m**) is designed and developed specially for this thesis. Section 2.4 describes the matching process. The matching program is also developed in this thesis. Section 2.5 describes the triangulation calculation. The calculation is done using Microsoft Excel work sheets.

2.2 Camera Calibration

The purpose of the camera calibration is to give the camera parameters described in chapter 1. Camera calibration is not necessary in stereo vision but leads to simpler rectification techniques once the cameras' internal parameters, mutual position and orientation are known. On the other hand, when reconstructing the 3-D shape of objects from stereo, calibration is mandatory in practice, and can be easily achieved [2,13].

To prevent time consuming camera calibration, this thesis uses the camera calibration toolbox for Matlab [12]. This toolbox works on Matlab 5.x (up to Matlab 5.3) and Matlab 6.x on Windows, Unix and Linux systems (platforms it has been fully tested) and does not require any specific Matlab toolbox (for example, the optimization toolbox is not required). The toolbox should also work on any other platform supporting Matlab 5.x and 6.x.

Camera calibration must begin with calibration of each camera as well as each stereo system. The work introduced in this thesis rectifies the image of adjacent cameras, from which pixel correspondences will be matched. In this case, the stereo systems are defined by a pair of cameras, not all five cameras as a whole.

2.2.1 Real Cameras and Pinhole Model

According to the web site, Camera Calibration [15], " Real Cameras deviate from the pinhole model in several respects. First in order to get enough light exposed to the film, the light is gathered across the entire surface of the lens. The most noticeable effect of this is that only a particular surface in space, called the focal plane, will be in perfect focus. In terms of camera calibration, each image point corresponds not to a single ray from the camera center but to a set of rays from across the front of the lens all converging on a particular point on the focal plan. Fortunately, the effects of this area sampling can be negligible by using a suitably small camera aperture. Second is lens distortion or radial distortion. A final and particularly insidious deviation from the pinhole camera model is that the imaged ray does not necessarily intersect at a point. As a result, there need not be a mathematically precise principal point."

2.2.2 Calibration grids

A calibration grid is produced by using an Excel work sheet. This can be achieved by adjusting rows and columns in the Excel worksheet to be equally sized, and manually place black to the blocks in a checkerboard way. The grid has of a block size 15.5x16.8 mm with total of 24x14 blocks. The calibration grid is then printed out with a high

accuracy printer on the paper size 11x17 inches and placed on a flat surface. Attention is paid to the flatness of the calibration grid, which should be very smooth or otherwise will give high error calibration parameters.

Figure 2.2 Calibration grid used in experiment.

2.2.3 Calibration method

Camera calibration may be done on a single camera. The camera calibration toolbox for Matlab can also allows stereo calibration where each pair of cameras are formed to be stereo systems.

Camera calibration starts with taking images of the calibration grid. At least 5 images are need for camera calibration purpose. Normally, 15 images for more accuracy are needed. With more images and varying orientations of the calibration grids, the software is able to locate more corners, which gives less variation in the parameter outputs. Images are stored in the same subdirectory with the camera name followed by the image index. The camera calibration toolbox for Matlab has a graphical user interface to load all images with the same name but different by index. The images are then displayed as shown in Figure 2.3.

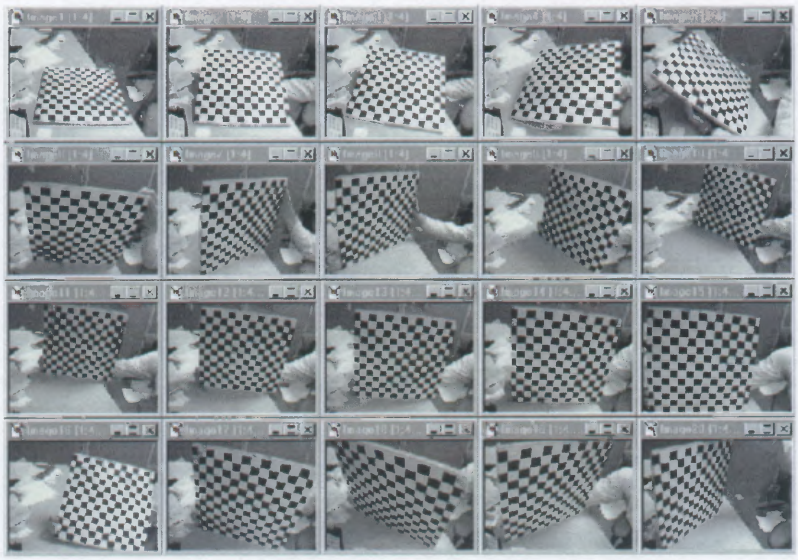


Figure 2.3 Example of camera calibration grid shown by camera calibration toolbox for Matlab.

The next step is extraction of grid corners from all image by setting parameters wintx and winty to a default value for which in this thesis wintx=winty=11. Calibration is begun with manually selecting the corner of the grid to locate the position of origin of the world reference. Because of the nature of this experiment, multiple stereo camera systems, the same grid pattern of the reference frame needs to be consistently selected for the different camera images. For each calibration grid image, the world reference frame is defined by the first click of that image when the corner extraction process is being done. Rotation and translation variables are then defined to be rotation and translation of the optical center of the camera with the world reference frame as shown in Figure 2.4.

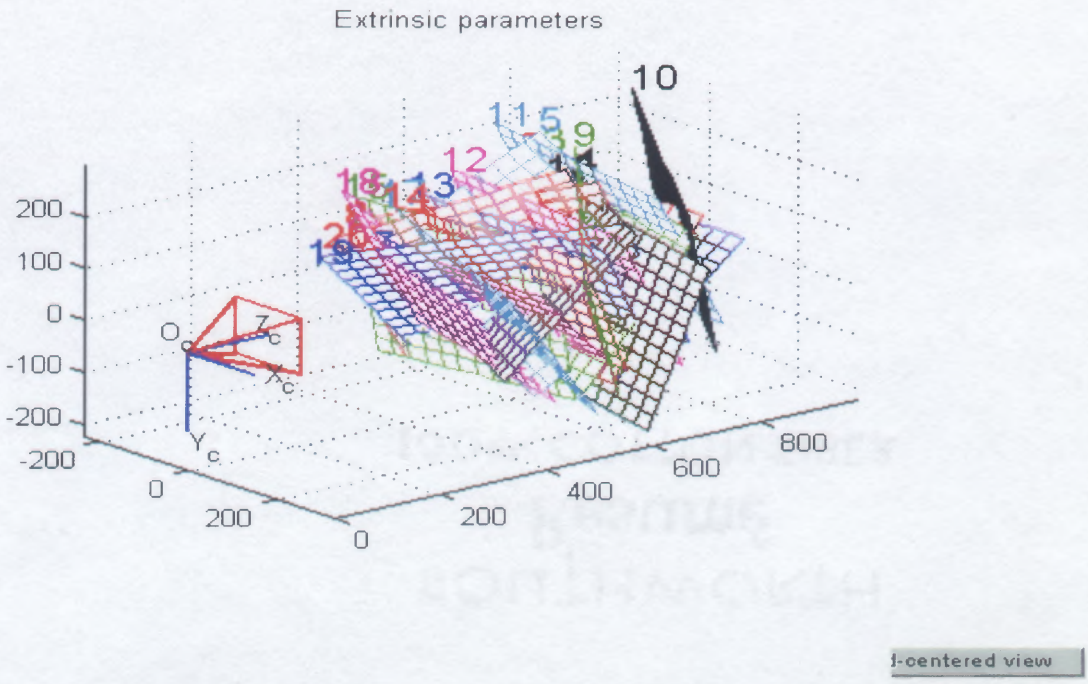


Figure 2.4 Extrinsic parameters, calibration grids and camera reference frame.

The main calibration steps can be done using a software available graphic user interface. After the first calibration, attention should be paid to pixel errors and tolerances of principal points. The calibration tool outputs all calibration parameters including intrinsic parameters such as focal length, principal point, skew factor, distortion, pixel error and extrinsic parameters such as rotation and translation matrices. The idea of good calibration lies in minimization of pixel error, which can be done by controlling the precision of calibration patterns and corner extraction.

In Figure 2.1 the focal lengths from the camera calibration toolbox for Matlab fc(1) and fc(2) are actually focal length times pixel size in the x and y direction. Vector CC gives the position of the principal point. After each camera is calibrated, the stereo system of the two cameras can be computed. This gives rotation and translation matrices

between the two cameras. Stereo calibration can be done by running the stereo_calib script in the camera calibration toolbox for Matlab.

Table 2.1 Camera A. calibration results. (Focal length shown is Focal length by pixel size)

Calibration results (with uncertainties):

Focal Length: fc = [2410.59382 2391.42953] ± [56.69217 60.46237]
Principal point: cc = [500.24104 209.06385] ± [38.38559 17.09149]
Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: kc = [-0.60010 1.68491 0.01623 -0.00452 0.00000] ± [0.04969 0.50549 0.00296 0.00658 0.00000]
Pixel error: err = [0.43780 0.40924]

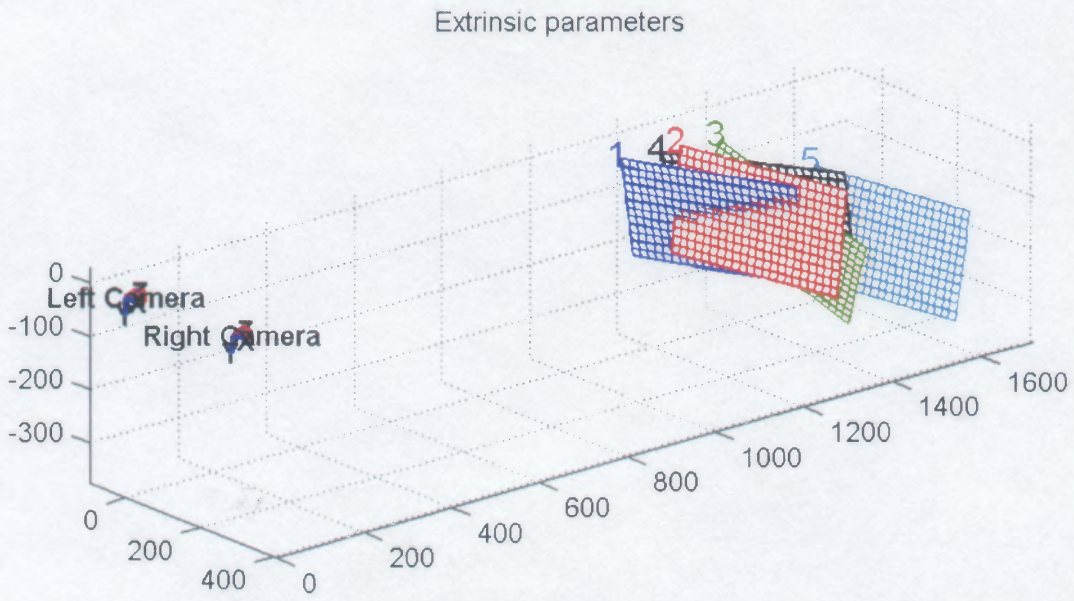


Figure 2.5 Results of stereo calibration, rotation and translation matrices between two cameras are defined.

2.3 Rectification Process

When two stereo images are not taken by the camera with parallel optical axes, the epipolar lines of the both images are not parallel. This means that the one dimensional searching along the epipolar line become difficult as one needs to know the epipolar line equation. Rectification is the method for projecting the entire image position back to the standard stereo diagram plane so that the all epipolar lines become parallel to the image horizontal axis. In this case, the searching for correspondences becomes a one dimension horizontal search, which is much easier to accomplish.

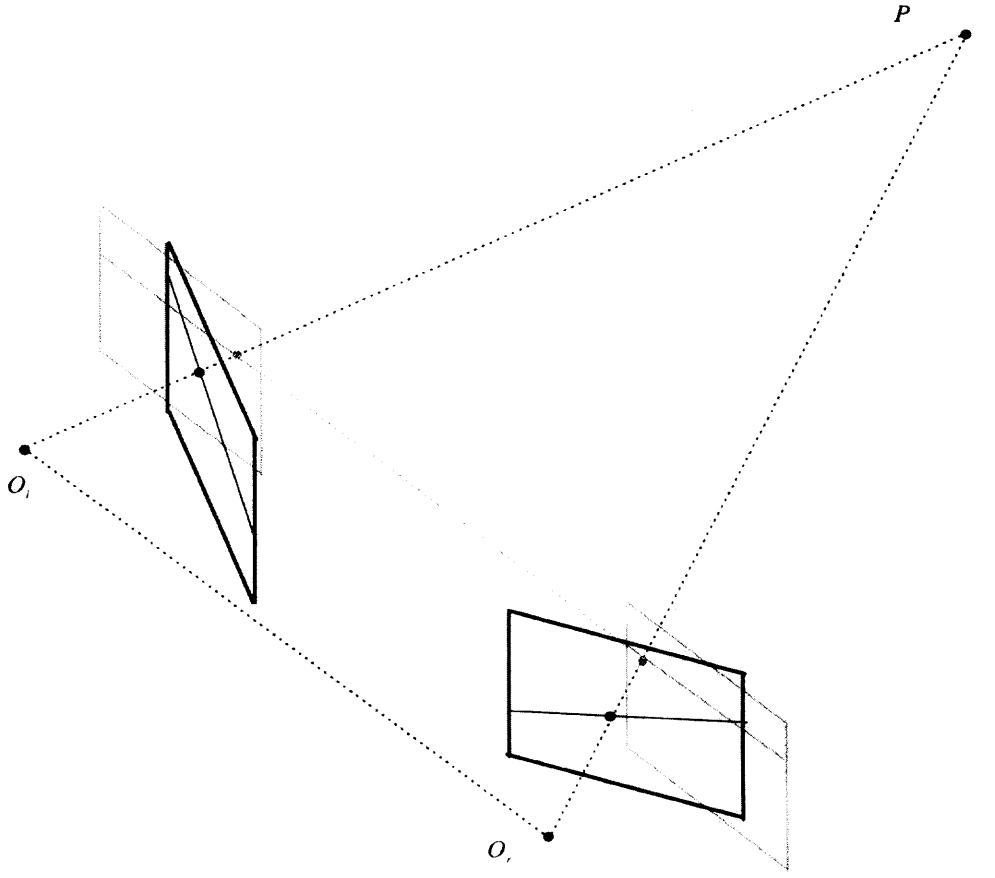


Figure 2.6 Rectification.
(Note: E. Trucco and A. Verri 1998 [7])

2.3.1 Rectified Coordinates

To implement rectification, the technique by A. Fusiello et al. [25] is chosen for implementation. Details for solving the rectifying projection matrices are discussed in [25]. The rectifying program, given from [25], is shown in Appendix A. In this algorithm, what one needs to know as an input is the perspective projective matrices of both cameras. The perspective projective matrix is defined by

$$P = A[R\,|T] \tag{2.1}$$

where A is the intrinsic matrix defined by eq (1.9).

R is the rotation matrix

T is the translation vector.

The algorithm results in four matrices two of which are the new perspective projective matrices and the other two are transformation matrices size 3 by 3 which transform the whole image plane position to the rectified image plane position. Mathematically,

$$\mathbf{M_n} = \mathbf{T}*\mathbf{M_o} \tag{2.2}$$

Where $\mathbf{M_n}$ is the image position in the rectified image plane

$\mathbf{T}$ is the transformation matrix

$\mathbf{M_o}$ is the original position in the original image plane

Note here that all position vectors use a homogeneous coordinate system.

2.3.2 Image Warping

Image positions in the rectified image plane computed using the method descried in section 2.3.1 usually have non-integer values. In order to be useful, the rectified image must be warped or interpolated to integer values. The idea of the program lies in starting with a new image plane and uses the inverse transformation

$$M_o = T^{-1} * M_n \tag{2.3}$$

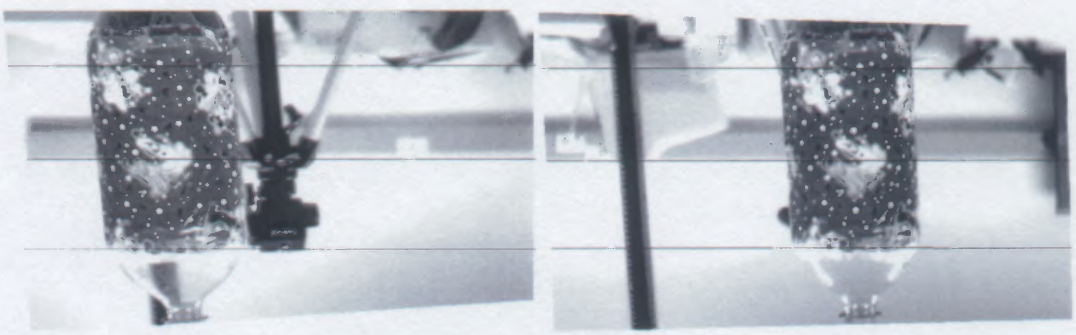
2.3.3 Implementation of Rectification Warping

Implementation is done using the following steps.

1. Pick all four corners of the original image and transform forward to the rectified position
2. Find the size of the rectified image by finding maximum X, minimum X, maximum Y and minimum Y of all four corners. Later these will be refered to as Xmax, Xmin, Ymax, Ymin respectively.
3. From position (Xmin, Ymin) apply the inverse transformation to every position in the rectified image plane to find the position of that point in the original image frame. Usually the position found is not an integer.
4. The X value of the position found and the Y value of the position found are kept in two position matrices. These matrices are then used for two dimension bilinear interpolation. The result of bilinear interpolation is the rectified image.
5. The rectified image and position matrices values are returned. This way one can restore a position of the original image given a position in the rectified image.

Programs written for warping rectified images are named **adjustt.m** and **warp_interp.m**. The whole routine for rectification is named **rec_routine_interp.m** See Appendix A for the detail source code. Figure 2.7 shows the result of the rectification. Detailed rectification results are discussed in section 3.2.

a) Unrectified stereo pair



b) Rectified stereo pair

Figure 2.7 Rectification results a) Original Images b) Rectified Images.

2.3.4 Using the Rectification Program

To use the rectification program introduced in this thesis, it is necessary to have all programs for rectification shown in appendix A in the same folder. The rectification can be done by running **rec_routine_interp.m**. It is necessary to have three files of stereo camera calibration results, left, right and stereo results, in the same folder. The stereo

images taken by stereo system must be in the same folder and manually change in the running **rec_routine_interp.m.**

2.4 Image Matching

After the stereo pairs have been rectified, image matching is done by a Sum of Squared Difference (SSD). A window size of 12x12 pixels is chosen because it offers good matching results while being small enough to give good depth map resolution. Given input position (x,y) in the left image, the program slides a block horizontally along the epipolar line in the right image to find the smallest SSD value. The position found with the smallest SSD within the search range is identified as the correspondence, and the difference between X1 (X position of the image point in the left image) and X2 (X position of the correspondence point in the right image) are called disparity. The program is shown in Appendix A named **testmatch.m**

This algorithm works fine in a textured area where there is no ambiguity. Figure 2.8 shows the program applied to a synthetic stereo pair, with good matching result.

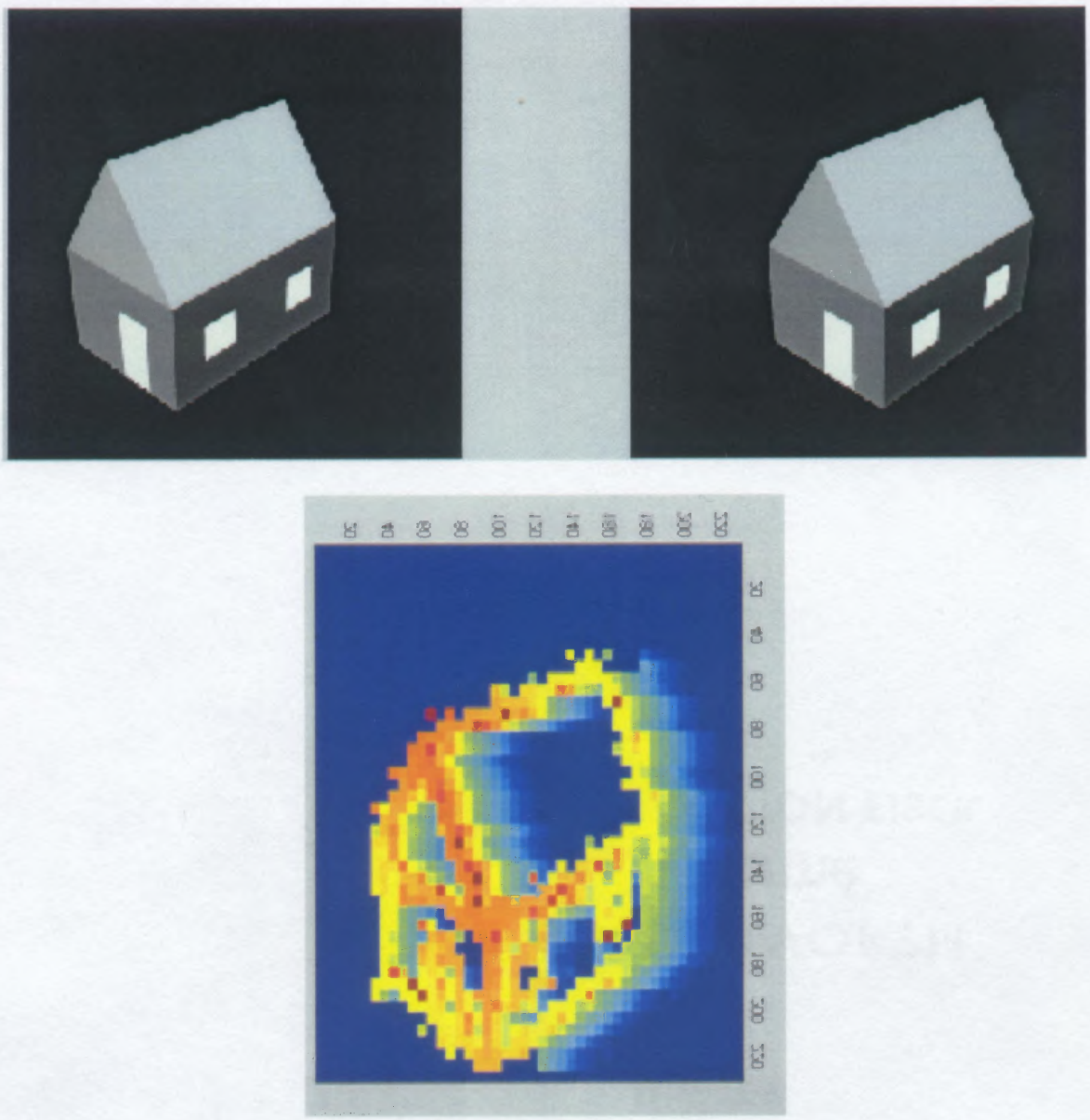


Figure 2.8 Dense disparity map found by application of SSD through synthesis stereo pairs. (Large disparities are shown red)

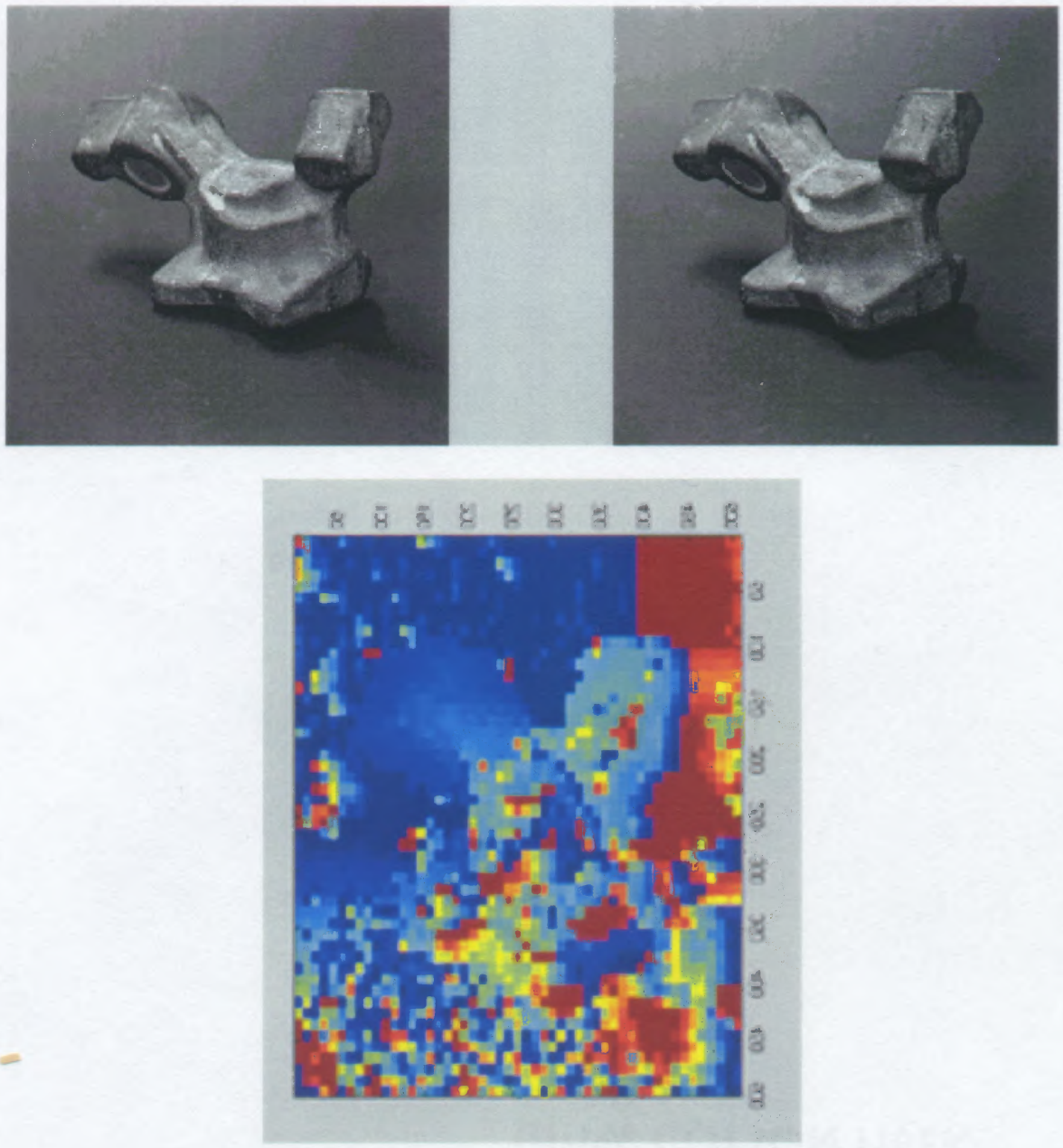


Figure 2.9 Dense disparity map found by application of SSD through a Renault stereo pairs.

Figure 2.9 shows the result of application of the SSD algorithm through a Renault stereo pair, which is commonly used in stereo research [24]. The result is good in the region where the texture is abundant. The SSD algorithm fails in many parts where

texture is not abundant. Figure 2.10 shows the correct intensity match results from figure 2.9.

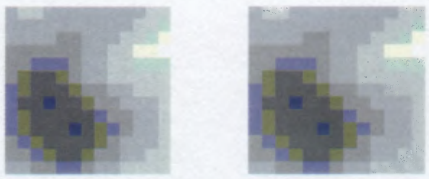


Figure 2.10 Correct intensity match found by searching rectified stereo pair shown in Figure 2.9.

2.5 Triangulation

Triangulation is done by eq (1.14). The angle theta can be found from the rotation matrix between the two cameras obtained from the camera calibration. From [2], the rotation by angle -θ about the Y axis

$$R = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.4)$$

The experiment was mechanically set up in such a way that the rotation between the Z axis and X axis is negligible. The angle θ in eq. (2.4) is approximately $\sin^{-1}[R(1,3)]$ where R(1,3) is the first row and third column of R. Baseline B can be found from the translation vectors. Because of the nature of the experimental setup, the baseline can be approximated by T(1) with sub millimeter precisions.

Triangulation for the proposed method can be done by summing the disparities among camera pairs. This is due to the fact that the rectified plane is the plane in which the epipolar lines become parallel. The pair wise disparities then lie in the same plane and thus may be summed directly.

2.6 Experimental Procedure

After completing the instrumentation and programming, an experimental procedure was conducted. Five cameras were mechanically mounted to an optical table along the same x line. Each camera was equipped with a Rainbow CCTV S16 1.4E-II with a focal length of 16mm. The lenses were set to full aperture.

Camera calibration is needed for all cameras. A camera calibration grid was made using the Excel program. The calibration grid was attached to a smooth surface. Ten images of the calibration grid were taken from different views. The cameras were named from the leftmost camera to rightmost camera, "A", "B", "C", "D", "E" respectively. One should try to minimize the pixel error in camera calibration. When every camera is calibrated, the stereo calibration can be done. Stereo calibration included the stereo pairs "A-B", "A-C", "A-E", "B-C", "C-D", and "D-E".

When stereo calibration is finished, the next step is taking pictures in all view of a subject, which was a cylinder wrapped by gift paper. The subject images are then rectified using calibration results of stereo pair "A-B", "A-C", "A-E", "B-C", "C-D", and "D-E" calibrated earlier.

The traditional matching method employs a direct matching of the A-C pair, A-D pair and A-E pair. To compare with the proposed method, traditional matching is done for six points of interest. The positions in the left image, disparity, and the new position in the right image are noted. Due to bad illumination condition found in camera E, the matching results in camera E are omitted.

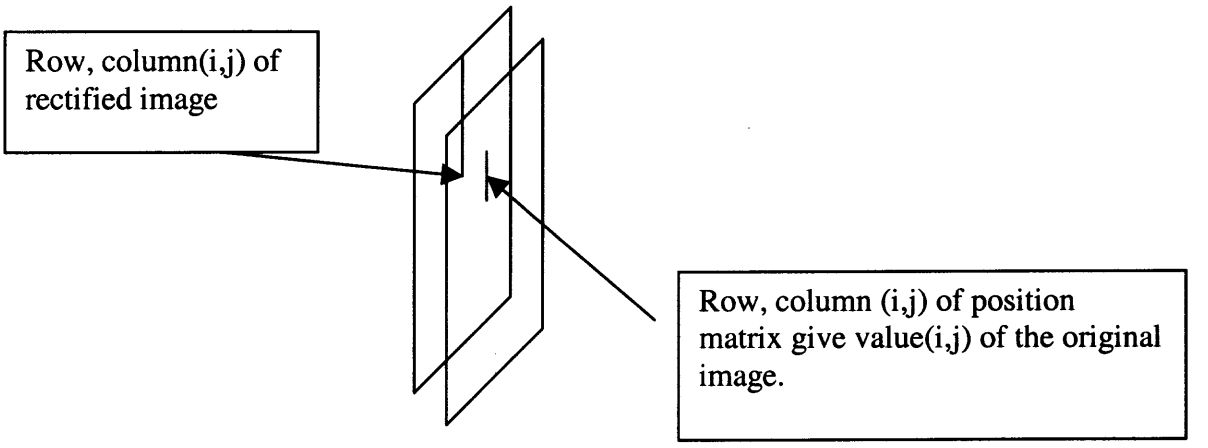


Figure 2.11 Tracing back the original image position given a known rectified image position.

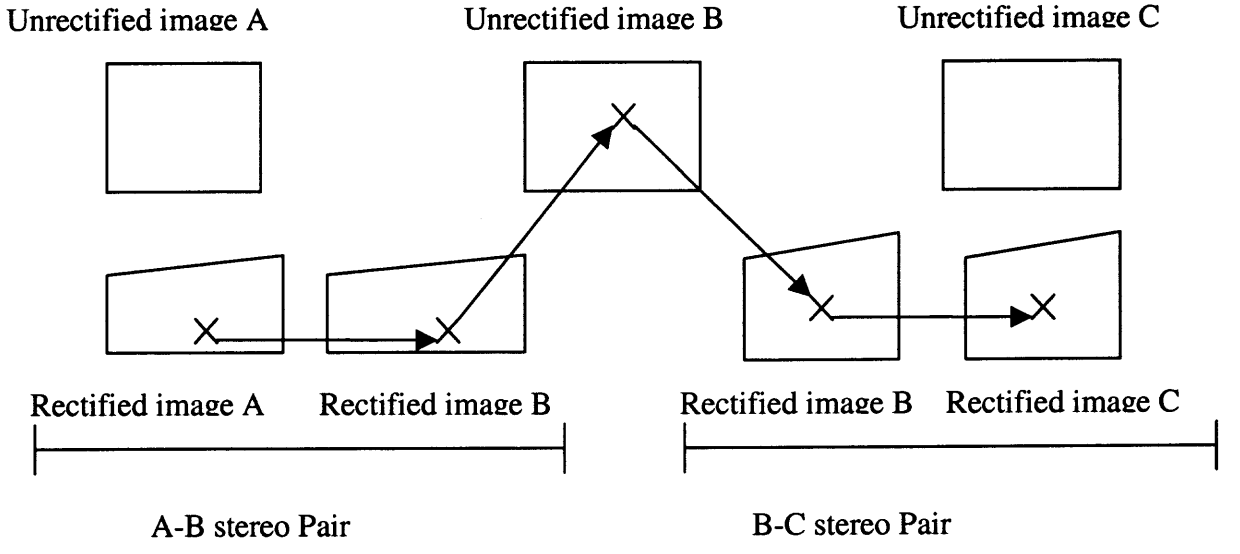


Figure 2.12 Proposed method, propagation of correspondences.

The proposed matching method employs a propagation of correspondences. In Figure 2.12, for example, to find the correspondence in A-C, a position in rectified image A was matched with image B. The position in rectified image A, disparity, and new position in rectified image B are recorded. When rectified, the position matrices of the left and right images are saved. These position matrices are used for tracking the position of the original image given a position in the rectified image. One can track back to the position of the original image B given a known rectified position B, row and column (i,j),

by finding the row and column (i,j) of the position matrices. The positions of the original image B are noted. The positions of the rectified image B in stereo pair B-C are found by using program **lookgo.m**. The program searches for a position (k,l) in rectified image B whose values are (i,j). Value k and l are returned as the position of a point of interest in rectified image B of the stereo pair B-C. SSD matching is done the same way it was done in A-B. The position and disparity of the path of the points are noted.

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Calibration Results

The calibration results are shown and discussed. All camera calibration results are shown in Appendix B. Table 3.1 shows the calibration results for camera A. The calibration results of camera A will be discussed as the stereotype of all camera results.

The focal length fc is actually the focal length divided by pixel size. Pixelink cameras use a 7.5μm pixel size. The actual focal length is therefore fc*7.5μm. In the case of camera A, this equals 18.07mm and 17.9mm for x and y axes respectively. The principal point is the point where the optical axes pass through the image plane. The distortion coefficient kc is expected to approach zero. One can improve pixel error and generate a more correct value of all parameters by reducing associated human error. These errors are smoothness of the calibration grids, and variation in camera position. Extrinsic parameters are more difficult to check. Normally one can check using two methods. The first method is to mechanically measure the distance between the origin point of the world coordinate system and the camera, and verify that distance with the magnitude of the translation vector. The result should be the same. For the case of camera A, the translation vector Tc_1 with the first coordination grid location #1, $[30\ 7\ 1290]^T$, which gives a magnitude of 1.29 m, which is in agreement with the distance measured. The second way is by checking the image with the extrinsic parameter image. The outcome should agree as well.

Table 3.1 Calibration result of camera A.

Focal Length: fc = [2410.59382 2391.42953] ± [56.69217 60.46237]
Principal point: cc = [500.24104 209.06385] ± [38.38559 17.09149]
Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: kc = [-0.60010 1.68491 0.01623 -0.00452 0.00000] ± [0.04969 0.50549 0.00296 0.00658 0.00000]
Pixel error: err = [0.43780 0.40924]

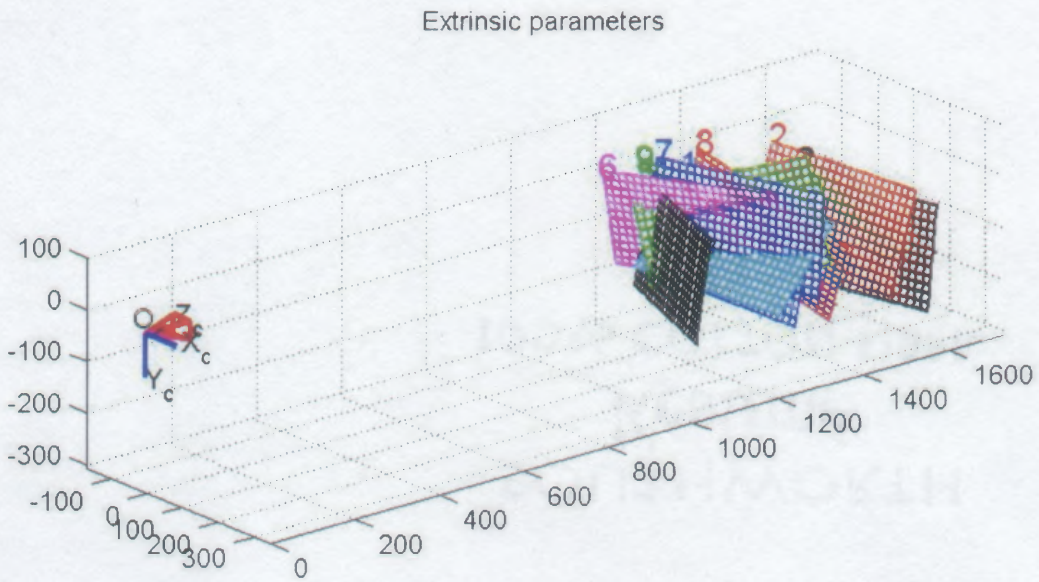


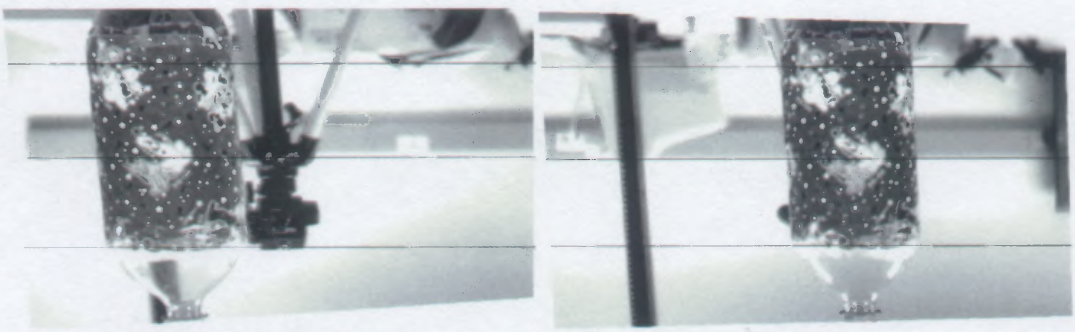
Figure 3.1 Extrinsic parameter for camera A.

The stereo calibration results could also be checked by the same two methods. If they are in agreement, it can be assumed that the results are sufficient for rectification.

3.2 Rectification Result

The rectification result for the A-C stereo system is shown in Figure 3.2. Rectification results of all image pairs are shown in Appendix B. From the rectification program, the image would not align vertically, but the epipolar lines are parallel. This is no surprise since the y coordinates of the principal points in each camera are not equal and there is also variation in the pitch angle. Therefore, there is a mismatch in the y position of the optical centers of the two rectified images. This misalignment is corrected by manually cropping the rectified images. Once that is done, searching for correspondences can be achieved using the horizontally sliding block.

a) Left and right images from camera A-C



b) Rectified stereo pair A-C

Figure 3.2 Original and Rectified image pair. Epipolar line become parallel after rectification. a) Left and right images from camera A-C, b) Rectified stereo pair A-C

The rectification results are not perfect. It is found to have maximum error of 1-2 pixels misalignment. This is due to the uncertainty in the camera calibration result. The misalignment may cause a problem if too small a window size is chosen for SSD matching. The window size is chosen to be 12x12 pixels to keep the maximum error around 10%. The inversion of the rectified images is an artifact of the algorithm, and causes no problem in matching. Therefore, there is no need to change the rectification result.

To use the rectification results with the proposed algorithm, one needs to crop the position matrices, specifying the original position of the unrectified image using the same crop parameters. Refer to Figure 2.12 in section 2.6 for detail of how the position matrices work.

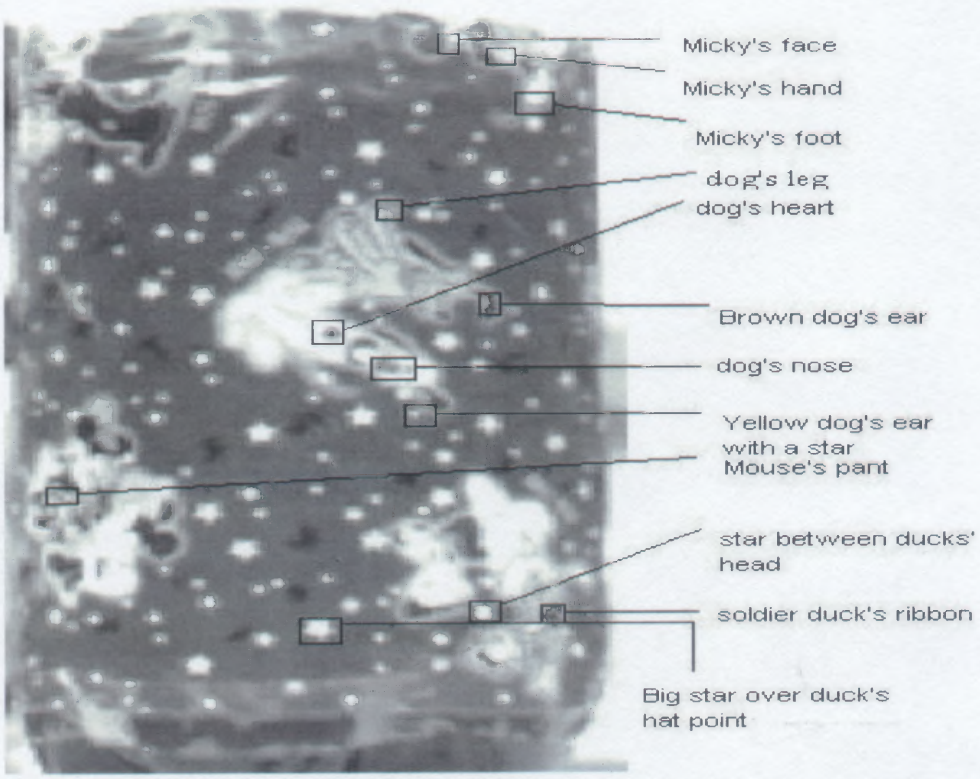
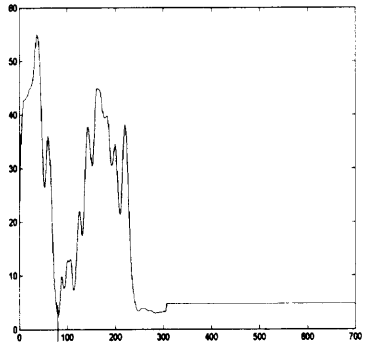


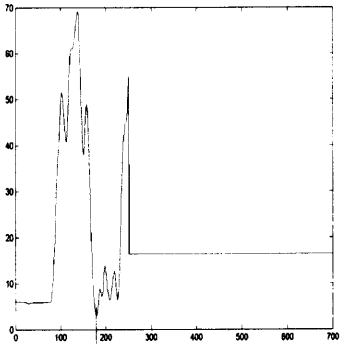
Figure 3.3 Image taken from each camera and the points of interest.

3.3 Matching Results

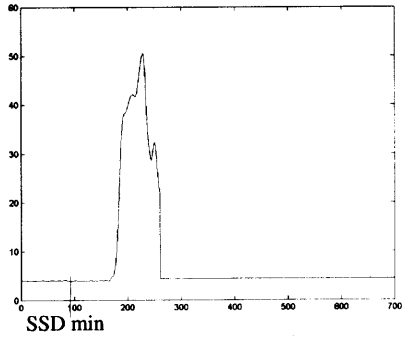
To demonstrate that the proposed algorithm works, the depth dimensions of 11 points on the surface are found using the traditional and proposed methods shown in Figure 3.3. Eleven points where correspondences are easy to find in the A-B stereo pair are chosen. The graphs in Figure 3.4 show matching result for the "Dog nose". In this case, an error occurred in A-C where the lowest SSD yields a incorrect disparity of 94 pixels. The SSD graph shows good matching from A-B and from B-C. The disparity of A-B, A-C reads 82 and 178 pixels respectively. This added up to 260 pixels. The visual approach of the SSD plots supports these findings, sharp, unambiguous SSD minima occur in the A-D and B-C matching. The plot of A-C shows much more ambiguity around its lowest value of SSD.



a) SSD graph for A-B stereo pair



b) SSD graph for B-C



c) SSD graph for A-C

Figure 3.4 SSD value versus candidate disparity for three stereo pairs.

The reconstruction of the propagation proposed method gives the result of 1100.22 mm, while the reconstruction of the matching method is 1683.77 mm. The object is placed at the position around 1100mm-1300 mm. This result will be verified in the reconstruction section. The reason for poor matches in camera A-C is because the image is slightly out of focus in camera C. In this case, the accuracy of matching is degraded when a larger baseline is used. This is called accuracy-precision trade off [8]. The proposed method gives a clearly better result in this case.

Table 3.2 Proposed method A-D disparity results

Position name	Disparity AB	Disparity BC	Disparity CD	Acumulate Disparity
Soldier duck's ribbon	80	176	30	286
Star between ducks' heads	82	#155	#30	267
Big star over duck's hat	82	#89		
Mouse's pant	75	#14		
Dog's heart	81	178	31	290
Dog's nose	82	178	32	292
Dog's leg	81	176	30	287
Brown dog's ear	81	176	30	287
Yellow dog's ear	82	177	30	289
Micky mouse's leg	79	175	29	283
Micky's face	80	176	29	285

In the experiment, camera C is out of focus and caused the matching between camera A and D to have better results than A-C. This is because the image should be large enough to gain the advantage of having many pixels representing small spots in the image. In the future, the lenses, which have smaller focal length, are need. This example shows that the proposed method is more robust with regards to this error.

Table 3.3 Direct matching method A-D disparity results

Position name	Position image A in AD		Disparity AD	Position image D in AD	
	x	y		x	y
Soldier duck's ribbon	134	83	#287	421	83
Star between ducks' heads	151	79	#182	333	79
Big star over duck's hat	214	76	#247	461	76
Mouse's pant	Occluded				
Dog's heart	209	226	300	509	228
Dog's nose	190	210	304	494	210
Dog's leg	189	287	299	488	287
Brown dog's ear	154	235	300	454	235
Yellow dog's ear	181	188	304	485	189
Micky mouse's leg	138	339	294	482	339
Micky's face	173	375	295	468	375

According to Tables 3.2 , 3.3 and the raw data provided in Appendix B, from A to D, the proposed algorithm has the same error as the long baseline method for three points. The proposed algorithm may have a problem in that it propagates some mistakenly matched points. Future work should concentrate on statistical analysis of many points using the proposed method and direct matching. From the matching result, the proposed method is a feasible method of solving correspondences in large baseline using propagation of correspondences.

3.4 Triangulation Results

Table 3.4 demonstrates the Z dimension triangulation results. For full triangulation results see appendix B. The triangulation is done according to the method described in section 2.5. The differences in distance values are found to be camera calibration related. This is because one needs to use the camera calibration results for triangulation. In the

experiment, it is approximated that the rotation is entirely about the Y axis. This assumption is valid in this experiment because the cameras were mechanically attached to the optical table in which case the X axis rotation and Z axis rotation are negligible. According to eq. (1.14) The angle θ in eq. (2.4) is found by $\sin^{-1}(R(1,3))$ where R(1,3) is the element(1,3) in the rotation matrix R.

Table 3.4, shown below, gives the triangulation results. For full reconstruction results see Appendix B. The "#" in front of number indicates that the triangulation is done under the mismatched result. Due to low camera calibration precision, the Z dimension has some error. This case can be prevented by better camera calibration, especially the camera calibration grid, which must be absolutely flat.

Table 3.4 Triangulation results

	Direct Matching Method					Proposed Method	
Position name	AB	AC	AD	BC	CD	AC	AD
	z	z	z	z	z	z	z
Dog's heart	1055.27	1052.39	1276.44	1133.69	1432.63	1102.31	1295.81
Dog's nose	1051.44	1048.59	1268.86	1133.69	1424.68	1100.22	1291.89
Yellow dog's ear	1051.44	#1262.85	1268.86	1138.13	1440.67	1102.31	1297.78
Dog's leg	1055.27	#1683.77	1278.35	1142.61	1440.67	1106.51	1301.74
Brown dog's ear	1055.27	#2045.39	1276.44	1142.61	1440.67	1106.51	1301.74
Soldier duck's ribbon	1059.12	#1604.70	1301.74	1142.61	1440.67	1108.62	1303.72
Micky's face	1059.12	#1299.57	1286.05	1142.61	1448.80	1108.62	1305.72
Micky mouse's leg	1063.01	1060.07	1287.99	1147.13	1448.80	1112.87	1309.72
Mouse's pant		#1069.82					
Star between ducks' heads	1051.44	#1171.19					
Big star over duck's hat	1051.44	#1273.92					

In Figure 3.5, it was shown that the result matching between A-C was not accurate due to the blurred image of C. Large baseline tends to give error for a variety of reasons. In this case, the proposed algorithm does work better as shown by triangulation

result for A-C. It is because of the matching accuracy that the proposed algorithm gives this precision. The object in this case was placed at 1.100 meters from camera C.

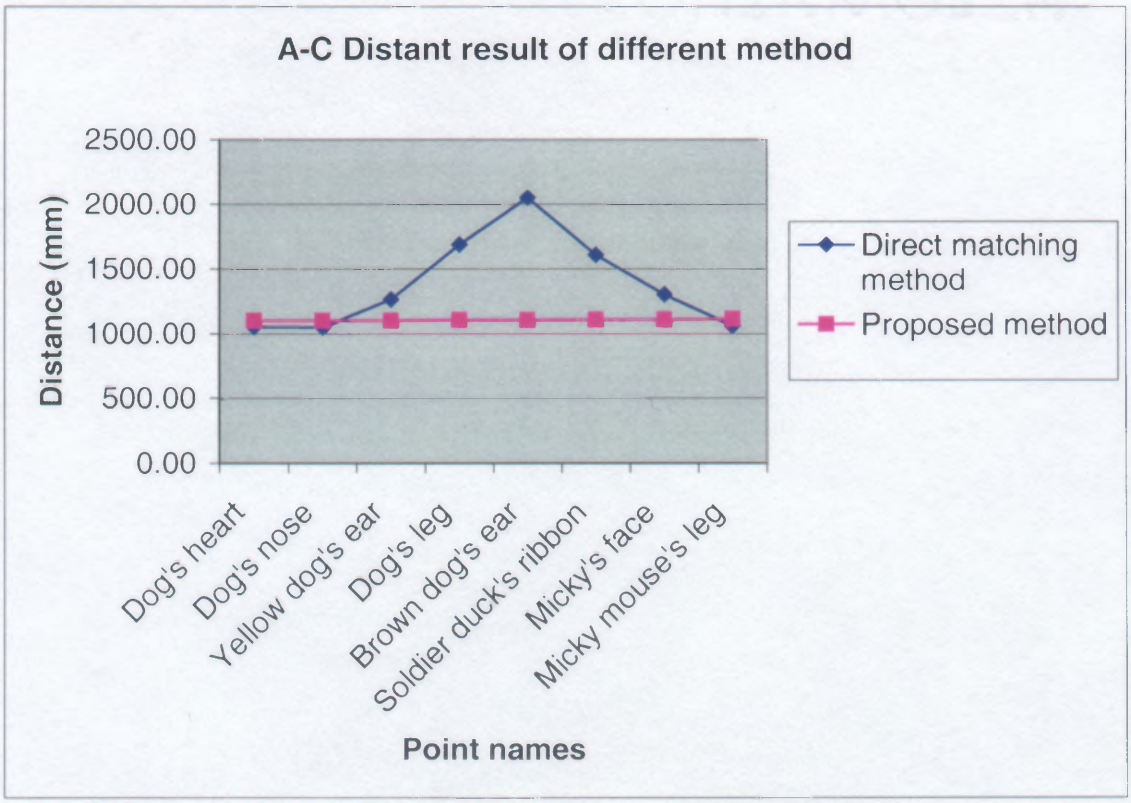


Figure 3.5 Proposed method gives superior result dealing with image blurred.

If the large baseline itself gives error due to a variety of reasons, the smaller baseline has its own problem. Though the accuracy of matching is high, the resolutions on measuring depth are low. This is demonstrated in Figure 3.6. The small baseline offers poor resolution. Taking C-D for example, there are step distances of four sizes. A-B, likewise, has a step distance of five sizes. The resolution for small baseline, in this experiment, is 3.83 mm/pixel, while the resolution of the large baseline A-D is 1.91 mm/pixel. From the experiment, the A-D matching results turn out to be good as demonstrated by Figure 3.7.

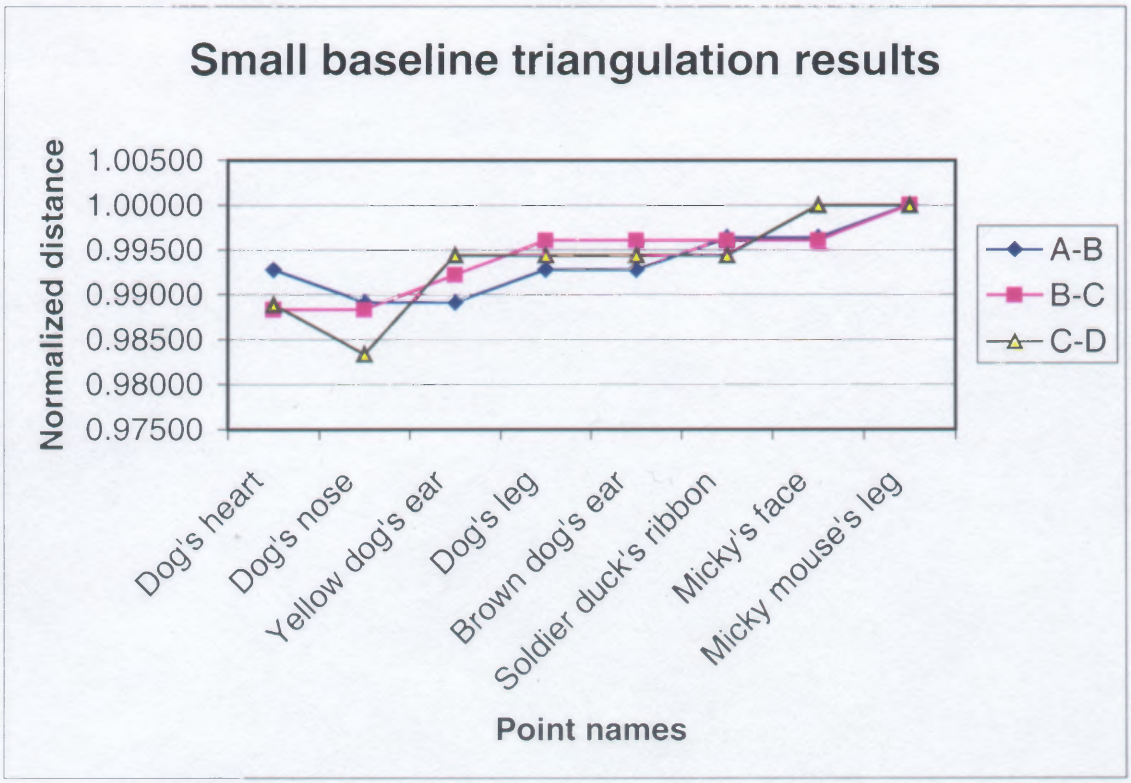


Figure 3.6 Small baseline triangulation results.

In the case in which the matching can be done correctly from A to D, there is no reason to use the proposed algorithm. However, it is likely that the larger baseline will results in correspondence matching error [8]. A-C is a good example of this situation where most of the matching using the direct matching method gives mostly incorrect results, while the proposed method gives consistently good results.

Figure 3.7 shows good triangulation results given by both the traditional direct matching and proposed methods. The correspondence found for the point named "Soldier duck's ribbon" in the direct matching method is mismatched and causes the resulting incorrect depth measurement at this point. The proposed method shows no mismatch in this figure, and gives better results.

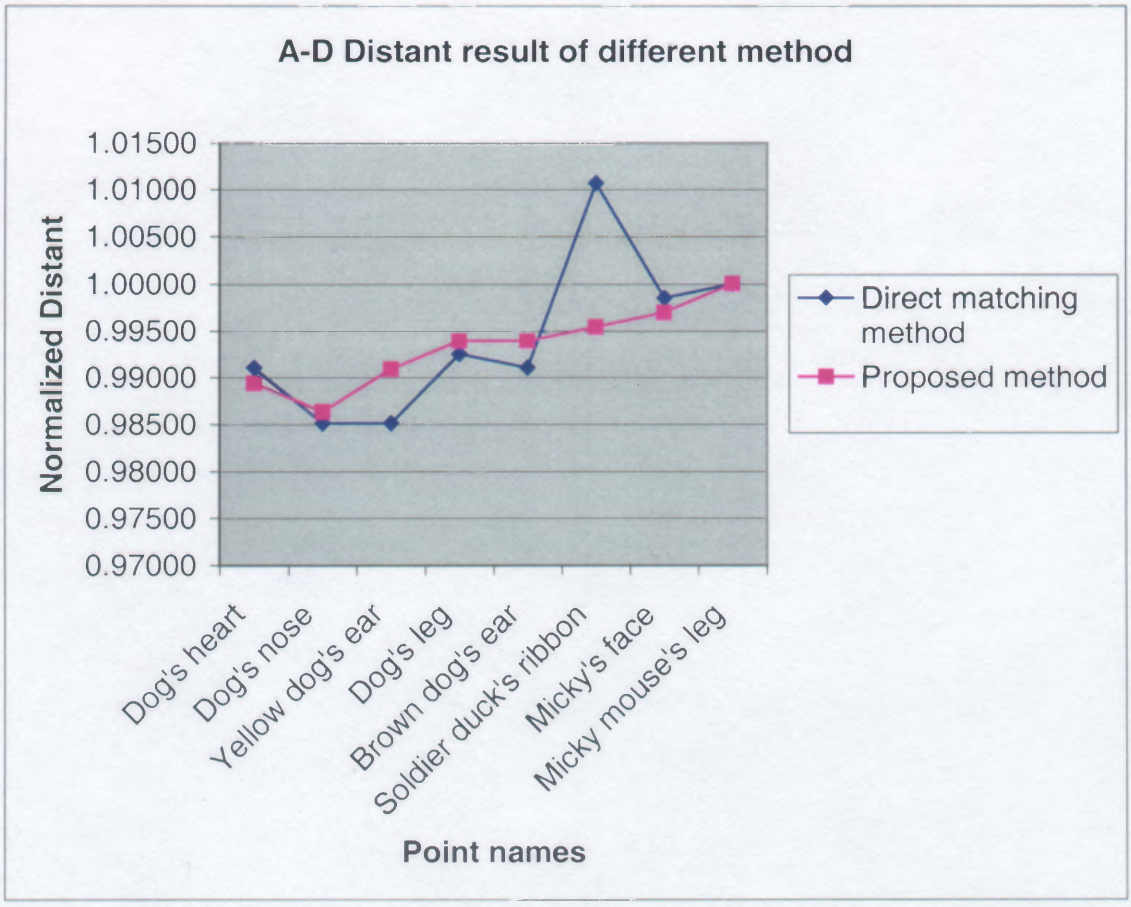


Figure 3.7 The precision of A-D is higher than A-B.

Figure 3.8 shows the triangulation results as normalized distances. Because the target points lie on the surface of the cylinder, the maximum difference in depth dimension among them is approximately 25 mm, as measured mechanically. Since the cylinder is placed at the position 1100 mm from camera C, the center camera in the array, it is no surprise that the normalized distances of the eight points appear to be near unity in this figure. It also gives an experimentally determined maximum difference in depth dimension of 19.14 mm, which very close to the measured value of 25 mm. The proposed method has better resolution than when using a small baseline, and gives better correspondence matching than the A-D direct matching. The proposed method also gives better resolution than with a small baseline direct matching method. The depth resolution

with a small base line is approximately 3.8 mm/pixel, and the proposed method resolution is 1.0 mm/pixel. These results confirm the main premise of this thesis.

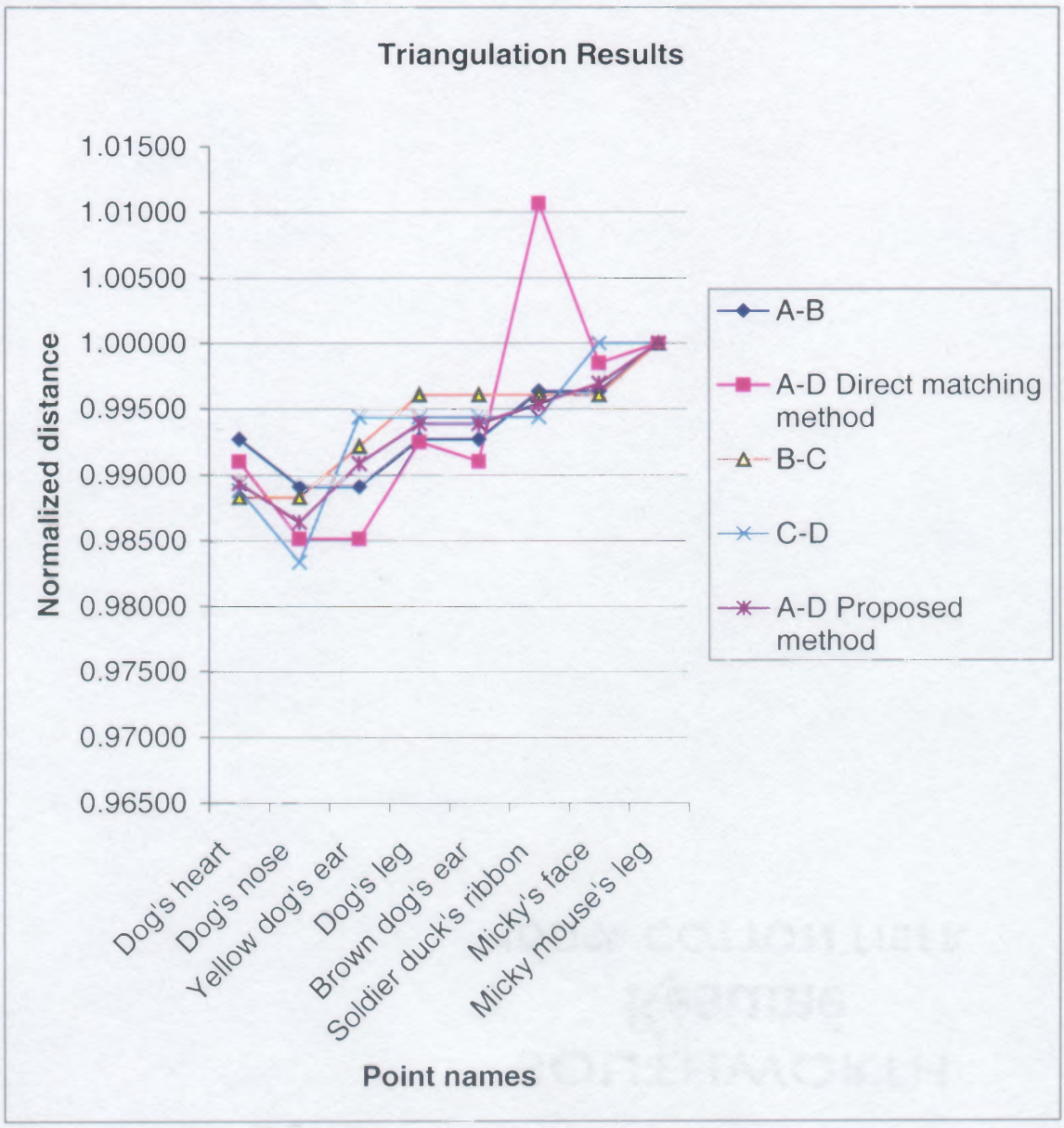


Figure 3.8 Triangulation result.

CHAPTER 4

CONCLUSIONS

The precision of the proposed method can equal that of large baseline, if the correspondences are matched correctly in both methods. The accuracy of matching in the proposed method in this experiment is better than the direct matching method. In the case of four cameras, A to D, the direct method resulted in a major error in a correspondence of one point, resulting in a substantial depth calculation. The proposed method of propagating correspondences determined from multiple small baselines minimizes this kind of error.

In the case of the slightly out of focus image of camera C, the proposed method is more robust than the direct matching method, which fails to achieve good matching. Due to the nature of summing disparities, the proposed method gives the results as if the object is viewed from four cameras simultaneously. The proposed method gives better distance resolution than the small baseline, while it enjoys the advantage of accurate matching that is typical of a small baseline. When compared to the large baseline, it enjoys the same advantage of more accurate depth resolution, while keeping correspondences matching accuracy high. The accuracy-precision tradeoff problem is resolved by the proposed method.

Future work requires a strategy of how one can precisely distinguish the high confidence corresponding points and use them to generate sparse representation of biological surfaces. Enhancement of the camera calibration precision, improvement of target textures, inclusion of camera rotation among x axes, and use of a more accurate triangulation algorithm are recommended for future continuation of this work.

APPENDIX A

PROGRAM USED IN THESIS

Rectification process

Program named **art.m** and **rectification.m** are found from A. Fusiello, E. Trucco, and A. Verri [25]. Program **rec_routine_interp.m** is the main routine, which calls **warp_interp.m** and **adjustt.m** for rectification. **Checkvert.m** check the rectification result manually to see if both image are allgned up.

art.m

```
function [A,R,t] = art(P)
 % ART: factorize a PPM as  P=A*[R;t]

 Q = inv(P(1:3, 1:3));
 [U,B] = qr(Q);

 R = inv(U);
 t = B*P(1:3,4);
 A = inv(B);
 A = A ./A(3,3);
```

rectification.m

```
% This is the rectification function


 function [T1,T2,Pn1,Pn2] = rectify(Po1,Po2,Xoff,Yoff)

 % RECTIFY: compute rectification matrices

 % factorize old PPMs
 [A1,R1,t1] = art(Po1);
 [A2,R2,t2] = art(Po2);

 % optical centers (unchanged)
 c1 = - inv(Po1(:,1:3))*Po1(:,4);
 c2 = - inv(Po2(:,1:3))*Po2(:,4);

 % new x axis (= direction of the baseline)
 v1 = (c1-c2);
```

```
% new y axes (orthogonal to new x and old z)
v2 = cross(R1(3,:)',v1);
% new z axes (orthogonal to baseline and y)
v3 = cross(v1,v2);

% new extrinsic parameters
R = [v1'/norm(v1)
   v2'/norm(v2)
   v3'/norm(v3)];
% translation is left unchanged

% new intrinsic parameters (arbitrary)
A = (A1 + A2)./2;
A(1,2)=0; % no skew
A(1,3)=A(1,3)-Xoff;
A(2,3)=A(2,3)-Yoff;

% new projection matrices
Pn1 = A * [R -R*c1 ];
Pn2 = A * [R -R*c2 ];

% rectifying image transformation
T1 = Pn1(1:3,1:3)* inv(Po1(1:3,1:3));
T2 = Pn2(1:3,1:3)* inv(Po2(1:3,1:3));
```

Rec_routine_interp.m

```
% Main Rectification Routine.

load calib_Results_left
QRc_left=Rc_1;
QTc_left=Tc_1;

load calib_Results_right
QRc_right=Rc_1;
QTc_right=Tc_1;

load calib_results_stereo
QKK_left=KK_left;
QKK_right=KK_right;

clear A*;clear B*;clear C*;clear D*;clear E*;clear F*;clear G*;clear H*;clear I*;
clear J*;clear K*;clear L*;clear M*;clear N*;clear O*;clear P*;clear R*;clear S*;
clear T*;clear U*;clear V*;clear W*;clear X*;clear Y*;clear Z*;clear a*;clear b*;
clear c*;clear d*;clear e*;clear f*;clear g*;clear h*;clear i*;clear j*;clear k*;
clear l*;clear m*;clear n*;clear o*;clear p*;clear q*;clear r*;clear s*;clear t*;
```

```
clear u*;clear v*;clear w*;clear x*;clear y*;clear z*;
PPM_left=QKK_left*[QRc_left QTc_left];
PPM_right=QKK_right*[QRc_right QTc_right];

%PPM_right=[9.767e+2  5.376e+1  -2.4e+2  4.003e+4;9.868e+1  9.31e+2  1.567e+2
2.517e+5;5.766e-1 1.141e-1 8.089e-1 1.174e+3]
%PPM_left=[9.765e+2  5.382e+1  -2.398e+2  3.875e+5;9.849e+1  9.333e+2  1.574e+2
2.428e+5;5.790e-1 1.108e-1 8.077e-1 1.118e+3]
leftcornx=200;leftcorny=250;spanx=800;spany=500;

[TL,TR,c,d]=rectification(PPM_left,PPM_right,0,0) % First rectification
Imlef=imread('dCoke22','tif');%imshow(Imlef);
Imleft=imcrop(Imlef(:,:,1),[leftcornx leftcorny spanx spany]);
Imrigh=imread('eCoke22','tif');%figure;imshow(Imrigh);
Imright=imcrop(Imrigh(:,:,1),[leftcornx leftcorny spanx spany]);


[Xoff_L,Yoff_L,Xmin_L,Ymin_L,Xmax_L,Ymax_L]=adjustt(TL,Imleft,leftcornx,leftcor
ny,spanx,spany);
[RecIm_L,PxL,PyL]=warp_interp(TL,Xmin_L,Ymin_L,Xmax_L,Ymax_L,Imleft,leftcor
nx,leftcorny);

[Xoff_R,Yoff_R,Xmin_R,Ymin_R,Xmax_R,Ymax_R]=adjustt(TR,Imright,leftcornx,left
corny,spanx,spany);
[RecIm_R,PxR,PyR]=warp_interp(TR,Xmin_R,Ymin_R,Xmax_R,Ymax_R,Imright,leftc
ornx,leftcorny);

RecIm_L=mat2gray(RecIm_L);
RecIm_R=mat2gray(RecIm_R);
FlipRec_L=fliplr(rot90(RecIm_L,3));FlipRec_R=fliplr(rot90(RecIm_R,3));
imshow(Imleft);figure;imshow(Imright);figure;imshow(RecIm_L);figure;imshow(RecIm
_R);figure;imshow(FlipRec_L);figure;imshow(FlipRec_R);

save deCoke22recIm;
%checkvert(FlipRec_L,FlipRec_R);
```

adjustt.m

```
% Function Adjust to give image position offset to rectify.m so that the
% image center is in the center and to warp_interp for postioning.
% Input to this function is T1 and the image.

function
[Xoff,Yoff,Xmin,Ymin,Xmax,Ymax]=adjustt(T,ImageIn,leftcornx,leftcorny,spanx,spany
)
```

```
Imsize=size(ImageIn);
a=T*[leftcornx leftcorny 1]';
b=T*[leftcornx+spanx leftcorny 1]';
c=T*[leftcornx leftcorny+spany 1]';
d=T*[leftcornx+spanx leftcorny+spany 1]';
Imedge(1,:)=(a./a(3))';
Imedge(2,:)=(b./b(3))';
Imedge(3,:)=(c./c(3))';
Imedge(4,:)=(d./d(3))';
Xmax=max(Imedge(:,1));
Ymax=max(Imedge(:,2));
Xmin=min(Imedge(:,1));
Ymin=min(Imedge(:,2));
ImCenter=[(Xmax+Xmin)/2 (Ymax+Ymin)/2];
Xoff=ImCenter(1)-512;
Yoff=ImCenter(2)-620;
```

Warp_interp.m

This function computes warping to warp image

```
function
[RecIm,Px,Py]=warp_interp(T,Xmin,Ymin,Xmax,Ymax,ImageOrig,leftcornx,leftcorny)

ImSize=size(ImageOrig);
ImageOrig=double(ImageOrig);
%xorg=[1:ImSize(2)];
%yorg=[1:ImSize(1)];
RecIm=zeros(Ymax,Xmax);
Tinv=inv(T);
XminIN=round(Xmin);
YminIN=round(Ymin);
XmaxIN=round(Xmax);
YmaxIN=round(Ymax);
icounter=0;jcounter=0;%Px=zeros((XmaxIN-XminIN),(YmaxIN-
YminIN));Py=zeros((XmaxIN-XminIN),(YmaxIN-YminIN));
for j=YminIN:1:YmaxIN
   jcounter=jcounter+1;
   icounter=0;
   for i=XminIN:1:XmaxIN
      i
      j
      icounter=icounter+1;
      a=Tinv*[i j 1]'
      P=double(a./a(3))
      Px(icounter,jcounter)=P(1);
```

```
      Py(icounter,jcounter)=P(2) ;
   end
end
Px=Px-leftcornx;
Py=Py-leftcorny;
size(Px)
size(Py)
RecIm=mat2gray(interp2(ImageOrig,Px,Py));

imshow(RecIm);title('RecIm from warp function');
Px
Py
```

Checkvert.m

```
function ver_mis=checkvert(Il,Ir)
figure;imshow(Il)
[n1,m1]=ginput(1)
figure;imshow(Ir);
[n2,m2]=ginput(1)
disans=n2-n1
ver_mis=m2-m1
K=imcrop(Il,[n1 m1 30 30]);
%figure;imshow(K)
L=imcrop(Ir,[n2 m2 30 30]);
%figure;imshow(L)
```

Matching Process

Matching is done by **testmatch.m** which calls function **SSDto.m**. Program named **matching.m** is used to create a dense disparity map. Lookgo.m helps search position of a point in the rectified image given a known value position in the unrectified image. **Checkdis.m** and **markerpointpoint.m** are used to check the result of matching. **Checkdis.m** allows a user to select 2 points manually and see the disparity while **markerpointpoint.m** needs position (x,y) of a point and displays that point in the image.

Testmatch.m

```
% Match only one block

load acCoke22RecIm;
BothSize=[870;532];
imL=imcrop(FlipRec_Lc,[0 0 BothSize']);
imLsize=size(imL);
imR=imcrop(FlipRec_Rc,[0 0 BothSize']);

wsize=[12 12];dcanmax=700;blockindex=0;MinSSD=10000000;icounter=0;
imshow(imL);
[i j]=ginput(1);
%i=214;j=76;
      K=imcrop(imL,[i j wsize(1) wsize(2)]);
      uplefpos=[i j];
      Dis=0;
      MinSSD=SSDto(imL,imR,wsize,BothSize,uplefpos,Dis);

      blockindex=blockindex+1;

         for dcan=0:dcanmax
         SSD_ij=SSDto(imL,imR,wsize,BothSize,uplefpos,dcan);

         if SSD_ij<MinSSD
         MinSSD=SSD_ij;
         Dis=dcan;
         DisPos=[i+dcan j];
         end
      end
L=imcrop(imR,[i+Dis j wsize(1) wsize(2)]);

figure;imshow(K);figure;imshow(L);figure;imshow(imR)
DisPos
Dis
i
j
PxR(DisPos(1),DisPos(2))
PyR(DisPos(1),DisPos(2))
```

Matching.m

```
% This is matching program for matching left and right images with the goal to generate
% disparity map.The stereo pairs used are expected to be calibrated and rectified.

filel='c:\Documents and settings\peerajak\My Documents\Stereo_database\renault\left.jpg';
filer='c:\Documents and settings\peerajak\My Documents\Stereo_database\renault\right.jpg';

surl='jpg';
surr='jpg';
imL=imread(filel,surl);
imLsize=size(imL);
imR=imread(filer,surr);
imRsize=size(imR);
%subplot(1,2,1);imshow(imL);subplot(1,2,2);imshow(imR);
wsize=[4 4];dcanmax=120;blockindex=0;MinSSD=10000000;icounter=0;
fti=[1 wsize(1) 232];ftj=[1 wsize(2) 232];
imshow(imcrop(imL,[fti(1) ftj(1) fti(3)-fti(1) ftj(3)-ftj(1)]));
for i=fti(1):fti(2):fti(3);
  jcounter=0;
  icounter=icounter+1;
   for j=ftj(1):ftj(2):ftj(3);
    jcounter=jcounter+1;
    uplefpos=[i j];
    MinSSD=SSDto(filel,surl,filer,surr,wsize,imRsize,uplefpos,0);
   Dis=0;

for dcan=1:dcanmax
  SSD_ij=SSDto(filel,surl,filer,surr,wsize,imRsize,uplefpos,dcan);
  if SSD_ij<MinSSD
    MinSSD=SSD_ij
    if MinSSD>=10000000
      Dis=0;
      else
        Dis=dcan
    end
    i
    j
  %else if SSD_ij==MinSSD
  %     Dis=dcan
  %     Hx='Anbiguity at x='
  %     i
  %     Hy='Anbiguity at y='
  %     j
```

```
  %  end
  end
end
%if Dis>160
 %  Dis=0
%end

blockdis=Dis*ones(wsize);
DisMap((icounter-1)*wsize(1)+1:icounter*wsize(1),(jcounter-
1)*wsize(2)+1:jcounter*wsize(1))=blockdis;
end
end
figure;mesh(DisMap);
figure;image(DisMap);
```

SSDto.m

```
% This is function SSD in 2D. Given upper left position vector(x,y) in left image,
% candidate disparity d and window size vector W(1x2),the function will calculate the
% and return sum of square different value (scalar)

function y=SSD(LeftIm,RightIm,winsize,Rpicsize,upleft,disp);
Fl=double(imcrop(LeftIm,[upleft(1) upleft(2) winsize(1)-1 winsize(2)-1]));

if upleft(1)<=Rpicsize(2)-disp-winsize(1)-10

  upleft_R(1)=upleft(1)+disp;
  upleft_R(2)=upleft(2);
else

 upleft_R(1)=15;
  %upleft_R(1)=Rpicsize(1)-disp-winsize(1)-10;
  upleft_R(2)=upleft(2);
end
Fr=double(imcrop(RightIm,[upleft_R(1) upleft_R(2) winsize(1)-1 winsize(2)-1]));
%disp

sum_sq_diff=sum(sum((Fl-Fr).^2));
y=sum_sq_diff;
```

lookgo.m

```
% Program to find back the position[i,j] given value
load deCoke22RecIm
PxLsize=size(PxL);
```

```
for j=1:PxLsize(2)-1
   for i=1:PxLsize(1)-1
      if PxL(i,j)>298 & PyL(i,j)>442 & PxL(i,j)<299 & PyL(i,j)<443
         %if PxL(j,i)=607
            PXLi=i;
            PXLj=j;
            % end
      end
   end
end
PXLi
PXLj
```

Checkdis.m

```
figure(1);
[n1,m1]=ginput(1);
figure(2);
[n2,m2]=ginput(1);
disans=n2-n1
K=imcrop(imL,[n1 m1 30 30]);
figure;imshow(K)
L=imcrop(imR,[n2 m2 30 30]);
figure;imshow(L)
```

markerpointpoint.m

```
load adCoke22RecIm
MarkerL=zeros(size(FlipRec_Lc));
j=214;i=76;
MarkerL(i,j)=1;
      MarkerL(i-1,j)=1;
      MarkerL(i+1,j+1)=1;
      MarkerL(i-1,j+1)=1;
      MarkerL(i+1,j-1)=1;
LeftImpose=imimposemin(FlipRec_Lc,MarkerL);
figure;imshow(LeftImpose);
figure;imshow(FlipRec_Lc);
```

APPENDIX B

RAW DATA

Appendix B shows raw data used in this thesis. Figure B.1 shows the target image and points of interest. Figure B.2 and B.3 show the target viewed by different camera and rectification results, respectively. Table B.1, B.2, and B.3 show positions and disparities of various stereo pairs. Table B.4 shows parameters used for triangulation. Table B.5 – B.9 show triangulation results. Table B.10 – B.15 show camera calibration results.

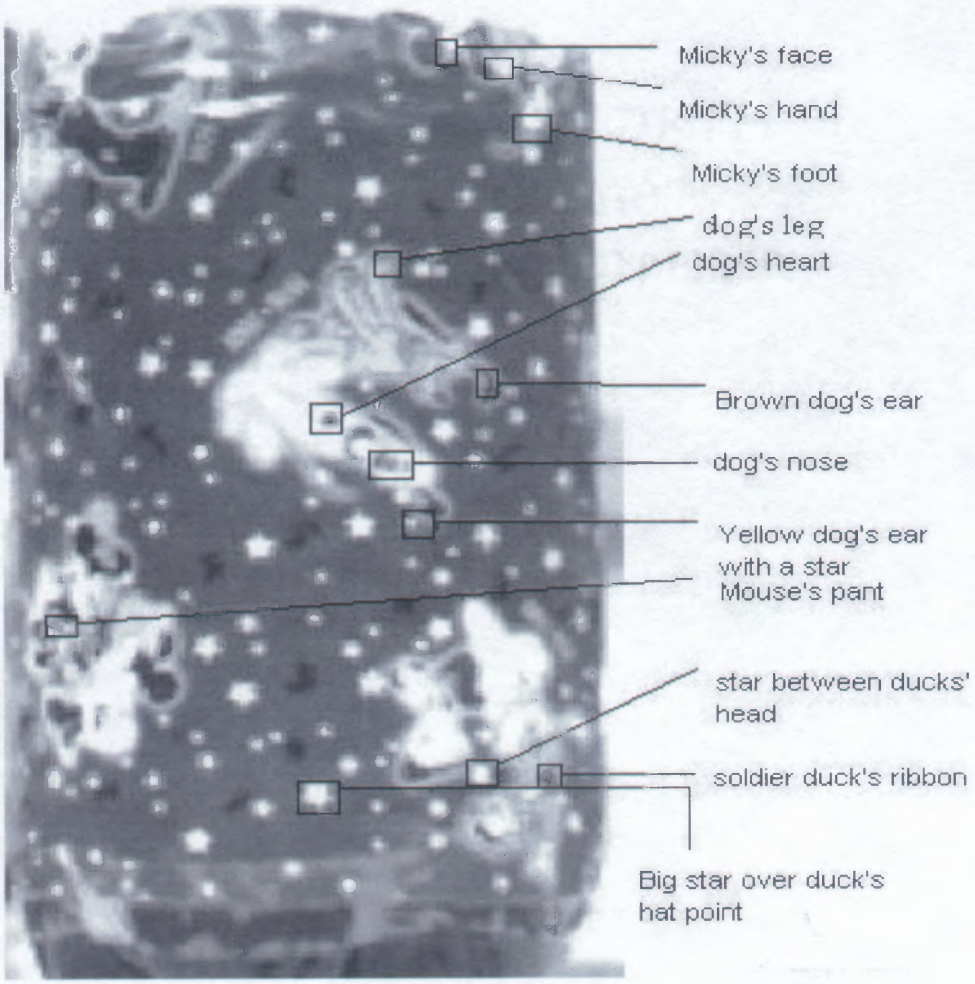
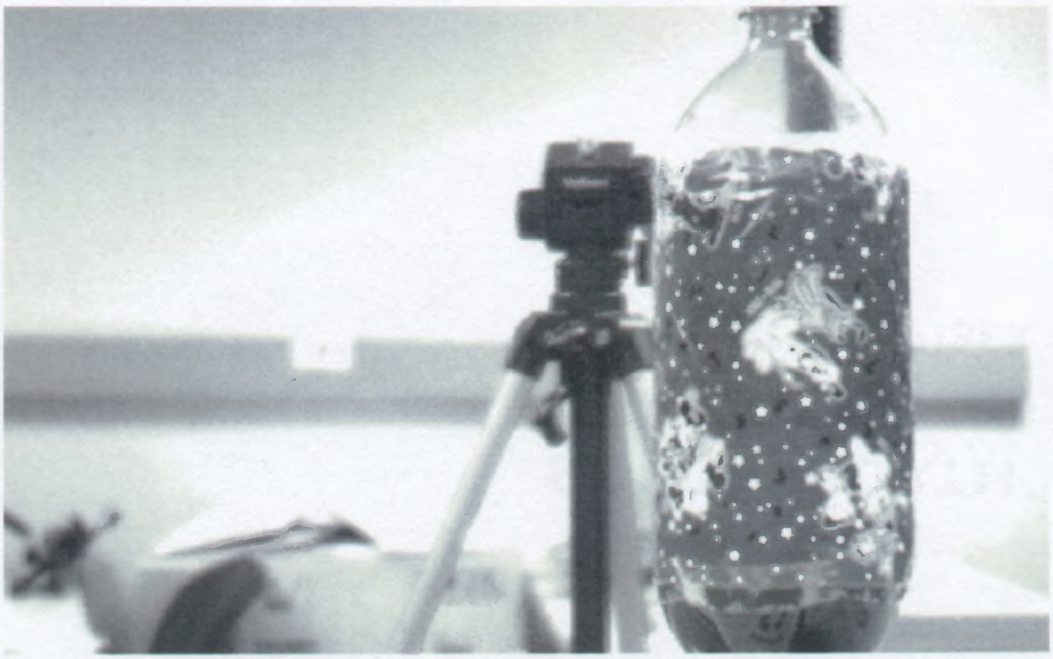


Figure B.1 Image taken from each camera and the points of interest.



a) Image taken from camera A, the left most camera

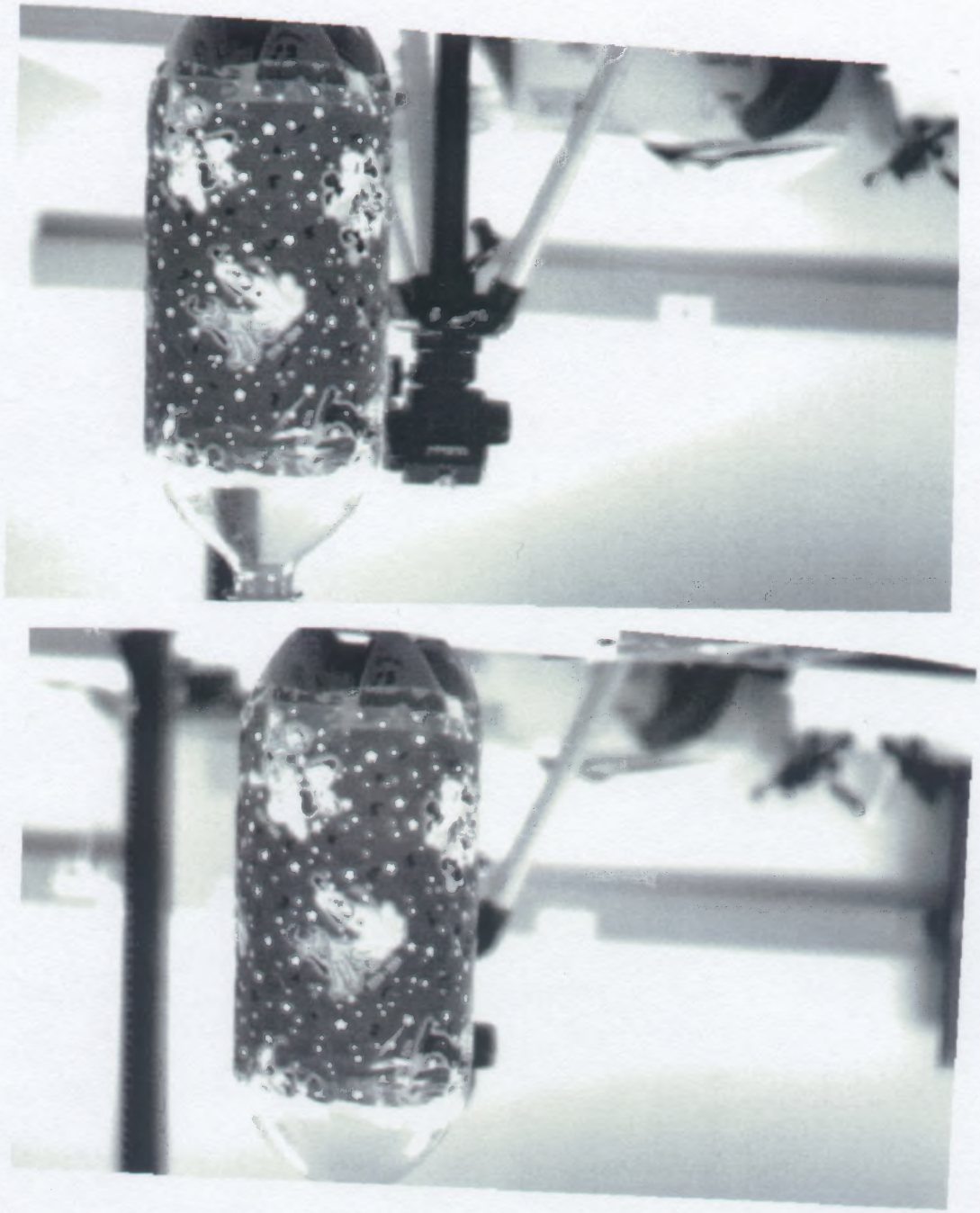
b) image taken from Camera B

Figure B.2 Image taken from all views.

c) image taken from Camera C

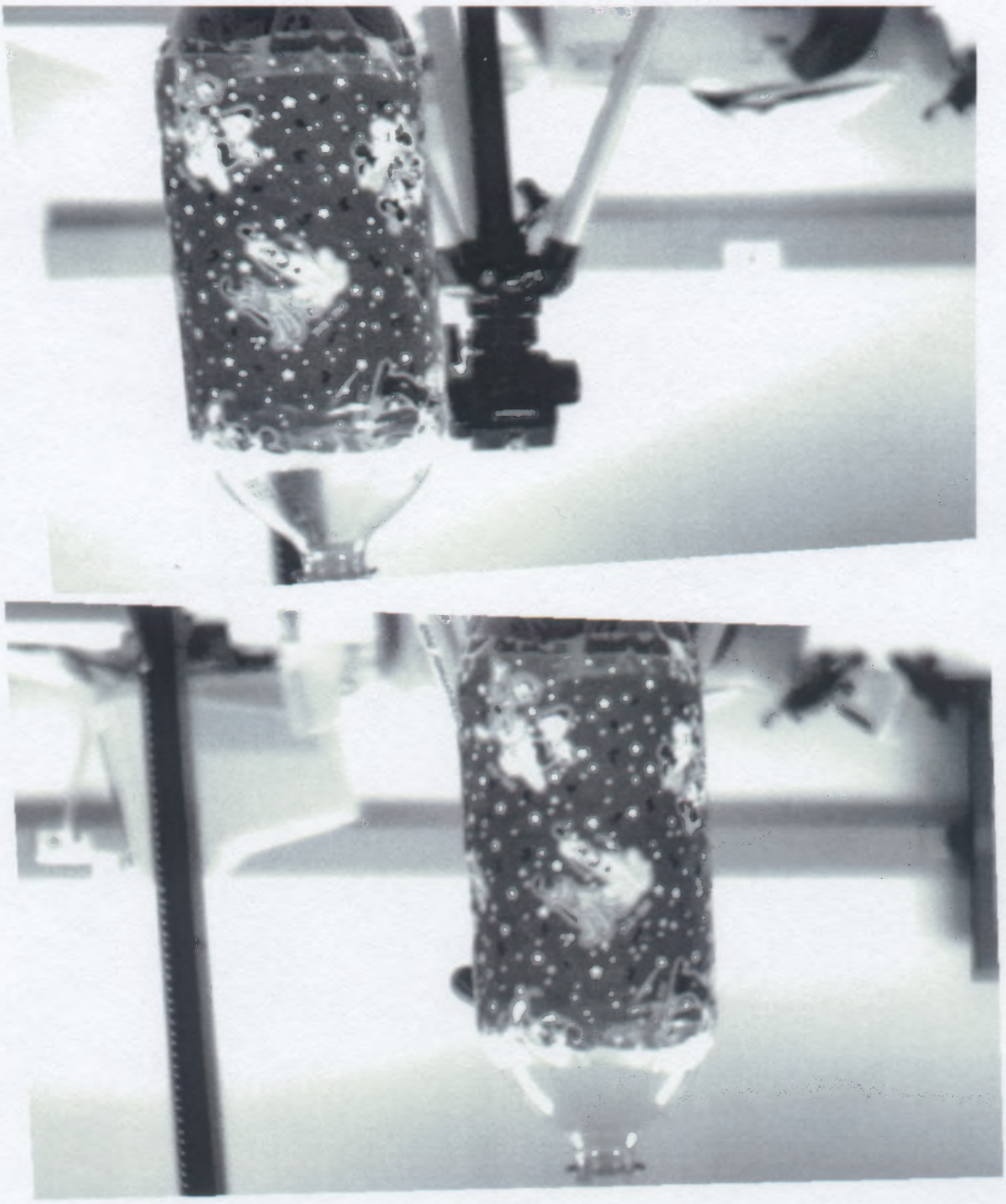
d) image taken from right most camera D.

Figure B.2 Image taken from all views. (Continued)



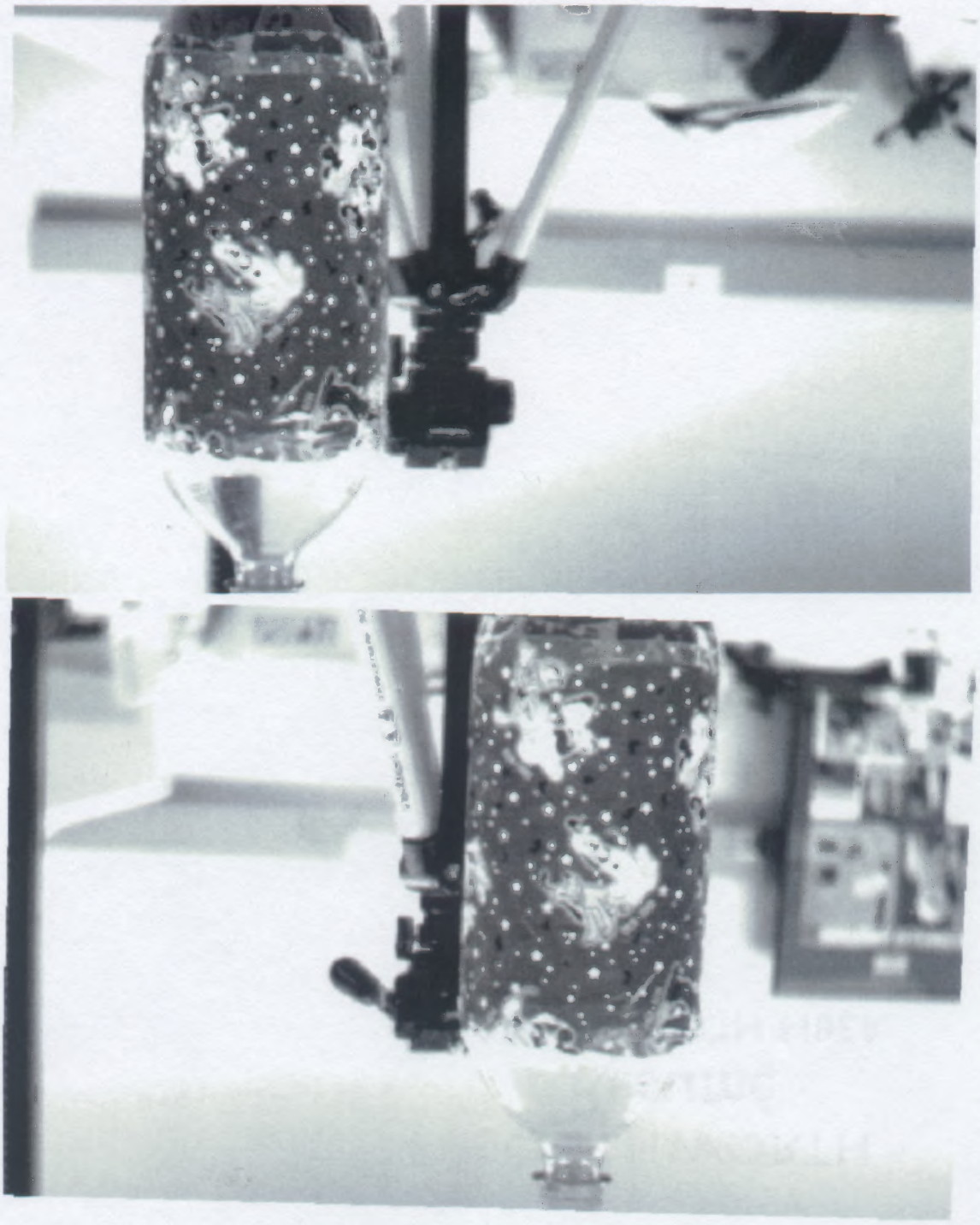
a) Rectified stereo pair A-B

Figure B.3 Rectified image all pairs.



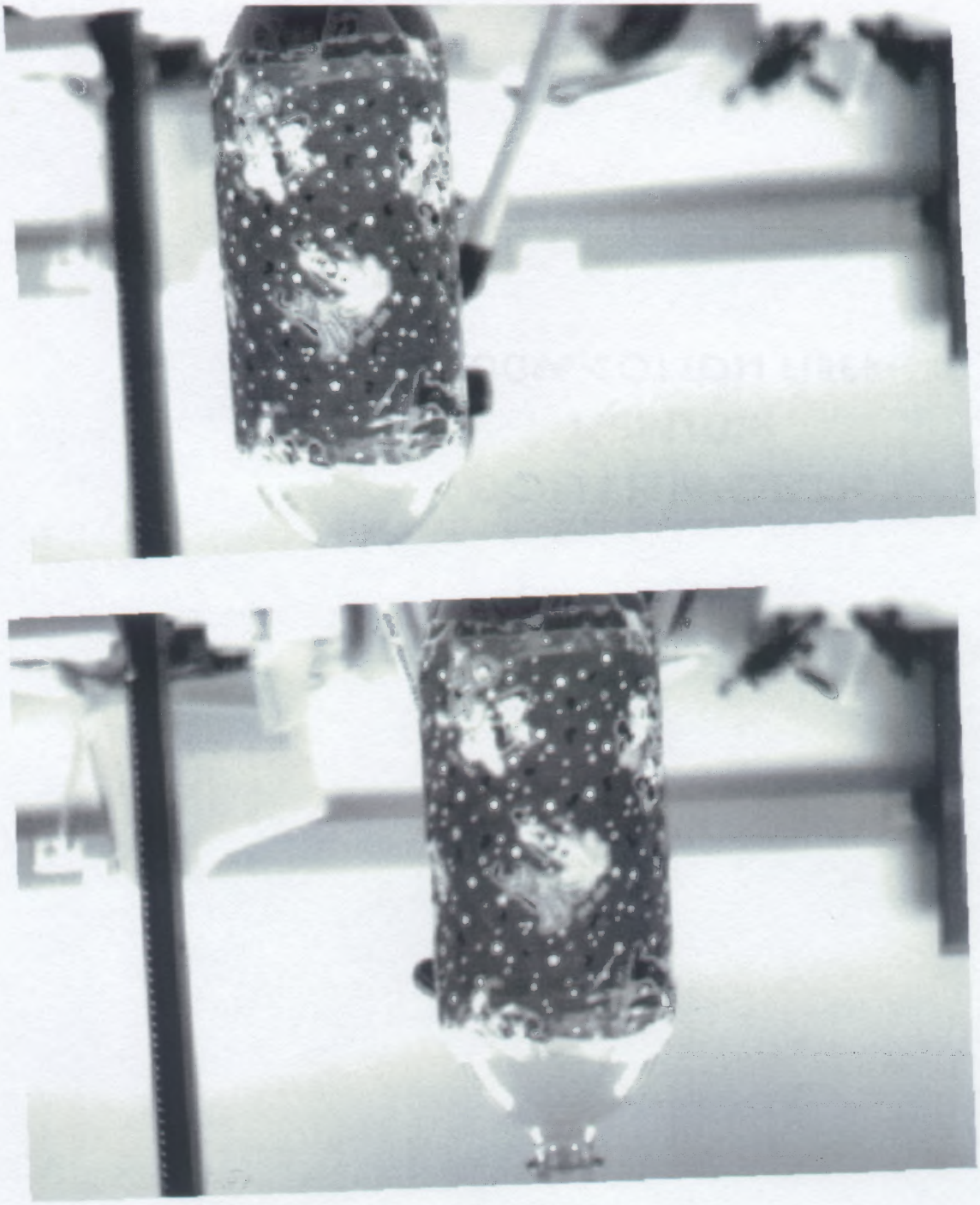
b) Rectified stereo pair A-C

Figure B.3 Rectified image all pairs. (Continued)



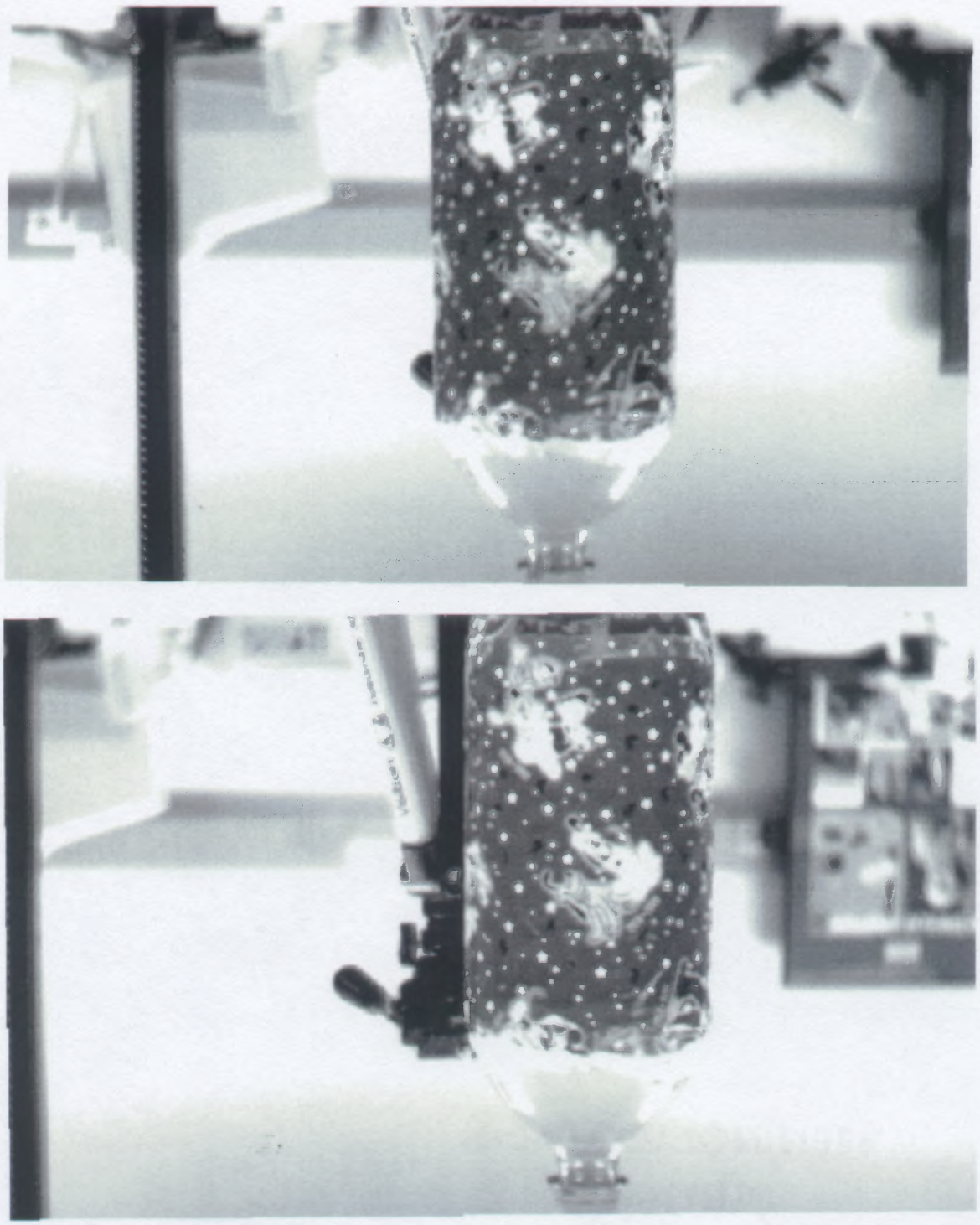
c) Rectified stereo pair A-D

Figure B.3 Rectified image all pairs. (Continued)



d) Rectified stereo pair B-C

Figure B.3 Rectified image all pairs. (Continued)



e) Rectified stereo pair C-D

Figure B.3 Rectified image all pairs. (Continued)

Table B.1 Position and disparity of a stereo pair A-B

Position name	Position image A in AB		Disparity AB	Position image B in AB	
	x	y		x	y
Soldier duck's ribbon	151	113	80	231	113
Star between ducks' heads	182	126	82	264	126
Big star over duck's hat	232	113	82	314	113
Mouse's pant	328	177	75	403	177
Dog's heart	230	263	81	331	263
Dog's nose	204	245	82	286	245
Dog's leg	209	326	81	289	326
Brown dog's ear	168	271	81	250	272
Yellow dog's ear	198	255	82	280	225
Micky mouse's leg	154	379	79	233	379
Micky's face	184	414	80	264	414

Table B.2 Position and disparity of a stereo pair B-C.

Position name	Position image A in AC		Disparity AC	Position image C in AC	
	x	y		x	y
Soldier duck's ribbon	170	97	284	454	97
Star between ducks' heads	198	97	286	484	97
Big star over duck's hat	264	87	#192	456	87
Mouse's pant	378	151	#77	455	151
Dog's heart	276	250	#16	292	250
Dog's nose	250	233	#94	344	233
Dog's leg	262	318	#179	441	318
Brown dog's ear	210	262	280	490	262
Yellow dog's ear	234	211	#275	510	211
Micky mouse's leg	203	379	#228	431	379
Micky's face	244	418	#188	432	418

(Note: Shown with # is mismatch result)

Table B.3 Position and disparity of stereo pair A-D

Position name	Position image A in AD		Disparity AD	Position image D in AD	
	x	y		x	y
Soldier duck's ribbon	134	83	#287	421	83
Star between ducks' heads	151	79	#182	333	79
Big star over duck's hat	214	76	#247	461	76
Mouse's pant	Occluded				
Dog's heart	209	226	300	509	228
Dog's nose	190	210	304	494	210
Dog's leg	189	287	299	488	287
Brown dog's ear	154	235	300	454	235
Yellow dog's ear	181	188	304	485	189
Micky mouse's leg	138	339	294	482	339
Micky's face	173	375	295	468	375

(Note: Shown with # is mismatch result)

Table B.4 Parameters from camera calibration

	Focal length by pixel size				
	fc a	fc b	fc c	fc d	
	2281.00	2263.00	2254.00	2265.00	
f(mm)	17.11	16.97	16.91	16.99	
Baseline	Bab	Bac	Bad	Bbc	Bcd
mm	127.07	254.69	374.38	128.31	113.89
Rotation matrix	Rab(1,3)	Rac(1,3)	Rad(1,3)	Rbc(1,3)	Rcd(1,3)
	0.08	0.12	0.16	0.03	0.07

Table B.5 Triangulation results. Shown red are mismatch points

Position name	AB			AC			AD		
	x	y	z	x	y	z	x	y	z
Soldier duck's ribbon	70.11	52.89	1059.12	79.37	45.07	1052.39	77.01	47.70	#1301.74
Star between ducks' heads	83.89	58.54	1051.44	92.11	44.91	1048.59	103.32	54.05	#1549.79
Big star over duck's hat	106.94	52.50	1051.44	147.91	48.51	#1262.85	130.98	46.51	#1386.26
Mouse's pant	155.13	84.38	1078.83	282.37	112.25	#1683.77			
Dog's heart	106.41	122.64	1055.27	250.46	225.76	#2045.40	117.78	127.36	1276.44
Dog's nose	94.03	113.83	1051.44	177.98	165.08	#1604.71	106.44	117.64	1268.86
Dog's leg	96.69	152.02	1055.27	151.06	182.46	#1299.58	106.67	161.98	1278.35
Brown dog's ear	77.72	126.37	1055.27	98.76	122.62	1060.07	86.79	132.43	1276.44
Yellow dog's ear	91.27	118.48	1051.44	111.06	99.66	#1069.83	101.40	105.32	1268.86
Micky mouse's leg	71.77	178.03	1063.01	105.48	195.98	#1171.20	78.47	192.77	1287.99
Micky's face	85.44	193.76	1059.12	137.91	235.10	#1273.93	98.23	212.92	1286.05

(Note: Shown with # is mismatch result)

Table B.6 Disparity of our method A-B-C

Position name	Position B unrectified in AB		Position B rectified in BC		Disparity BC	Position C in rectified BC	
	x	y	x	y		x	y
Soldier duck's ribbon	607	371	201	97	176	377	97
Star between ducks' heads	576	361	230	103	#155	#385	#103
Big star over duck's hat	531	377	275	85	#89	#364	#85
Mouse's pant	446	323	363	135	#14	#377	#135
Dog's heart	528	234	286	228	178	464	228
Dog's nose	552	250	260	214	178	438	214
Dog's leg	544	172	272	292	176	488	292
Brown dog's ear	584	222	230	243	176	406	243
Yellow dog's ear	558	268	253	196	177	403	196
Micky mouse's leg	595	118	224	348	175	399	348
Micky's face	564	86	256	378	176	432	378

Table B.7 Proposed method B-C-D

Position name	Position C in unrectified BC		Position C in rectified CD		Disparity CD	Position D in rectified CD		Acumulate Disparity	Disparity AB (for comparison)
	x	y	x	y		x	y		
Soldier duck's ribbon	422	420	392	66	30	406	116	286.00	80
Star between ducks' heads	#414	#415	#399	#75	#30	#414	#127	#267.00	82
Big star over duck's hat	#434	#433							82
Mouse's pant	#423	#383							75
Dog's heart	339	268	474	206	31	503	243	290.00	81
Dog's nose	365	301	448	190	32	476	231	292.00	82
Dog's leg	357	233	455	259	30	485	259	287.00	81
Brown dog's ear	398	274	415	217	30	445	217	287.00	81
Yellow dog's ear	372	320	441	171	30	471	171	289.00	82
Micky mouse's leg	408	169	403	323	29	432	323	283.00	79
Micky's face	376	137	435	355	29	464	355	285.00	80

(Note: Shown with # is mismatch result)

Table B.8 Triangulation proposed method

Position name	BC			CD		
	x	y	z	x	y	z
Soldier duck's ribbon	101.49	49.17	1142.61	240.96	61.70	1440.67
Star between ducks' heads	#126.59	#56.92	#1245.54	#246.08	#65.51	#1440.67
Big star over duck's hat	#211.13	#65.52	#1737.44			
Mouse's pant	#505.60	#188.78	#3151.99			
Dog's heart	143.28	114.68	1133.69	294.92	144.21	1432.63
Dog's nose	130.25	107.64	1133.69	276.85	134.61	1424.68
Dog's leg	137.34	148.02	1142.61	311.91	185.73	1440.67
Brown dog's ear	116.13	123.18	1142.61	259.50	154.56	1440.67
Yellow dog's ear	127.24	98.97	1138.13	257.58	124.67	1440.67
Micky mouse's leg	113.55	177.11	1147.13	256.46	222.60	1448.80
Micky's face	129.26	191.62	1142.61	277.68	241.79	1448.80

Table B.9 Triangulation of proposed method (cont.)

Position name	AC			AD			AB		
	x	y	z	x	y	z	x	y	z
Soldier duck's ribbon	73.39	55.36	1108.62	86.31	64.59	1303.72	70.11	52.89	1059.12
Star between ducks' heads	#91.78	#64.05	#1150.33	107.13	74.17	#1342.67	83.89	58.54	1051.44
Big star over duck's hat							106.94	52.50	1051.44
Mouse's pant							155.13	84.38	1078.83
Dog's heart	111.15	128.11	1102.31	130.66	149.41	1295.81	106.41	122.64	1055.27
Dog's nose	98.40	119.11	1100.22	115.54	138.76	1291.89	94.03	113.83	1051.44
Dog's leg	101.39	159.40	1106.51	119.27	186.04	1301.74	96.69	152.02	1055.27
Brown dog's ear	81.50	132.51	1106.51	95.88	154.66	1301.74	77.72	126.37	1055.27
Yellow dog's ear	95.68	124.21	1102.31	112.65	145.08	1297.78	91.27	118.48	1051.44
Micky mouse's leg	75.13	186.38	1112.87	88.42	217.62	1309.72	71.77	178.03	1063.01
Micky's face	89.43	202.81	1108.62	105.33	236.99	1305.72	85.44	193.76	1059.12

(Note: Shown with # is mismatch result)

Table B.10 Calibration result of camera A.

Focal Length: fc = [2410.59382 2391.42953] ± [56.69217 60.46237]
Principal point: cc = [500.24104 209.06385] ± [38.38559 17.09149]
Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: kc = [-0.60010 1.68491 0.01623 -0.00452 0.00000] ± [0.04969 0.50549 0.00296 0.00658 0.00000]
Pixel error: err = [0.43780 0.40924]

Table B.11 Calibration result of camera B.

Focal Length: fc = [2282.32932 2266.86396] ± [47.93600 49.34110]
Principal point: cc = [449.46831 275.35311] ± [33.68075 15.24367]
Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: kc = [-0.51899 1.18991 0.01154 0.00501 0.00000] ± [0.03942 0.32066 0.00237 0.00505 0.00000]
Pixel error: err = [0.38697 0.40938]

Table B.12 Calibration result of camera C

Focal Length: fc = [2279.86858 2268.92749] ± [35.79652 36.68829]
Principal point: cc = [437.90009 372.01523] ± [24.72611 13.81380]
Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: kc = [-0.44748 1.64914 0.00317 -0.00372 0.00000] ± [0.04005 0.58075 0.00181 0.00260 0.00000]
Pixel error: err = [0.35492 0.41640]

Table B.13 Calibration result of camera D

Focal Length: fc = [2272.06613 2261.01983] ± [32.20768 32.49057]
Principal point: cc = [494.86567 437.78209] ± [20.03213 12.67956]
Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion: kc = [-0.35473 0.87491 0.00173 -0.00666 0.00000] ± [0.04075 0.85849 0.00143 0.00134 0.00000]
Pixel error: err = [0.30312 0.37641]

Table B.14 Stereo calibration result, translation vectors between cameras

Translation vector between camera coordinates			
	x	y	z
Tab	-126.36	-2.5797	-13.1762
Tac	-254.612	3.3888	5.1835
Tad	-371.353	-0.3189	-4.2171
Tbc	-127.903	3.5186	-9.5656
Tcd	-113.864	-0.075	2.282
Tde	-125.209	-1.3128	3.8447

TableB.15 Stereo calibration result, Rotation matrices between cameras

Rotation Matrices between camera coordinates

Rab			Rac			Rad		
0.9979	0.0225	0.06	0.995	0.0343	0.0936	0.987	0.0169	0.1597
-0.0195	0.9985	-0.0505	-0.029	0.9979	-0.0578	-0.0061	0.9977	-0.0675
-0.0611	0.0493	0.9969	-0.0954	0.0548	0.9939	-0.1605	0.0657	0.9848

Rbc			Rcd		
0.9993	0.0104	0.0345	0.9976	-0.0209	0.0656
-0.0101	0.9999	-0.0077	0.0217	0.9997	-0.0116
-0.0346	0.0074	0.9994	-0.0653	0.013	0.9978

REFERENCES

[1] P. Witoonchart, and R. Foulds. "Three-Dimensional Surface and Volume Measurements Using a Camera Array." *IEEE 28th Annual Northeast Bioengineering Conference* (2002): 145-146.

[2] R. Klette and others, eds., *Computer vision, Three-Dimensional Data from Images,* (Singapore, Springer, 1998).

[3] M. Yachida, Y. Kitamura, and M. Kimachi. "Trinocular Vision: New Approach for Correspondence Problem," *Proc. 8th International Conference on Pattern Recognition* (1986): 1041-1044.

[4] D. Papadimitriou and T. Dennis, "Epipolar Line Estimation and Rectification for Stereo Image Pairs." *IEEE Trans. Pattern Analysis and Machine Intelligence,* (1996), Vol 5 No 4: 672-676.

[5] W. E. L. Grimson. "A computer implementation of a theory of human stereo vision." *Phil. Trans. Royal Soc. London* (1981) Vol. B292: 217-253.

[6] D. Marr and T. Poggio. "A computational theory of human stereo vision." *Proc. Royal Soc. London* (1979) Vol. B204: 301-328.

[7] E. Trucco and A. Verri , *Introductory Techniques for 3-D Computer Vision,* (New Jersey, Prentice-Hall,1998).

[8] T. Kanade and M. Okutomi. "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment." *Proc. IEEE Trans. Pattern Analysis and Machine Intelligence* (1994) Vol.16 No.9: 920-932.

[9] C. Lawrence Zitnick and T. Kanade. "A Cooperative Algorithm for Stereo Matching and Occlusion Detection." *Proc. IEEE Trans. Pattern Analysis and Machine Intelligence* (2000) Vol 22. No7: 675-684.

[10] R. Tsai. "A Versatile Camera Calibration Techniques for High- Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses." *IEEE Journal of Robotics and Automation* (1987) Vol. RA-3 No.4: 323-344.

[11] M. Okutomi and T. Kanade. "A Multiple-Baseline Stereo." *IEEE Trans. Pattern Analysis and Machine Intelligence* (1993) Vol.15 No.4: 353-363.

[12] Intel Corp, "Camera calibration Toolbox for Matlab," http://newbologna.vision.caltech.edu/bouguetj/calib_doc/ (25 February 2002).

[13] S. B. Kang, John A. Webb, C. Lawrence Zitnick, Takeo Kanade. "An Active Multi-baseline Stereo System with Real-Time Image Acquisition." *International Conf. on Computer Vision* (1995): Cambridge, MA.

[14] Z. Zhang. "A flexible new technique for camera calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000): 1330-1334

[15] Camera Calibration
http://www.debevec.org/~debevec/Thesis/debevec-phdthesis-1996_ch4_calib.pdf

[16] J. P. Frisby and J. E. W. Mayhe. "The role of spatial frequency tuned channels in vergence control." *Vision Res.* (1981) Vol.20: 721-732.

[17] J.E.W. Mayhew and J.P. Frisby. "Psychophysical and computational studies towards a theory of human stereopsis." *Artificial Intell.* (1981) Vol. 17: 349-385.

[18] P. Mowforth, J. E. W. Mayhew and J. P. Frisby. "Vergence eye movements made in response to frequency filtered random dot stereograms." *Perception* (1981) Vol.10: 299-304.

[19] S. B. Pollard, J. E. W. Mayhew and J. P. Frisby. "PMF: A stereo correspondence algorithm using a disparity gradient limit." *Perception* (1981) Vol. 14: 449-470.

[20] U. R. Dhond and J. K. Aggrawal. "Structure from Stereo-A Review." *IEEE Transactions on Systems, Man, And Cybernetics* (1989) Vol. 19 No.6: 1489-1510.

[21] H.P. Moravec. "Towards automatic visual obstacle avoidance." i*n Proc. 5th Int. Joint Conf. Artificial Intel.* (1977): 584.

[22] D. B. Gennery. "Object detection and measurement using stereo vision." *in Proc. ARPA Image Understanding Workshop, College Park, MD* (1980): 161-167.

[23] M. J. Hannah, "Bootstrap Stereo," *in Proc. ARPA Image Understanding Workshop* (1980): 201-208.

[24] S. D. Cochran."3-D Surface Description from Binocular Stereo." *IEEE Trans. On Pattern Analysis and Machine Intelligence* (1992)Vol. 14 No 10: 981-994.

[25] A. Fusiello, E. Trucco, and A. Verri,. "A compact algorithm for rectification of stereo pairs." *Machine Vision and Applications* (2000)*:* 400-408.

[26] Luca Iocchi, "Stereo Vision: Triangulation," *Dipartimento di Informatica e Sistemistica Università di Roma "La Sapienza"*, Italy, http://www.dis.uniroma1.it/~iocchi/stereo/triang.html (12 November 2001).

[27] E. R. Devies. *Machine Vision Theory Algorithms Practicalities 2nd edition*, (London, Academic Press, 1997).