

Fall 1-31-2007

Implementation of stereo correspondence algorithms for multi-baseline vision system

Navaneetha Shivram
New Jersey Institute of Technology

Follow this and additional works at: <https://digitalcommons.njit.edu/theses>



Part of the [Biomedical Engineering and Bioengineering Commons](#)

Recommended Citation

Shivram, Navaneetha, "Implementation of stereo correspondence algorithms for multi-baseline vision system" (2007). *Theses*. 386.

<https://digitalcommons.njit.edu/theses/386>

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen



The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

IMPLEMENTATION OF STEREO CORRESPONDENCE ALGORITHMS FOR MULTI – BASELINE VISION SYSTEM

by

Navaneetha Shivram

The purpose of this study is to determine the impact of varying camera baselines on correspondence and depth estimation. As the baseline increases the resolution of depth increases but finding correspondences is very tedious. On the other hand when cameras are placed close to each other and thereby reducing the baseline distance points in the images are easily matched however depth resolution deteriorates. This problem is solved by using a multi-camera system is used in which the cameras are spaced close to each other along a straight line. Calculating the correspondence along successive camera pairs is relatively easy. These correspondences are later propagated to the pairs with larger baselines.

Another important aspect of this study was to implement different area based stereo correspondence algorithms. Advantages and disadvantages are noted in each case. It is noticed that in some cases simple SSD and SAD worked very well but failed in other cases where Second Order Correlations worked well.

**IMPLEMENTATION OF STEREO CORRESPONDENCE ALGORITHMS FOR
MULTI – BASELINE VISION SYSTEM**

by

Navaneetha Shivram

**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 Biomedical Engineering**

Department of Biomedical Engineering

January 2007

January 2007

APPROVAL PAGE

**IMPLEMENTATION OF STEREO CORRESPONDENCE ALGORITHMS FOR
MULTI - BASELINE VISION SYSTEM**

Navaneetha Shivram

Dr. Richard Foulds, Thesis Advisor
Associate Professor of Biomedical Engineering, NJIT

Date

Dr. Sergei Adamovich, Committee Member
Associate Professor of Biomedical Engineering, NJIT

Date

Dr. Lisa Simone, Committee Member
Associate Research Professor of Biomedical Engineering, NJIT

Date

BIOGRAPHIC SKETCH

Author: Navaneetha Shivram

Degree: Master of Science

Undergraduate and Graduate Education:

- Master of Science in Biomedical Engineering,
New Jersey Institute of Technology, Newark, NJ, 2007
- Bachelor of Engineering in Medical Electronics,
Dayanand Sagar College of Engineering, Bangalore, India, 2007

Major: Biomedical Engineering

This thesis is dedicated to
my father Prabhakar Reddy, mother Suvarna,
sisters Shilpa and Pooja

ACKNOWLEDGMENT

I would like to thank my thesis advisor, Dr. Richard Foulds, for his patience, technical-guidance and unwavering support. This project would have been exceedingly difficult if not for his understanding and encouragement. I would also like to thank Dr. Sergei Adamovich and Dr. Lisa Simone for graciously agreeing to be members of the thesis committee. I would like to thank the Office of Graduate Studies for their effort in ensuring the success of the thesis.

I would also like to thank my teachers and New Jersey Institute of Technology; especially Dr. Rockland for his astute mentorship. I am also grateful to my colleagues – Ms. Lisa Schmidt and Ms. Pooja Maini for their ideas and encouragement. Most of all, I would like thank my family, for their love and belief in me which has been a tremendous motivation throughout my life. Lastly, my special thanks to my friends Pradeep and Vishal.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
1.1 Objective	1
1.2 Background	2
1.2.1 Principle of Stereovision.....	2
1.2.2 Calibration.....	5
1.2.3 Rectification.....	6
1.2.4 Correspondence.....	8
1.2.5 Triangulation.....	9
1.3 Multi-camera System.....	10
2 CORRESPONDENCE.....	12
2.1 Overview.....	12
2.1.1 Area Based Matching.....	12
2.1.2 Feature Based Matching	13
2.1.3 Difficulties in Stereo Correspondence	13
2.2 Algorithms	14
2.2.1 Sum of Absolute Difference (SAD).....	15
2.2.2 Sum of Squared Differences (SSD)	17
2.2.3 Normalized SSD	18
2.2.4 First Order Cross-correlation	20
2.2.5 Second Order Cross-correlation.....	21

TABLE OF CONTENTS
(Continued)

Chapter	Page
2.3 Range Limiting	23
2.3.1 Start Range	24
2.3.2 End Range	24
2.4 Edge Enhancement.....	26
2.5 Evaluation	29
2.5.1 Intensity Difference	29
2.5.2 Change in Shape	31
2.5.3 Surfaces.....	33
2.5.4 Occlusion	35
2.5.6 Uniqueness.....	37
3 PROCEDURE.....	40
3.1 Apparatus	40
3.2 STEP 1 – Image Acquisition.....	41
3.3 STEP 2 – Calibration	41
3.4 STEP 3 – Rectification.....	44
3.5 STEP 4 – Correspondence	46
3.6 STEP 5 – Triangulation	47
4 RESULTS	48
4.1 AB.....	48
4.1.1 Captured Images	48
4.1.2 Rectified Images	49

TABLE OF CONTENTS
(Continued)

Chapter	Page
4.1.3 Correspondence Tabulation	49
4.2 BC	50
4.2.1 Captured Images	50
4.2.2 Rectified Images	50
4.2.3 Correspondence Tabulation	51
4.3 CD	51
4.3.1 Captured Images	51
4.3.2 Rectified Images	52
4.3.3 Correspondence Tabulation	52
4.4 AC	53
4.4.1 Captured Images	53
4.4.2 Rectified Images	53
4.4.3 Correspondence Tabulation	54
4.5 AD	54
4.5.1 Captured Images	54
4.5.2 Rectified Images	55
4.5.3 Correspondence Tabulation	55
4.6 Propagation	56
4.7 Effect of Baseline	57
5 CONCLUSION	59
REFERENCES	60

LIST OF TABLES

Table	Page
1 Correspondence Result for AB	49
2 Correspondence Result of BC.....	51
3 Correspondence Result of CD	52
4 Correspondence Result for AC	54
5 Correspondence Result for AD.....	55
6 Propagation Results	56
7 Triangulation Results.....	58

LIST OF FIGURES

Figure	Page
1 Stereo imaging setup.....	3
2 Reference frames	6
3 Epipolar Geometry.....	6
4 Non - rectified Images	7
5 Rectified Images	8
6 Stereo correspondence	8
7 Triangulation.....	10
8 Proposed multi-camera system	11
9 Left and Right Images.....	15
10 SAD log plot	16
11 SSD log plot.....	18
12 Normalized SSD log plot.....	19
13 Cross correlation plot.....	21
14 Normalized Cross correlation plot.....	22
15 Summary of correlation plots.....	23
16 Range $x_r < x_l$	24
17 Full range with possible false correspondences.....	25
18 Limited range with false correspondences removed.....	26
19 Edge Enhanced Images	27
20 Plot for original images with false correspondence.....	28

**LIST OF FIGURES
(Continued)**

Figure	Page
21 Plot for edge enhanced images	29
22 Left and Right Images with intensity difference highlighted	30
23 Plot in case of intensity difference.....	31
24 Left and Right Images showing changed shape.....	31
25 Plot in case of shape change	32
26 Surface example.....	33
27 Plot for surface correspondence.....	34
28 Occlusion Example 1	35
29 Occlusion Example 2	36
30 Plot in case of occlusion	36
31 Occlusion Example 1	37
32 Incorrect correspondence due to lack of uniqueness	38
33 Non-unique region where Second Order works better	39
34 Experimental Setup.....	40
35 Captured images.....	41
36 Calibration Toolbox GUI.....	42
37 Checkerboard images used for calibration.....	43
38 Corner selection and Extracted corners	44
39 Stereo rectification interface	44
40 Stereo Calibration parameters.....	45

LIST OF FIGURES
(Continued)

Figure	Page
41 Rectified images.....	45
42 Example correlation plot.....	46
43 Stereo triangulation function call.....	47
44 Images from cameras A & B	48
45 Rectified images from cameras A & B.....	49
46 Images from cameras B & C.....	50
47 Rectified images from cameras B & C	50
48 Images from cameras C & D	51
49 Rectified images from cameras C & D.....	52
50 Images from cameras A & C	53
51 Rectified images from cameras A & C.....	53
52 Images from cameras A & D	54
53 Rectified images from cameras A & D.....	55
54 Propagation of Correspondence.....	56
55 Correspondence Mismatch.....	57
56 Second Order Algorithms for the point shown in the previous figure.....	58

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of this thesis is to investigate different aspects of computer vision as applicable to a 3 dimensional multi camera system. In particular, the thesis concentrates on the problem of *correspondence* between images from multiple cameras. It also explores in lesser detail - *triangulation* (depth calculation based on correspondence data) and effect of *camera baseline* on correspondence and triangulation.

For the problem of stereo-correspondence, the following algorithms are reviewed in detail: SAD, SSD, normalized SSD and first/second order cross-correlation. The algorithms considered are largely area based, with improvements such as pre-processing for edge-enhancement. The intent of the thesis is to highlight the advantages and drawbacks of each approach and provide a platform for comparison.

A dilemma often faced in computer vision is with respect to the optimal baseline distance between cameras. Small baselines yield accurate correspondence at the cost of inaccurate depth estimation. Larger baselines improve quality of the latter but make correspondence more difficult. The thesis also evaluates the possibility of using a camera array with small baselines between individual cameras and a larger end-to-end baseline to overcome this problem

1.2 Background

Stereo in Greek means *relating to space*. Hence, stereovision represents vision as related to space. It refers to three-dimensional information obtained from one or more images [1]. The concept of stereovision dates back to the 15th century when the Italian polymath, Leonardo da Vinci also described as the Renaissance man, realized that eyes project different distances of the object at different horizontal positions; which in essence is depth perception [9]. However, stereovision was first scientifically described by Charles Wheatstone in 1838. He invented the stereoscope to display pictures to the left and right eyes separately. It was based on the fact that each individual eye views the same object placed at different distances from slightly different horizontal positions. This in turn gives the depth cue of horizontal disparity to the brain and allows us to perceive depth by triangulation [2].

1.2.1 Principle of Stereovision

An important aspect of understanding stereovision is to be familiar with the mathematics behind it. The following illustration along with the listed geometric equations provides a better insight into the fundamental principle of depth perception. The figure shows a setting with two parallel cameras which are placed at a distance 'd' from each other. The line joining the 2 cameras is called the *baseline* and is perpendicular to the line of sight of the cameras. The origin 'o' of the setup will be at the centre of the baseline

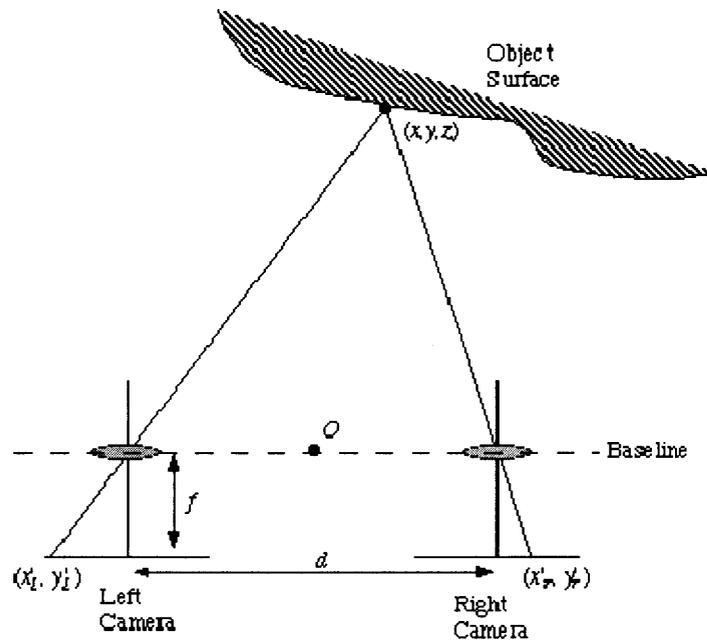


Figure 1 Stereo imaging setup

Source: David Marshall [10]

Assume a point (x, y, z) in the 3 d co-ordinate system. Let this point have image coordinates (x_l, y_l) and (x_r, y_r) in the left and right image planes of the respective cameras. Let 'f' be the focal length of both cameras and 'd' the perpendicular distance between the lens centre and the image plane. Then by similar triangles:

$$\frac{x_l}{f} = \frac{x - d/2}{z}$$

$$\frac{x_r}{f} = \frac{x + d/2}{z}$$

$$\frac{y_r}{f} = \frac{y_l}{f} = \frac{y}{z}$$

Solving for (x, y, z) gives:

$$x = \frac{d(x_l + x_r)}{2(x_l - x_r)}$$

$$y = \frac{d(y_l + y_r)}{2(x_l - x_r)}$$

$$z = \frac{df}{2(x_l - x_r)}$$

Source: David Marshall [10]

The important thing in these equations is the denominator $(x_l - x_r)$. It is called the *disparity* of the system. It is the critical parameter that characterizes the depth of the point P.

In a practical system, the process of depth perception is a multi step process [1]. Listed below is a brief description of each step. The following sub-section explains them in more detail.

Calibration – This step characterizes each individual camera in the system with a set of parameters. These parameters uniquely identify the optical and spatial characteristics of the cameras [4] [3]

Rectification – It is the process of aligning the epipolar lines of images captured by different cameras, taken one pair at a time. This is equivalent to aligning the image planes of a given camera pair.

Correspondence – It involves identifying reference points in both the captured images. This is often challenging as each point in the reference frame appears slightly differently in both the images.

Triangulation – In this step, the *disparity* data is used to calculate the depth of each point in the reference frame.

1.2.2 Calibration

It is the process of determining the values of certain parameters that characterize the behavior of a camera. The parameters are of two types [a] *intrinsic* parameters and [b] *extrinsic* parameters.

The intrinsic parameters represent the optical, geometric and digital characteristics of a camera [3] [4]. They are independent of the spatial position and orientation of the camera. The intrinsic parameters are [8]

- Focal length
- Principal point - The point at which the ray passing through the focal point intersects the image plane perpendicularly
- Skew coefficient – It defines the angle between the x and y pixel axes
- Optical Distortion of lens – It represents the optical distortions of the lens

The extrinsic parameters represent the spatial characteristics of the camera. It describes the camera reference frame in relation to a world reference frame [3]. It consists of a translation vector (T) and a rotation vector (R).

In order to better understand the concept of world reference frame, consider the following figure. The Cartesian co-ordinates (X_w, Y_w, Z_w) represent the world reference frame and the camera reference frame is represented by (X_k, Y_k, Z_k) . Extrinsic parameters T and R describe the mapping of the former to the latter.

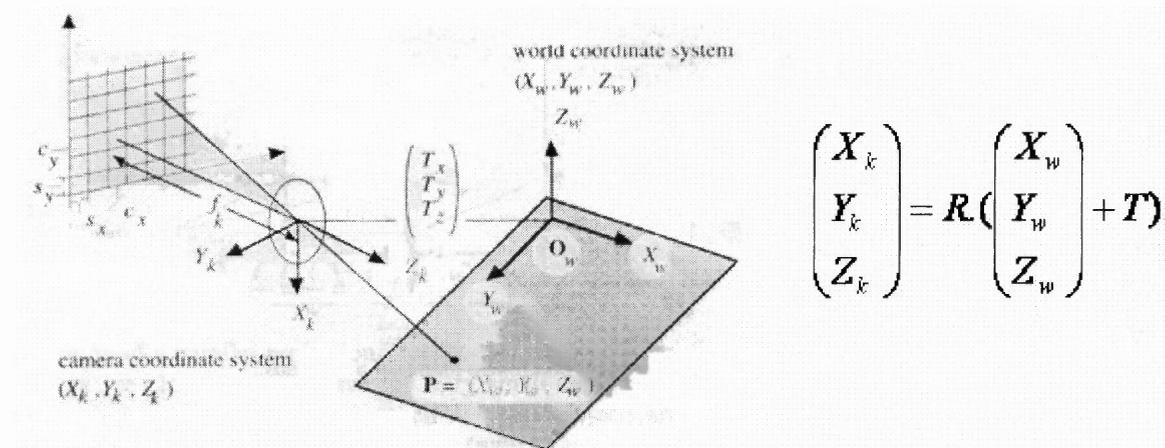


Figure 2: Reference frames

Source: Klette [1]

1.2.3 Rectification

Rectification attempts to align the epipolar lines of a given image pair in order to reduce the correspondence search to one axis [11]. To understand the concept of epipolar lines consider the following illustration.

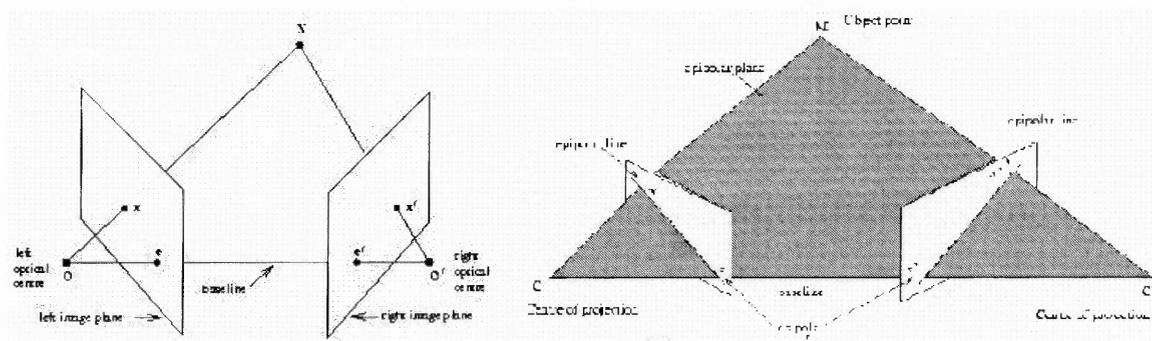


Figure 3: Epipolar Geometry

Source: David Marshall [10]

The figure on the left in the above example shows two cameras with 'O' and 'O'' as left and right optical centers respectively. In the right image the details of the epipolar geometry is described. The epipolar plane passes through the object point and the optical centers 'C' and 'C''. The epipolar line is then defined as the intersection of the epipolar

plane and the individual image planes. It depends only on the intrinsic and extrinsic parameters of the camera pair [5]

The concept of stereo rectification is to bring the two camera planes to be coplanar i.e. to some common plane in space [4]. To satisfy such a condition one image is rotated with respect to the other image's optical centre such that the focal planes are coplanar and thereby containing the baseline. Thus the new x-axis is parallel to the baseline and these form the horizontal epipolar lines.



Figure 4: Non - rectified Images
Source: Marc Pollefeys [12]

A criterion used to verify rectification of the images is the fact that conjugate points in rectified images should have the same vertical coordinates i.e. lie on the same line when placed next to each other. For the purpose of illustration stereo-images of a mansion are presented here. It may be observed in the non-rectified images that the base of the door is not aligned in the two images. This is typical of a practical setup as the cameras are not aligned as required. The process of rectification eliminates this problem. As a result, in the rectified images the base of the door is clearly aligned.

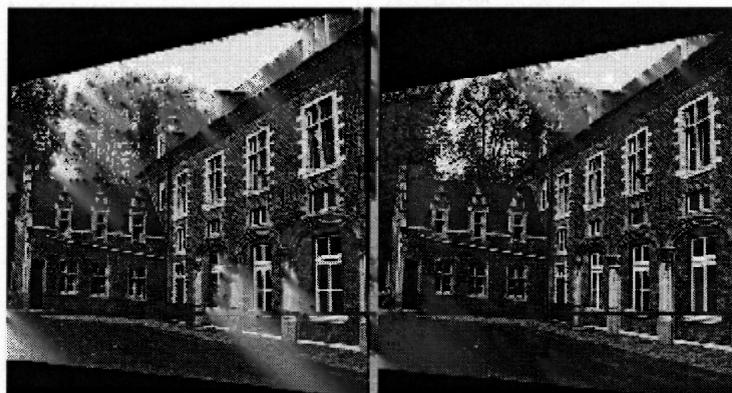


Figure 5: Rectified Images
Source: Marc Pollefeys [12]

Note that the rectification warps the image in 2-D

1.2.4 Correspondence

Correspondence is defined as the problem of finding sets of pixels in the left and right images such that they represent same point(s) in space.

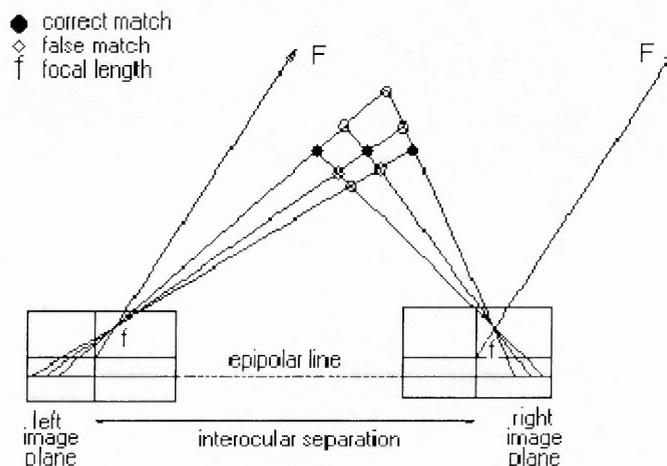


Figure 6: Stereo correspondence
Source: Jesse Hoey [14]

In the figure the left image plane and the right image plane correspond to two human eyes or two cameras placed a certain baseline distance apart. The image planes and the optical centers are along the epipolar plane. The reason for the image planes to be along the epipolar plane is that rectification reduces the correspondence problem to be along a single dimension (horizontal). The example shows a target containing four dots.

The problem is to determine which one of these dots as seen by the left eye correspond to the dots as seen by the right eye. There are nine possible combinations or possible matches that may be inferred from this experiment. But the ones marked black are the correct matches and the rest are incorrect. The ability to choose the correct dots as seen by each eye is trivial in case of humans but, providing this intelligence to a mechanical system is a challenge.

The reason for the importance of correspondence matching is that only when the images from both the eyes/cameras are matched will an individual be able to perceive the depth map of the environment. There are a few methods that have been proposed to solve this problem, but none of these algorithms have been able to produce the results of desired quality.

1.2.5 Triangulation

3-D imaging is one of the most explored subjects in the imaging world in the recent past. The third dimension corresponds to the depth of the object in the scene. To calculate the depth perception images are always taken in pairs by two different sources (cameras) with slightly different orientation (angles). The images are then compared for common points. The corresponding points are triangulated to obtain an estimate of the depth.

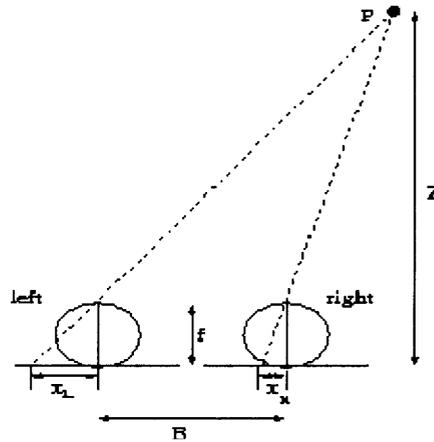


Figure 7: Triangulation

Source: Jesse Hoey [14]

The figure below depicts cameras at positions X_l and X_r . B is the baseline and F is the focal length of the cameras. The depth Z of the object 'P' from the baseline of the two cameras and the disparity or the difference between these positions can be related by the following equation.

$$\text{Disparity} = (x_l - x_r) = \frac{B * F}{Z}$$

1.3 Multi-camera System

Consider a point M that has projects M_L and M_R on a pair of cameras with focal length f & optical centers C_L and C_R respectively. The depth Z of the point is then given by

$$Z = \frac{(C_R - C_L) * f}{((M_L - C_L) (M_R - C_R))}$$

Source: Witoonchart & Foulds [15]

From the equation it is seen depth is equal to the focal length times the ratio of the baseline to the disparity [2]. The disparity when measured in pixels becomes dependent on baseline. Hence, farther away the cameras are placed, larger is the pixel disparity. This

provides a greater range (in terms of number of pixels) for measuring depth. Therefore, it may be concluded that larger baselines provide improved accuracy in measuring z .

The downside of increasing baselines is the reduction in the quality of correspondence between the images. This is because as the separation between the cameras increases, the nature of the images also changes. To overcome this dilemma it has been proposed that a multi-camera system be used where the baseline between successive cameras is small, but the end-to-end baseline is sufficiently large [15]. This is the approach that has been adopted in this thesis.

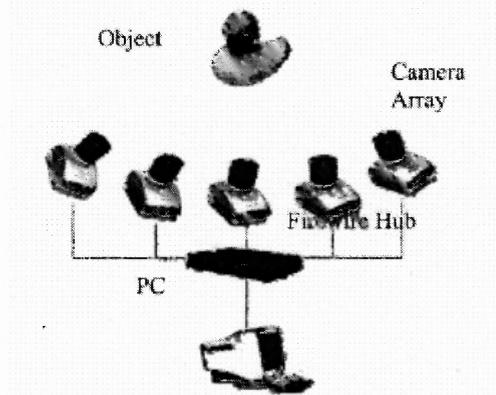


Figure 8: Proposed multi-camera system
Source: Witoonchart & Foulds [15]

CHAPTER 2

CORRESPONDENCE

2.1 Overview

Stereo correspondence is one of the biggest challenges in the world of computer vision. There are many people who have dedicated their entire research years to this field. Over the years, there have been different approaches to stereo matching, the most widely accepted ones being *area based* matching and *feature based* matching [1].

2.1.1 Area Based Matching

Area based matching as the name suggests attempts to match a certain area in the left image to a similar area in the right image. It is a popular approach and one that is investigated in detail in this thesis [6].

In area based matching a small portion of the images represented by a 5x5, 7x7 or 9x9 size window are compared for intensity, positional or statistical correspondence using functions. The functions commonly used are SAD (sum of absolute difference), SSD (sum squared difference), and normalized SSD, first order cross-correlation and second order cross-correlation. Since the images being compared are rectified i.e. the corresponding points in both the images lie on the same horizontal-scan (epipolar) lines. Hence, the search may be limited to the horizontal dimension only.

The advantage of this method is that a *dense* disparity map is the output. Like all the other methods area based matching also has the occlusion problem where some parts of the scene or objects are not covered by either of the 2 cameras.

2.1.2 Feature Based Matching

In feature based stereo matching there are two processes the first one is called preprocessing where the features that are stable under the change of the viewpoint are extracted and the second step is to apply the matching techniques to these features. The features extracted include edges, corners, line segments, and curve segments are stable and do not change with the position of the camera. This approach is not investigated in detail here.

The advantage of feature based matching is that by doing so the number of search elements will be reduced thus increasing the accuracy of the matches. The drawbacks with this method is that edges and corners are easy to detect but suffer from occlusion, lines and curves require extra computation time and other features such as circles, ellipses and polygonal regions are restricted to indoor scenes only.

2.1.3 Difficulties in Stereo Correspondence

A quality solution for stereo correspondence has been elusive for a long time, largely due certain critical constraints that are inherent in the task. Listed are some of the assumptions that are required to hold for correspondence algorithms to work.

Uniqueness [2]: Correspondence requires each pixel or feature from one image to match with no more than one pixel or feature in the other image being compared. This constraint fails at points where portions of an object in the scene are occluded in one of the images. Hence, there may be pixels or features in one image that do not have a unique match in the other.

Similarity [16]: Another requirement of algorithms is that an object in a scene should be projected on to the cameras such that pixel values or features due to the projection in both the images must be similar. This may not be the case due to dissimilar displacements of the camera frames with respect to the reference frame. Hence, certain portions of an object project may not be similar in both the compared images.

Continuity [2]: Algorithms generally are based on the fact that the images are composed of cohesive matter which means that the disparity of the matches should vary smoothly. But this constraint fails at depth discontinuities and this causes an abrupt change in disparity.

2.2 Algorithms

In the thesis the following algorithms are evaluated [13]

- Sum of Absolute Difference (SAD)
- Sum of Squared Difference (SSD)
- Normalized SSD
- First Order Cross-correlation
- Second Order Cross-correlation

For the purpose of illustration, consider the following left and right images of a target captured using a pair of cameras. The images have been rectified. In the presented algorithms the point being correlated has projections (740, 602) on the left image and (689, 602) on the right image. It may be observed that the y coordinates in both the projected points are identical. This is a result of the process of rectification. To

take advantage of this fact, matching algorithms fix a window (generally 7x7) about the projection in the first image and look for the corresponding projection in the second image, but only along the same epipolar line i.e. only search horizontally along a fixed line parallel to the y axis.

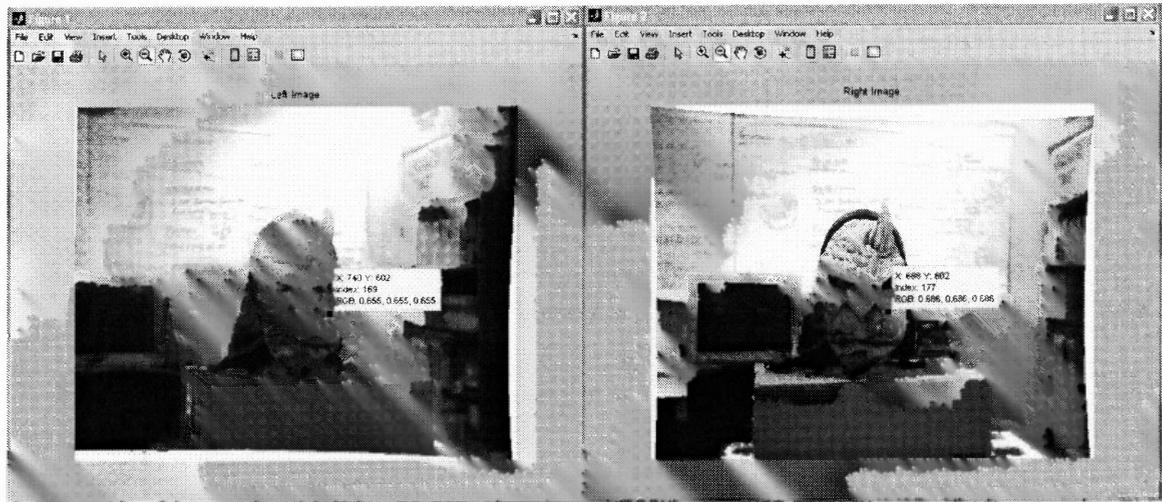


Figure 9: Left and Right Images

In the following algorithms, a 7x7 window is placed around the point (742, 602) in the left image which represents a point on the tip of the target mask's tooth. Another window is slid along the $y = 602$ line in the right image. The window regions are compared at each point for correspondence and the result is plotted. If successful, correspondence must be obtained at $x = 689$.

2.2.1 Sum of Absolute Difference (SAD)

SAD algorithm as the name suggests, is the summation of the difference between the intensity values of pixels in the windowed portions of the two images. It is represented mathematically as follows

$$SAD(d) = \sum_{u,v} |I_1(u,v) - I_2(u+d,v)|$$

In the formula $I_1(u, v)$ represents the intensity value of pixel in the 1st image at the u^{th} row and v^{th} column and $I_2(u, v)$ represents the intensity value of pixel in the 2nd image at the $(u + d)^{\text{th}}$ row and v^{th} column.

This method attempts to determine the dissimilarity between the windowed regions by taking the difference between the intensity values of each pixel. Hence, more the difference in the windowed regions, greater the difference between the intensity values of the pixels. Inversely, more alike the windowed regions, smaller will be the difference in the intensity values. Thus, the algorithm states that minimum value of the SAD array calculated from the stated formula, represents the point of maximum correspondence.

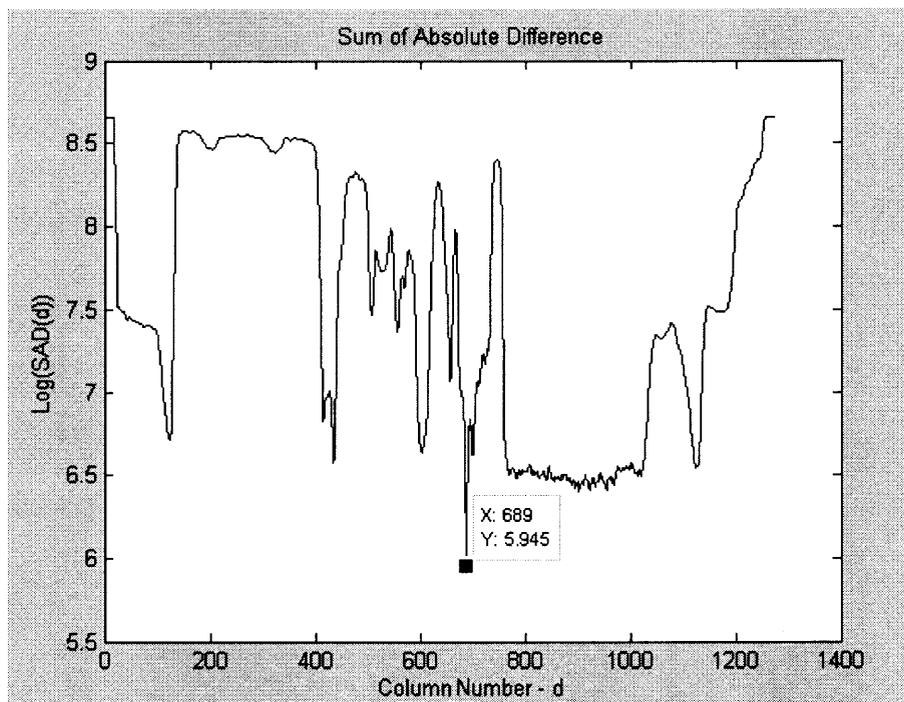


Figure 10: SAD log plot

The above plot shows the result of the SAD algorithm applied to the left and right images shown before. It can be clearly observed that there is a minimum at $d = 689$. This may be manually cross verified by looking for the corresponding point (tooth tip) in the right image. This has been highlighted at $x = 689$. Hence, the correspondence is clearly identified. It may also be noted that the log of the result is plotted to reduce the dynamic range.

2.2.2 Sum of Squared Differences (SSD)

It is another algorithm that is based on the difference in the intensity values of the pixels. However, in this method the sum of the squares of the differences is considered. This is equivalent to the *error energy* of the windowed regions.

$$\text{SSD}(d) = \sum_{u,v} (I_1(u,v) - I_2(u+d,v))^2$$

In the formula $I_1(u, v)$ represents the intensity value of pixel in the 1st image at the u th row and v th column and $I_2(u, v)$ represents the intensity value of pixel in the 2nd image at the $(u + d)$ th row and v th column.

The biggest difference between SAD and SSD is the way intensity difference is treated. In SAD it is magnitude summed, while in SSD it is squared and summed. Hence, it may be said that SAD varies linearly with difference while SSD varies non-linearly with square of difference.

The biggest difference between SAD and SSD is the way intensity difference is treated. In SAD it is magnitude summed, while in SSD it is squared and summed. Hence, it may

be said that SAD varies linearly with difference while SSD varies non-linearly with square of difference.

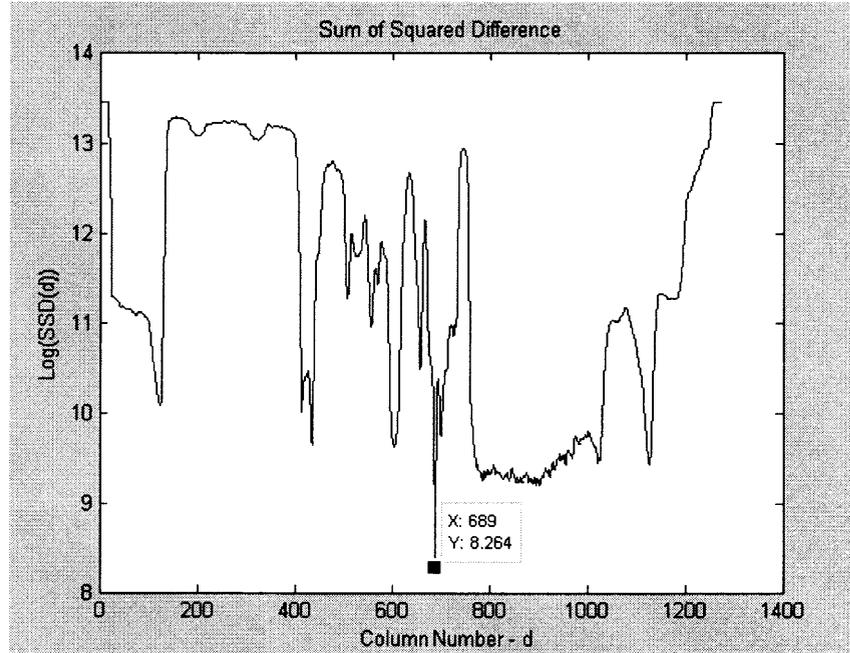


Figure 11: SSD log plot

The correspondence indicated by the minimum is at the expected value $d = 689$. Further, like the SAD algorithm, log of the result is plotted to reduce the dynamic range.

2.2.3 Normalized SSD

This is a second order matching algorithm i.e. it attempts to find the similarity in the intensity *deviation* in the two windows. This is as opposed to other first order algorithms that attempt to determine the similarity in the intensity values.

$$\text{Normalized SSD}(d) = \sum_{u,v} \left(\frac{(I_1(u,v) - \bar{I}_1)}{\sqrt{\sum_{u,v} (I_1(u,v) - \bar{I}_1)^2}} - \frac{(I_2(u+d,v) - \bar{I}_2)}{\sqrt{\sum_{u,v} (I_2(u+d,v) - \bar{I}_2)^2}} \right)^2$$

In the formula $I_1(u, v)$ represents the intensity value of pixel in the 1st image at the u^{th} row and v^{th} column and $I_2(u, v)$ represents the intensity value of pixel in the 2nd image at the $(u + d)^{\text{th}}$ row and v^{th} column. Further, \bar{I}_1 represents the mean of I_1 and \bar{I}_2 represents the mean of I_2

There are two important differences between SSD and Normalized SSD. Firstly, the pixel intensity values are subtracted from their respective means in both the image windows. This indicates that the entity being compared is not the intensity value but its deviation. Secondly, the pixel intensity values are normalized. This helps in comparing regions that have different brightness levels in the two images.

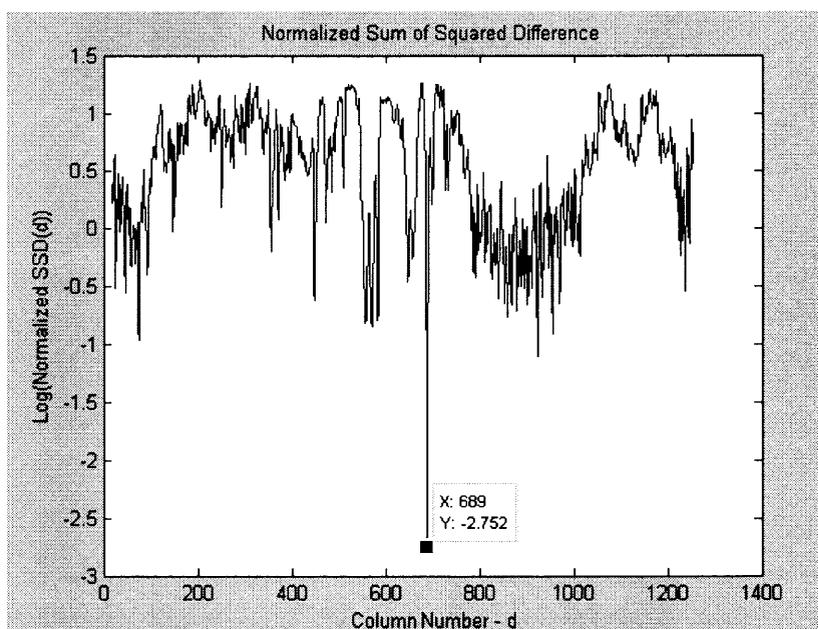


Figure 12: Normalized SSD log plot

The correspondence again indicated by the minimum is at the expected value $d = 689$. Also, log of the result is plotted to reduce the dynamic range. The change in plot shape is the effect of matching deviation instead of intensity.

2.2.4 First Order Cross-correlation

Cross correlation is often used in signal processing to determine the similarity between two images. In this method, the product of the intensity values is considered instead of difference as in the above methods. It is given by

$$\text{Cross-Correlation}(d) = \sum_{u,v} \left(\frac{I_1(u,v)}{\sqrt{\sum_{u,v} I_1(u,v)}} * \frac{I_2(u+d,v)}{\sqrt{\sum_{u,v} I_2(u+d,v)}} \right)^2$$

In the formula $I_1(u, v)$ represents the intensity value of pixel in the 1st image at the u^{th} row and v^{th} column and $I_2(u, v)$ represents the intensity value of pixel in the 2nd image at the $(u + d)^{\text{th}}$ row and v^{th} column.

The rationale behind using product is that if the intensity values of the pixels are similar, then the product will be roughly equal to the square of the intensity value of the reference pixel i.e. a large number. However, if the intensity values are dissimilar, then the product will be a smaller number. Hence, the algorithm states that maximum value of the cross correlation array represents the point of maximum correspondence.

The correspondence indicated at the maximum at $d = 689$ is within the expected range. However, it is easily observed that the distinction of correlation is not very high even after raising the result to the power of 10000 (to increase the dynamic range).

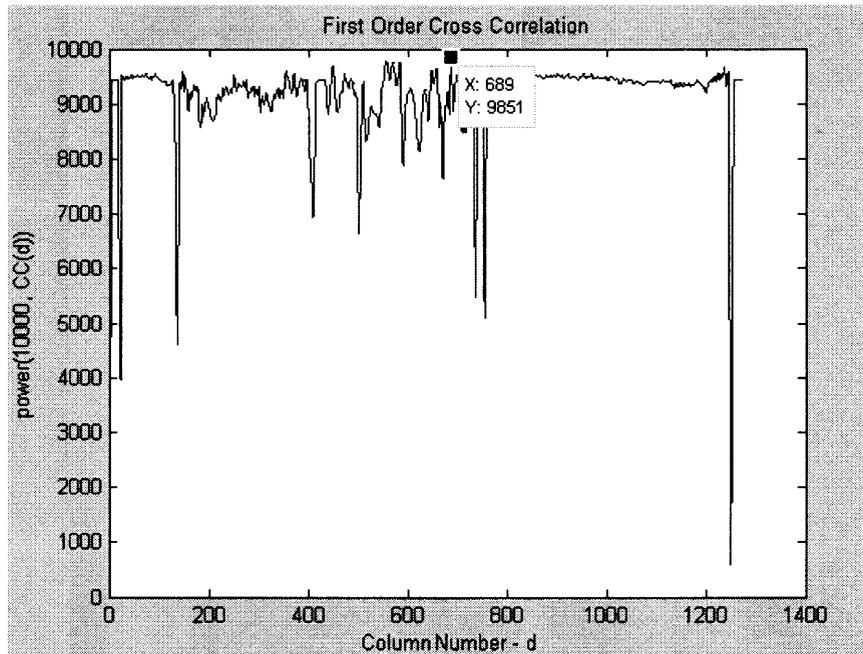


Figure 13: Cross correlation plot

2.2.5 Second Order Cross-correlation

This method is also called Normalized Cross-correlation. It attempts to match the deviation in the intensity values of the pixels. It is similar to Normalized SSD in the sense that they are both second order algorithms.

$$\text{Normalized CC}(d) = \frac{\sum_{u,v} (I_1(u,v) - \bar{I}_1) \cdot (I_2(u+d,v) - \bar{I}_2)}{\sqrt{\sum_{u,v} (I_1(u,v) - \bar{I}_1)^2 \cdot (I_2(u+d,v) - \bar{I}_2)^2}}$$

In the formula $I_1(u, v)$ represents the intensity value of pixel in the 1st image at the u^{th} row and v^{th} column and $I_2(u, v)$ represents the intensity value of pixel in the 2nd image at the $(u + d)^{\text{th}}$ row and v^{th} column.

This is a robust algorithm that makes use of the statistical properties of the windowed regions under comparison. This is particularly important as the way an object

appears in the left and right images is always slightly different due to the relative orientations of the cameras. By considering the statistical properties of the regions, the algorithm allows for this slight difference during comparison.

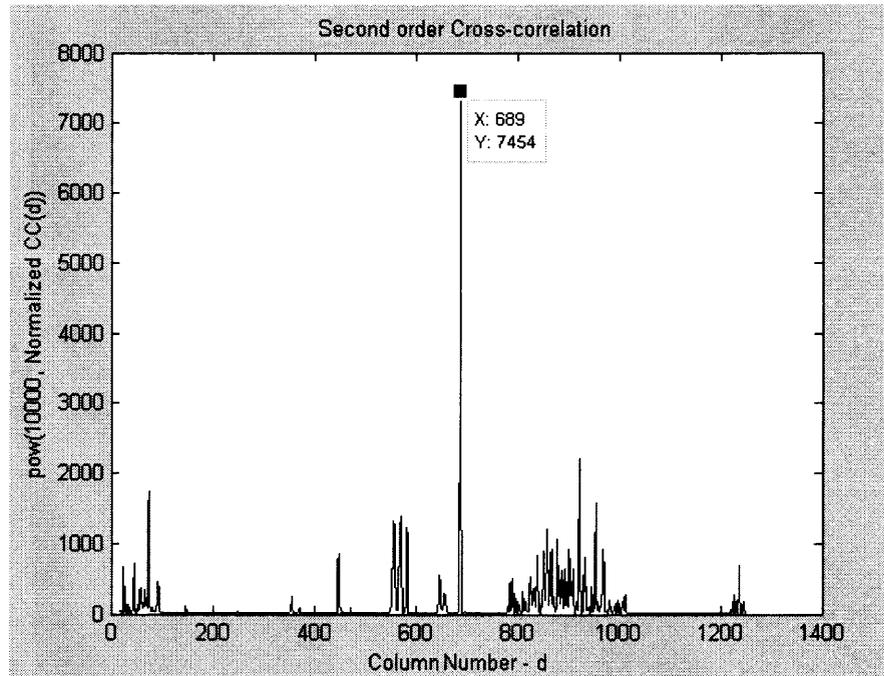


Figure 14: Normalized Cross correlation plot

The correspondence as expected is at $d = 689$. The correlation value actually varies between -1 and +1 with +1 indicating maximum correlation. However, it is raised to the power of 10000 to increase the dynamic range

To summarize the algorithms, the following sub-plot aggregates all the individual plots. The correspondence at $d = 689$ is highlighted in each plot for clarity.

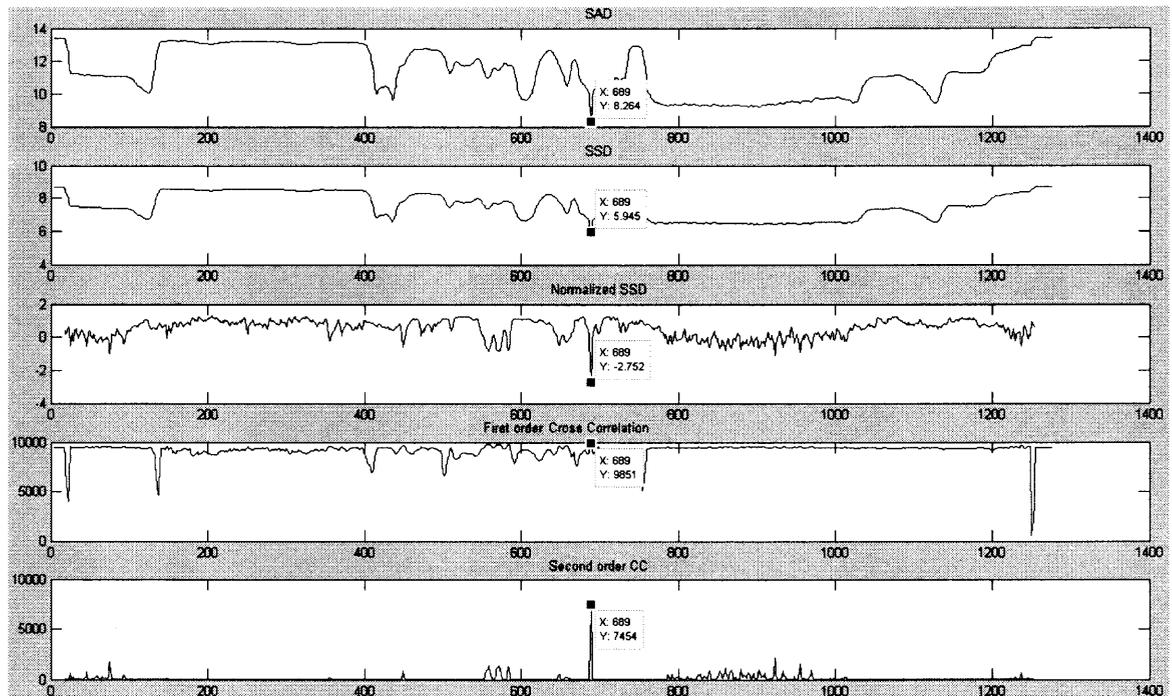


Figure 15: Summary of correlation plots

2.3 Range Limiting

In the previous section an attempt was made to match a set of pixels in the left image represented by a fixed 7x7 window to a set of pixels in the right image represented by a sliding 7x7 window that was shifted by one pixel every iteration. Possible correspondence was calculated after each shift. By default the entire horizontal dimension (1280p) was searched for possible correspondence. This sometimes gives rise to false correspondences when there are similar objects in the scene. Further, in the real world such a wide range is generally not necessary. Hence, it is essential to find valid search ranges.

2.3.1 Start Range

In this project the search range has been limited to just the width of the target mask. As a result, the start of the search range is set to 350 which is the lowest pixel at which the mask appears in all the camera images.

2.3.2 End Range

In rectified stereo images it can be observed that for any given point $x_r < x_l$, where x_r & x_l are the x coordinates (by pixel number) of the projections of the point on the right and left images. This is because of the lateral shift in the positions of the cameras relative to the object.

This may be visualized better in terms of a stationary point and a camera sliding along the horizontal axis. If the camera is moving to the right, the projection of the stationary point in the image will move left and vice versa. Extending the concept to two cameras, if there is a camera C_1 to the right of the given camera C_2 then it will be observed that the projection of a stationary point appears to shift left in the image from camera C_2 as compared to the image from C_1 .

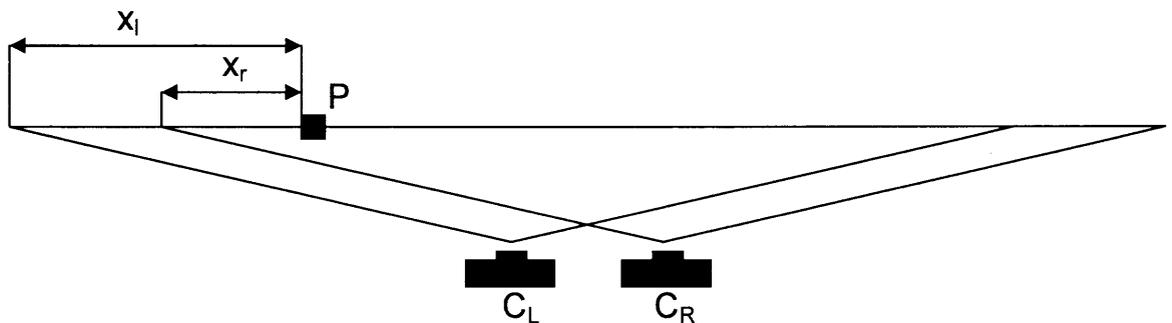


Figure 16: Range $x_r < x_l$

To illustrate, consider a point 'P' in a scene captured by left and right cameras C_L and C_R respectively. It is clearly seen from the figure that due to the shift in camera position, the point P has relatively shifted projections in the captured images. Further, as discussed before, projection of the point appears to shift left in C_R 's image. Hence, it is closer to the image origin in the right image than the left i.e. $x_r < x_l$. It is obvious from the discussion that when searching in the right image for a portion that corresponds to the windowed portion in the left image, it is sufficient to search till x_l as it is known that the projection in the right image would have shifted to the left of projection in the left image.

The following example clearly establishes the advantages of range limiting. In this example a match is attempted to be found for a point at (559, 642) in the right image. Manually, the corresponding point is found to be at (516, 642) in the left image. The algorithm used is SSD.

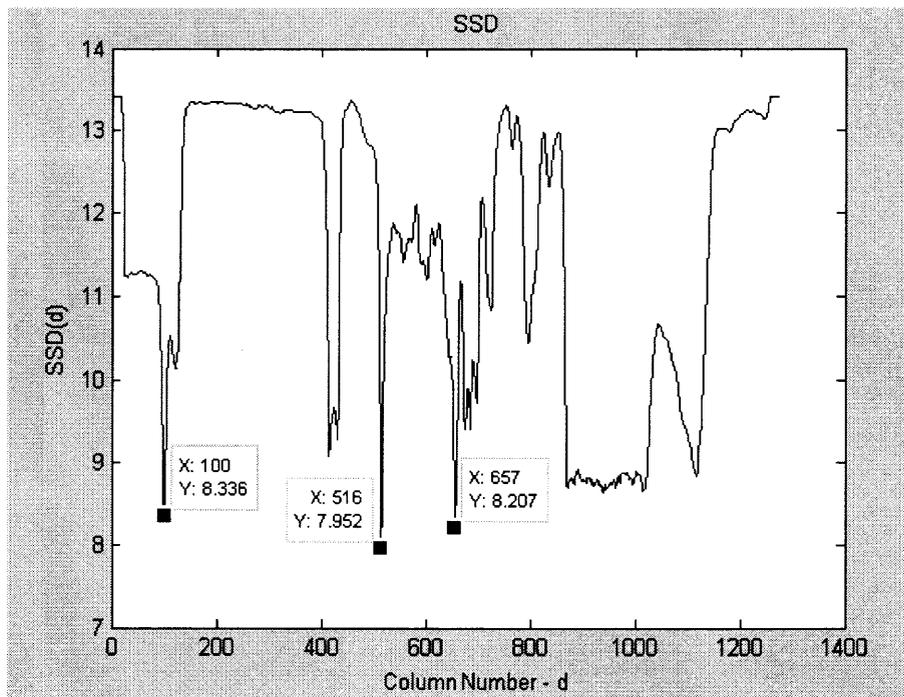


Figure 17: Full range with possible false correspondences

In the full range plot the minimum at $x = 516$ is seen, but there are close competitors at $x = 100$ and $x = 657$. This may lead to false identifications in case a threshold is applied. By applying the range limiting techniques described above, the following plot is obtained. The search range starts at 350 as it is only required to search within the region of the mask. Also, the search range ends at $d = 559$ (i.e. x_l) as $x_r < x_l$.

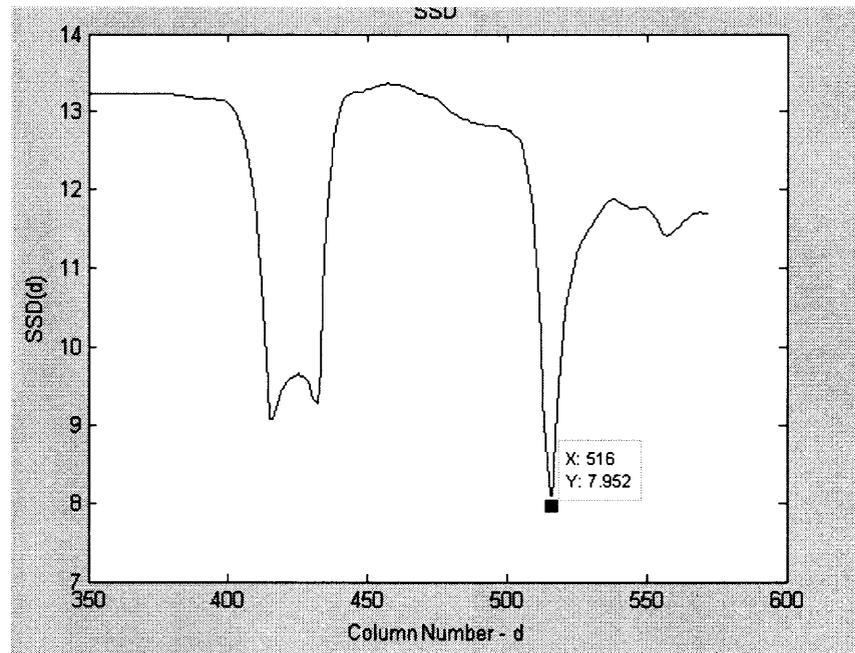


Figure 18: Limited range with false correspondences removed

2.4 Edge Enhancement

Edge enhancement of images generally involves boosting the high frequency components of the image. This tends to highlight the intensity changes in an image. This in-turn improves the performance of the second order matching algorithms as it accentuates deviation from the mean.

This preprocessing is achieved using the 'sobel' filter in the 'fspecial' function in MATLAB. The vertical and horizontal edges are extracted individually and combined to obtain edge enhanced images as shown.

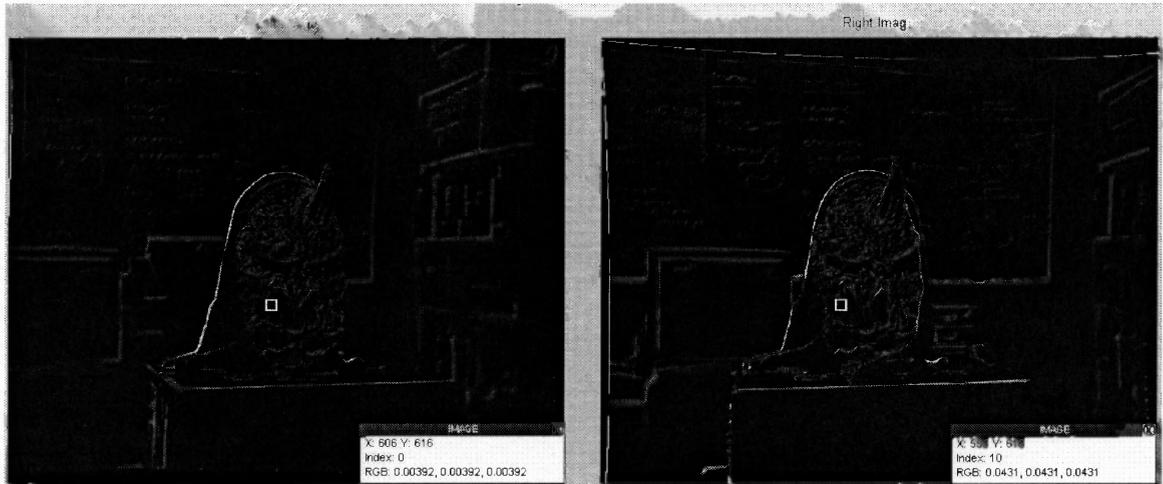


Figure 19: Edge Enhanced Images

The points (606, 616) in the left image and (555, 616) in the right image both represent a relatively smooth surface with small deviations from mean. In case of the original images this causes Normalized SSD and CC methods to fail. This is seen by the false correspondence at $x = 600$ in the plot below.

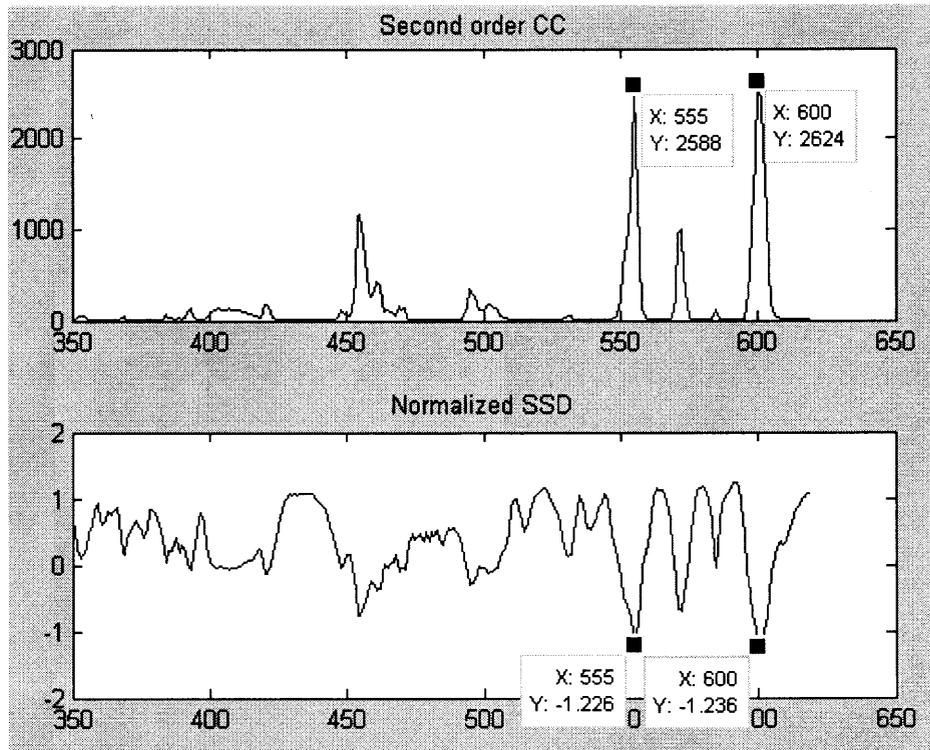


Figure 20: Plot for original images with false correspondence

Edge enhancing the images increases the deviation from mean and allows Normalized SSD and CC a better chance of finding a match. The plot below shows the false correspondence eliminated in the case of preprocessed images.

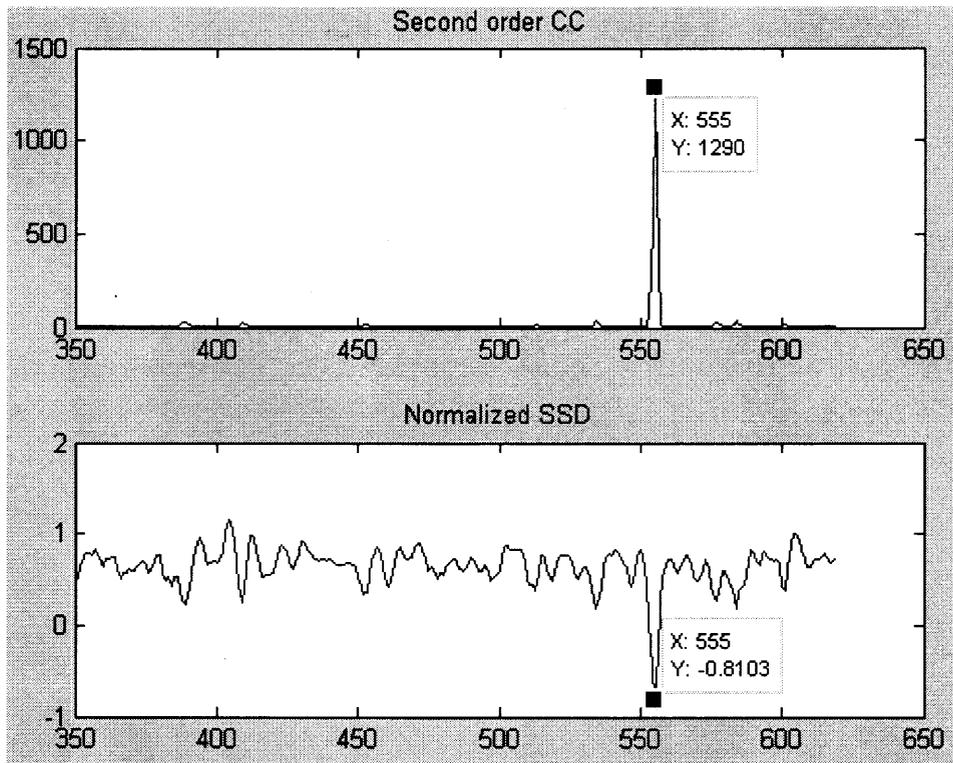


Figure 21: Plot for edge enhanced images

The problem with this method of preprocessing is its inconsistency amongst algorithms. It improves performance at certain surface-like regions for the second order algorithms. But it provides less than desirable output for SAD and SSD.

2.5 Evaluation

This section presents an analysis of the matching algorithms.

2.5.1 Intensity Difference

The intensities of certain portions of a scene are captured differently in stereo images. This is generally due to relative change in the position of a light source with respect to the image. This is shown in the figure.

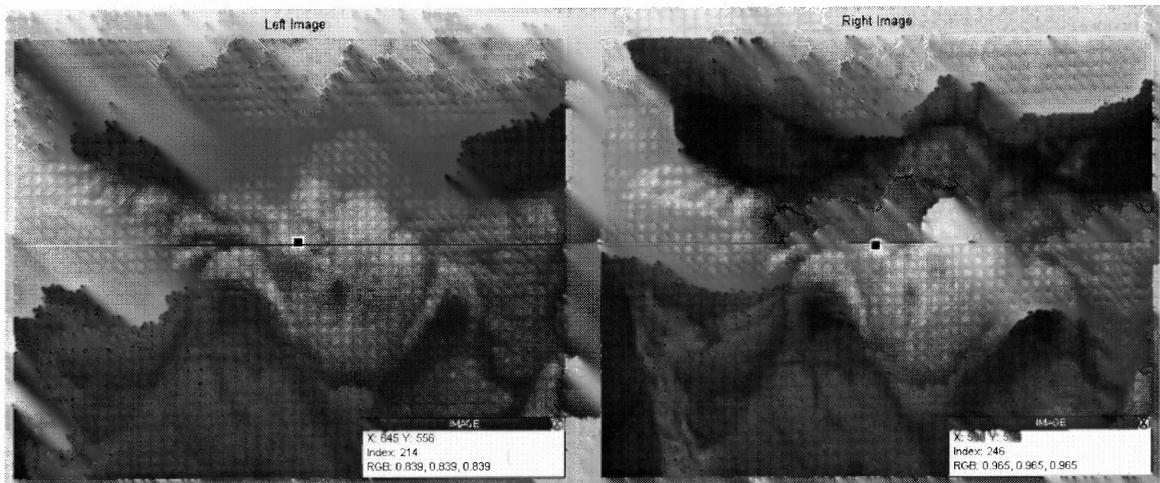


Figure 22: Left and Right Images with intensity difference highlighted

It is evident that the point at (643, 556) in the left image is less bright than the corresponding point at (588, 556) in the right image. This misleads matching algorithms such as SAD and SSD that simply compare the intensity values of the pixels. However, algorithms such as Normalized Cross-correlation and Normalized SSD tend to perform better in these circumstances due to the fact that they are *normalized* i.e. each pixel in the window is divided by a factor proportional to the energy in the window. In brighter windows, this factor is large and in less bright windows the same is small. This *normalizes* the pixel values for fairer comparison. The plot confirms this line of reasoning. Normalized CC and SSD indicate correspondence at $x = 589$ which is acceptable, while SAD and SSD return $x = 623$ which is clearly wrong. Note the epipolar line is highlighted for clarity.

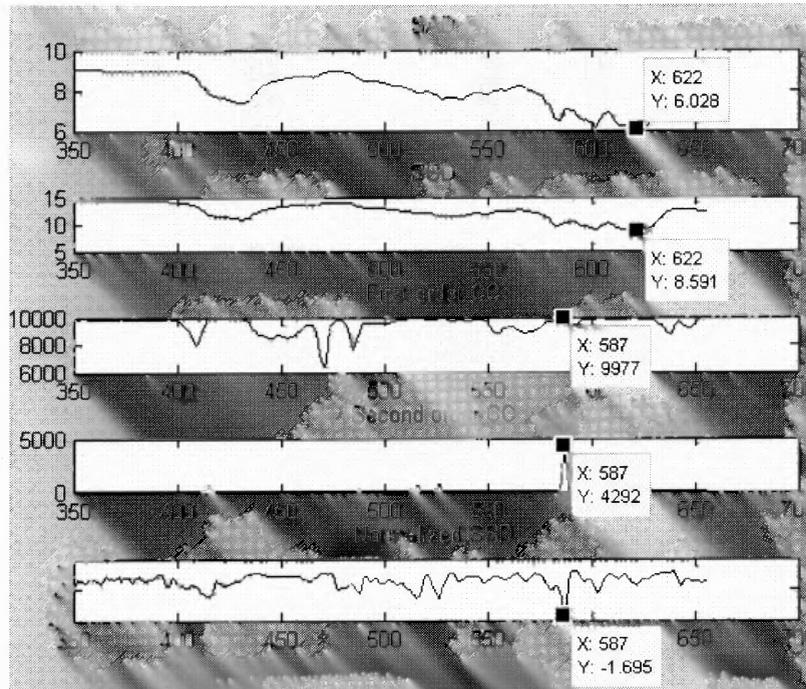


Figure 23: Plot in case of intensity difference

2.5.2 Change in Shape

The shape of certain portions of an object changes slightly in projections on the stereo images. This is due to the different orientations of the cameras relative to the target. This is clear in the figure.



Figure 24: Left and Right Images showing changed shape

Consider the highlighted crease in the target mask at (657, 630) in the left image and (605, 630) in the right image. It is prominent and dark in the right image. However, in the left image it is distorted and lighter. This causes algorithms such as SAD and SSD which are first order in nature to fail. A simple comparison of intensity values is inadequate to identify the correspondence of these points. Hence, Normalized SSD and Cross-correlation use deviation from mean to determine similarity. This gives a crude statistic measure of intensity distribution in the window pixels. This enables these matching methods to identify correspondences even when first order techniques fail miserably. This is confirmed by the plot below. SAD and SSD give non-coherent correspondences at 554 and 433 respectively. Normalized CC and SSD indicate correspondence at 605 as expected.

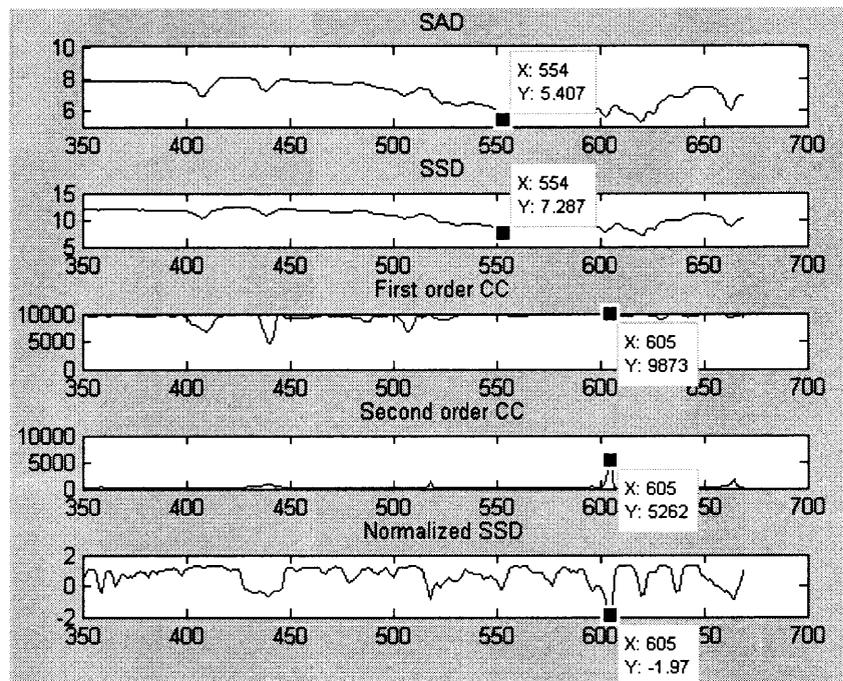


Figure 25: Plot in case of shape change

2.5.3 Surfaces

In portions of images that represent images, the gradient is smooth. Hence, the variation is gradual and small. This causes the second order methods that compared deviation from the mean to fail.

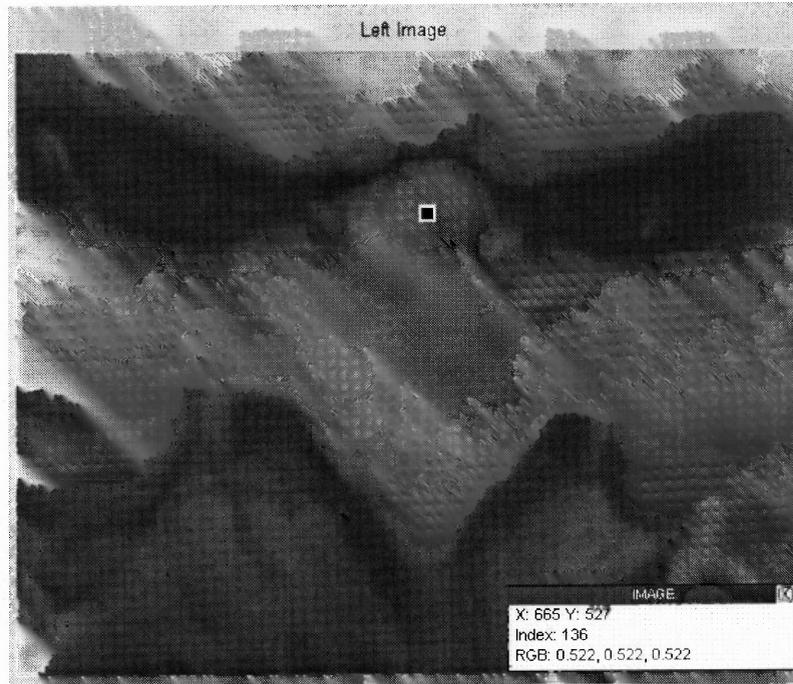


Figure 26: Surface example

The surface around the highlighted point in the image is pretty smooth. To illustrate the following is the intensity values of the 7x7 window of pixels around the point.

```

window_pixels =
  136  134  139  139  138  138  130
  133  136  140  139  137  134  128
  135  138  139  138  135  133  132
  136  137  137  136  134  135  134
  135  137  134  132  132  131  131
  134  135  132  131  131  129  128
  134  135  132  134  133  131  129

```

It is clear to see that there is very little variation in the windowed pixels with a maximum of 140 and minimum of 128 (range being 0 to 255). Hence, the deviation from the mean is very small.

```

deviation_from_mean =
    2     0     5     5     4     4    -4
   -1     2     6     5     3     0    -6
    1     4     5     4     1    -1    -2
    2     3     3     2     0     1     0
    1     3     0    -2    -2    -3    -3
    0     1    -2    -3    -3    -5    -6
    0     1    -2     0    -1    -3    -5

```

Algorithms trying to match this deviation from mean find fail due to these small values as even small differences lead to large error values (due to normalization). This is the main reason it is found that SAD and SSD perform better than Normalized SSD and Cross-correlation in surfaces. In the plot below, SAD and SSD indicate correct correspondence at $x = 612$.

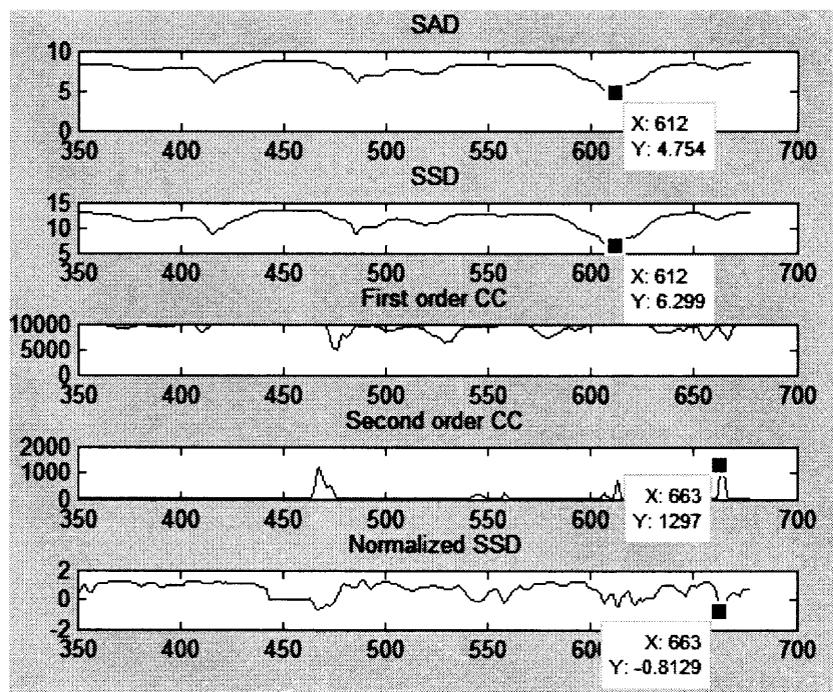


Figure 27: Plot for surface correspondence

2.5.4 Occlusion

A major problem in stereo correspondence is occlusion. This occurs generally at points of depth transitions due to relative change in the position of a light source with respect to the image. This is shown in the figure.

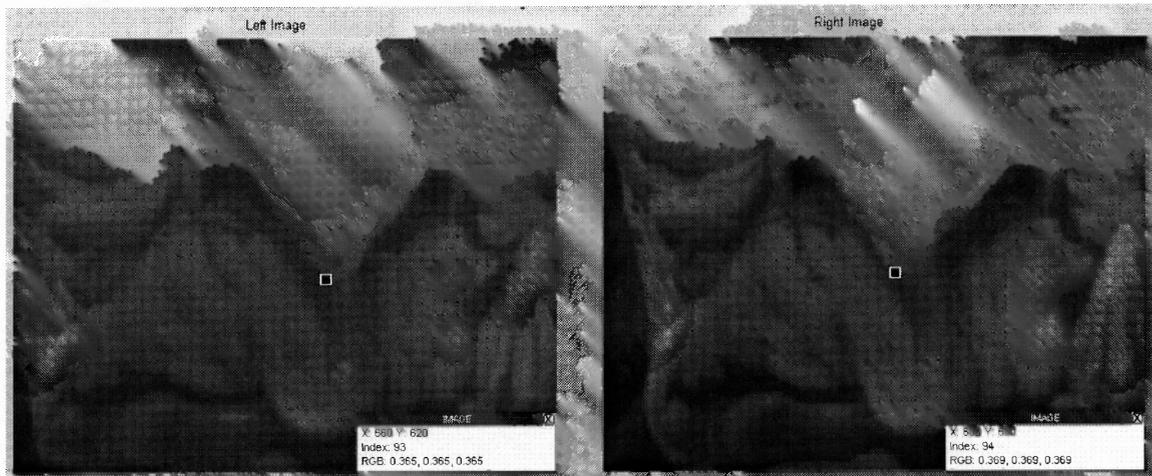


Figure 28: Occlusion Example 1

In the pictures displayed, the tip of the nose and upper lip below it are clearly at two different distances from the camera pair. This has an interesting effect on the captured images. The distance the tip of the nose is displaced in the two images is different from the distance the upper lip is displaced. This can be observed by noting the distance between the tip of the nose and crease in the surface below it at the highlighted point. In the left image, the crease is distinctly to the left of the tip of the nose. However, in the right image the crease is occluded or partially covered by the tip of the nose in the right image. Hence, a point that is seen in the left image has no corresponding point in the right image. This confuses the matching algorithms to a great extent. Worse it causes them to return wrong correspondences. Presented below is another example of occlusion. The point (701, 383) on the left image shows a point on the left image to the right of the

horn in the mask. However, this is completely unsighted bin the right image. This is reflected in the result of the matching algorithms in the plot. Each of the matching algorithms provides different possible points of correspondence. Note the epipolar line marked in black.

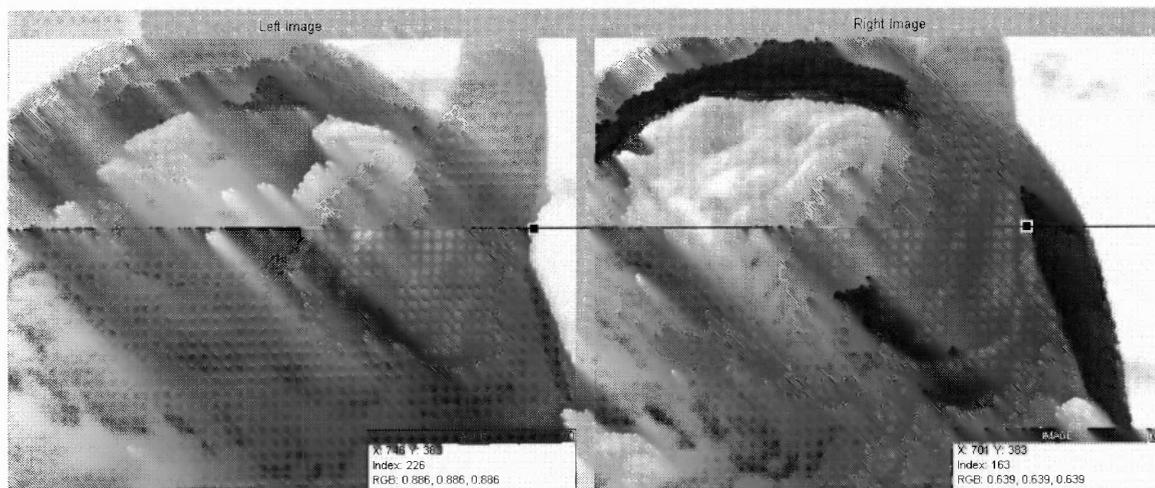


Figure 29: Occlusion Example 2

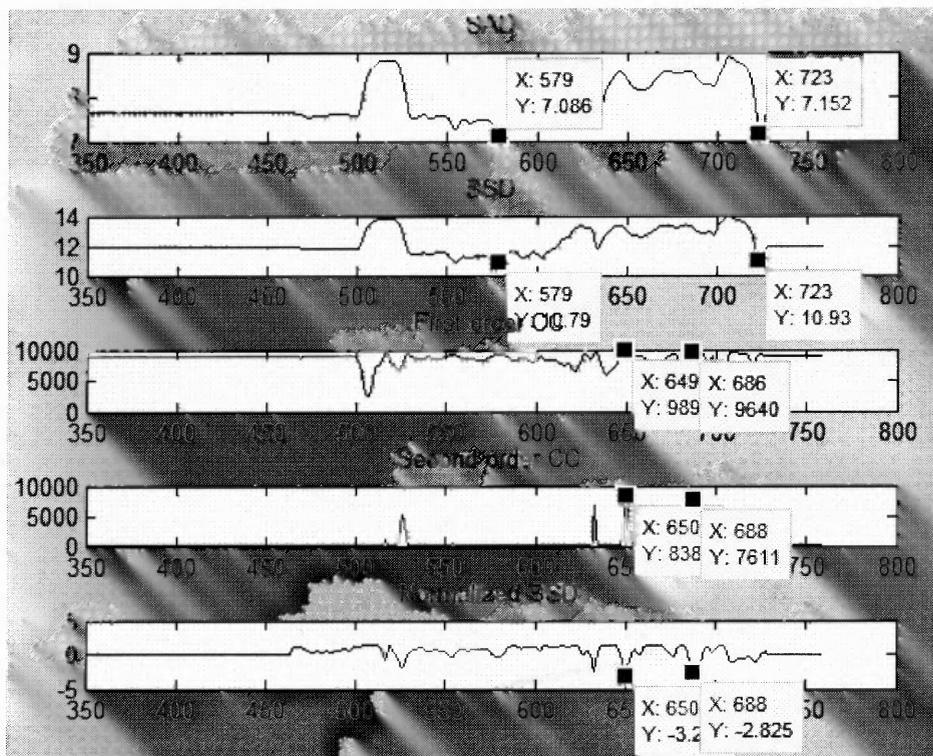


Figure 30: Plot in case of occlusion

But if the occluded part of the image is seen by the rest of the images it can be calculated using the method of propagation.

2.5.6 Uniqueness

Another common problem of correspondence is lack of uniqueness. To find the correspondence of a point, the considered point has to be *unique* to a certain extent. If this criterion is not satisfied the matching algorithms tend to find a large number of possible matches. For the sake of understanding, consider the following extreme case.

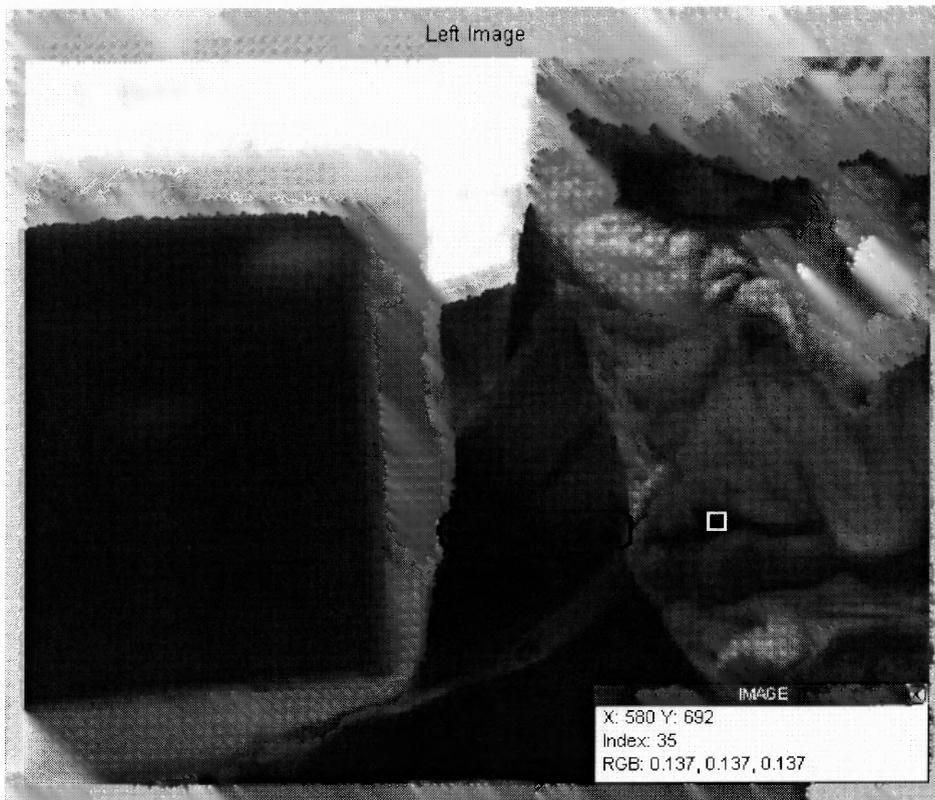


Figure 31: Occlusion Example 1

In the picture, the highlighted point (crease in the upper lip) is predominantly black. This poses a problem as the background has objects that are dark (blue box). The lack of uniqueness of the considered point in the surrounding region or stated differently, the similarity with certain background objects causes the correspondence to fail. As

shown in the plot below, the match at $x = 470$ is overshadowed by the match at $x = 423$.

The latter match is the region of the veil and is incorrect.

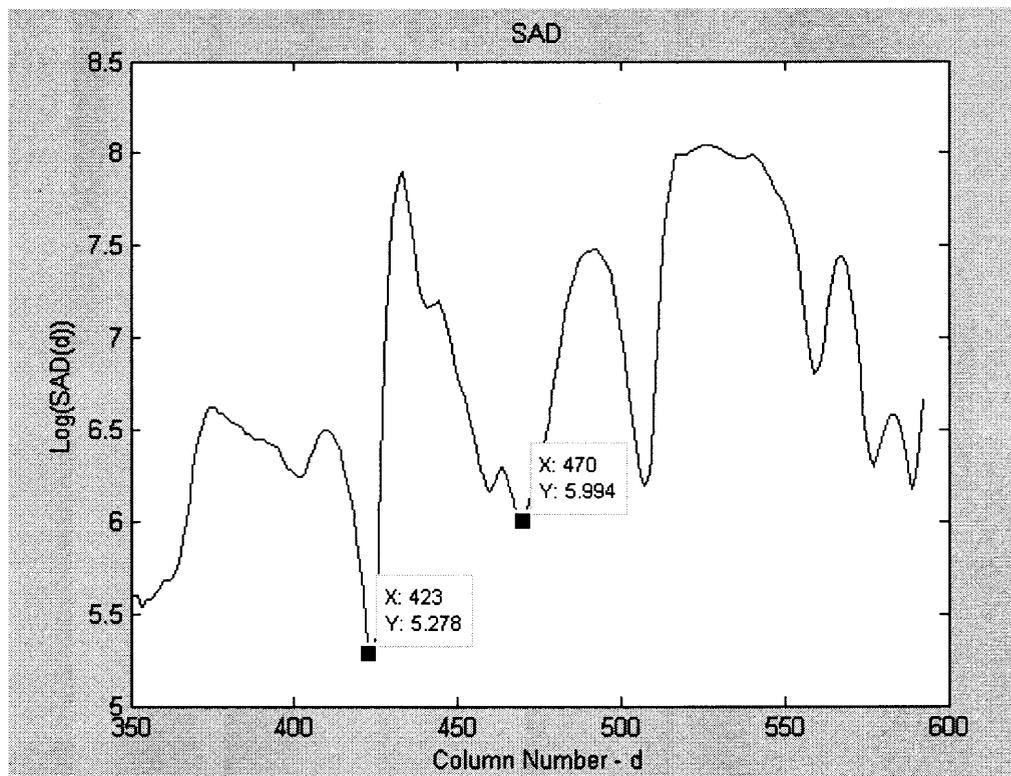


Figure 32: Incorrect correspondence due to lack of uniqueness

Second Order CC and Normalized SSD work relatively better even in these cases.

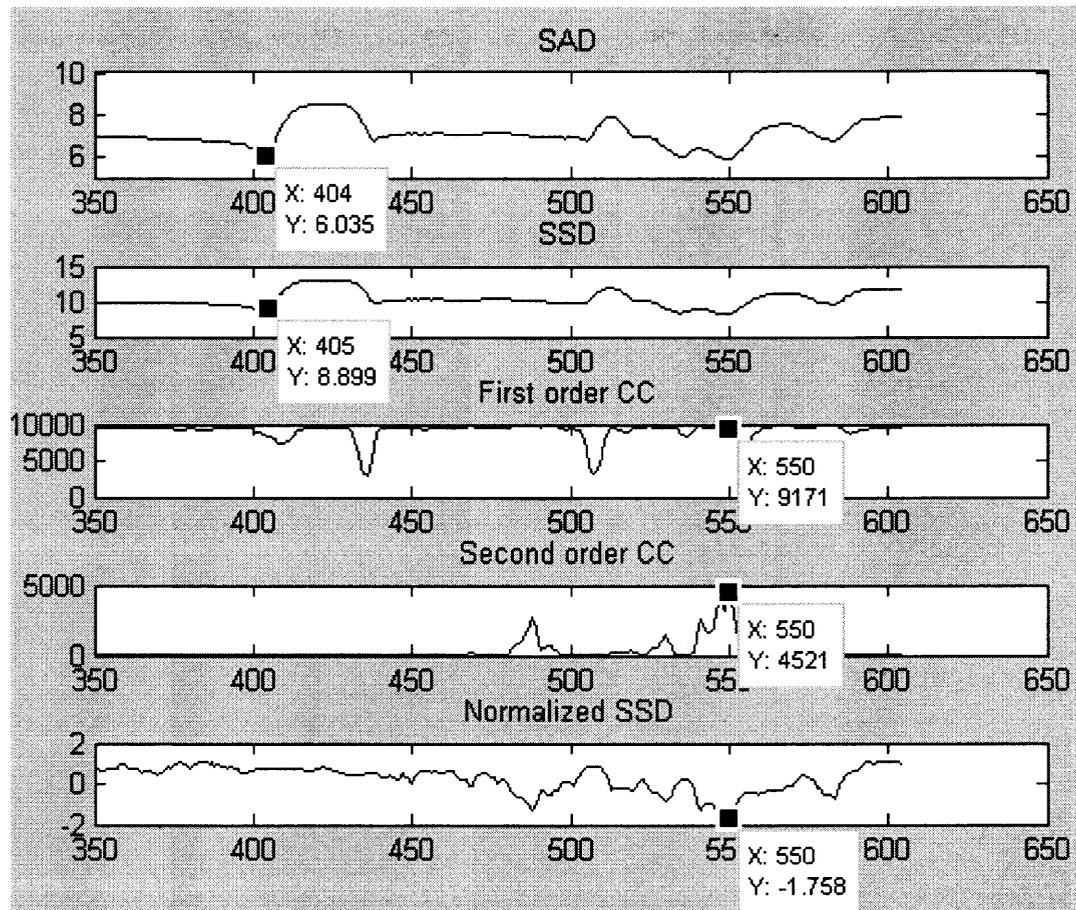


Figure 33: Non-unique region where Second Order works better

CHAPTER 3

PROCEDURE

3.1 Apparatus

The experimental setup for this project involves four cameras placed along a straight line. Placed in front of them is a target, a Halloween mask. As a result of their different orientations relative to the target (the mask), the cameras capture images of the mask from different angles. For ease of reference, the cameras are name A to D from left to right.

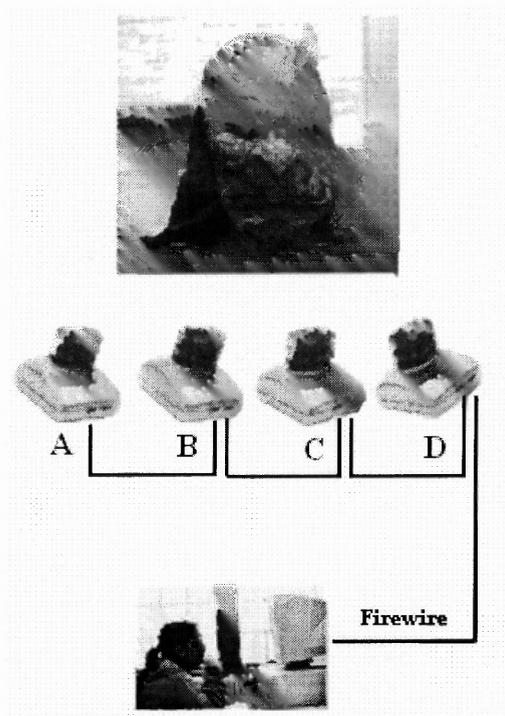


Figure 34: Experimental Setup

- **Hardware:** Four Pixelink A642 cameras connected via Fire-wire to a PC running Windows 2000
- **Software:**

- Frame grabber MATLAB interface drivers from Pixelink
- Camera Calibration Toolbox from Caltech University [8]
- MATLAB 7.0 development environment

3.2 STEP 1 – Image Acquisition

The camera drivers provided by Pixelink were compatible with MATLAB 7.0 Frame Grabber Interface (pIFGI). As the images had to be obtained from all the four cameras simultaneously to avoid any manual error during the process, a script named *capture_image.m* was created.

Images may be captured and saved in the work folder of MATLAB simply by running this script. It is written such that ‘n’ (user specified) number of images can be captured from each camera and stored. The figure depicts the four images taken by running the script.

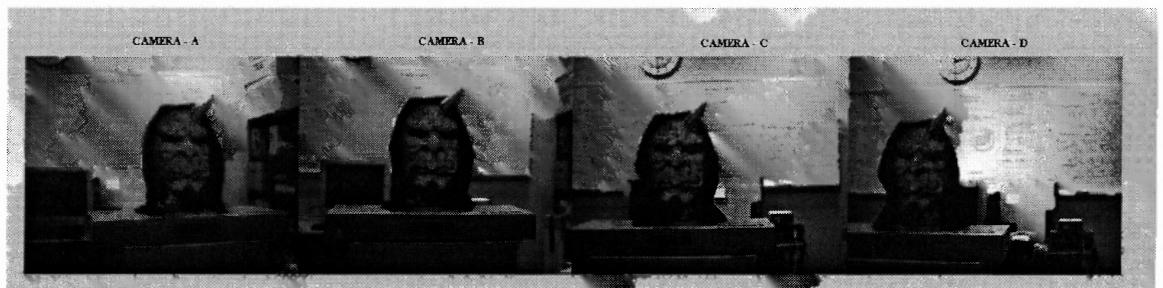


Figure 35: Captured images

3.3 STEP 2 – Calibration

Calibration as mentioned before is a technique used to determine a camera’s intrinsic and extrinsic parameters. In this experiment the calibration and rectification are done using

the Camera Calibration Toolbox by Klaus Strobl, Wolfgang Sepp, Stefan Fuchs, Cristian Paredes and Klaus Arbter from the Institute of Robotics and Mechatronics [8].

The Calibration toolbox GUI is launched by running *calib* command from the command prompt. The toolbox's calibration algorithm is based on images of a checkerboard in different orientations. It attempts to match squares in different images to determine the necessary parameters.

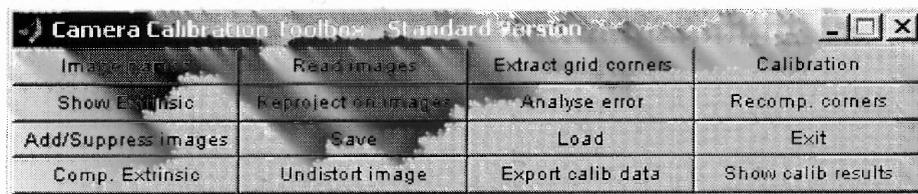


Figure 36: Calibration Toolbox GUI

The toolbox requires a minimum of fourteen different images per camera for this purpose. The large number is to minimize the error that may occur in some orientations during the process. All of the above images are placed in a folder and the path is set. The Image Names function in the toolbox reads all the images in the folder. To rectify the images each camera set has to be calibrated separately and hence the base names need to be different. In this case, respective camera names are used as the base name

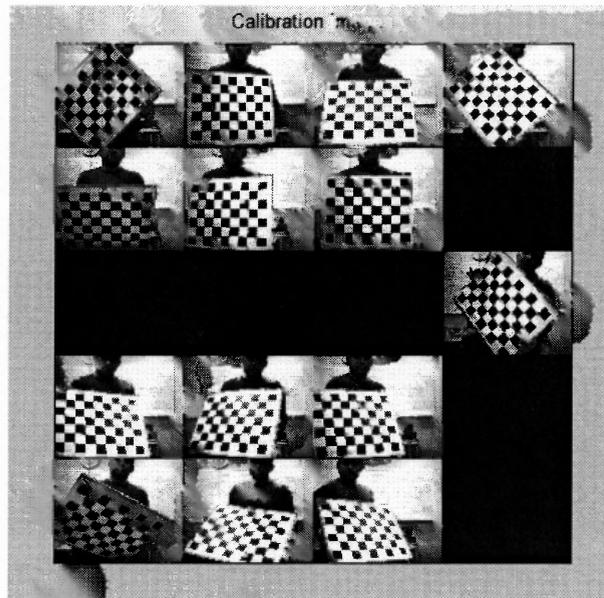


Figure 37: Checkerboard images used for calibration

The images read are individually chosen for corner extraction. The user has to select 4 extreme corners on the checkerboard. Once the corners are marked manually, the remaining corners are automatically generated and the calculations are saved for further analysis. It is important to note that the 1st corner has to be marked with respect to the origin point of the reference frame attached to the grid and the rest of the 3 points follow the clockwise direction. This is done to calibrate external multiple cameras such that the relative position of these cameras in space can be computed with a common reference frame. Calibration is per camera. The calculated parameters are saved as *Calib_Results.mat* in the work folder by default.

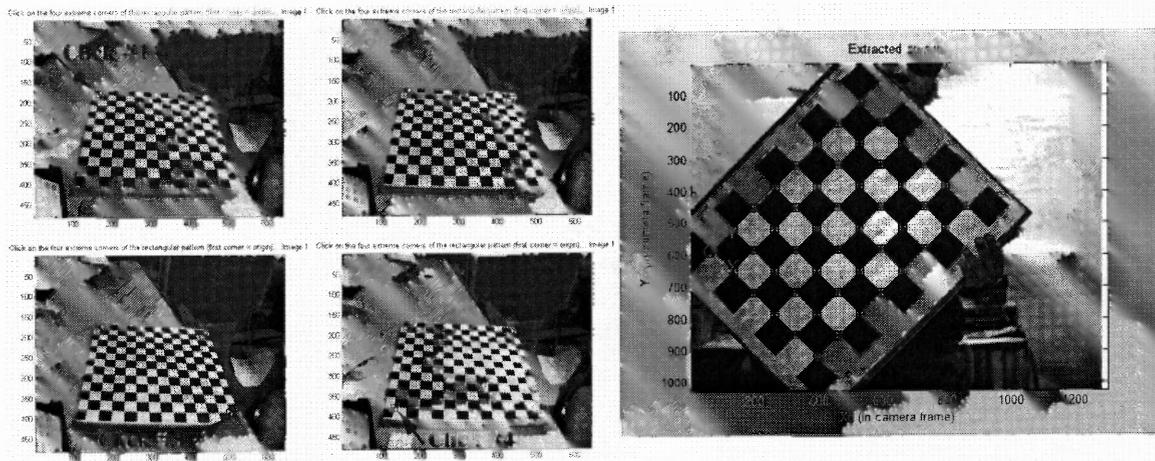


Figure 38: Corner selection and Extracted corners

3.4 STEP 3 – Rectification

Rectification is achieved using version 5 of the camera calibration toolbox. The GUI is launched by typing `stereo_gui` in the MATLAB command window. For rectification it is necessary to recalibrate the camera now with respect to the 2 cameras being compared.

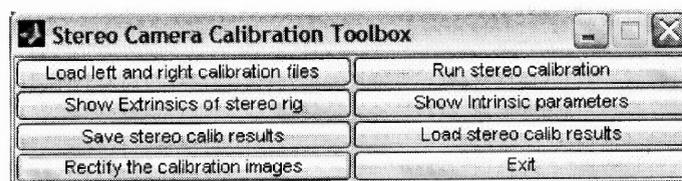


Figure 39: Stereo rectification interface

The toolbox requires the intrinsic parameters and extrinsic parameters of the camera pair being rectified. These are available in the `.mat` files created by the calibration toolbox. The figure shows the new set of parameters (rotation and translation vectors) calculated by the toolbox for the given camera pair.

Stereo calibration parameters after optimization:

Intrinsic parameters of left camera:

```
Focal Length:      fc_left = [ 2217.40469  2207.38815 ] ± [ 15.86385  15.80230 ]
Principal point:   cc_left = [ 707.48062  522.77477 ] ± [ 37.04758  26.84945 ]
Skew:             alpha_c_left = [ 0.00000 ] ± [ 0.00000 ] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:       kc_left = [ -0.22532  -0.19260  0.00652  0.00172  0.00000 ] ± [ 0.07178
                                                                0.89554  0.00242  0.00280  0.00000 ]
```

Intrinsic parameters of right camera:

```
Focal Length:      fc_right = [ 2232.03191  2219.58251 ] ± [ 16.23541  16.34061 ]
Principal point:   cc_right = [ 647.30415  539.14576 ] ± [ 39.64906  28.33761 ]
Skew:             alpha_c_right = [ 0.00000 ] ± [ 0.00000 ] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:       kc_right = [ -0.24727  -0.06544  0.00436  0.00153  0.00000 ] ± [ 0.06887
                                                                0.73612  0.00287  0.00259  0.00000 ]
```

Extrinsic parameters (position of right camera wrt left camera):

```
Rotation vector:   om = [ -0.01934  0.15514  -0.03315 ] ± [ 0.01574  0.02229  0.00178 ]
Translation vector: T = [ -175.88357  6.26818  17.95907 ] ± [ 1.29415  1.05156  9.11968 ]
```

Figure 40: Stereo Calibration parameters

The rectified images corresponding to the two cameras are created and stored in the same folder as the acquired images and the calibration results. The figures below are the rectified images of Camera 70216 (A) and Camera 70215 (B). A laser beam was focused to indicate the projection of a laser point on both images (highlighted by white box). It can be seen that the y-axis is the same for both. Also, straight lines are drawn at the tip of the horn and beard. It is seen that the y-axes coincides for these points coincide too. This proves that the epipolar lines of both images are aligned satisfying the rectification condition.



Figure 41: Rectified images

3.5 STEP 4 – Correspondence

The code directory contains a script called *generate_result.m*. On execution, the script takes the image names and pixel co-ordinates from the user. Shown below is an example.

```
>> generate_result
Enter name of left image:Cam_70216_Pic_rectified2.bmp
Enter name of right image:Cam_70215_Pic_rectified2.bmp

Enter the y co-ordinate value (y):602
Enter the x co-ordinate value (x):742
Enter the rectification compensation value:0
```

It generates five graphs for the result of each algorithm.

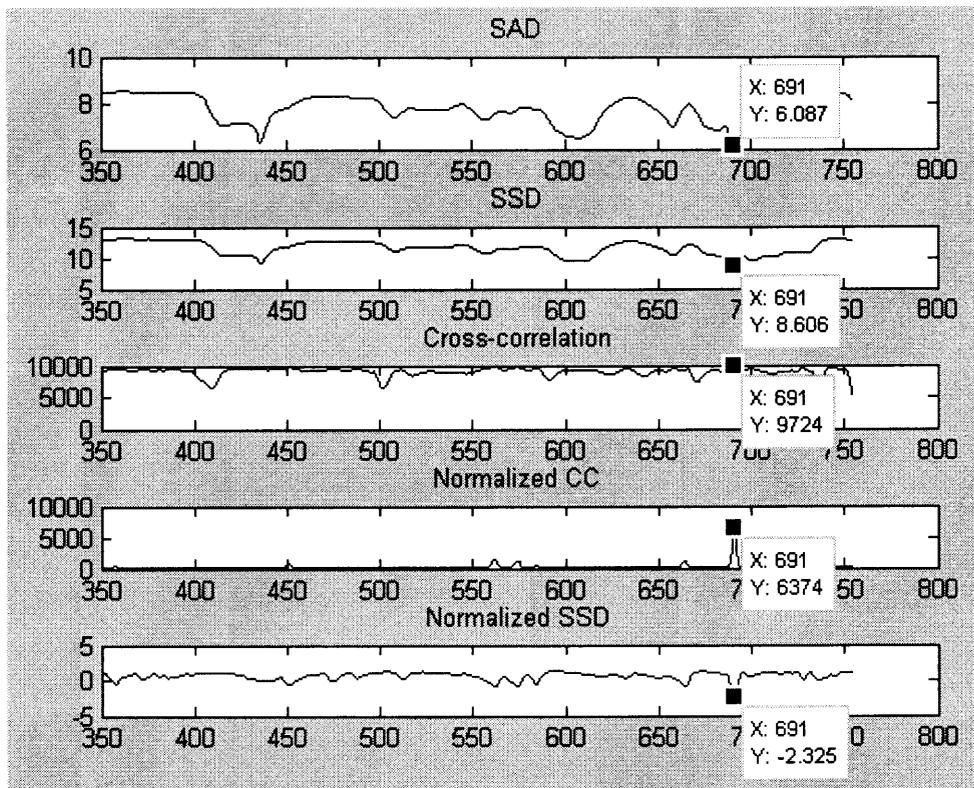


Figure 42: Example correlation plot

3.6 STEP 5 – Triangulation

Triangulation is also obtained using a function in the Caltech Calibration Toolbox. There are two steps involved in the process. The first is to load the calibration

Calib_Results_stereo_rectified.mat files using the ‘Load’ function. The second is to execute the *stereo_triangulation* function.

```
xL =  
    497  
    738  
>> xR=[497 494] '  
xR =  
    497  
    494  
>> [XL, XR] = stereo_triangulation(xL,xR,om,T,fc_left,cc_left,kc_left,  
                                   alpha_c_left,fc_right,cc_right,kc_right,alpha_c_right)  
  
XL =  
   -64.2611  
    2.0786  
   743.4700  
XR =  
   -66.5952  
    30.0227  
   742.9270
```

Figure 43: Stereo triangulation function call

CHAPTER 4

RESULTS

The 3D camera system contains four cameras labeled A, B, C and D (from left to right). There are results presented here for 5 pairs of cameras viz. AB, BC, CD, AC and AD. The following notation has been used

- $x(L)$: x-co-ordinate of the point of the interest in the left image
- $x(R)$: x-co-ordinate of the point of the interest in the right image
- Y : y-co-ordinate of the point of interest in both the images due to epipolar geometry.
- W : Algorithm Working
- NW : Algorithm fails to find the $x(R)$ in the 2nd image

4.1 AB

4.1.1 Captured Images



Figure 44: Images from cameras A & B

4.1.2 Rectified Images



Figure 45: Rectified images from cameras A & B

4.1.3 Correspondence Tabulation

Table 1: Correspondence Result for AB

Camera Pair – AB								
No.	x (L)	y	x (R)	SAD	SSD	N. SSD	CC	N. CC
1.	719	611	326	W	W	W	W	W
2.	691	578	579	W	W	W	W	W
3.	663	553	428	NW	NW	W	W	W
4.	720	604	539	NW	NW	W	W	W
5.	732	612	421	NW	NW	W	W	W
6.	810	706	353	W	W	W	W	W
7.	729	618	379	W	W	W	W	W
8.	712	608	296	W	W	W	W	W
9.	772	663	426	W	W	W	W	W
10.	771	654	365	W	W	NW	W	NW

It may be seen from the table that correspondence for the camera pair AB is on the stronger side. This is largely because of the relatively small distances between the cameras. Certain entries in the table show SAD and SSD failing. This is generally due to intensity difference or change in shape as discussed the preceding chapters.

4.2 BC

4.2.1 Captured Images

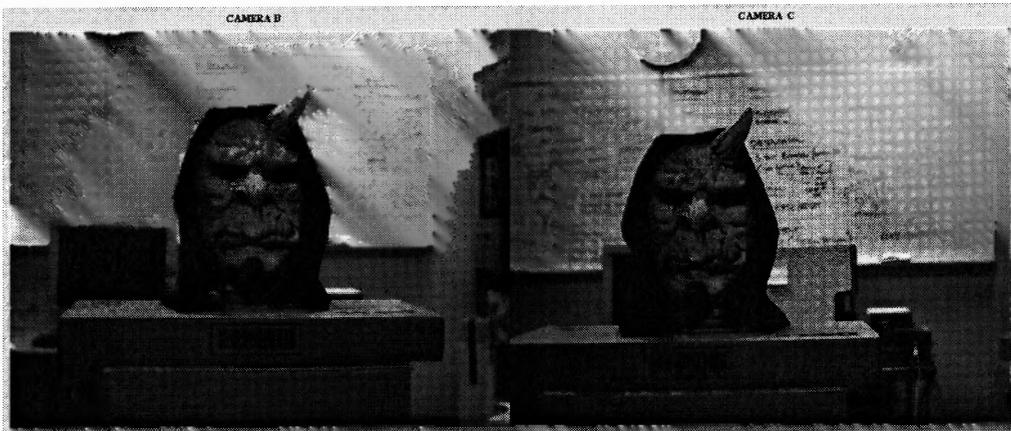


Figure 46: Images from cameras B & C

4.2.2 Rectified Images

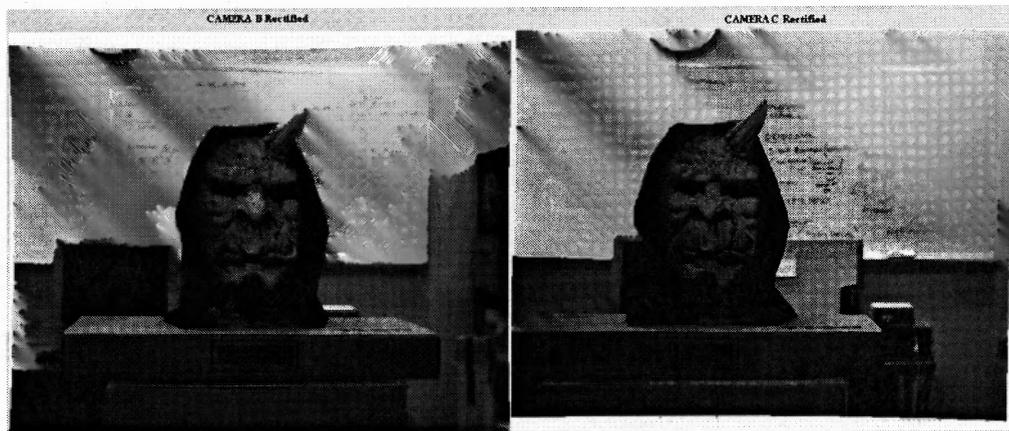


Figure 47: Rectified images from cameras B & C

4.2.3 Correspondence Tabulation

Table 2: Correspondence Result of BC

Camera Pair – BC								
No.	x (L)	Y	x (R)	SAD	SSD	N. SSD	CC	N. CC
1.	703	587	488	NW	NW	NW	NW	NW
2.	622	599	440	NW	NW	W	NW	W
3.	616	506	355	W	W	NW	NW	NW
4.	639	534	300	W	W	W	W	W
5.	630	521	329	W	W	NW	W	NW
6.	607	489	558	W	W	NW	W	NW
7.	723	610	568	W	W	NW	W	NW
8.	668	549	621	W	W	W	W	W
9.	662	552	357	W	W	W	W	W
10.	552	443	373	W	W	W	W	W

The results for camera pair BC are very similar to that for camera pair AB.

Correspondence was strong as before.

4.3 CD

4.3.1 Captured Images

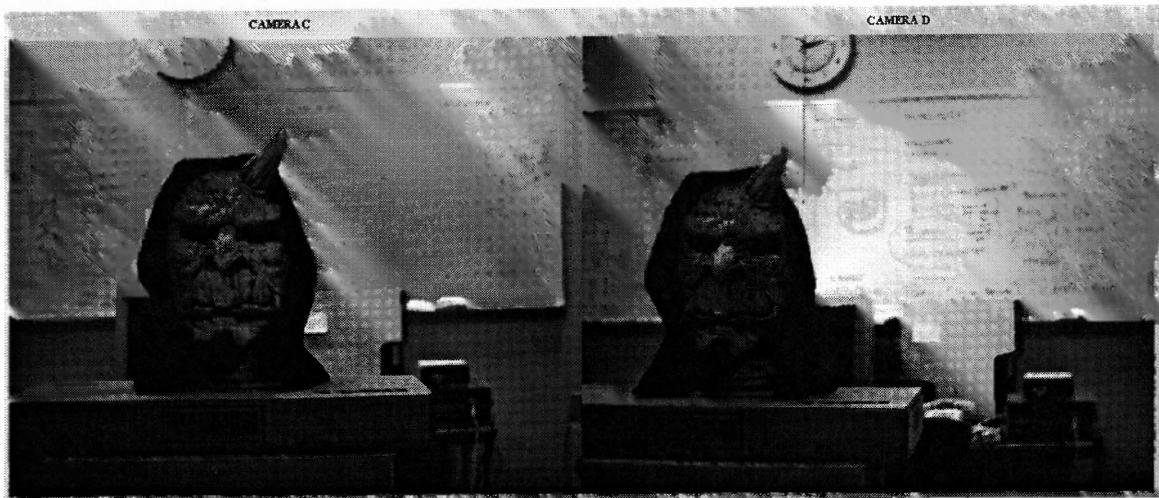


Figure 48: Images from cameras C & D

4.3.2 Rectified Images



Figure 49: Rectified images from cameras C & D

4.3.3 Correspondence Tabulation

Table 3: Correspondence Result of CD

Camera Pair – CD								
No.	x (L)	Y	x (R)	SAD	SSD	N. SSD	CC	N. CC
1.	579	422	525	W	W	W	W	W
2.	536	383	468	NW	NW	W	W	W
3.	623	499	440	NW	NW	W	W	W
4.	586	470	508	NW	NW	W	W	W
5.	568	457	436	W	W	W	W	W
6.	672	560	449	W	W	W	W	W
7.	612	493	511	W	W	W	W	W
8.	596	483	423	W	W	W	W	W
9.	670	552	568	W	W	W	W	W
10.	487	329	600	W	W	W	W	W

The results for camera pair CD are very similar to that for camera pairs AB and BC.

4.4 AC

4.4.1 Captured Images



Figure 50: Images from cameras A & C

4.4.2 Rectified Images



Figure 51: Rectified images from cameras A & C

4.4.3 Correspondence Tabulation

Table 4: Correspondence Result for AC

Camera Pair – AC								
No.	x (L)	Y	x (R)	SAD	SSD	N. SSD	CC	N. CC
1.	666	446	464	W	W	W	W	W
2.	768	550	463	NW	NW	W	NW	W
3.	726	500	424	W	W	W	W	W
4.	735	490	475	W	W	NW	NW	NW
5.	801	588	453	NW	NW	W	W	W
6.	687	461	522	NW	NW	W	NW	W
7.	643	428	447	NW	NW	W	NW	W
8.	721	486	577	NW	NW	W	W	W
9.	688	458	632	NW	NW	W	W	W
10.	653	435	582	NW	NW	W	W	W

The quality of the results camera pair AC are inferior compared to the above cases. This is due to the larger baseline of the cameras.

4.5 AD

4.5.1 Captured Images



Figure 52: Images from cameras A & D

4.5.2 Rectified Images

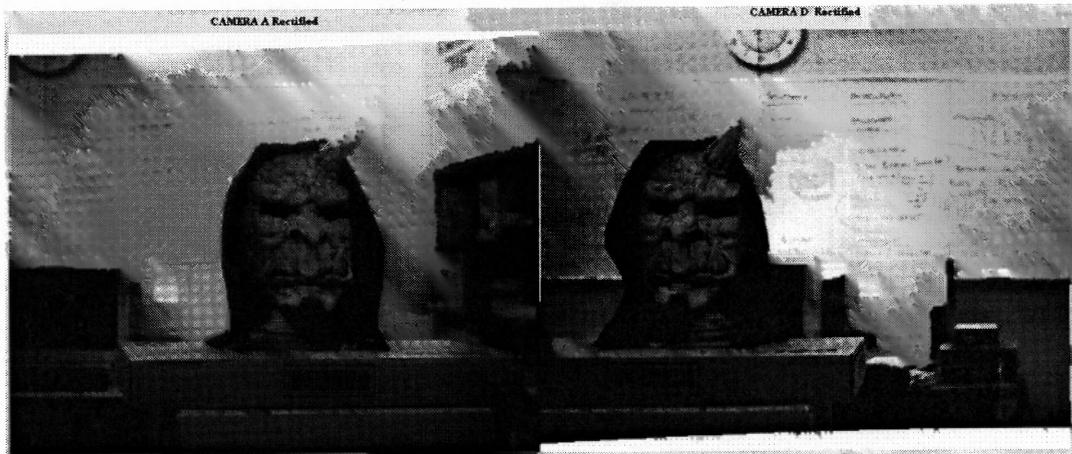


Figure 53: Rectified images from cameras A & D

4.5.3 Correspondence Tabulation

Table 5: Correspondence Result for AD

Camera Pair – AD								
No.	x (L)	Y	x (R)	SAD	SSD	N. SSD	CC	N. CC
1.	718	360	372	NW	NW	W	NW	W
2.	804	425	512	W	W	W	W	W
3.	664	300	471	NW	W	W	W	W
4.	716	328	585	W	W	W	W	W
5.	659	301	385	NW	NW	NW	NW	NW
6.	810	462	396	W	W	W	W	W
7.	672	299	627	W	W	NW	W	NW
8.	747	398	328	W	W	NW	NW	NW
9.	747	398	329	W	W	W	NW	W
10.	630	273	553	NW	NW	W	W	W

The quality of the results camera pair AD are the worst of all the pairs as the baseline is the largest between these two cameras. This is evident on a closer examination of the figures. The orientation of the cameras causes any portion of scene to appear very differently in both the images. Often the portions are completely occluded in one of the images.

4.6 Propagation

A camera array uses multiple cameras with correspondence being performed on two successive cameras at a time. Disparity is found pair-wise. The total disparity with respect to the extreme pair is equal to the sum of the disparities of the intermediate pairs.

In the table below, disparities of a given point is calculated for pairs AB, BC & CD. The sum of these is shown to be equal to the disparity of the same point in AD.

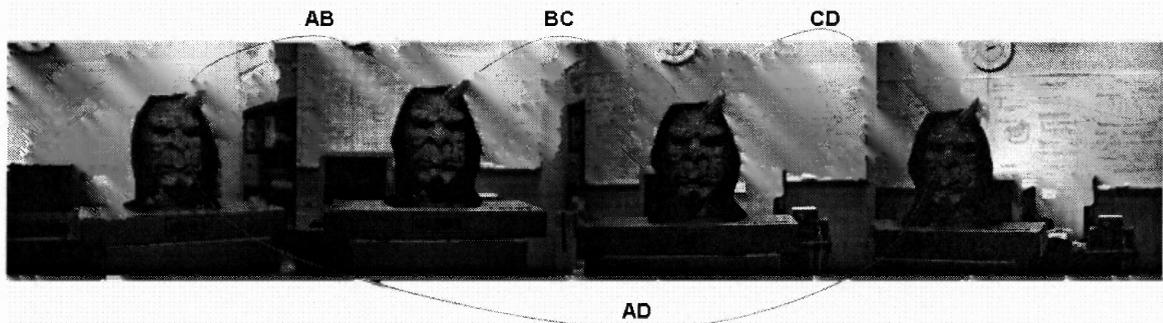


Figure 54: Propagation of Correspondence

Table 6: Propagation Results

No.	Region of Interest	AB Disparity ($x_L - x_R$)	BC Disparity ($x_L - x_R$)	CD Disparity ($x_L - x_R$)	Sum of Disparities	AD Disparity ($x_L - x_R$)	Propagation Match $AB+BC+CD=AD$
1.	Point on the Nose	723-613=110	624-510=114	500-347=153	377	723-348=375	Match
2.	Point on the Chin	745-634=111	643-525=118	517-361=156	385	749-366=383	Match
3.	Point on Forehead	717-610=107	623-516=107	503-355=248	363	716-356=360	Match
4.	Point on Lip	718-602=114	613-494=119	483-325=158	391	718-329=389	Match

4.7 Effect of Baseline

As the baseline increases the correspondence matching deteriorates but increases the accuracy of the Z-depth. This resulted in placing 4 cameras with certain baseline distance apart from one another, such that finding the correspondences is easy for cameras right next to each other. These correspondences can then be propagated to the other cameras as explained in the propagation section. The two extreme cameras with large baseline distance can calculate the Z-depth accurately.

The plots below show the mismatch in correspondences as the baseline increases. The triangulation table shows that as the baseline increases the depth is calculated more accurately.

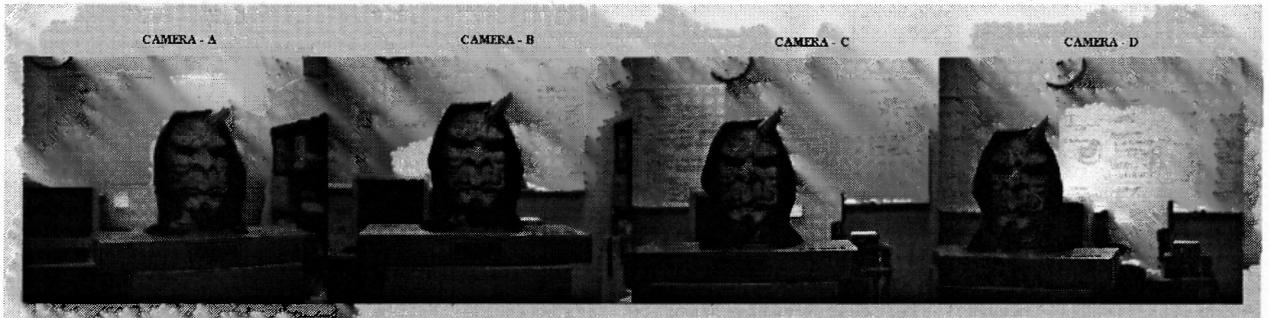


Figure 55: Correspondence Mismatch

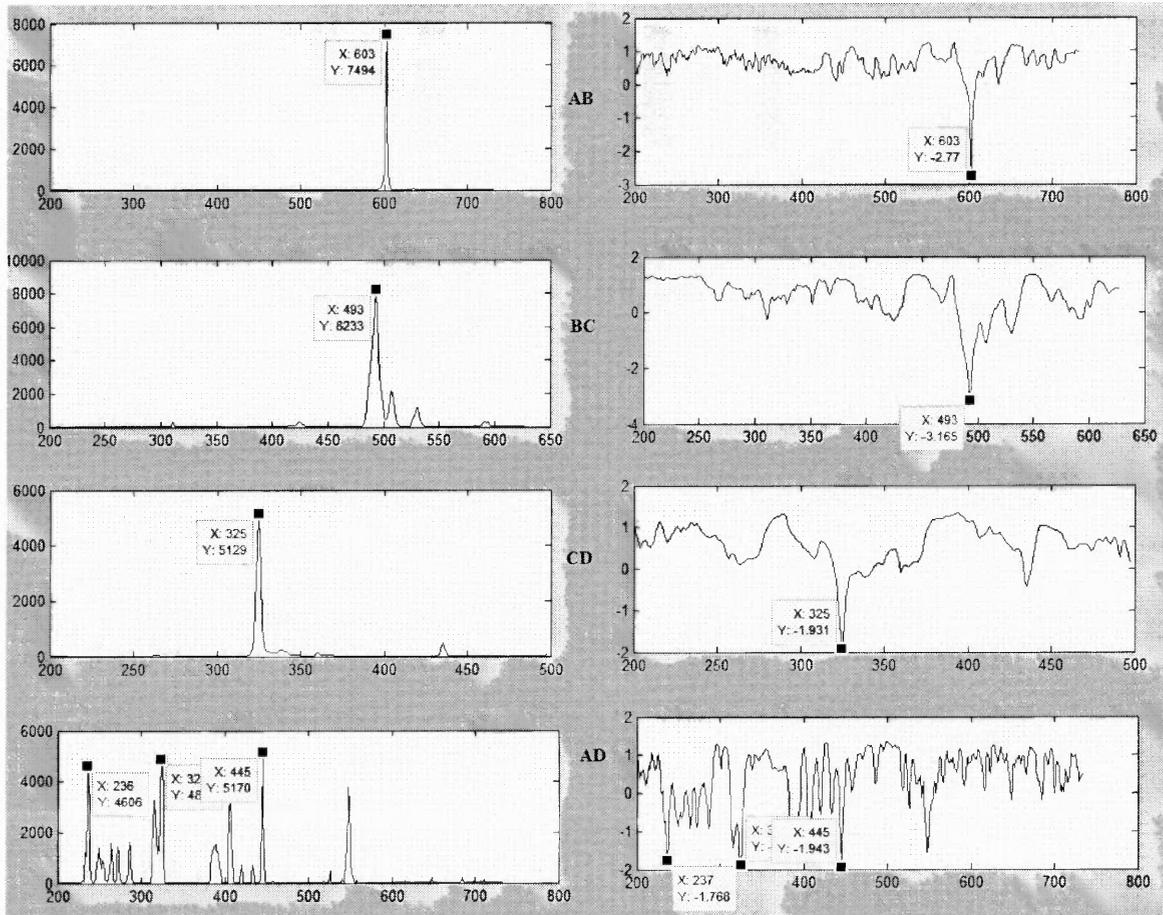


Figure 56: Second Order Algorithms for the point shown in the previous figure

Table 7: Triangulation Results

Camera 1		Camera 2		Depth (mm)	Expected Depth (mm)
B	(703, 489)	C	(587, 489)	953.7955	1110
C	(579, 525)	D	(422, 525)	927.69	1110
A	(806, 466)	B	(692, 466)	1031	1110
A	(806, 503)	C	(577, 503)	1140	1110
A	(804, 511)	D	(425, 511)	1104	1110

CHAPTER 5

CONCLUSION

The thesis evaluated five types of area-matching algorithms. Largely second order cross-correlation and normalized SSD were the most successful. However, an interesting observation that was noted was the effectiveness of algorithms with respect to nature of the portion of the image being matched. It was observed that when the considered portion was surface-like i.e. smooth, with small deviations from mean the second order methods generally performed badly. SAD and SSD techniques were more reliable. This may be overcome by using a texture check that classifies the nature of the image region and uses a suitable matching algorithm.

In Chapter 3 various issues that confront correspondence algorithms are presented. Researchers have proposed a large number of techniques ranging from simple to extremely complex, to overcome this problem. The approach taken in this thesis was to use camera arrays with smaller baselines to enable accurate correspondence. This setup ensured that very complex algorithms are not necessary to find the correspondences. Simple algorithms can be used to find the correspondences with smaller baselines and propagate them to the further ones. As the baseline increases the depth resolution increases which was proved by calculating the depth between different pairs. The ones with greater baseline had better resolution when compared to the ones with shorter baseline.

REFERENCES

- [1] R. Klette and others, eds., *Computer Vision, Three-Dimensional Data from Images*. Singapore: Springer, 1998.
- [2] William Eric Leifur Grimson, *From Images to Surfaces, A Computational Study of The Human Early Visual System*. Massachusetts: The MIT Press, 1986.
- [3] E. Trucco and A. Verri , *Introductory Techniques for 3-D Computer Vision*. New Jersey: Prentice-Hall, 1998.
- [4] Olivier Faugeras, *Three-Dimensional Computer Vision, A Geometric Viewpoint*. Massachusetts: The MIT Press, 1996.
- [5] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*. New York: Cambridge University Press, 2004.
- [6] Nicholas Ayache, translated from the French by Peter T. Sanders, *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*. Massachusetts: The MIT Press, 1991.
- [7] The MathWorks, Inc, “Matlab Documentation”.
<http://www.mathworks.com/access/helpdesk/help/techdoc/> (February 2006).
- [8] Intel Corp, “Camera calibration Toolbox for Matlab”.
http://newbologna.vision.caltech.edu/bouguetj/calib_doc/ (February 2006).
- [9] Wikipedia, “Stereopsis”.
<http://en.wikipedia.org/wiki/Stereopsis/> (February 2006).
- [10] Lecture Notes, “Vision Systems” by David Marshall.
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARSHALL/Vision_lecture_caller.html (February 2006).

- [11] 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.
- [12] Tutorial Notes, "Visual 3D Modeling from Images" by Marc Pollefeys
<http://www.cs.unc.edu/~marc/tutorial/tutorial02.html> (February 2006).
- [13] M. Z. Brown, D. Burschka, G.D. Hager, "Advances in Computational Stereo." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.25, No. 8, August 2003
- [14] Tutorial Notes, "Stereoscopic Vision: Natural and Artificial" by Jesse Hoey
<http://www.cs.ubc.ca/spider/jhoey/review/review.html> (February 2006).
- [15] P. Witoonchart, and R. Foulds. "Three-Dimensional Surface and Volume Measurements Using a Camera Array." *IEEE 28th Annual Northeast Bioengineering Conference* (2002): 145-146.
- [16] Grimson, W. E. L. "*From Images to Surfaces: A Computational Study of the Human Early Visual System*". MIT Press, Cambridge, MA., (1981).