1978

Measurement of three dimensional information using stereo views.

Vyomesh F. Shah
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation
https://scholar.uwindsor.ca/etd/2685

This online database contains the full-text of PhD dissertations and Masters’ theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000 ext. 3208.
NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur. S.R.C. 1970, c. C-30. Veuillez prendre connaissance des formulaires d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS RÊUE
MEASUREMENT OF THREE DIMENSIONAL INFORMATION USING STEREO VIEWS

by

VYOMESH F. SHAH

A Thesis

Submitted to the Faculty of Graduate Studies through the Department of Electrical Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science at the University of Windsor

Windsor, Ontario
1978
ABSTRACT

The accurate measurement of the three dimensional surface coordinates of a solid object has many applications in industrial automation and control. A stereo pair of images of the object allows extraction of such information. This work is devoted to the development of an algorithm for measurement of such information using narrow angle stereo views.

First, the problem of matching points from a pair of stereo views by using statistical measures has been investigated. A correlation technique has been found optimal and an algorithm based on this is presented for matching. An efficient search procedure based on the Laplacian operator, hill-climbing method, etc. has been constructed to minimize computer time. Secondly, the problem of calculating three dimensional coordinates from two dimensional coordinates is considered. The perspective transformation for a camera model under real world conditions is investigated. An algorithm based on the homogeneous transformation technique has been developed for calculating three dimensional coordinates from two dimensional coordinates of a matched point pair.

The complete algorithm has been implemented in FORTRAN-V and a comparison of calculated and actual value is given. For the narrow angle stereoscopy, these values are in satisfactory agreement for many applications.
ACKNOWLEDGEMENTS

The author is deeply indebted to Dr. J. J. Soltis for his advice and whole-hearted assistance in this work. Dr. Soltis has been a source of profound inspiration during this work. The author also expresses his sincere gratitude to Dr. W. C. Miller and Dr. G. A. Jullien for their suggestions and encouragement. Thanks are also due to other members of department who co-operated and assisted the author.
# TABLE OF CONTENTS

<p>| ABSTRACT | I |
| Acknowledgements | II |
| TABLE OF CONTENTS | III |
| LIST OF ILLUSTRATIONS | V |
| LIST OF TABLES | VII |
| <strong>CHAPTER 1: INTRODUCTION</strong> | |
| 1.1 Introduction | 1 |
| 1.2 Previous Work in 3-D Measurement | 1 |
| 1.3 Problem Statement | 3 |
| 1.4 Scope of Present Investigation | 5 |
| 1.5 Outline of the Present Study | 7 |
| 1.6 Organization | 7 |
| <strong>CHAPTER 2: METHODS FOR FINDING 3-D INFORMATION</strong> | 9 |
| 2.1 Various Methods for Measuring Three Dimensional Information | 9 |
| 2.2 Ultrasonic Sound Ranging Technique | 9 |
| 2.3 Laser Ranging Technique | 12 |
| 2.4 Axial Tomography - Related Methods | 14 |
| 2.5 Spatial Modulation and Volume Scanning | 14 |
| 2.6 Stereo View Processing | 15 |
| 2.7 Justification of Stereoview Method Selection | 18 |
| <strong>CHAPTER 3: STEREOVIEW PROCESSING</strong> | 20 |
| 3.1 Theory of Stereo Views | 20 |
| 3.2 Principle of Stereo Views | 21 |
| 3.3 Problems of Real World Systems | 23 |
| 3.4 Approach Utilized for Real World Case | 25 |
| <strong>CHAPTER 4: EFFICIENT MATCH FINDING</strong> | 27 |
| 4.1 Match Finding | 27 |
| 4.2 Techniques for Finding Match | 29 |
| 4.3 Comparison of Different Statistical Measures | 36 |
| 4.4 Autocorrelation for Correct Match | 43 |
| 4.5 Efficient Match Finding | 50 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6 Preprocessing Scheme</td>
<td>51</td>
</tr>
<tr>
<td>4.7 Accurate Initial Guess</td>
<td>54</td>
</tr>
<tr>
<td>4.8 Hill Climbing Method</td>
<td>57</td>
</tr>
<tr>
<td>CHAPTER 5: FINDING THREE DIMENSIONAL INFORMATION FROM MATCHED POINT PAIR</td>
<td>60</td>
</tr>
<tr>
<td>5.1 Camera Model</td>
<td>61</td>
</tr>
<tr>
<td>5.2 Real World Camera Model</td>
<td>67</td>
</tr>
<tr>
<td>5.3 Homogeneous Transformation Technique</td>
<td>74</td>
</tr>
<tr>
<td>CHAPTER 6: RESULTS FOR SIMULATED AND REAL WORLD OBJECT</td>
<td>85</td>
</tr>
<tr>
<td>6.1 Program</td>
<td>85</td>
</tr>
<tr>
<td>6.2 Error Analysis of Proposed Algorithm</td>
<td>88</td>
</tr>
<tr>
<td>6.3 Experimental Set-up</td>
<td>90</td>
</tr>
<tr>
<td>6.4 Results</td>
<td>96</td>
</tr>
<tr>
<td>CHAPTER 7: SUMMARY AND CONCLUSION</td>
<td>104</td>
</tr>
<tr>
<td>APPENDIX A.</td>
<td>108</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>136</td>
</tr>
<tr>
<td>VITA AUCTORIS</td>
<td>140</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>1-1</td>
<td>Stereo Views Processing Scheme.</td>
</tr>
<tr>
<td>2-1</td>
<td>Block Diagram for Measuring Coordinates of a Point in Space Using Sound Ranging Technique</td>
</tr>
<tr>
<td>2-2</td>
<td>Block Diagram for Measuring Three Dimensional Information Using Laser Ranging Technique</td>
</tr>
<tr>
<td>2-3</td>
<td>Block Diagram for Measuring Three Dimensional Information Using Spatial Modulation and Volume Scanning Principle</td>
</tr>
<tr>
<td>2-4</td>
<td>Block Diagram for Measuring Three Dimensional Information Using Stereo Views</td>
</tr>
<tr>
<td>3-1</td>
<td>Principle of Stereo Views</td>
</tr>
<tr>
<td>3-2</td>
<td>Approach Utilized for Measuring Three Dimensional Information</td>
</tr>
<tr>
<td>4-1</td>
<td>Window of [(2m+1)x(2n+1)] Pixels Around a Pixel</td>
</tr>
<tr>
<td>4-2</td>
<td>Windows with Centers at ((i_o,j_o)) and ((i_o',j_o')) in Two Pictures</td>
</tr>
<tr>
<td>4-3</td>
<td>Variation of 1.0-RMSD for Real World Image Point Pair ((60,121),(65,144))</td>
</tr>
<tr>
<td>4-4</td>
<td>Variation of Correlation for Simulated Image Point Pair ((46,50),(45,39))</td>
</tr>
<tr>
<td>4-5</td>
<td>Variation of Correlation for Real World Image Point Pair ((60,121),(65,144))</td>
</tr>
<tr>
<td>4-6</td>
<td>Variation of Correlation for Real World Image Point Pair ((70,131),(75,154))</td>
</tr>
<tr>
<td>4-7</td>
<td>Variation of 1.0-Autodifference for Real World Image Point ((60,121))</td>
</tr>
<tr>
<td>4-8</td>
<td>Variation of Autocorrelation for Simulated Image Point ((46,50))</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>4-9</td>
<td>48</td>
</tr>
<tr>
<td>Variation of Autocorrelation for Real World Image Point (60,121)</td>
<td></td>
</tr>
<tr>
<td>4-10</td>
<td>49</td>
</tr>
<tr>
<td>Variation of Autocorrelation for Real World Image Point (70,131)</td>
<td></td>
</tr>
<tr>
<td>4-11</td>
<td>53</td>
</tr>
<tr>
<td>Flow-Chart for Thinning and Cleaning Algorithm</td>
<td></td>
</tr>
<tr>
<td>4-12</td>
<td>55</td>
</tr>
<tr>
<td>Determination of Number of Edges Forming a Corner</td>
<td></td>
</tr>
<tr>
<td>4-13</td>
<td>55</td>
</tr>
<tr>
<td>Hill Climbing Method</td>
<td></td>
</tr>
<tr>
<td>5-1</td>
<td>62</td>
</tr>
<tr>
<td>Camera Model with Physical and Ideal Image</td>
<td></td>
</tr>
<tr>
<td>5-2</td>
<td>62</td>
</tr>
<tr>
<td>Camera Model with Axis Along Z Axis and Focal Center at (0,0,-F)</td>
<td></td>
</tr>
<tr>
<td>5-3</td>
<td>64</td>
</tr>
<tr>
<td>Ideal Stereo Views with Relations Between Different Coordinates</td>
<td></td>
</tr>
<tr>
<td>5-4</td>
<td>66</td>
</tr>
<tr>
<td>Comparison of Actual and Found Boundary for Simulated Ideal Stereo Views</td>
<td></td>
</tr>
<tr>
<td>6-1</td>
<td>91</td>
</tr>
<tr>
<td>Variation of Maximum Percentage Error of Y with Number of Pixels</td>
<td></td>
</tr>
<tr>
<td>6-2</td>
<td>92</td>
</tr>
<tr>
<td>Variation of Maximum Percentage Error of X with Number of Pixels</td>
<td></td>
</tr>
<tr>
<td>6-3</td>
<td>93</td>
</tr>
<tr>
<td>Variation of Maximum Percentage Error of Z with Number of Pixels</td>
<td></td>
</tr>
<tr>
<td>6-4</td>
<td>94</td>
</tr>
<tr>
<td>Variation of Percentage Mean Square Error of Y with Number of Pixels</td>
<td></td>
</tr>
<tr>
<td>6-5</td>
<td>95</td>
</tr>
<tr>
<td>Variation of Total Percentage Mean Square Error with Number of Pixels</td>
<td></td>
</tr>
<tr>
<td>6-6</td>
<td>97</td>
</tr>
<tr>
<td>Digitized Real world Stereo Views</td>
<td></td>
</tr>
<tr>
<td>6-7</td>
<td>98</td>
</tr>
<tr>
<td>Simulated (With Noise) Stereo Views</td>
<td></td>
</tr>
<tr>
<td>6-8</td>
<td>102</td>
</tr>
<tr>
<td>Comparison of Actual and Found Boundary for Real World Stereo Views</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Comparison of Number of Correlations Required for Direct Search and Hill Climbing Method</td>
<td>59</td>
</tr>
<tr>
<td>5.1</td>
<td>Comparison of Actual Values and Values found by Homogeneous Transformation Technique</td>
<td>83</td>
</tr>
<tr>
<td>6.1</td>
<td>Comparison of Actual Values and Found Values for Simulated (Without Noise) Stereo Views</td>
<td>99</td>
</tr>
<tr>
<td>6.2</td>
<td>Comparison of Actual Values and Found Values for Simulated (With Noise) Stereo Views</td>
<td>100</td>
</tr>
<tr>
<td>6.3</td>
<td>Comparison of Actual Values and Found Values for Real World Stereo Views</td>
<td>101</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Introduction

An object can be visually defined by specifying intensity and $(X,Y,Z)$ coordinates for different points on the outer surface, i.e., three-dimensional (3-D) information. This information plays an important role in analysis of objects and various control and automation applications. For example, we find many processes and systems where 3-D information would be of great help. Particular examples are numerically controlled machines, industrial robotic systems, and aerial photography. An important aspect of research on an intelligent robot is accurate measurement of 3-D information. In order to make an intelligent robot perform various complicated tasks, a visual device is required to recognize the object and measure its position according to robot's purpose.

Three-dimensional display using stereoview technique has been known since 1877. A large number of papers [12, 19, 21, 24, 26] have been published and much research has been done in this area. However, the technique of measuring 3-D information using stereo views is quite new and recent and little work has been done in this area. Three-dimensional display systems and measurement of 3-D information using stereo views are entirely different problems.

1.2 Previous Work in 3-D Measurement

A 3-D measurement technique should produce intensity and numerical value of coordinates for different points on the outer surface of the
object. Very little work has been done on 3-D information measurement using stereo views. Work in this area started with B. Julesz and J. Miller[11] in 1961. They derived equations for parallax between two image points which are the image of the same point in ideal stereo views. If the cameras are ideal pinholes with their axes exactly parallel to depth axis, ideal stereo views can be obtained. This can be utilized to measure information about an object made up of a few discrete points, when ideal stereo views are available. It is not applicable to real world objects, as ideal stereo views are practically impossible to obtain. In order to find information, matched point pairs, i.e. points in the stereo views which correspond to the image of the same point in space should be obtained. Their work does not give any information about finding matched point pairs.

Recently R. Nevatia[17] reported his work in 3-D measurement using stereo views. His work deals with finding 3-D information about an object when a number of views (9 to 13) are available. Here matched point pairs are found with the help of mean square difference and 3-D information is found by a camera model similar to that used by Julesz. This method requires more than 2 views and hence more memory storage and computation time. Also, the angle of rotation of camera has to be maintained very accurate (by computer control), otherwise it will give erroneous results.

K. Mori, M. Kikode and H. Asada[14] have also reported their work in this area. Their work is mainly concerned with the problem of depth measurement in aerial photography. This work deals with finding the elevation of the ground surface when two aerial views of the surface are taken, when information about a few well-contrasting parts of the image
is known a priori. Here a first prediction of elevation is carried out for well-contrasting parts and then the prediction is expanded to neighboring parts by using an equivalent camera model and prediction-correction. Thus elevation of the ground surface is obtained. It is not possible to utilize this work for finding 3-D information about arbitrary objects as the object must have at least a few sharp edges and the coordinates of points on the object should be known. Also, results obtained by this have poor accuracy.

M. Levine, D. O'Handley and G. Yagi[13] have also reported their work in this area. Their work deals with finding depth-maps in given stereo views. This work fails to give numerical value of the depth. It gives a number of regions with different depth contours. It cannot be utilized to obtain 3-D information about the object.

1.3 Problem Statement

The problem being investigated in this work can be defined:

Given an object (with some restrictions), obtain the information about intensity and \((X,Y,Z)\) coordinates of the outer surface of the object.

The aim of this work is to find an algorithm (processing scheme) for measuring information about intensity and coordinates of the outer surface of an object using stereo views. Among the various methods for measuring 3-D information, stereoview processing method gives accurate results and requires moderate instrumentation. The problem of measuring 3-D information using stereo views is thus investigated in this work.

When a picture of a 3-D object is taken, it is projected onto the two-dimensional (2-D) space of the image plane. This projection of space
from three dimensions to two dimensions results in a loss of information about the third dimension. Human eyes presented with a photograph of a 3-D object are able to infer depth information with little or no difficulty. Among the various cues believed to be used by human eyes are textured gradients, shadow, occlusion, size of familiar objects, etc. Depth inferences from such cues are complex and poorly understood. It is a difficult task for machines to find 3-D information from a 2-D image. Some processing should be done to obtain information about the third dimension which is lost during the standard picture taking process.

Measurement of information for different points on the 3-D surface is a difficult task, especially when one is dealing with large class of objects of different kinds. For a restricted class of objects, an algorithm can be developed dependent upon the characteristics of the class.

The present work is restricted to narrow-angle stereo views. The stereo views can be broadly classified in two classes.

1. Narrow-angle stereo views.
2. Wide-angle stereo views.

If the cameras are separated by a small distance in lateral direction, narrow-angle stereo views can be obtained. Total angle subtended by camera axes at the nearest object point does not exceed 15°. The narrow-angle stereo views are characterized by a large area common to both views. Portion of an object visible in one view and invisible in another view is small. Wide-angle stereo views are characterized by a larger area visible in one view and invisible in another view. If views are taken from widely different angles, i.e., cameras are separated by larger distance in lateral direction, wide-angle stereo views can be obtained.
The present work is restricted to narrow-angle stereo views only. The term stereo views should be interpreted as the narrow angle stereo views. The developed algorithm may not be valid for wide-angle stereo views. The present investigation describes an efficient method for measuring 3-D information using stereoview processing. A block diagram of the processing scheme is shown in figure 1-1.

1.4 Scope of Present Investigation

Main applications of three dimensional measurement are:

(1) Computer graphics.
(2) Numerically controlled machines.
(3) Industrial robotic systems.
(4) Limbs to operate radioactive material.
(5) Automatic material handling.
(6) Aerial photography.
(7) Various automation applications.

During the last few years, the use of two dimensional computer I/O devices has become common place. More recently three dimensional computer I/O devices have become of prominent interest. It is obvious that corresponding 3-D computer input devices might be interesting and useful. So far there have been a few laboratory devices of limited utility and speed. One of the major problems in this area is the lack of a method for accurate measurement of 3-D information about objects.

Thus taking into consideration all the above, it is obvious that if there is some method for measuring 3-D information, it would of great help from an industrial and research point of view.
FIGURE 1-1: Stereo Views Processing Scheme
1.5 Outline of The Present Study

The present investigation gives an algorithm for measuring 3-D information using stereo views. A processing scheme has been developed which can be utilized to obtain 3-D information about an object if digitized stereo views of the object are available. This processing scheme finds information efficiently.

A comparison of the various methods for measuring 3-D information has been undertaken. The selection of the stereoview method has been explained and justified. An algorithm for finding matched point pairs has been developed. An algorithm is developed which finds matched point pairs by using the hill-climbing method to find the match efficiently. The speed of processing has been increased by first finding a match for corner points and then for other points. Autocorrelation has been used as a check for finding an accurate and reliable match. The homogeneous transformation technique[1,18,31] has been used to calculate 3-D coordinates from the position of matched point pairs.

Programs were written in FORTRAN-V. The algorithm was used to measure 3-D information from simulated(with/without noise) and real world stereo views. Results obtained have been compared with the actual values.

1.6 Organization

Chapter 2 describes the various known methods for measuring 3-D information. A critical comparison is made of the different methods. The stereoview method has been justified for its selection.

Chapter 3 explains the stereoview processing method. A processing scheme utilized for measuring 3-D information is described. Various
problems associated with this method to utilize it in the real world have been described. A processing scheme has been suggested to solve the real world problems:

Chapter 4 describes methods for finding matched point pairs. The various methods that can be utilized are given. A method using correlation has been determined as optimal. A processing scheme is thence developed to find the match efficiently with minimum computation time.

Chapter 5 gives the method for calculating 3-D information from matched point pairs position. The homogeneous transformation technique utilized for calculating 3-D coordinates from a matched point pair has been explained.

Chapter 6 relates to the programs developed and results. A brief description of the programs and various subroutines written to implement the overall algorithm is provided. An error analysis of the algorithm is given. The experimental set-up to obtain real world stereo views is shown. Results obtained for simulated(with/without noise) and real world objects are given and compared with actual values.

Chapter 7 gives the summary and conclusions of the present investigation.
CHAPTER 2

METHODS FOR FINDING 3-D INFORMATION

In recent years, a large number of methods have been suggested and implemented for measuring 3-D information. However, most of these methods require elaborate instrumentation. Some of these are limited to a certain class of objects whereas some have poor accuracy. There has been no method available which requires moderate instrumentation, gives accurate results and can be applied to all normal objects. From the various methods utilized to measure 3-D information, the stereoview method is compared with others and the reasons for its adoption are explained.

2.1 Various Methods for Measuring Three Dimensional Information

The various methods reported for measuring 3-D information are:

(1) Ultrasonic sound ranging technique.

(2) Laser ranging technique.

(3) Axial tomography - Related methods.

(4) Spatial modulation and volume scanning.

(5) Stereoview processing.

2.2 Ultrasonic Sound Ranging Technique

This method calculates coordinates of a point in space by using the principle that the time required for an ultrasonic sound pulse to travel the distance between a transmitter and a receiver is dependent on the distance between them. If the distance between transmitter and receiver is $\Delta z$, speed of sound $c$, then the time required for a sound pulse to travel the
distance between transmitter and receiver is

$$\Delta t = \frac{\Delta z}{c}$$  \hfill (2.1)

The earliest device for measuring coordinates of a point in space using this principle was designed by L. G. Roberts in 1963. This device used an ultrasonic sound generator and four microphones mounted at the corners of a rectangle. The length from a point source of sound to each of the microphones was determined by the arrival time of the pulse at each microphone.

Recently a system for measuring the cartesian coordinates of a point in space was designed by W. Moritz and P. Shreve[16]. This method uses an ultrasonic sound pulse generator and 3 microphones placed along 3 coordinate axes. The time required for a sound pulse to travel the distance from a point source of an ultrasonic sound pulse generator placed on the object to the microphones is measured. A block diagram of the system utilized is shown in figure 2-1. Note that the pulse generator is physically, in turn, placed at the object points to be measured. A transit time counter counts time elapsed between the pulse transmitted and received. By measuring the transit time, the coordinates of a point where the sound generator was placed is calculated.

Though this method can be utilized to obtain the coordinates of a point in space, it has several weaknesses. It calculates only a few points per minute. One has to put an ultrasonic sound generator at the point in space where one wants to calculate the coordinates. Since this method calculates coordinates by using the sound velocity; as the sound velocity changes with
FIGURE 2-1: Block Diagram for Measuring Coordinates of a Point in Space Using Sound Ranging Technique[16]
temperature and air flow, resolution and accuracy are affected accordingly. To measure various points on the surface of an object, one has to move the transmitter on the surface of the object. This complicates automated measurement, especially when the object is moving. Its most serious drawback for many applications is the excessive time required.

2.3 Laser Ranging Technique

This method uses the principle that it takes time for a laser beam to travel from front to the back of an object. If the object depth is \( \Delta z \), it takes back scattered light \( 2\Delta z/C \) longer to return from the back of an object than from the front of the object, where \( C \) is the speed of light. One can measure the thickness of such a scatterer with a laser pulse of duration

\[
\Delta t \ll \frac{2\Delta z}{C} \tag{2.2}
\]

Although this method can be utilized to measure the coordinates of a point in space, it requires elaborate instrumentation. One has to use a mode-locked laser with pulse duration \( \Delta t \ll 2\Delta z/C \) and instruments which are capable of measuring the time interval \( 2\Delta z/C \) very accurately such as a streak-camera. H. Caulfield, T. Hirschfeld, J. Weinberg and R. Herron\([4,5]\) have described a method for measuring 3-D coordinates of the outer surface of an object using this method. Figure 2-2 shows their block diagram. By using a mode locked laser and a transmitting telescope, a laser beam is split using a beam splitter. Laser beam exposes the desired object. The beam reflected from the object is passed through an imaging lens and a slit to a streak camera. The output of the camera is further processed
FIGURE 2-2: Block Diagram for Measuring Three Dimensional Information Using Laser Ranging Technique[4,5]
to give a space profile of the object. By moving the slit in the vertical
direction, one can scan the whole object and thereby obtain 3-D information
about the object.

Though this method can be utilized to measure 3-D information, one
should note that it requires highly sophisticated instruments. There is
error in measurement due to the collimation effect. If there is a lack
of synchronization between the streak camera and the pulse generated by the
laser beam, there will be completely wrong information. This method cannot
find intensity information. Its most serious drawback is the extensive
instrumentation required.

2.4 Axial Tomography - Related Methods

Axial tomography is a special method used in medical analysis for
measuring 3-D information about an object. In this method, 2-D cross-section
of a 3-D object are found by processing various X-ray images taken from
different angles of a 2-D plane. These images can be utilized to calculate
information about the outer and inner surface of the object. Although this
method can be utilized to measure 3-D information, its use just to get
information about the outer surface of an object still requires very large
memory storage and processing power. Also the whole object has to be scanned.
Examination of the techniques used shows no practical extension to our
study.

2.5 Spatial Modulation and Volume Scanning

A further method for measuring 3-D information is by using spatial
modulation and volume scanning principles. Recently a paper published by
J. Hamasaki, Y. Nagata, H. Higuchi and M. Okada[10] describes a method for
transmission of a 3-D image using this principle and a varifocal mirror.

Figure 2-3 shows a block diagram for the system for measuring 3-D information using this principle. This method uses a grid, lenses and computer for signal processing. With a lens system, a grid is focussed on a particular sectional plane (depth level) of the object. The object is viewed by a vidicon camera. The output video signal is sent to an A/D converter which digitizes the signal for computer processing. The image of the object on the vidicon tube consists of a sharp image of a particular focussed plane and a blurred image corresponding to the remaining sectional planes. Thus spatial modulation is achieved by focussing the grid on a particular sectional plane of the object. The output signal can be processed to determine which points belong to a particular focussed sectional plane. By moving the lens, the grid is focussed on different sectional planes, thereby achieving volume scanning. By focussing the grid on different sectional planes, 3-D information about the object is obtained.

This method has limited depth resolution since sectional planes are necessarily quantized and scanned in turn. Further development appears required for this technique to find broad applications.

2.6 Stereo Views Processing

Two views of an object taken by two cameras, separated by some distance in the lateral direction are called stereo views. The stereoview processing method determines 3-D information by using the principle that the projection of a point in space on the image plane depends upon the coordinates of the point and the distance between the camera and point. A block diagram of the processing scheme is shown in figure 2-4. As shown in the figure,
FIGURE 2-3: Block Diagram for Measuring Three Dimensional Information Using Spatial Modulation and Volume Scanning Principle[10]
FIGURE 2-4: Block Diagram for Measuring Three Dimensional Information Using Stereo Views
two views of an object are taken. These 2 views are digitized to get a
digital picture of the image for processing. These digitized views are
provided to a computer which contains the programs for image processing to
give 3-D information about the object. Thus by using the stereo views, 3-D
information about the object is measured. It is the algorithm that enables
one to measure 3-D information about the object.

Compared to other methods, the stereo views processing method has
certain advantages which one cannot underestimate. It requires moderate
instrumentation, less memory storage, gives accurate results and gives all
the 3-D information on coordinates and intensity. It can be utilized
without mechanical scanning and light signal can be normal background
lighting.

2.7. Justification of Stereoview Method Selection

The stereoview method is selected for measuring 3-D information as
it has several advantages over other methods. It requires moderate insru-
mentation. As compared to the laser ranging technique, it does not require
special instruments like a streak camera and a mode locked laser. It does
not require a special grid and lens system as in the spatial modulation
and volume scanning method. Nor does it require an ultrasonic sound
generator as in the sound ranging technique. As compared to spatial modulation
and volume scanning and axial tomography, it requires much less memory
storage capacity. This method is quite easy and simple to apply if a digital
computer and digitized stereo views of an object are available. The images
can be digitized using digitizer, e.g. flying-spot-scanner, vidicon and A/D
converter, image dissector, etc. Also as compared to other methods, this
method gives more accurate results. In addition, since stereoview measuring system also innately handles intensity information, this method leads itself naturally to those applications in which pattern recognition is of concern.
CHAPTER 3

STEREOVIEW PROCESSING

The stereoview processing method depends on the principle that the projection of a point in space on the image plane of a camera depends upon the distance between the camera and the object point. Use of stereo views for measuring 3-D information was suggested by Julesz[11] in 1961 in his work. However, his work is applicable to ideal stereo views which are practically impossible to obtain. This chapter describes the principle of stereoview processing. Problems encountered in the real world and the approach utilized for solving them are described.

3.1 Theory of Stereo Views

Two views of an object taken by cameras, separated by some distance in the lateral direction are called stereo views. An object, via reflected light, provides intensity information for its 3-D coordinates on its outer surface. By taking a single picture, third dimensional(depth) information is lost. Now if one tries to reconstruct 3-D object from only one 2-D picture on the image plane, one can get two equations in terms of three unknown coordinates. Thus the system of equations has no unique solution. Therefore, at least one more equation is required to find all the three unknown coordinates X, Y and Z.

This problem is solved by taking two views of an object. By taking two views, four equations for three unknowns can be obtained. In this case, it is possible to find all the three unknowns by a least squares fit. Two views should be taken in such a way that four equations for three
unknowns are formed. This is achieved by stereo views. In stereo views as there is a large area common to both views, it is possible to obtain four equations for the three unknowns.

3.2 Principle of Stereo Views

Projection of a point in space on the image plane of a camera depends upon the coordinates \((X, Y, Z)\) of a point and the camera position. When stereo views of a point are taken, the image of a point in the stereo views depends upon the coordinates of a point and the separation between the two cameras. This principle is utilized for measuring 3-D information using stereo views.

As shown in figure 3-1, consider two points 1 and 2 which have coordinates \((X_1, Y_1, Z_1)\) and \((X_2, Y_2, Z_2)\) in space. Two cameras whose focal length is \(F\), focal centers at \(F_1\) and \(F_2\), with their axes parallel to the \(Z\) axis and \(Y\) axis parallel to and having the same sense as the image horizontal take two views of these points. The image of points 1 and 2 on the image plane of camera 1 is \(u_1\) and \(u_2\) and that on the camera 2 is \(v_1\) and \(v_2\), respectively. It should be noted that in the figure, the cameras are replaced by an equivalent camera model. Though the points 1 and 2 have the same \(X\) and \(Y\) coordinates, their image on the image plane is different depending on the \(Z\) coordinates of the points and the position of the camera. It is also to be noted that the separation between the two image points will be different depending on the separation between the two cameras and the coordinates of the point. If \((X_L, Y_L)\) are the coordinates of the image of a point in L.H.S. view, \((X_R, Y_R)\) the coordinates of the image of the same point in R.H.S. view, then \(X, Y\) and \(Z\) coordinates of the source point in
FIGURE 3-1: Principle of Stereo Views
space can be calculated by the following equations.

\[
X = \frac{B \cdot (x_L + x_R)}{2 \cdot (x_L - x_R)} \tag{3.1}
\]

\[
Y = \frac{y_L \cdot B}{(x_L - x_R)} \tag{3.2}
\]

\[
Z = H - \frac{B \cdot F}{(x_L - x_R)} \tag{3.3}
\]

where \( F \) is the focal length of each camera.

\( B \) is the separation between two cameras.

\( H \) is the distance of focal center from the center of coordinate axes.

So far as only one point is concerned, this method seems to be easy and trivial. However when one deals with real world images, there are numerous problems in utilizing this method. The following assumptions:

(1) both camera axes are exactly parallel to \( Z \) axis,

(2) focal centers have the same \( Z \) coordinates,

(3) focal length \( F \) and separation \( B \) and \( H \) can be determined exactly, are practically impossible to maintain. Results obtained by using eqns. (3.1)-(3.3) for real world objects will have inevitable errors.

3.3 Problems for Real World Systems

If the scene contains only a few points and ideal stereo views can be obtained, the above method can be applied and gives accurate results. But in practice, an object consists of a large number of planes and surfaces of different and varying intensities. Also, the camera axes are rotated by some angles about the \( X, Y \) and \( Z \) axes and it is difficult to find the
camera axis position on the image plane. The main problems in a real
world setting are now elaborated upon.

(1) Finding Matched Point Pairs:

A matched point pair is formed by the two points in the stereo views
of the same original object point in space. To calculate 3-D coordinates
of an object point in space, it is necessary to find the corresponding matched
point pairs in the stereo views. The problem in finding a matched point
pair is to find which particular pixel in the right view corresponds to a
certain pixel in the left view. The matched point pairs can be found by
the intensity and the coordinates of the points in the stereo views. If
one deals with an image with only a few discrete points, the match can be
found easily. But in real world as a large number of points are involved,
it is difficult to find matched point pairs, also the quantization in the
intensity (e.g. 8 bits/pixel in our case) and noise associated with the real
world image has to be considered.

(2) Finding 3-D Coordinates from Matched Point Pairs:

The second problem is to calculate 3-D coordinates, given matched
point pairs coordinates. If both the camera axes were parallel to the Z
axis and in the horizontal plane, the coordinates can be calculated using
the eqns. (3.1)-(3.3). But in the real world, the camera axes may be rotated
about X and Y axis during the picture taking process and during the
digitization, there may be rotation of the image about the Z axis. Thus
it is difficult to determine the true values of \((x_L, y_L)\) and \((x_R, y_R)\) coordinates
of a typical image point in L.H.S. and R.H.S. views. Proper equations should
be obtained to find the true values of X, Y and Z coordinates even if all
conditions which occur in the real world are taken into account.
3.4 Approach Utilized for Solving Real World Case

The approach can be explained:

An algorithm is developed which will find accurate matched point pairs from the digitized stereo views. A matched point pair means \((x, y)\) coordinates in each of the stereo views which correspond to the same point in space. This algorithm is capable of finding matched point pairs in a real world setting. Another algorithm is developed which will calculate coordinates from the matched point pairs coordinates values. This algorithm is capable of calculating coordinates even if all conditions which occur in the real world are taken into account. The intensity of a point is found by taking average intensity of its matched point pair. A block diagram of the processing scheme utilized for measuring 3-D information for the real world is shown in figure 3-2.
FIGURE 3-2: Approach Utilized for Measuring Three Dimensional Information
CHAPTER 4

EFFICIENT MATCH FINDING

It is necessary to find the correct match for accurate measurement of 3-D information. Not only this is necessary, but the match should be found efficiently to save computer time. Finding an accurate match is the main problem that has restricted the use of stereo views for measuring 3-D information for such a long time. A proper algorithm should be developed so that the computer can find an accurate and reliable match from the digitized stereo views. This chapter describes the various techniques that can be utilized for match finding. A correlation technique, which has been utilized for match finding is described. The selection of correlation has been justified. A search procedure has been developed to find the match efficiently. A hill-climbing method for efficient searching is explained.

4.1 Match Finding

Match finding is the process of finding, for a given sub-area of an image X, the sub-area of image Y which contains point for point the same information for a point in space. Suppose one has been given two digitized photographs which are taken of the same object but differ in some aspect such as point of view. Then matching means finding the area of image X and image Y which corresponds to the same area of the actual object. Geometrically two areas match if both are projections of the same 3-D piece of the scene. Intuitively two areas match if they both look the same. Consider the problem of using a computer to determine which area of picture Y best matches a given area of picture X so that it can be utilized for
automatic matching of the stereo views. Computationally two areas match if a calculated measure of match between them is sufficiently optimal.

The digitized version of the stereo views provide the intensity values for 2-D coordinates on the image of each view. The problem is finding proper means to find the matched area between two views. If one is dealing with a restricted class of objects, an algorithm can be developed dependent upon the characteristics of that class. But as this investigation should handle general objects, a statistical measure of match is utilized. Statistical measures are applicable to all normal objects. Previous work done in this area shows that statistical measures are accurate to give reliable match[17].

Definitions

Some of the terms from the field of computer vision which are used in this thesis are defined below.

Picture -- a two dimensional array of integer value which represent the light intensities of a scene at some set of grid points.

Point -- one of the array element of a picture.

Pixel -- (Contraction of picture element) a point in a picture.

Image -- a picture representing a photograph.

Conventions of Picture Processing

In keeping with the conventions used in the field of computer vision, pixels are identified by there \((I,J)\) positions with respect to the upper left-hand corner of the picture, which has position \((1,1)\). The \(I\) dimension increases downward; \(J\) dimension increases right.
The intensities at each pixel (value of a pixel) are represented by numbers from 0 through $2^n-1$ (n represents the number of bits representing the intensity of a pixel; e.g. 8 in our case), with 0 no light, or black and $2^n-1$ representing full light or white.

The intensity of a pixel in picture $X$ in $i$th row and $j$th column is represented by $X_{i,j}$.

4.2 Techniques for Finding Match

There are different statistical measures which can be applied to a picture. Our problem is that of finding proper matched point pair for a given pixel in one view. One should also determine how to use a statistical measure for finding the match. For this, a certain window is selected around a pixel of interest in both views. A statistical measure is applied to these two windows to find match. The window in one view is moved along two directions. When an optimum value of statistical measure occurs, it is taken as the matched point pair. For example one is interested in finding the match for a pixel which is in $i_o$th row and $j_o$th column in the first view. A window of $[(2n+1)X(2m+1)]$ pixels is selected around this pixel as shown in figure 4-1. A window of the same size is selected around the (test) pixel of interest in the second view. The window in the second view is moved along two directions to find the proper match. When a match occurs, the statistical measure will have an optimum value. When this occurs, the position of the window in the second view and hence the position of the pixel which is the center of the window in the second view is determined. This pixel and the center of the window in the first view gives a matched point pair.
FIGURE 4-1: Window of $[(2m+1)\times(2n+1)]$ Pixels Around a Pixel
The various statistical measures that can be applied to find a match are:

1. Absolute difference.
2. Root mean square difference.
3. Correlation.

**Symbols**

- $X_{i,j}$ is the intensity of a pixel of the picture $X$ in $i$th row and $j$th column.
- $Y_{i,j}$ is the intensity of a pixel of the picture $Y$ in $i$th row and $j$th column.
- $\bar{X}$ is the average intensity of the pixels in the selected window for the picture $X$.
- $\bar{Y}$ is the average intensity of the pixels in the selected window for the picture $Y$.
- $\theta$ is the angle between the normal to the surface at a point and direction of observation.
- $K$ is the coefficient of reflection assigned to a particular point on the surface of an object.
- $\theta_1$ is the angle between normal to the surface at a particular point and direction of observation for the first view.
- $\theta_1 + \theta_2$ is the angle between normal to the surface at a point and direction of observation for the second view.
- $I_0$ is the intensity of primary light source.
- $A_0$ is the intensity of ambient light source.
- $I_1$ is the intensity of a point in the first view.
\( I_2 \) is the intensity of the same point as \( I_1 \) in the second view.

\( \Delta i \) is average difference of \( i \) coordinates between the stereo views.

\( \Delta j \) is average difference of \( j \) coordinates between the stereo views.

\( AD \) is absolute difference.

\( \text{RMSD} \) is root mean square difference.

\( \text{CORR} \) is correlation.

\( \text{SQRT} \) is square root.

**Absolute Difference**

Absolute difference measures the absolute value of difference between the samples over two areas. Equation for the normalized value of absolute difference with windows of \([(2n+1)\times(2m+1)]\) pixels centered at \((i_0, j_0)\)th and \((i_0', j_0')\)th pixel in two pictures as shown in figure 4-2 is given by

\[
AD = \left( \frac{1}{(2n+1) \cdot (2m+1)} \right) \left[ \sum_{i=-n}^{m} \sum_{j=-m}^{m} \left| \frac{X_{i+i_0, j+j_0} - \bar{X}}{\bar{X}} \right| \right] \left( \frac{Y_{i+i_0', j+j_0'} - \bar{Y}}{\bar{Y}} \right)
\]

(4.1)

Normalized value means its absolute value is less or equal to 1. Point pairs, which have minimum value of absolute difference is considered as matched point pairs.

**Root Mean Square Difference**

This is a measure defined as the root of the mean of the square of the difference between samples over two areas. The normalized value of root mean square difference with windows of \([(2n+1)\times(2m+1)]\) pixels centered at \((i_0, j_0)\)th and \((i_0', j_0')\)th pixel in two pictures as shown in figure 4-2 is given by
FIGURE 4.2: Windows with Centers at $(i_0, j_0)$ and $(i_0', j_0')$ in Two Pictures.
\[
\text{RMSD} = \text{SQRT} \left( \frac{1}{(2n+1)(2m+1)} \sum_{i=-n}^{n} \sum_{j=-m}^{m} \frac{X_{i+i_0,j+j_0} - \bar{X}}{\bar{X}} \cdot \frac{Y_{i+i_0,j+j_0} - \bar{Y}}{\bar{Y}} \right)^2 \right) (4.2)
\]

Point pairs which have minimum value of root mean square difference is considered as matched point pairs. Figure 4-3 shows the variation of 1-RMSD (1-Normalized value of root mean square difference) for a point pair in the real world stereo views as the window in the second view is moved along two directions. 1-RMSD is plotted instead of RMSD, so that it can be compared with correlation. It has a hill kind of nature. When a match occurs, RMSD has minimum value, i.e. 1-RMSD has maximum value.

Correlation

Correlation can also be utilized as a measure of match. Correlation measures equality between the samples over two windows. The equation for correlation[6] between two pictures is given by

\[
\phi_{xy}(k,l) = \frac{\sum_{i} \sum_{j} X_{i,j} \cdot Y_{i+k,j+l}}{\sum_{i} \sum_{j} X_{i,j} \cdot \sum_{i} \sum_{j} Y_{i,j}} (4.3)
\]

Normalizing the equation (4.3) by average value and second moment of the samples,

\[
\phi_{xy}'(k,l) = \frac{\sum_{i} \sum_{j} (X_{i,j} - \bar{X}) \cdot (Y_{i+k,j+l} - \bar{Y})}{\text{SQRT} \left[ \sum_{i} \sum_{j} (X_{i,j} - \bar{X})^2 \cdot \sum_{i} \sum_{j} (Y_{i+k,j+l} - \bar{Y})^2 \right]} (4.4)
\]

Equation (4.4), which is also known as correlation coefficient is used
FIGURE 4-3: Variation of 1.0-RMSD for Real World Image Point Pair (60,121),(65,144)
in this work; as it has an absolute value which is less or equal to one.

The normalized value of correlation, which is used in this work and denoted by CORR, with windows of \([(2n+1) \times (2m+1)]\) pixels centered at \((i_0,j_0)\)th and \((i_0',j_0')\)th pixel in two pictures as shown in figure 4-2 is given by

\[
\text{CORR} = \frac{\sum_{i=-n}^{n} \sum_{j=-m}^{m} \left( (X_{i+i_0}+j+j_0-\bar{X}) \cdot (Y_{i+i_0'}+j+j_0'-\bar{Y}) \right)}{\sqrt{\left( \sum_{i=-n}^{n} \sum_{j=-m}^{m} (X_{i+i_0}+j+j_0-\bar{X})^2 \right) \left( \sum_{i=-n}^{n} \sum_{j=-m}^{m} (Y_{i+i_0'}+j+j_0'-\bar{Y})^2 \right)}}
\] (4.5)

Figure 4-4 shows variation of correlation for simulated stereo views as the window in the second view is moved along two directions. Figures 4-5 and 4-6 show the variation of correlation for the real world stereo views as the window in the second view is moved along two directions.

Correlation attains its highest value when match occurs. It also has a hill-like variation. The value of correlation at points other than the matched point pair is less than that at the matched point pair. It should be noted that absolute difference and root mean square difference are difference measures, i.e. when match occurs, they will have a minimum value. The minimum value is zero in the ideal case. Correlation is an equality measure, i.e. when match occurs, it will have maximum value. The maximum value is one in the ideal case.

4.3 Comparison of Different Statistical Measures

As described, there are mainly 3 statistical measures which can be
FIGURE 4-4: Variation of Correlation for Simulated Image Point Pair (46,50),(45,39)
FIGURE 4-5: Variation of Correlation for Real World Image Point Pair (60,121),(65,144)
FIGURE 4-6: Variation of Correlation for Real World Image Point Pair (70,131),(75,154)
utilized for finding match. One should select the proper one taking into consideration the situation in which one is working, desired accuracy and other practical considerations.

The simplest measure in terms of number of instructions required to implement and computer processing time is absolute difference. Absolute difference will have a zero value when

$$Y_{i+i_0', j+j_0'} = X_{i+i_0, j+j_0}$$

(4.6)

for each value of i and j in the window.

The variation of the intensity of pixels over two areas in the real world is quite complex. Absolute difference gives reliable match only if the intensity variation of pixels over two areas is approximately related by equation (4.6). Hence though absolute difference method seems attractive in terms of computer time, it has poor accuracy, i.e. the probability of a match found by absolute difference of being wrong is high for the real world stereo views.

Root mean square difference is a measure which requires more computer processing time compared to absolute difference but less compared to correlation. When match occurs, root mean square difference will have minimum value. The minimum value will be zero only when the intensity for each pixel in a window is a sum of a constant and the intensity of the corresponding pixel in the other window, i.e.

$$Y_{i+i_0', j+j_0'} = \sqrt{X_{i+i_0, j+j_0}^2 + b}$$

(4.7)

where b is a constant.

for each value of i and j in the window.
Although root mean square difference requires more computation time than absolute difference, it has better accuracy. Probability of a match found by root mean square difference of being wrong is less than that found by absolute difference. When the intensity variation in two views is approximately given by equation (4.7), it is advisable to use this measure. However, when intensity variation in two views is given by

\[ Y_{i+1, j+1} = a \cdot X_{i+1, j+1} + b, \]  

(4.8)

where \(a\) and \(b\) are constants.

Root mean square difference fails to find an accurate match.

Correlation is the most expensive in terms of the number of instructions required to implement and computer processing time. Value of correlation will be 1 when intensity of each pixel in window is a sum of a constant and factor of the intensity of the corresponding pixel in the other view, i.e. equation (4.8). Though correlation is the most elaborate, it has been selected as it gives a match even if the variation of intensity is given by equation (4.8), whereas the other two fail. It will be shown that the intensity variation of the stereo views in the real world approximately follows equation (4.8).

The intensity of a particular point on the surface of an object is a function of

1. Angle between the normal to the surface and the direction of observation.
2. The coefficient of reflection \(K\) assigned to that particular point on the surface of the object.

Intensity of a particular point will be

\[ I = I_0 \cdot K \cdot \cos \theta + A_0 \]  

(4.9)
Assume stereo views of an object are taken. If $I_1$ is the intensity of a point in the first view and $I_2$ the intensity of the same point in the second view, then

$$I_1 = I_0 \cdot K \cdot \cos \theta_1 + A_0$$  \hspace{1cm} (4.10)

$$I_2 = I_0 \cdot K \cdot \cos(\theta_1 + \theta_2) + A_0$$  \hspace{1cm} (4.11)

Simplifying equation (4.11) and substituting equation (4.10),

$$I_2 = I_1 \cdot (\cos \theta_2 - \tan \theta_1 \cdot \sin \theta_2) + A_0 \cdot (1 - \cos \theta_2 + \tan \theta_1 \cdot \sin \theta_2)$$  \hspace{1cm} (4.12)

In case of narrow angle stereo views, $\theta_2$ will be nearly constant. If the window is over a small area compared to total object area, $\theta_1$ can be considered constant over window. So in this narrow angle case, the relation for the intensity variation for all pixels in windows selected in the real world stereo views is approximately given by

$$I_2 = a \cdot I_1 + b$$  \hspace{1cm} (4.13)

i.e. $Y_{i,j} = a \cdot X_{i,j} + b$  \hspace{1cm} (4.14)

Correlation gives a perfect match when all the pixels in two windows are given by equation (4.14). However, equation (4.14) is the approximate relation by which intensity of pixels of real world stereo views are related. Hence the probability of a match found by correlation of being wrong is small and it has been adopted in this work. When match occurs, correlation will have maximum value. The maximum value of correlation is 1 if the intensity of each pixel in two windows are exactly related by equation (4.14). It should be noted that eqns. (4.14) is not the exact relation by which the intensity of pixels of stereo views are related in the real world. As the equation (4.14) is not the exact
relation in the real world, correlation will not attain value of 1. Maximum value is in the range of 0.80 - 0.94. Figures 4-5 and 4-6 show variation of correlation for real world stereo views. When match occurs, it does not reach value of 1.0, but it does have a maximum value when match occurs.

A subroutine CORREL has been written which calculates correlation between two windows of two pictures. For finding match for a ceratin pixel in L.H.S. view, a pixel is selected in that view. Let \((i_0, j_0)\) be the position of a selected pixel. A section of the program gives \(\Delta i\) and \(\Delta j\), which is the average difference of \(I\) and \(J\) respectively between stereo views. An initial guess for match in the second view is taken as \((i_0 + \Delta i, j_0 + \Delta j)\).

A window of the desired size is formed around the pixel \((i_0, j_0)\) in the first view. A second window is formed around the guessed match point in the second view. This window is moved along \(I\) and \(J\) direction and the nature of variation of correlation is determined by calculating correlation with the help of subroutine CORREL. When correlation achieves a maximum, that value of correlation and the position of the center of the window is stored, which gives matched point pair.

4.4 Autocorrelation for Correct Match

If just the hill-like variation of correlation is utilized as a criterion to find match, it might lead to a wrong match. It is possible that at a pixel other than the perfect match, the correlation would show a similar nature and that might lead to a wrong match. Hence some check should be applied to the maximum value of correlation for finding match. Autocorrelation has been utilized as a criterion to provide check to the maximum value of correlation. Let the equation for correlation be defined
\[ \text{CORR} = C[X, i_o, j_o; Y, i_o', j_o'] \]  

so that

\[
\mathcal{X}_{\text{CORR}} = \frac{n \sum_{i=-n}^{m} \sum_{j=-m}^{m} \left( X_{i+i_o, j+j_o} - \bar{X} \right) \left( Y_{i+i_o', j+j_o'} - \bar{Y} \right)}{\sqrt{n \sum_{i=-n}^{m} \sum_{j=-m}^{m} (X_{i+i_o, j+j_o} - \bar{X})^2 \sum_{i=-n}^{m} \sum_{j=-m}^{m} (Y_{i+i_o', j+j_o'} - \bar{Y})^2}}
\]

Let \( A[i_o, j_o; di, dj] \) denote autocorrelation between the windows centered in picture \( X \) at \((i_o, j_o)\) and \((i_o + di, j_o + dj)\). In the notations of equation (4.15), this can be expressed as

\[ A[i_o, j_o; di, dj] = C[X, i_o, j_o; X, i_o + di, j_o + dj] \]  

(4.16)

If two images were identical except for a constant translation \((ti, tj)\), gain \(a\) and offset \(b\), i.e.

\[ Y_{i, j} = a \cdot X_{i+ti, j+tj} + b \]  

(4.17)

for all \(i\) and \(j\) in the images, then the correlation and autocorrelation surfaces would be exactly identical. For a pair of areas centered at \((i_o, j_o)\) and \((i_o', j_o')\), which are an intuitive match, we would have

\[ A[i_o, j_o; di, dj] = C[X, i_o, j_o; Y, i_o' + di, j_o' + dj'] \]  

(4.18)

for all \((di, dj)\) within two images. This is rarely the case, since most data of interest will have more meaningful changes between the two images than a constant translation, gain and offset. However, when we assume that there is little or no distortion over windows of the size being correlated, we effectively postulated that the changes between the two images are small locally. Consequently, while the equation (4.18) usually will not hold for
all \((di, dj)\) within the two images, it might be expected to hold within the immediate vicinity of the matching area centers.

Now we know that the correlation of \((i_o', j_o')\) with the areas-centered at points around \((i_o', j_o')\) yields values not greater than the correlation with \((i_o', j_o')\), i.e. for \(0 < |(di, dj)| < 2\),

\[
C[X, i_o, j_o; Y, i_o' + di, j_o' + dj] \leq C[X, i_o, j_o; Y, i_o', j_o']
\]

(4.19)
i.e. that the match correlation is not less than any of the immediate neighboring \(A[i_o, j_o; di, dj]\). Consequently, we would expect that the correlation is not less than the maximum of these autocorrelations, that is

\[
C[X, i_o, j_o; Y, i_o', j_o'] \geq \text{MAX} \left[ A[i_o, j_o; di, dj] \right]
\]

(4.20)

where \(0 < |(di, dj)| < 2\)

Equation (4.20) is applied to the maximum value of correlation to check whether correct match has been reached. Equation (4.20) is applicable when there is no noise in the digitizer or error in digitization. Due to the quantization error, noise, and nonlinearity of digitization, a constant \(C\) with a value \(C < 1\) should be included which will take in to account different errors and noise. So equation (4.20) becomes

\[
C[X, i_o, j_o; Y, i_o', j_o'] \geq C \cdot \text{MAX} \left[ A[i_o, j_o; di, dj] \right]
\]

(4.21)

where \(0 < |(di, dj)| < 2\)

Equation (4.21) is applied to the maximum value of correlation for the real world stereo views for finding correct match. Figure 4-7 shows variation of [1-Autodifference] for the real world stereo views. Figures 4-8 shows variation of autocorrelation for simulated(with noise) stereo views. Figures 4-9 and 4-10 show variation of autocorrelation for the real world stereo views.
FIGURE 4-7: Variation of 1.0-Autodifference for Real World Image Point (60,121)
FIGURE 4-8: Variation of Autocorrelation for Simulated Image Point (46,50)
FIGURE 4-9: Variation of Autocorrelation for Real World Image
Point (60, 121)
FIGURE 4-10: Variation of Autocorrelation for Real World Image Point (70,131)
A subroutine OTOCOR is written which calculates autocorrelation between two windows. Another subroutine MAXOTO is written which calculates autocorrelation at all 8 pixels which are around a given pixel, by calling subroutine OTOCOR. The maximum of the value of autocorrelation for eight pixels which are around a given pixel is determined. The maximum value is multiplied by a factor C to take into account quantization, nonlinearity and noise. Thus subroutine MAXOTO gives right hand side of equation (4.21), whereas the maximum value of correlation is determined as described in section 4.3. Equation (4.21) is applied to the value obtained by MAXOTO and the maximum value of correlation. When this is satisfied, it shows that a matched point pair has been achieved and another subroutine is called to obtain 3-D information for the matched point pair.

4.5 Efficient Match Finding

If one tries to find the match by calculating correlations at all pixels of the other view, one will have to compute a very large number of correlations. For example, for 128X128 pixels picture, one will have to compute $128 \times 128 \times 128 \times 268 \approx 268$ million correlations. This is exceedingly large. Some means should be taken to find the match efficiently in terms of computer time. Various techniques and algorithms have been utilized to find the match efficiently and save computer time. The processing scheme utilized to reduce computer time is now described.

Before we go further, one should note that this work dels with narrow-angle stereoscopy. The stereo views are taken by two cameras which are separated by a small distance in the lateral direction. The total angle subtended by the two cameras at the nearest point does not exceed $15^\circ$. 
Hence that portion of an object visible in one view and invisible in another view is very small. All points which are visible in one view and invisible in another view are simply neglected when finding 3-D information about them.

To find the match efficiently, use is made of the Laplacian operator, a cleaning algorithm, initial guess by average difference and hill-climbing method. This efficient match finding processing scheme can be broadly divided into 3 sections.
(a) Preprocessing scheme.
(b) Accurate initial guess.
(c) Hill-climbing method.

4.6 Preprocessing Scheme

In the preprocessing, a processing scheme is utilized on the digitized picture so as to obtain the average difference of I and J coordinates for stereo views efficiently. The average difference can be further utilized for an accurate initial guess to find the match efficiently.

Laplacian Operator:

Initially the Laplacian operator \( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \) is applied to the picture, viz.,

\[
\left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) P(I,J) = (4 \cdot P(I,J) - \left[ P(I-1,J) + P(I+1,J) + P(I,J-1) + P(I,J+1) \right]) \tag{4.22}
\]

where \( P(I,J) \) is the intensity of a pixel in the Ith row Jth column.

When the Laplacian operator is applied to the picture, the resultant picture will have a higher value at the edges in picture where the intensity
changes abruptly and edges of the object are obtained. After applying the Laplacian operator, the picture is thresholded by an appropriate value. By applying Laplacian operator and proper thresholding, a lined version of the picture is obtained.

**Thinning and Cleaning**

Even after the Laplacian operator is applied and thresholded, the picture will not have thin edges and sharply defined corners. It will generally have thick lines and it is difficult to find the exact position of corner points. Hence a thinning and cleaning algorithm is applied to extract sharp edges and corners of the object. This thinned and cleaned picture will have sharp edges and corners. The algorithm used is a modification and extension of Rutovitz's algorithm[27]. Two features are added to Rutovitz's algorithm.

1. Isolated points are removed.
2. Zig-zag lines are straightened.

Figure 4-11 shows the flow chart for this algorithm. The picture, after preprocessing by this algorithm will contain sharp edges and corner points. The picture version is hereafter termed the lined picture.

**Determining Corner Points**

An algorithm is applied to the lined version of the picture to find the corner points of the object. As the corners are sharply defined, it is easy to find a match for corners. By determining the number of nearby points which the lined version of the picture has, it finds corners and classifies them. The corners are stored differently depending upon number of edges forming them. For example, as shown in figure 4-12(a), when three edges
P(i,j) is picture element in ith row and jth column in a picture of size NIXNJ.

FIGURE 4-11: Flow-Chart for Thinning and Cleaning Algorithm
form a corner, it will have 3 nearby pixels around a corner in the region 
0<||(di, dj)||<2. In figure 4-12(b), when 4 edges form a corner, it will have 
4 nearby pixels around a corner in the region 0<||(di, dj)||<2. Thus each 
corner point is classified depending upon whether it is formed by 5, 4 or 
3 edges. Corners formed by 2 edges are neglected. Along with the position 
of corners, an index is stored which indicates the position of corners 
when the picture is scanned from top to bottom with movement from L.H.S. 
to R.H.S.

4.7 Accurate Initial Guess

In this section, the average difference of I and J coordinates between 
stereo views is found. By finding the average difference of I and J 
coordinates, one can make an accurate initial guess for matched point pairs 
in two views which will result in a large savings of computer time.

Finding the Average Difference

With the help of the preprocessing scheme, one knows the position of 
corners formed by different number of edges. For finding the average 
difference, a pixel which is a corner formed by 5 edges in the first view 
is selected. Also its index is known. A pixel which is in the same class 
and has the same index in another view is selected. The correlation 
between them is found and the algorithm checks whether they are matched point 
pair. If a matched point pair is found, 3-D information about that matched 
point pair is found and the difference of I and J coordinate is noted. If 
a match is not found, then correlation is found between that pixel in the 
first view and each corner point in the second view. The maximum of 
correlation and the position of that pixel in the second view is determined.
FIGURE 4-12:  Determining Number of Edges Forming a Corner

FIGURE 4-13:  Hill Climbing Method
Then a subroutine is called which searches for a match point in the region around that pixel and a matched point pair is found. The same procedure is applied to the corner points in other classes. As soon as 4 matched point pairs are found, the program jumps to a section of the program which finds the average difference of I and J coordinates from these 4 matched point pairs. As an object contains few corner points compared to total pixels in a picture, a matched point pair for corners can be found accurately with small computation time.

**Initial Guess by Average Difference**

With the help of the average difference obtained, substantial computation time is saved. This difference gives the approximate value by which each pixel in the first view is shifted in the other view. Average difference does not give the exact value of shift between two views, but it does give the approximate difference of a matched point pair in two views. With the help of average difference, one can calculate an initial guess for a matched point pair in the second view. Let \( (u_1, v_1) \) be the position of a pixel in the first view for which a matched point pair is to be found. If \( \Delta i \) and \( \Delta j \) are average difference of I and J coordinates, then the initial guess for the matched point pair is taken as \( (u_2, v_2) \), where

\[
\begin{align*}
  u_2 &= u_1 + \Delta i \quad (4.23) \\
  v_2 &= v_1 + \Delta j \quad (4.24)
\end{align*}
\]

A search is done around the pixel \( (u_2, v_2) \) for finding the matched point pair of \( (u_1, v_1) \). Using the average difference concept, a substantial savings in computer time is achieved. If one tries to find the match for a pixel in one view by brute force of calculating correlation at each
pixel in the second view, he will have to calculate $M^2$ correlations for a (MXM) pixels picture for each pixel in the first view. But with the difference scheme, this reduces to the range of 9 to 25.

4.8 Hill-Climbing Method

With the help of the average difference method, the approximate position for a matched point pair in the second view is obtained. To find the exact position of the pixel where maximum value of correlation occurs, a search around the initial guessed position should be carried out. A search procedure called hill-climbing is utilized to find the exact position of the pixel where this maximum value occurs. It finds the match with minimum computer time.

If one searches around a pixel in the region of $\pm n$ pixels for finding the position of maximum correlation, one will have to compute $(2n+1)^2$ correlations. The value of $n$ has to be judiciously chosen. If the maximum value of correlation occurs at a pixel which is separated from the guessed pixel by $n_1$ pixels in the horizontal direction and $n_2$ pixels in the vertical direction, then

$$n = n_1 + 1 \quad \text{if } n_1 > n_2$$
$$n = n_2 + 1 \quad \text{if } n_2 > n_1$$

For a real-world picture, if $n$ is taken as 4, one will have to compute $N_{COR} = (2 \cdot 4 + 1)^2 = 81$, i.e., 81 correlations if a direct search is applied. With hill-climbing, the number of correlations required to be performed reduces to $[9 + 5 \cdot (n-1)]$ for a given value of $n$. If $n$ is taken as 4, one will have to compute

$$N_{COR} = [9 + 5 \cdot (4-1)] = 24 \quad , \text{i.e.}$$
24 correlations if hill-climbing method is utilized. Thus for \( n=4 \), number of correlations required to be performed reduces to 24 from 81. In the hill-climbing method, it is not necessary to fix the value of \( n \); \( n \) will be individually selected depending upon pixel position. With the hill-climbing, the savings in computer time is achieved. The principle of the hill-climbing can be explained as follows:

Let \( P_2 \) be the initial guess given by the average difference method. Let \( P_3 \) be the position of a pixel where the maximum value of correlation occurs and is separated from \( P_2 \) by 2 pixels in the horizontal and 2 pixels in the vertical as shown in figure 4-13. If one tries to find the position of maximum correlation by finding correlation at all pixels in the region of \( \pm 3 \) pixels around the guessed pixel, one will have to compute 49 correlations.

In the hill-climbing method, initially the program calculates correlations at 9 pixels in the region defined by \( L_1 \) and also stores their value. Then it finds position of a pixel having the maximum correlation. As \( P_2' \) is nearer to \( P_3 \) than \( P_2 \), \( P_2' \) will have a higher value of correlation than \( P_2 \). \( P_2' \) becomes the modified point of match. Then correlations for pixels defined by region \( L_2 \) are calculated. As the previous values of correlations were stored, the program need not calculate correlations at 4 pixels which are common to region \( L_1 \) and \( L_2 \). It will compute only 5 correlations. Now \( P_3 \) is taken as a modified matched pixel. Correlation at all pixels defined by the region \( L_3 \) are computed. Here also it performs only 5 more correlations. As \( P_3 \) is a point of the highest correlation, it need not compute any more correlations. The program will give \( P_3 \) as the point of the highest correlation. Table 4.1 shows a comparison of correlations required to
be performed for the direct search method and the hill-climbing method for different values of \( n \).

### Table 4.1

<table>
<thead>
<tr>
<th>Value of ( n )</th>
<th># of Correlations Required to be Performed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Direct Search</strong></td>
</tr>
<tr>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>81</td>
</tr>
<tr>
<td>5</td>
<td>121</td>
</tr>
<tr>
<td>6</td>
<td>169</td>
</tr>
</tbody>
</table>

With \( n=3 \), the program will be required to calculate 19 instead of 49 correlations. In practice, the value of \( n \) should be taken of the order 4 to 5 to find a match. In that case, the hill-climbing method will result in about a 70% savings in computer time.

A large savings in computer time is achieved with the help of the processing scheme described in this chapter. It would have taken unrealistically long to find a match without this savings of time.
CHAPTER 5

FINDING THREE DIMENSIONAL INFORMATION FROM MATCHED POINT PAIR

Once a matched point pair has been found, the problem reduces to determining three-dimensional (3-D) coordinates from two-dimensional (2-D) coordinates of a matched point pair. In order to calculate complete information about an object, the intensity and 3-D coordinates for points on the outer surface of an object should be obtained. A matched point pair (MPP) gives the position of the image of a single object point in the stereo views. If the camera model (Section 5.1) is known, by applying the inverse camera model to a MPP, the corresponding 3-D coordinates can be calculated. A general MPP is defined by the coordinates \((u_1, v_1)\) and \((u_2, v_2)\) in the first and the second views, respectively. Two equations in terms of the three unknown coordinates can be obtained for the position of an image point in the first view. Two more equations are obtained from the coordinates \((u_2, v_2)\) of the MPP in the second view. With two views, four equations for three unknowns are obtained and unknowns \((X, Y, Z)\) can be solved uniquely.

This chapter describes:

1. Method utilized for determining 3-D information from a MPP.
2. A camera model which transforms 3-D space into a 2-D image in the image plane.
3. Equations which represent the camera in the real world.
4. An algorithm based on homogeneous transformation has been developed which can be utilized for calculating 3-D coordinates from 2-D coordinates of a matched point pair.
5.1 Camera Model

When dealing with 3-D problems, it becomes necessary to relate 2-D pictures of the image plane to the 3-D object in the real world. The method used to transform the 3-D object coordinates to 2-D image coordinates and vice-a-versa is called perspective transformation. The purpose of the camera model is to facilitate the derivation of a perspective transformation to map the world coordinates of the objects to the image coordinates and its inverse that maps the image coordinates to the object coordinates. The camera model used by photogrammetrists[22,29,32] is used in this work. In this model, camera is represented by a pinhole lens and distortions in the image due to lens aberrations and other hardware are neglected. The camera can be thought of as forming an image which is a point projection of its field of view. This is schematically represented in figure 5-1 by a point and a plane. The point C is known as focal center or center of projection. The plane which is called the ideal image plane is a reflected version of the physical image plane formed by the lens. The ray through C and normal to the ideal image plane is called the camera axis or principal ray. Assume for every point in the view, a ray from the focal center to the point in question. The image of a point is the intersection of this ray with the ideal image plane.

Figure 5-2 illustrates the camera model with the cartesian coordinate system centered at the center of the image plane. The Z axis points along the principal ray towards the image plane and the X axis is parallel to and
**FIGURE 5-1:** Camera Model with Physical and Ideal Image

**FIGURE 5-2:** Camera Model with Axis Along Z Axis and Focal Center at (0,0,-F)
has the same sense as the image horizontal. Let \( F \) be the focal length of the camera and \( V_0 \) be a point in the outside world whose coordinates are \( X \), \( Y \) and \( Z \). Then the image of this point \( V_0 \) will be formed on the image plane at a point which is the intersection of the image plane with a line joining focal center \( P \) and point \( V_0 \). If \( V_1 \) is the image of point \( V_0 \), then the coordinates of point \( V_1 \) are given by

\[
\begin{align*}
x_1 &= \frac{X \cdot F}{F + Z} \\
y_1 &= \frac{Y \cdot F}{F + Z}
\end{align*}
\tag{5.1}
\tag{5.2}

Equations (5.1)-(5.2) give the coordinates of the image of a point on image plane. They describe a camera model wherein the image plane is in the \( X-Y \) plane at \( Z=0 \) and the camera axis is along the \( Z \) axis. This is a primitive model. Consider a camera with focal length \( F \) and center point at \( X',Y',Z' \) and axis along \( Z \) axis. Then the coordinates of a point \( P \) with coordinates \( X,Y,Z \) are given by

\[
\begin{align*}
x_1 &= X' + \frac{(X-X') \cdot F}{(Z-Z')} \\
y_1 &= Y' + \frac{(Y-Y') \cdot F}{(Z-Z')}
\end{align*}
\tag{5.3}
\tag{5.4}

Now consider an ideal case with two cameras separated by distance \( B \) between them. Their axes are parallel to \( Z \) axis with focal length \( F \) and focal centers at \((B/2,0,H)\) and \((-B/2,0,H)\). Figure 5-3 shows this configuration. If \((x_L,y_L)\) is coordinates of a image point in L.H.S. view and \((x_R,y_R)\) is coordinates of a image point in R.H.S. view, then \( x_L, y_L, x_R \) and \( y_R \) are given by
FIGURE 5-3: Ideal Stereo Views with Relation Between Different Coordinates
\[ x_L = \frac{(X+B/2) \cdot F}{H-Z} \quad (5.5) \]
\[ y_L = \frac{Y \cdot F}{H-Z} \quad (5.6) \]
\[ x_R = \frac{(X-B/2) \cdot F}{H-Z} \quad (5.7) \]
\[ y_R = \frac{Y \cdot F}{H-Z} \quad (5.8) \]
\[ y_L = y_R \quad (5.9) \]

When matched point pair is found, \((x_L, y_L)\) and \((x_R, y_R)\) are known.

If \(F\), \(F\) and \(B\) are known, \(X\), \(Y\) and \(Z\) coordinates can be calculated using the following equations:

\[ X = \frac{(x_L+x_R) \cdot B}{x_L-x_R} \quad (5.10) \]
\[ Y = \frac{y_L \cdot B}{x_L-x_R} = \frac{y_R \cdot B}{x_L-x_R} \quad (5.11) \]
\[ Z = H - \frac{B \cdot F}{x_L-x_R} \quad (5.12) \]
\[ p = \frac{p_1+p_2}{2} \quad (5.13) \]

where \(p_1\) is the intensity of a matched point pair in the first view.

\(p_2\) is the intensity of a matched point pair in the second view.

A program was written to simulate stereo views of a cube by using eqns. (5.5)-(5.8). Another program was written to calculate \(X\), \(Y\) and \(Z\) coordinates for different pixels by using eqns. (5.10)-(5.12).

Figure 5-4 shows the results of this program for a particular sectional plane of cube, i.e. for the same value of \(y_L\) and \(y_R\).
However, it should be noted that when one tries to calculate coordinates for a real world object by using eqns. (5.10)-(5.12), they fail to give accurate values of 3-D coordinates information. This is because eqns. (5.10)-(5.12) are applicable only for ideal stereo views which assumes that both camera axes are exactly parallel to the Z axis and in the horizontal plane. In the real world system this is not true as it is practically impossible to keep both camera axes exactly parallel to the Z axis and in the horizontal plane. Hence further equations, which take into account these nonideal conditions of the real world should be used to determine 3-D information.

In the real world, there is some rotation of the camera axis about the X and Y axes. During digitization, there may be rotation of the image about the Z axis, as the digitizer may not scan in the horizontal of the camera. Also it is difficult to determine the exact value of camera focal length and separation between cameras. Due to digitization, there might be an error in determining the exact position of the image point on the image plane. These conditions should be taken into account when finding 3-D information from a matched point pair of the real world object. The camera model and correspondingly its inverse should take into consideration all of the above facts.

5.2 Real World Camera Model

Eqns. (5.1)-(5.2) give the image of a point in space on the image plane when the camera axis is along the Z axis with focal center at \((0,0,-F)\). These equations can be modified so as to include the various conditions of the real world like rotation about X, Y and Z axes, inaccuracy in determining the position of axis on the image plane, etc. Eqns. (5.14)-(5.39) provide the coordinates of a point on the image plane for different conditions.
Symbols

$F$ is the focal length of the camera.

$x, y, z$ is the coordinates of a point in space.

$x', y', z'$ is the coordinates of the focal center of the camera.

$X_1, Y_1, Z_1$ is a coordinate system whose center is at $x', y', z'$ and its axes parallel to the $X, Y$ and $Z$ axes.

$x_1$ is $X$ coordinate of image of a point on the image plane referred to the $X_1, Y_1$ and $Z_1$ axes.

$y_1$ is $Y$ coordinate of image of a point on the image plane referred to the $X_1, Y_1$ and $Z_1$ axes.

$a_1$ is an angle of rotation of the camera axis about the $X_1$ axis.

$a_2$ is an angle of rotation of the camera axis about the $Y_1$ axis.

$a_3$ is an angle of rotation of the camera axis about the $Z_1$ axis.

$x_1'$ is the error in determining the camera axis position along the $X$ direction.

$y_1'$ is the error in determining the camera axis position along the $Y$ direction.

$(u_1, v_1)$ is the coordinates of a matched point pair in the first view.

$(u_2, v_2)$ is the coordinates of a matched point pair in the second view.

$T$ is the camera transfer matrix for the first view.

$T'$ is the camera transfer matrix for the second view.
5.2 A

No rotation of axis about X, Y, Z axes. Camera focal center at 
(x',y',z'). No error in determining camera axis position on image plane.

\[ x_1 = \frac{(x-x') \cdot F}{(z-z')} \]  
\[ y_1 = \frac{(y-y') \cdot F}{(z-z')} \]  

5.2 B

No rotation of axis about X, Y, Z axes. Camera focal center at 
(x',y',z'). Error in determining camera axis position on image plane.

\[ x_1 = \frac{(x-x') \cdot F}{(z-z')} - x_1' \]  
\[ y_1 = \frac{(y-y') \cdot F}{(z-z')} - y_1' \]  

5.2 C

Camera axis rotated about X axis by a degree. Camera focal center 
at (x',y',z'). Error in determining camera axis position on image plane.

\[ z_3 = z' + (z-z') \cdot \cos \alpha_1 + (y-y') \cdot \sin \alpha_1 \]  
\[ x_1 = \frac{(x-x') \cdot F}{(z_3-z')} - x_1' \]  
\[ y_1 = \frac{(y_3-y') \cdot F}{(z_3-z')} - y_1' \]  

Substituting eqns. (5.18) and (5.19) in eqns. (5.20) and (5.21),

\[ x_1 = \frac{(x-x') \cdot F}{(z-z') \cdot \cos \alpha_1 + (y-y') \cdot \sin \alpha_1} - x_1' \]  

\[ y_1 = \frac{(y-y') \cdot F}{(z-z') \cdot \cos \alpha_1 + (y-y') \cdot \sin \alpha_1} - y_1' \]
\[ y_1 = \frac{[(y-y') \cdot \cos a_1 - (z-z') \cdot \sin a_1] \cdot F}{(z-z') \cdot \cos a_1 + (y-y') \cdot \sin a_1} - y_1' \]  \hspace{1cm} (5.23)

5.2.D

Camera axis rotated about X1 and Y1 axes by \( a_1 \) and \( a_2 \) degrees, respectively. Camera focal center at \((x', y', z')\). Error in determining camera axis position on the image plane.

\[ z_3 = z' + (z-z') \cdot \cos a_1 + (y-y') \cdot \sin a_1 \]  \hspace{1cm} (5.24)

\[ y_3 = y' + (y-y') \cdot \cos a_1 - (z-z') \cdot \sin a_1 \]  \hspace{1cm} (5.25)

\[ z_4 = z' + (z_3-z') \cdot \cos a_2 - (x-x') \cdot \sin a_2 \]  \hspace{1cm} (5.26)

\[ x_4 = x' + (x-x') \cdot \cos a_2 + (z_3-z') \cdot \sin a_2 \]  \hspace{1cm} (5.27)

\[ y_4 = y_3 \]  \hspace{1cm} (5.28)

Substituting eqns. (5.24)-(5.25) in eqns. (5.26)-(5.27)

\[ x_4 = x' + (x-x') \cdot \cos a_2 + (z-z') \cdot \cos a_1 \cdot \sin a_2 + (y-y') \cdot \sin a_1 \cdot \sin a_2 \]  \hspace{1cm} (5.29)

\[ z_4 = z' + (z-z') \cdot \cos a_1 \cdot \cos a_2 + (y-y') \cdot \sin a_1 \cdot \cos a_2 - (x-x') \cdot \sin a_2 \]  \hspace{1cm} (5.30)

\[ y_4 = y_3 = y' + (y-y') \cdot \cos a_1 - (z-z') \cdot \sin a_1 \]  \hspace{1cm} (5.31)

\[ x_1 = \frac{(x_4-x') \cdot F}{(z_4-z')} - x_1' \]  \hspace{1cm} (5.32)

\[ y_1 = \frac{(y_4-y') \cdot F}{(z_4-z')} - y_1' \]  \hspace{1cm} (5.33)

Substituting eqns. (5.29)-(5.31) in eqns. (5.32)-(5.33),
\[ x_1 = \frac{[(x-x') \cdot \cos a_2 + (z-z') \cdot \cos a_1 \cdot \sin a_2 + (y-y') \cdot \sin a_1 \cdot \sin a_2] \cdot F}{(z-z') \cdot \cos a_1 \cdot \cos a_2 + (y-y') \cdot \sin a_1 \cdot \cos a_2 - (x-x') \cdot \sin a_2} - x_1' \quad (5.34) \]

\[ y_1 = \frac{[(y-y') \cdot \cos a_1 - (z-z') \cdot \sin a_1] \cdot F}{(z-z') \cdot \cos a_1 \cdot \cos a_2 + (y-y') \cdot \sin a_1 \cdot \cos a_2 - (x-x') \cdot \sin a_2} - y_1' \quad (5.35) \]

5.2 E

Camera axis rotated about X1 and Y1 axes by \( a_1 \) and \( a_2 \) degrees, respectively. Image is rotated by \( a_3 \) degrees during digitization. Camera focal center at \((x', y', z')\). Error in determining camera axis position on the image plane.

\( x_3, y_3, z_3, x_4, y_4 \) and \( z_4 \) are the same as those in eqns. (5.24)-(5.31).

\[ x_5 = \frac{(x_4 - x') \cdot F}{(z_4 - z')} \]

\[ y_5 = \frac{(y_4 - y') \cdot F}{(z_4 - z')} \]

\[ x_1 = x_5 \cdot \cos a_3 + y_5 \cdot \sin a_3 - x_1' \quad (5.36) \]

\[ y_1 = y_5 \cdot \cos a_3 - x_5 \cdot \sin a_3 - y_1' \quad (5.37) \]

Substituting proper values in eqns. (5.36) and (5.37),

\[ x_1 = \frac{[[((x-x') \cdot \cos a_2 + (z-z') \cdot \cos a_1 \cdot \sin a_2 + (y-y') \cdot \sin a_1 \cdot \sin a_2] \cdot F]}{(z-z') \cdot \cos a_1 \cdot \cos a_2 + (y-y') \cdot \sin a_1 \cdot \cos a_2 - (x-x') \cdot \sin a_2} \cdot \cos a_3 + \]

\[ \left[ \frac{[(y-y') \cdot \cos a_1 - (z-z') \cdot \sin a_1] \cdot F}{(z-z') \cdot \cos a_1 \cdot \cos a_2 + (y-y') \cdot \sin a_1 \cdot \cos a_2 - (x-x') \cdot \sin a_2} \right] \cdot \sin a_3 - x_1' \quad (5.38) \]
\[ y_1 = \frac{[(y-y') \cdot \cos a_1 - (z-z') \cdot \sin a_1] \cdot F}{(z-z') \cdot \cos a_1 \cdot \cos a_2 + (y-y') \cdot \sin a_1 \cdot \sin a_2 - (x-x') \cdot \sin a_2} \cdot \cos a_3 - \]
\[ \frac{[(x-x') \cdot \cos a_2 + (z-z') \cdot \sin a_1 \cdot \sin a_2 + (y-y') \cdot \sin a_1 \cdot \sin a_2] \cdot F}{(z-z') \cdot \cos a_1 \cdot \cos a_2 + (y-y') \cdot \sin a_1 \cdot \sin a_2 - (x-x') \cdot \sin a_2} - \sin a_3 \cdot \]

(5.39)

Eqns. (5.38)-(5.39) give the coordinates of an image of a point on the image plane whose coordinates are \((x, y, z)\), so that

1. Camera focal center is at \((x', y', z')\) with focal length \(F\).
2. Camera axis is rotated \(a_1\) degrees about \(X\) axis, \(a_2\) degrees about \(Y\) axis, and image is rotated \(a_3\) degrees about \(Z\) axis.
3. Error in finding image of camera axis on image plane is \(x_1'\) in \(X\) coordinates and \(y_1'\) in \(Y\) coordinate.

Equations (5.38) and (5.39) are quite complex, but they represent a real world system. In the real world, \(x_1\) and \(y_1\) are converted into integer numbers by some scale factor. There is some digitization error in \(x_1\) and \(y_1\) in real world data. If \(a_1, a_2, a_3, x, y, z, F, x_1'\) and \(y_1'\) are known, values of \(x_1\) and \(y_1\) can be calculated for known values of \(x, y\) and \(z\). However, it is not possible to find \((x, y, z)\) just from eqns. (5.38)-(5.39), as in that case, there are only 2 equations for 3 unknowns. Two more equations are obtained by taking a second view of the object. Two more equations in \(x, y\) and \(z\) are obtained by finding \((x_2, y_2)\), which is the matched point pair for point \((x_1, y_1)\) of the first view, i.e. image of the
same point in space in second view. Equations similar to (5.38) and (5.39)
can be obtained for the second view. For the second view
(1) Camera focal center is at \((x'_2, y'_2, z'_2)\) with focal length \(F_2\).
(2) Camera axis is rotated \(a'_1\) degrees about \(X\) axis, \(a'_2\) degrees about \(Y\)
axis and during digitization, image is rotated \(a'_3\) degrees about \(Z\) axis.
(3) Error in finding image of camera axis on image plane is \(x'_1\)" in \(X\)
coordinates and \(y'_1\)" in \(Y\) coordinates.

Then equations for \(x_2\) and \(y_2\) are given by
\[
x_2' = \frac{[(x-x'_2) \cdot \cos a'_2 + (z-z'_2) \cdot \cos a'_1 \cdot \sin a'_2 + (y-y'_2) \cdot \sin a'_1 \cdot \sin a'_2] \cdot F_2}{(z-z'_2) \cdot \cos a'_1 \cdot \cos a'_2 + (y-y'_2) \cdot \sin a'_1 \cdot \cos a'_2 - (x-x'_2) \cdot \sin a'_2} \cdot \cos a'_3 +
\]
\[
y_2' = \frac{[(y-y'_2) \cdot \cos a'_1 - (z-z'_2) \cdot \sin a'_1] \cdot F_2}{(z-z'_2) \cdot \cos a'_1 \cdot \cos a'_2 + (y-y'_2) \cdot \sin a'_1 \cdot \cos a'_2 - (x-x'_2) \cdot \sin a'_2} \cdot \sin a'_3 -
\]
\[
(5.40)
\]
\[
x_1'' = \frac{[(x-x'_1) \cdot \cos a'_2 + (z-z'_1) \cdot \cos a'_1 \cdot \sin a'_2 + (y-y'_1) \cdot \sin a'_1 \cdot \sin a'_2] \cdot F_2}{(z-z'_1) \cdot \cos a'_1 \cdot \cos a'_2 + (y-y'_1) \cdot \sin a'_1 \cdot \cos a'_2 - (x-x'_1) \cdot \sin a'_2} \cdot \cos a'_3 -
\]
\[
y_1'' = \frac{[(y-y'_1) \cdot \cos a'_1 - (z-z'_1) \cdot \sin a'_1] \cdot F_2}{(z-z'_1) \cdot \cos a'_1 \cdot \cos a'_2 + (y-y'_1) \cdot \sin a'_1 \cdot \cos a'_2 - (x-x'_1) \cdot \sin a'_2} \cdot \sin a'_3 -
\]
\[
(5.41)
\]

Eqns. (5.40) and (5.41) represent the second view system.

If all the parameters of the camera model, i.e. \(x', y', z', a_1, a_2, a_3, F, F_2\),
\(x'_1, y'_1, x'_2, y'_2, z'_2, a'_1, a'_2, a'_3, x''_1, y''_1\) are known and \((x'_1, y'_1)\) and \((x'_2, y'_2)\), which
are the coordinates of a matched point pair in two views are known,
coordinates \((x, y, z)\) can be calculated from the sets of eqns. (5.38)-(5.41).
However solving eqns. (5.38)-(5.41) is a tedious task. A new approach based on the homogeneous transformation technique[18] is utilized for calculating \((x,y,z)\) coordinates from \((x_1,y_1)\) and \((x_2,y_2)\). This approach is believed unique for our application. This approach is much simpler and easier.

5.3 Homogeneous Transformation Technique

Homogeneous transformation technique is utilized to transform an \(N\) dimensional space to \(N+1\) dimensional space. When a point \([x_1,x_2,x_3,\ldots,x_N]\) in an \(N\) dimensional space is transformed to a \(N+1\) dimensional space using the homogeneous transformation, its homogeneous coordinates in \(N+1\) dimensional space becomes \([x_1\cdot w, x_2\cdot w, x_3\cdot w, \ldots, x_N\cdot w, w]\), where \(w\) is a non-zero constant.

When a point \([x_1,x_2,x_3,\ldots,x_P]\) in \(P\) dimensional space is transformed to a \(P+1\) dimensional space, its homogeneous coordinates becomes \([x_1/x_P, x_2/x_P, x_3/x_P, \ldots, x_{P-1}/x_P]\).

For calculating 3-D coordinates from 2-D coordinates of a matched point pair, the camera is considered as a device which transforms three dimensional space to two dimensional space. The technique for calculating 3-D coordinates from coordinates of a matched point pair using homogeneous transformation is now described. The camera is considered as a [4X3] matrix which transforms a four dimensional point (produced by transforming a 3-D point in space to a 4-D space using homogeneous transformation) to a three dimensional point (produced by transforming a 2-D point on the image plane to a 3-D space using homogeneous transformation).

Let there be a point \((x,y,z)\) in space. Image of this point on the image plane of camera 1 is \((u_1,v_1)\). Let camera transformation matrix for view 1 be \(T\).
\[
T = \begin{bmatrix}
T_{11} & T_{12} & T_{13} \\
T_{21} & T_{22} & T_{23} \\
T_{31} & T_{32} & T_{33} \\
T_{41} & T_{42} & T_{43}
\end{bmatrix}
\]

then,

\[
\begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix} = \begin{bmatrix}
T_{11} & T_{12} & T_{13} \\
T_{21} & T_{22} & T_{23} \\
T_{31} & T_{32} & T_{33} \\
T_{41} & T_{42} & T_{43}
\end{bmatrix} \begin{bmatrix}
x \\
y \\
z
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix} = \begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix}
\]

where \( u_1 = \frac{x'}{w} \), \( v_1 = \frac{y'}{w} \)

Carrying out multiplication shown in equation (5.44),

\[
T_{11} \cdot x + T_{21} \cdot y + T_{31} \cdot z + T_{41} = w \cdot u_1
\]

\[
T_{12} \cdot x + T_{22} \cdot y + T_{32} \cdot z + T_{42} = w \cdot v_1
\]

\[
T_{13} \cdot x + T_{23} \cdot y + T_{33} \cdot z + T_{43} \cdot z = w
\]

Substituting expression for \( w \) from equation (5.47) in to eqns. (5.45) and (5.46),

\[
(T_{11} - T_{13} \cdot u_1) \cdot x + (T_{21} - T_{23} \cdot u_1) \cdot y + (T_{31} - T_{33} \cdot u_1) \cdot z +
(T_{41} - T_{43} \cdot u_1) = 0
\]

\[
(T_{12} - T_{13} \cdot v_1) \cdot x + (T_{22} - T_{23} \cdot v_1) \cdot y + (T_{32} - T_{33} \cdot v_1) \cdot z +
(T_{42} - T_{43} \cdot v_1) = 0
\]

If three space coordinates \((x,y,z)\) and transformation matrix \(T\) are known, eqns. (5.48) and (5.49) give values of \( u_1 \) and \( v_1 \), two dimensional
coordinates in the photograph. If the transformation matrix \( T \) and \( (u_1, v_1) \) are known, eqns. (5.48) and (5.49) have the form

\[
a \cdot x + b \cdot y + c \cdot z + d = 0
\]

which is the equation of a plane in three dimensional space. Eqns. (5.48) and (5.49) can be written as

\[
a_1 \cdot x + b_1 \cdot y + c_1 \cdot z + d_1 = 0 \quad (5.51)
\]

\[
a_2 \cdot x + b_2 \cdot y + c_2 \cdot z + d_2 = 0 \quad (5.52)
\]

where

\[
a_1 = T_{11} - T_{13} \cdot u_1 \quad a_2 = T_{12} - T_{13} \cdot v_1
\]

\[
b_1 = T_{21} - T_{23} \cdot u_1 \quad b_2 = T_{22} - T_{23} \cdot v_1
\]

\[
c_1 = T_{31} - T_{33} \cdot u_1 \quad c_2 = T_{32} - T_{33} \cdot v_1
\]

\[
d_1 = T_{41} - T_{43} \cdot u_1 \quad d_2 = T_{42} - T_{43} \cdot v_1
\]

Let \( T' \) be the transfer matrix for the second camera and the image of a point on the image plane of the second camera be \((u_2, v_2)\).

\[
T' = \begin{bmatrix}
T_{11}' & T_{12}' & T_{13}' \\
T_{21}' & T_{22}' & T_{23}' \\
T_{31}' & T_{32}' & T_{33}' \\
T_{41}' & T_{42}' & T_{43}'
\end{bmatrix}
\]

Two more equations similar to eqns. (5.51) and (5.52) for the second view can be written as

\[
a_3 \cdot x + b_3 \cdot y + c_3 \cdot z + d_3 = 0 \quad (5.53)
\]

\[
a_4 \cdot x + b_4 \cdot y + c_4 \cdot z + d_4 = 0 \quad (5.54)
\]

where

\[
a_3 = T_{11}' - T_{13}' \cdot u_2 \quad a_4 = T_{12}' - T_{13}' \cdot v_2
\]
\[ b_3 = T_{21} - T_{23} \cdot u_2; \quad b_4 = T_{22} - T_{23} \cdot v_2 \]

\[ c_3 = T_{31} - T_{33} \cdot u_2; \quad c_4 = T_{32} - T_{33} \cdot v_2 \]

\[ d_3 = T_{41} - T_{43} \cdot u_2; \quad d_4 = T_{42} - T_{43} \cdot v_2 \]

Writing eqns. (5.51)-(5.54) in matrix form,

\[
\begin{bmatrix}
  a_1 & b_1 & c_1 \\
  a_2 & b_2 & c_2 \\
  a_3 & b_3 & c_3 \\
  a_4 & b_4 & c_4 \\
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
\end{bmatrix}
=
\begin{bmatrix}
  -d_1 \\
  -d_2 \\
  -d_3 \\
  -d_4 \\
\end{bmatrix}
\tag{5.55}
\]

or \([C] \cdot [X] = [D]\) \tag{5.56}

where

\[
[C] =
\begin{bmatrix}
  a_1 & b_1 & c_1 \\
  a_2 & b_2 & c_2 \\
  a_3 & b_3 & c_3 \\
  a_4 & b_4 & c_4 \\
\end{bmatrix}
\tag{5.57}
\]

\[
[X] =
\begin{bmatrix}
  x \\
  y \\
  z \\
\end{bmatrix}
\tag{5.58}
\]

\[
[D] =
\begin{bmatrix}
  -d_1 \\
  -d_2 \\
  -d_3 \\
  -d_4 \\
\end{bmatrix}
\tag{5.59}
\]

\[
[C] \cdot [X] = [D]
\]

\[
[c^T] \cdot [C] \cdot [X] = [c^T] \cdot [D]
\tag{5.60}
\]
\[ [CPR] \cdot [X] = [C^T] \cdot [D] \]  

where  
\[ [CPR] = [C^T] \cdot [C] \]  

\[ [X] = [CPR]^{-1} \cdot [C^T] \cdot [D] \]  

If \((u_1, v_1), (u_2, v_2)\) and camera transformation matrix \(T\) and \(T'\) of both cameras are known, \((x, y, z)\) can be calculated using eqns. (5.51)-(5.62).

Solution obtained by this is the same as that obtained by solving eqns. (5.51)-(5.54) by a least squares fit.

**How to Get Camera Transformation Matrix**

Each point \((x, y, z)\) in space gives two independent equations if the camera transformation matrix is known. The camera transformation matrix has 12 unknowns. Since a homogeneous system is considered, the matrix will include an arbitrary scale factor and one of the unknowns can be freely set to any non-zero value. To obtain 11 unknowns, 11 independent eqns. are required. But as one point gives 2 independent equations, at least 5 1/2 points will be needed to obtain \(T\). As fractional points cannot be taken, at least 6 points in each picture are required to determine the transformation matrix. There should be at least 6 points whose \((x, y, z)\) coordinates and their positions \((u, v)\) coordinates in both views are known a priori. Let there be 6 points \(a, b, c, d, e, f\) and set \(T_{43} = 1\). The resulting 12 equations with the help of eqns. (5.48) and (5.49) can be written in matrix form.
\[
\begin{pmatrix}
\mathbf{u}_a & \mathbf{u}_b & \mathbf{u}_c & \mathbf{u}_d & \mathbf{u}_e & \mathbf{v}_a & \mathbf{v}_b & \mathbf{v}_c & \mathbf{v}_d & \mathbf{v}_e & \mathbf{v}_f
\end{pmatrix}
= \begin{pmatrix}
T_{11} & T_{12} & T_{13} & T_{14} & T_{15} & T_{16}
T_{21} & T_{22} & T_{23} & T_{24} & T_{25} & T_{26}
T_{31} & T_{32} & T_{33} & T_{34} & T_{35} & T_{36}
\end{pmatrix}
\begin{pmatrix}
\mathbf{T} \mathbf{1} \mathbf{1} \\
\mathbf{T} \mathbf{1} \mathbf{2} \\
\mathbf{T} \mathbf{1} \mathbf{3} \\
\mathbf{T} \mathbf{1} \mathbf{4} \\
\mathbf{T} \mathbf{1} \mathbf{5} \\
\mathbf{T} \mathbf{1} \mathbf{6}
\end{pmatrix}
\]
\[ [A] \cdot [T_1] = [B] \]
\[ [A^T] \cdot [A] \cdot [T_1] = [A^T] \cdot [B] \]
(5.55)

Let \([APR] = [A^T] \cdot [A] \)
\[ [APR] \cdot [T_1] = [A^T] \cdot [B] \]
(5.66)
\[ [T_1] = [APR]^{-1} \cdot [A^T] \cdot [B] \]
(5.67)

If 6 points whose \((x, y, z)\) coordinates and their position on the image planes are known, the elements of camera transfer matrix can be found from eqns. (5.63)-(5.67). Once a transfer matrix is known, 3-D coordinates of a point from 2-D coordinates can be obtained using eqns. (5.51)-(5.62).

For calculating camera transfer matrix and 3-D coordinates from 2-D coordinates of a matched point pair, the following steps are taken.

1. Find out at least 6 points whose coordinates \((x_a, y_a, z_a), (x_b, y_b, z_b), \ldots, (x_f, y_f, z_f)\) are known.
2. Obtain position of these points in both views.
3. Calculate camera transfer matrix \(T\) and \(T'\) for both views using eqns. (5.63)-(5.67).
4. Determine a matched point pair in two views and find its coordinates \((u_1, v_1)\) and \((u_2, v_2)\).
5. Using eqns. (5.51)-(5.62), calculate 3-D coordinates of a point.

It can be shown that camera model described in section 5.2 is a version of the camera transfer matrix. By using the proper transformation matrix, the model for a camera can be obtained. The transformation matrix for the different conditions described in section 5.2 are given.
(A) For 5.2 A
\[
T = \begin{bmatrix}
-\frac{F}{z^t} & 0 & 0 \\
0 & -\frac{F}{z^t} & 0 \\
0 & 0 & -\frac{1}{z^t} \\
\frac{F \cdot x'}{z^t} & \frac{F \cdot y'}{z^t} & 1
\end{bmatrix}
\]  \hspace{1cm} (5.68)

(B) For 5.2 B
\[
T = \begin{bmatrix}
-\frac{F}{z^t} & 0 & 0 \\
0 & 0 & 0 \\
-\frac{-x'}{z^t} & -\frac{y'}{z^t} & 0 \\
\frac{F \cdot x'}{z^t} + x' & \frac{F \cdot y'}{z^t} + y' & 1
\end{bmatrix}
\]  \hspace{1cm} (5.69)

(C) For 5.2 C
\[
T = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \text{Cosa}_1 & \text{Sina}_1 & 0 \\
0 & -\text{Sina}_1 & \text{Cosa}_1 & 0 \\
0 & y' - y' \cdot \text{Cosa}_1 & z' - y' \cdot \text{Sina}_1 & 1
\end{bmatrix}
\]
\[
\cdot \begin{bmatrix}
-\frac{F}{z^t} & 0 & 0 \\
0 & -\frac{F}{z^t} & 0 \\
-\frac{-x'}{z^t} & -\frac{y'}{z^t} & 0 \\
\frac{F \cdot x'}{z^t} + x' & \frac{F \cdot y'}{z^t} + y' & 1
\end{bmatrix}
\]  \hspace{1cm} (5.70).
(D) For 5.2 D

\[
T = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos a_1 & \sin a_1 & 0 \\
0 & -\sin a_1 & \cos a_1 & 0 \\
0 & y' - y' \cdot \cos a_1 & z' - y' \cdot \sin a_1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
\cos a_2 & 0 & -\sin a_2 & 0 \\
0 & 1 & 0 & 0 \\
\sin a_2 & 0 & \cos a_2 & 0 \\
x' - x' \cdot \cos a_2 & 0 & z' - z' \cdot \cos a_2 & 1 \\
-\sin a_2 & \cos a_2 & 0 & x' \cdot \sin a_2 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
-\frac{F}{z'} & 0 & 0 \\
0 & \frac{F}{z'} & 0 \\
\frac{-x_1}{z'} & \frac{-y_1}{z'} & \frac{-1}{z'} \\
\frac{F \cdot x'}{z'} + x_1 & \frac{F \cdot y'}{z'} + y_1 & 1 \\
\end{bmatrix}
\]

(E) For 5.2 E

\[
T = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos a_1 & \sin a_1 & 0 \\
0 & -\sin a_1 & \cos a_1 & 0 \\
0 & y' - y' \cdot \cos a_1 & z' - y' \cdot \sin a_1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
\cos a_2 & 0 & -\sin a_2 & 0 \\
0 & 1 & 0 & 0 \\
\sin a_2 & 0 & \cos a_2 & 0 \\
x' - x' \cdot \cos a_2 & 0 & z' - z' \cdot \cos a_2 & 1 \\
-\sin a_2 & \cos a_2 & 0 & x' \cdot \sin a_2 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
-\frac{F}{z'} & 0 & 0 \\
0 & -\frac{F}{z'} & 0 \\
\frac{-x_1}{z'} & \frac{-y_1}{z'} & \frac{-1}{z'} \\
\frac{F \cdot x'}{z'} + x_1 & \frac{F \cdot y'}{z'} + y_1 & 1 \\
\end{bmatrix}
\]
A program was written which calculates the transformation matrix $T$ and $T'$ for two views from the known 6 points and then calculates value of $(x, y, z)$ coordinates for given values of $(u_1, v_1)$ and $(u_2, v_2)$ by using eqns. (5.51)-(5.62). The position of a few discrete points were simulated and the program was used to obtain $(x, y, z)$ coordinates from the positions of the simulated points. Table 5.1 shows the results.

<table>
<thead>
<tr>
<th>X Coordinate</th>
<th>Y Coordinate</th>
<th>Z Coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Value</td>
<td>Found Value</td>
<td>Actual Value</td>
</tr>
<tr>
<td>7.50</td>
<td>7.499</td>
<td>4.00</td>
</tr>
<tr>
<td>7.50</td>
<td>7.499</td>
<td>0.00</td>
</tr>
<tr>
<td>2.75</td>
<td>2.750</td>
<td>3.00</td>
</tr>
<tr>
<td>2.75</td>
<td>2.750</td>
<td>0.00</td>
</tr>
<tr>
<td>9.75</td>
<td>9.749</td>
<td>0.00</td>
</tr>
<tr>
<td>9.75</td>
<td>9.749</td>
<td>3.00</td>
</tr>
<tr>
<td>9.00</td>
<td>8.999</td>
<td>2.00</td>
</tr>
<tr>
<td>6.50</td>
<td>6.499</td>
<td>1.00</td>
</tr>
<tr>
<td>6.50</td>
<td>6.499</td>
<td>7.00</td>
</tr>
<tr>
<td>5.50</td>
<td>5.499</td>
<td>8.00</td>
</tr>
</tbody>
</table>

Maximum percentage error in $X = 0.00623$

Maximum percentage error in $Y = 0.06760$

Maximum percentage error in $Z = 0.01259$

Percentage mean square error of $X = 0.00437$
Percentage mean square error of $Y = 0.05199$
Percentage mean square error of $Z = 0.01032$
Total percentage mean square error = 0.03071

Various statistics about error are shown below Table 5.1. The maximum percentage error is 0.06760, maximum percentage mean square error is 0.05199 and total percentage mean square error is 0.03071. As the results show, values found are quite accurate. It can be concluded from the results that homogeneous transformation technique is quite accurate. It should work efficiently for many practical applications. Typical applications are:

(1) Computer Analysis of 3-D objects:

Given orthographic views (PLAN and ELEVATION, or PLAN and SIDEVIEW, or ELEVATION and SIDEVIEW) of parts, the transformation matrix can be determined. By finding 2-D coordinates manually or with the help of the proper digitizer, 3-D coordinates can be determined easily. This would facilitate computer analysis of 3-D objects for various engineering applications.

(2) Computer Display:

The design, delineation and display of arbitrary objects for arbitrarily chosen viewpoints can be done conveniently on the graphic display console of a computer. For arbitrarily chosen viewpoints, the transformation matrix can be determined readily. Once the transformation matrix is known, the object can be displayed from known coordinates of object points using equation (5.44).
CHAPTER 6

RESULTS FOR SIMULATED AND REAL WORLD OBJECTS

Thus far an algorithm has been described for measuring three dimensional (3-D) information. Two major sections of the algorithm were verified by the results shown in Chapters 4 and 5. However, the complete algorithm should be checked for its validity and accuracy. This chapter describes:

1. Program for the implementation of complete algorithm.
2. Error analysis of the proposed algorithm.
3. Experimental set-up to get data for real world stereo views.
4. Results obtained for both a simulated and real world stereo views.

6.1 Program

The algorithm consists of two major sections. Section I determines matched point pairs from stereo views (Chapter 4). The second section evaluates 3-D information from matched point pairs (Chapter 5). A program to implement the entire algorithm was written in FORTRAN-V for a Data General NOVA-840 computer. The program assumes that the digitized stereo views are stored in two files. The program mainly consists of a mainline program and a number of subroutines. The entire program has been stored in a directory named STEREO. A complete listing of the program is given in appendix A. A brief description of the program including the subroutines is now given.

Mainline Program

The file name for the mainline program is PROCESS. It assumes that stereo views of an object are digitized with 8 bits grey level/pixel and
stored in two files. The number of columns in a view should be 128 or 256. The number of rows should be a multiple of 4 for views with 256 columns and a multiple of 8 for views with 128 columns. Initially the program reads \((X,Y,Z)\) coordinates and their positions in stereo views for 12 different points, which are used for calculating the camera transformation matrix. The Laplacian operator is applied to the digitized views and stored in another file. Then a thinning and cleaning algorithm is applied to picture. By determining the end points of the object, the program determines the limit along two axes between which the actual object is present. This is repeated for the second view. The program classifies different corners of the edged version of the picture by determining the number of edges forming them. Then in turn, matched point pairs formed by 5 edges, 4 edges and 3 edges are found. When a matched point pair is found, the program calls a subroutine(COCOR), which calculates 3-D information about that matched point pair. As soon as 4 matched point pairs are found, the program branches to a section of program which finds the average difference of the \(I\) and \(J\) coordinates between matched point pairs. After the average difference is calculated, the program finds the matched point pair for each pixel of the edged version of picture by searching for a match around the pixel formed by adding the average difference of \(I\) and \(J\) to the desired pixel. Whenever a matched point pair is found, the program calls a subroutine(COCOR), which finds 3-D information for this matched point pair. When 3-D information about all pixels of edged version of picture is found, the program finds 3-D information for each pixel of the object.
Subroutines:

TRANSF

This subroutine calculates camera transformation matrix $T$ and $T'$ for both views. A set of 3-D coordinates and their positions on stereo views for $n$ points, $6 < n < 12$, is provided to this subroutine and it calculates the camera transformation matrix for both views.

COCOR

This subroutine calculates 3-D information for a given matched point pair. Position of the matched point pair is provided to the subroutine which uses the algorithm described in Chapter 5. Intensity is found by taking average intensity.

CORREL

This subroutine calculates correlation for a given point pair by using the equation (4.5).

OTOCOR

This subroutine calculates autocorrelation for a given point pair.

MAXOTO

This subroutine is utilized to obtain maximum value of autocorrelation in the region around a pixel so that it can be utilized as a check to the maximum value of correlation for finding correct match.

HILL

This subroutine implements a hill-climbing technique. Position of a pixel is provided to the subroutine. It searches around that pixel and finds the position of a pixel which has maximum correlation.
MINVRD

This subroutine is utilized to invert a double precision matrix. Inverse matrix is also a double precision matrix.

PRMA

This subroutine is utilized to calculate product of two matrices.

TRANP

This subroutine finds transpose of a matrix.

DOUBLE

This subroutine converts a single precision matrix to a double precision matrix by adding zeros for less significant digits.

SINGLE

This subroutine converts a double precision matrix to a single precision matrix by truncating less significant digits.

SUBSTR

Performs subtraction of two matrices.

DRAW

This subroutine prints edged version of a digitized view. Given a two dimensional array, which contains properly thresholded view of an object, it prints it.

6.2 Error Analysis of Proposed Algorithm

There are mainly two causes which would give erroneous results.

(1) Incorrect matched point pair.

(2) Quantization error.

Incorrect Matched Point Pair

If the correct matched point pair is not found, it will give wrong values of coordinates because in that case 2 of 4 equations used for finding 3-D
coordinates of a point refer to one point whereas the other two equations refer to some other point which has different coordinates. Correlation has been used to avoid this kind of error as the probability of a match found by correlation of being wrong is small. However, due to noise, digitization of image and blurring of image during digitization, there will be a few points which have this kind of error. Probability of this kind of error is quite small (less than 1% for a good digitizer, about 5% in our digitizer). When this kind of error is present, there will be a large error (about 20% in our case) in the results obtained by such a point pair. Accurate matched point pairs should be found to avoid this kind of error. **Quantization Error**

Here quantization error means positional quantization error. Since the coordinates of points are available which are those at integer points at the grid points on the picture, this error is introduced. This is the main source of error in measurement. It is introduced due to the digitization of the image. The position of a point on the image plane is not its exact position, but its digitized representation. Error due to quantization in position is introduced at two places. First the transformation matrix T (and T') will not be exact, but will have error since it is determined from the digitized representation of image point position. Inherently there is some error in the transformation matrix. Next, quantization error will be present in the position of matched point pairs itself. As matched point pairs coordinates are a digitized representation of their exact coordinates, there will be error in \((u_1, v_1)\) and \((u_2, v_2)\) of eqns. (5.51)-(5.54). This error is magnified when one tries to calculate 3-D coordinates using
eqns. (5.51)-(5.62). Quantization error is a function of the number of
digits to which a view is digitized. As the number of pixels to which a
view is digitized increases, the quantization error is reduced. It is a
complicated function of the number of pixels to which a view is digitized,
as it is introduced at two places.

To find the nature of variation of quantization error as a function
of the number of pixels, a few discrete points were simulated on the stereo
views. For this simulation, it was assumed that all pixels are within a
space of 12"X10"X8". The cameras were assumed to be at a distance of 30"
from the nearest edge, i.e. 40" from the farthest edge. The Y axis was
assumed to be along depth direction and the Z axis along the height direction.
Simulated stereo views were digitized to different number of pixels. The
algorithm described in Chapter 5 was used to calculate the coordinates of
these points. Figures 6-1 to 6-5 show the variation of the maximum percentage
error of Y, maximum percentage error of X, maximum percentage error of Z,
percentage mean square error of Y and total percentage mean square error,
respectively. All of these have similar nature of variation. Y, i.e. the
depth direction has maximum error, as this information was lost during the
picture taking process and is derived by the algorithm. It should be noted
that percentages are calculated as percentage of full scale, i.e. size of
the space 12"X10"X8". Hence if percentage error is calculated as percentage
of the distance between camera and object, shown error should be divided
by 4 to obtain percentage error.

6.3 Experimental Set-up

The algorithm was checked for a real world object. To get data for
the real world object, stereo views of an object (a rectangular box) were
FIGURE 6-1: Variation of Maximum Percentage Error of $Y$ with Number of Pixels
FIGURE 6-2: Variation of Maximum Percentage Error of X with Number of Pixels

MAXIMUM PERCENTAGE ERROR OF X

NUMBER OF PIXELS
FIGURE 6-3: Variation of Maximum Percentage Error of Z with Number of Pixels
FIGURE 6-4: Variation of Percentage Mean Square Error of Y with Number of Pixels
FIGURE 6-5: Variation of Total Percentage Mean Square Error with Number of Pixels
taken. Six thin sticks with blackened ends were placed at different places in a space of 12"X10"X8". The object was placed in this space. Stereo views of that object were taken. Transparencies of the stereo views were prepared and images were digitized using a flying-spot-scanner. Views were digitized to 256X256 pixels, but the effective number of pixels representing the space of 12"X10"X8" was 150X105. Pixels outside this region do not contain any useful information. Stereo views were trimmed to insert to the program. The images of the end points of the sticks were determined and provided to the program. Figure 6-6(a) and (b) show left and right views of the object. As can be seen from these figures, there is lot of noise and blurring of images in digitization of picture using our flying-spot-scanner.

Stereo views of an object(a rectangular box with a slot) were simulated so that the program could be used with unblurred images. These views were hand-digitized. A gaussian noise with signal to noise ratio of 200(23 db.) was added to these views. Figure 6-7(a) and (b) show left and right view of these simulated stereo views.

6.4 Results

A computer program(Section 6.1) written to implement the algorithm was used to measure 3-D information from the simulated and real world stereo views. For simulated views, a space of 12"X10"X8" corresponded to 80X50 pixels. But the effective number of pixels representing the object was only 50X30. The program was used to measure 3-D information about the object from simulated noise-free stereo views. Table 6.1 shows comparison of the actual and found values for several discrete points.
FIGURE 6-6: Digitized Stereo Views of Real World Image

(a) LEFT VIEW

(b) RIGHT VIEW
TABLE 6.1

<table>
<thead>
<tr>
<th>X Coordinate</th>
<th>Y Coordinate</th>
<th>Z Coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Value</td>
<td>Found Value</td>
<td>Actual Value</td>
</tr>
<tr>
<td>1.40</td>
<td>1.427</td>
<td>7.50</td>
</tr>
<tr>
<td>6.80</td>
<td>6.623</td>
<td>6.50</td>
</tr>
<tr>
<td>1.40</td>
<td>1.247</td>
<td>6.00</td>
</tr>
<tr>
<td>1.70</td>
<td>1.686</td>
<td>6.50</td>
</tr>
<tr>
<td>7.50</td>
<td>7.632</td>
<td>4.00</td>
</tr>
<tr>
<td>3.00</td>
<td>3.009</td>
<td>7.30</td>
</tr>
<tr>
<td>5.00</td>
<td>5.166</td>
<td>4.00</td>
</tr>
<tr>
<td>1.80</td>
<td>1.691</td>
<td>7.20</td>
</tr>
<tr>
<td>5.20</td>
<td>5.173</td>
<td>7.00</td>
</tr>
<tr>
<td>3.20</td>
<td>3.099</td>
<td>4.50</td>
</tr>
</tbody>
</table>

Maximum percentage error of Y was found to be 15% of full scale.
The mean square error of Y was found to be about 4% of full scale. This error is mainly due to the quantization error. Though the maximum percentage error is more, considering the small number of pixels to which the views were digitized, it appears to be acceptable.

A gaussian noise with signal to noise ratio of 200 (23 db.) was added to the noise-free simulated stereo views. The program was used to measure 3-D information about this object. Table 6.2 shows comparison of actual and found values for several points.


<table>
<thead>
<tr>
<th>X Coordinate Actual Value</th>
<th>X Coordinate Found Value</th>
<th>Y Coordinate Actual Value</th>
<th>Y Coordinate Found Value</th>
<th>Z Coordinate Actual Value</th>
<th>Z Coordinate Found Value</th>
<th>Intensity</th>
<th>Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.77</td>
<td>5.675</td>
<td>6.65</td>
<td>7.443</td>
<td>6.00</td>
<td>6.038</td>
<td>80</td>
<td>82</td>
</tr>
<tr>
<td>4.95</td>
<td>4.887</td>
<td>9.00</td>
<td>9.573</td>
<td>0.30</td>
<td>0.362</td>
<td>50</td>
<td>53</td>
</tr>
<tr>
<td>4.76</td>
<td>4.598</td>
<td>7.50</td>
<td>7.482</td>
<td>6.00</td>
<td>5.878</td>
<td>80</td>
<td>83</td>
</tr>
<tr>
<td>4.50</td>
<td>4.641</td>
<td>8.90</td>
<td>9.561</td>
<td>0.10</td>
<td>0.231</td>
<td>70</td>
<td>73</td>
</tr>
<tr>
<td>6.40</td>
<td>6.202</td>
<td>7.50</td>
<td>7.392</td>
<td>6.00</td>
<td>5.922</td>
<td>80</td>
<td>83</td>
</tr>
<tr>
<td>5.00</td>
<td>4.893</td>
<td>9.00</td>
<td>9.548</td>
<td>0.10</td>
<td>0.231</td>
<td>50</td>
<td>52</td>
</tr>
<tr>
<td>6.30</td>
<td>6.702</td>
<td>3.80</td>
<td>3.319</td>
<td>5.70</td>
<td>5.538</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>6.50</td>
<td>6.471</td>
<td>7.50</td>
<td>7.382</td>
<td>6.00</td>
<td>5.930</td>
<td>50</td>
<td>54</td>
</tr>
<tr>
<td>4.95</td>
<td>4.881</td>
<td>9.00</td>
<td>9.598</td>
<td>0.59</td>
<td>0.493</td>
<td>50</td>
<td>54</td>
</tr>
<tr>
<td>4.90</td>
<td>4.870</td>
<td>8.20</td>
<td>7.439</td>
<td>6.00</td>
<td>5.754</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>5.00</td>
<td>4.899</td>
<td>9.10</td>
<td>9.522</td>
<td>0.00</td>
<td>0.099</td>
<td>90</td>
<td>92</td>
</tr>
<tr>
<td>2.65</td>
<td>2.754</td>
<td>7.00</td>
<td>7.653</td>
<td>6.00</td>
<td>5.962</td>
<td>50</td>
<td>53</td>
</tr>
<tr>
<td>5.10</td>
<td>5.514</td>
<td>9.10</td>
<td>9.507</td>
<td>0.00</td>
<td>0.097</td>
<td>90</td>
<td>92</td>
</tr>
<tr>
<td>5.90</td>
<td>6.085</td>
<td>3.69</td>
<td>3.415</td>
<td>5.80</td>
<td>5.672</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>3.20</td>
<td>3.537</td>
<td>7.20</td>
<td>7.858</td>
<td>6.00</td>
<td>5.981</td>
<td>80</td>
<td>82</td>
</tr>
</tbody>
</table>

Maximum percentage error of Y was found to be 15% of full scale. The mean square error of Y was about 4% of full scale. Though the maximum percentage error of Y is large, considering the small number of pixels to which views were simulated, it is acceptable.
Stereo views of an object (a rectangular box) were taken by using a stereo camera. These views were digitized using a flying-spot-scanner. The flying-spot-scanner used, has poor linearity, adds noise and blurs the image. Views were digitized to 256X256 pixels, but the effective number of pixels representing the object was only 80X80. The program was used to obtain 3-D information about the object. Figure 6-8 shows comparison of the actual and found boundary for a small section of the object. Table 6.3 shows comparison of actual and found values for a few discrete points.

<table>
<thead>
<tr>
<th>X coordinate</th>
<th>Y Coordinate</th>
<th>Z Coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Value</td>
<td>Found Value</td>
<td>Actual Value</td>
</tr>
<tr>
<td>5.50</td>
<td>5.589</td>
<td>6.28</td>
</tr>
<tr>
<td>8.50</td>
<td>8.732</td>
<td>7.40</td>
</tr>
<tr>
<td>7.10</td>
<td>7.370</td>
<td>8.06</td>
</tr>
<tr>
<td>5.55</td>
<td>5.676</td>
<td>6.32</td>
</tr>
<tr>
<td>3.50</td>
<td>3.485</td>
<td>4.00</td>
</tr>
<tr>
<td>6.00</td>
<td>6.212</td>
<td>6.80</td>
</tr>
<tr>
<td>5.70</td>
<td>5.941</td>
<td>6.50</td>
</tr>
<tr>
<td>4.10</td>
<td>4.059</td>
<td>4.72</td>
</tr>
<tr>
<td>6.50</td>
<td>6.761</td>
<td>7.40</td>
</tr>
<tr>
<td>8.30</td>
<td>8.448</td>
<td>7.62</td>
</tr>
<tr>
<td>6.30</td>
<td>6.503</td>
<td>7.18</td>
</tr>
<tr>
<td>5.70</td>
<td>5.941</td>
<td>6.50</td>
</tr>
<tr>
<td>9.00</td>
<td>9.328</td>
<td>6.86</td>
</tr>
<tr>
<td>6.35</td>
<td>6.600</td>
<td>7.20</td>
</tr>
<tr>
<td>5.65</td>
<td>5.853</td>
<td>6.46</td>
</tr>
</tbody>
</table>
The maximum percentage error of Y was found to be 10% of full scale. Mean square error of Y was found to be about 3% of full scale. Although the maximum percentage error seems large, considering noise and blurring introduced by the scanner and small number of pixels representing the actual object, this error is acceptable. There were a few (about 5%) pixels which have error due to incorrect matched point pair. This is mainly due to excessive noise and blurring introduced in the data during digitization. If stereo views were digitized to a larger number of pixels using a higher resolution digitizer, more accurate results would be obtained. Results obtained are satisfactory considering the limitations present.
CHAPTER 7

SUMMARY AND CONCLUSION

The objective of this work was to develop an efficient algorithm for measurement of 3-D information using stereo views. A major requirement was that the algorithm must function with a real world environment, where it is practically impossible to obtain ideal stereo views and noise is present. The algorithm must be efficient in terms of computer time and accuracy of measuring real world information to be acceptable.

An algorithm is developed for measurement of 3-D information using stereo views. The algorithm first finds a matched point pair using correlation and then calculates 3-D coordinates of a matched point pair using homogeneous transformation. Comparison of various methods for measuring 3-D information was undertaken. The stereoview processing method was selected because of its moderate instrumentation requirements and capability of giving accurate information. An investigation of the problems encountered in the use of this method in real world was undertaken. A processing scheme has been suggested to overcome these problems. Correlation has been selected for match finding as it is the most reliable as compared to absolute difference and mean square error based upon intensity variation of the stereo views in the real world. An equation has been developed based on autocorrelation to check the validity of each match found by the correlation technique. A complete algorithm has been implemented for match finding. A special search procedure, which utilizes Laplacian operator, hill-climbing, average difference, etc. has been used for efficient match finding. This match
finding algorithm gives accurate results with the real world and simulated stereo views while minimizing processing time.

A camera model (in real-world conditions) for perspective transformation has been developed. Also, the homogeneous transformation was used because of its analytical simplifications. An algorithm based on homogeneous transformation was constructed for calculating 3-D coordinates from 2-D coordinates of a matched point pair. An error analysis of the proposed algorithm has been given. Sources of error in the results are described. The complete algorithm was implemented in FORTRAN-V and utilized to measure information about the real world and simulated (with/without noise) objects. Though the percentage error was large, useful results would be obtained if the images had been digitized to a larger number of pixels using a higher quality digitizer.

The major conclusions and contributions of this research can be summarized in the following:

1. An algorithm has been developed and successfully implemented for measuring 3-D information using stereo views. The algorithm was used to measure the 3-D information about the real world and simulated (with/without noise) objects.

2. The stereoview method has certain advantages compared to other methods for measuring 3-D information.

3. Suggestion of correlation for matching: application of correlation gives the most reliable match compared to absolute difference and root mean square difference techniques. The validity of the match found by correlation can be checked with autocorrelation.
(4) A special search procedure using Laplacian operator, average difference, hill-climbing, etc. has been suggested for efficient match finding.

(5) The camera model for the perspective transformation of space into the image plane of the real world has been investigated. An algorithm based on the homogeneous transformation has been developed to obtain 3-D information from 2-D coordinates of a matched point pair. This algorithm can be used with real world objects. Determination of the camera model with the aid of six known (a priori) points amounts to a calibration.

(6) The stereoview method for measurement of 3-D information in real world appears practical. The error in measurement can be significantly reduced if the image is digitized to a sufficiently large (e.g., 1024x1024) number of pixels.

Areas of Further Investigation

In the process of this investigation, a number of areas have been discovered which warrant further study.

(1) Reduction of Error by Using Modified Matched Point Pair Coordinates:

The area of modification of matched point pairs coordinates for error reduction in results has been untouched. A major source of error in the results is the quantized representation of matched point pairs coordinates. This error is magnified as the processing proceeds. An investigation should be carried out to modify the coordinates of a matched point pair from the nature of the variation of correlation around that pixel and 3-D coordinates of nearby pixels and then using this modified coordinates for calculating 3-D information. This investigation could result in a
significant reduction in error. This work would be particularly helpful when the images are digitized to a small number of pixels.

(2) Effects of Noise on Matching:

A more detailed analysis of noise should be undertaken. The effects of noise introduced into the data during the digitization and other stages and its effect on matching should be investigated. This work would allow one to eliminate the error introduced in the results if sources of noise are known.

(3) Finding Information About Unmatched Pixel:

As the present investigation deals with narrow-angle stereoscopy, no information is found about the pixel for which a matched point pair cannot be found. The problem of finding information about the pixel for which a matched point pair cannot be found (with the help of some kind of extrapolation) should be investigated. This would seem necessary in wide-angle stereoscopy.

We leave these as challenges to the future investigators.
APPENDIX A

PROGRAM LISTING

The algorithm developed in this thesis was implemented in FORTRAN-V on Data General NOVA-840 computer. Listing of the entire program is given here. To use the program, number of columns in stereo views should be 128 or 256. Number of rows should be multiple of 8 for views with 128 columns and multiple of 4 for views with 256 columns.

PROGRAM

DIMENSION NAME1(5), NAME2(5), NAMES(5), NAME3(5), I*(256),
L*(256), IS(256), IX(256), IY(256)
C******
C*****PROGRAM FOR PROCESSING OF STEREO VIEWS TO OBTAIN
C*****THREE DIMENSIONAL INFORMATION.
C******
DIMENSION I*HIND(1024),
DIMENSION I*:MEJ(12), IMEJ(12), XM(12), YM(12), ZM(12),
DIMENSION IPAST(256), I*(8), Q*(8), KLIND(10),
C******
C******
C******
DIMENSION I*:K12, K12(10), K2(10), K2(10), U12(10),
DIMENSION CV(12), U2(12), V2(12), IIND1(12), JIND1(12), IIND2(12),
DIMENSION CJIND(12),
DIMENSION B15(20), B14(50), B13(200), B12(1000),
DIMENSION B25(20), B24(50), B23(200), B22(1000),
DIMENSION I*:DIFF(4), IIYDIF(4), I*:NOPTY(4), I*:NOPT(4),
DIMENSION RES(12), I*:KSX(4), I*:SY(4),
DIMENSION TL(4), T1(4), T2(4), T3(4),
COMMON /E1/ I*HIND,
COMMON /CC/ TL, T2,
COMMON /DD/ N1, N2, N3, LLM, NAME1, NAME2, MMLEN, MMREC,
COMMON /EE/ N112, NIJ2, NAMES, NAME3,
COMMON /FF/ NABCD,
COMMON /GG/ CCBAD
COMMON /HH/ LLM22
C******INPUT SECTION OF PROGRAM
C******READ 12 DATA CARDS CONTAINING COORDINATES OF 12 DISCRETE
C******POINTS AND THEIR POSITION IN STEREO VIEWS.
C******DO 992 I=1, 12
C******READ(3, 992) X(I), Y(I), Z(I),
C******CONTINUE

108
END OF LISTING
*** APPLY LAPLACIAN OPERATOR TO FIRST VIEW AND STORE IN ANOTHER FILE.
THRESHOLD:

DO 204 J=1,NMLEN
201 J=J+1
WRITE(C) (IN4(J), J=1,NMLEN)
DO 202 I=2,NL1
IMIN1=1
CALL fseek(A, IMIN1)
READ(C) (IN1(LZ), LZ=1,NMLEN)
READ(3) (IN2(LZ), LZ=1,NMLEN)
DO 203 J=2,NJ1
IN4(J)=ABS(4+IN2(J)-IN1(J)+1)+IN2(J-1)+IN2(J)+IN1(J)
IF(IN4(J) .GT. 11THRS) GO TO 205
IN4(J)=0
GO TO 202
205 IN4(J)=1
202 CONTINUE
IN4(J)=0
IN4(NJ1)=0
WRITE(C) (IN4(J), J=1,NMLEN)
206 CONTINUE
DO 207 J=1,NMLEN
204 IN4(J)=0
WRITE(C) (IN4(J), J=1,NMLEN)
CLOSE 3
CLOSE 1
NMLEN=1
CALL crank(NMVIEW)
NL1=NL1-1
NJ1=NJ1-1
DO 227 J=1,NJ1
IPAST(J)=0
227 CONTINUE
*** APPLY THINNING AND CLEANING ALGORITHM TO THRESHOLDED FIRST VIEW:
OPEN 2, NAME=LEN2=NMLEN, REC=NMREC
229 KOUNT=0
NJ2=NJ1-1
NJ2=NJ1-1
DO 222 I=2,NL1
IMIN1=1
CALL fseek(A, IMIN1)
READ(C) (IN1(LZ), LZ=1,NMLEN)
READ(3) (IN2(LZ), LZ=1,NMLEN)
READ(2) (IN2(LZ), LZ=1,NMLEN)
DO 221 J=2,NJ2
IF(IN2(J) .EQ. 0) GO TO 221
P(1)=IN1(J)
P(2)=IN2(J-1)
P(3)=IN2(J)
P(4)=IN2(J+1)
P(5)=IN2(J)
P(6)=IN2(J-1)
P(7)=IN1(J-1)
P(8)=IN1(J-1)
Q(2)=P(2)
Q(2)=P(2)
Q(2)=P(2)
Q(2)=P(2)
Q(2)=P(2)
Q(2)=P(2)
Q(2)=P(2)
Q(2)=P(2)
M=P(1)+P(2)+P(2)+P(2)+P(2)+P(2)+P(2)+P(2)
N=0
NL=0
DO 225 K=L-1
   IF(CEK(K), EQ. 0) AND. (CEK(K+1), EQ. 1) JMP NNN=NN+1
221 IF(CPK(K), EQ. 0) AND. (CPK(K+1), EQ. 1) JMP NNN=NN+1
   IF(CPK(K), EQ. 0) AND. (CPJ(K), EQ. 1) JMP NN=NN+1
   IF(NN LE. 1) GO TO 225
   IF(NN LE. 0) GO TO 227
   GO TO 221
227 IF(NN, EQ. 1) GO TO 229
   GO TO 240
229 IF(CPK(K)+CPJ(K)+CP(K), EQ. 0) OR. (NNPLOT(J), NE. 1) GO TO 343
   GO TO 231
242 IF(CPK(K)+CPJ(K)+CP(K), EQ. 0) OR. (NN, NE. 1) GO TO 245
   GO TO 231
245 CONTINUE
   NNPLOT(J)=N
   INZ(J)=0
   KOUNT=KOUNT+1
   GO TO 231
241 BABOR=0.
   DO 247 K=1, L
      BABOR=BABOR+10.+PK(K)
   IF(BABOR, EQ. 1101100. 0) GO TO 225
   IF(BABOR, EQ. 11000110. 0) GO TO 225
   IF(BABOR, EQ. 101110001. 0) GO TO 225
   IF(BABOR, EQ. 11011. 0) GO TO 225
   IF(BABOR, EQ. 101100. 0) GO TO 225
   IF(BABOR, EQ. 11000010. 0) GO TO 225
   IF(BABOR, EQ. 11011110. 0) GO TO 225
   IF(BABOR, EQ. 10110000. 0) GO TO 225
   IF(BABOR, EQ. 11010000. 0) GO TO 225
   IF(BABOR, EQ. 10100000. 0) GO TO 225
   IF(BABOR, EQ. 11010000. 0) GO TO 225
   IF(BABOR, EQ. 10100000. 0) GO TO 225
   IF(BABOR, EQ. 10100000. 0) GO TO 225
   IF(BABOR, EQ. 10100000. 0) GO TO 225
   GO TO 231
235 INZ(J)=0
231 CONTINUE
   INZ(J)=0
   INZ(J+2)=0
   INZ(J+1)=0
   CALL FSEEK(2, 1)
   WRITE(2) (INZ(J), J=1, NJ)
222 CONTINUE
   IF(KOUNT, GT. 0) GO TO 229
   DO 249 J=1, NJ
      INZ(J)=0
249 CONTINUE
   CALL FSEEK(2, 1)
   WRITE(2) (INZ(J), J=1, NJ)
   GO 240 J=1, NJ
240 INZ(J)=0
230 CALL FSEEK(3, NJ)
   WRITE(3) (INZ(J), J=1, NJ)
   CLOSE 2
   CALL DRAWVIEW
C****** CALCULATE TRANSFER MATRIX BY CALLING SUBROUTINE TRANSF.
   NNMML=12
   DO 195 I=1, NNML
      CALL TRANSF(IINCI, JINCI, IINCI+V, Z, NNML)
195 CONTINUE

8
C********** FIND BOUNDARY FOR FIRST VIEW.
DO 521 I=1,6
IF(V1(I).GT.MAX) MAX=V1(I)
521 CONTINUE
V1L=MAX
TYPE"VALUE OF V1L = " V1L
MIN=V1L
DO 522 I=7,12
IF(V1(I).LT.MIN) MIN=V1(I)
522 CONTINUE
V1M=MIN
TYPE"VALUE OF V1M = " V1M
MIN=V1M
DO 523 I=13,18
IF(V1(I).LT.MIN) MIN=V1(I)
IF(V1(I).GT.MAX) MAX=V1(I+6)
523 CONTINUE
V1R=MIN
TYPE"VALUE OF V1R = " V1R
MIN=V1R
DO 524 I=19,24
IF(V1(I).LT.MIN) MIN=V1(I)
IF(V1(I).GT.MAX) MAX=V1(I+6)
524 CONTINUE
V1R=MIN
TYPE"VALUE OF V1R = " V1R
MIN=V1R
C********** FIND CORNER POINTS IN FIRST VIEW AND STORE THEM.
OPEN 2, FILE=1, LEN=20, REC=20, EOC=20
I12=0
I13=0
I14=0
I15=0
DO 501 I1=UL1,ULM
I1M1=I1-1
CALL FSEEK(2, I1M1)
READ(C) (IX1,LZC,LZ1=NJ)
READ(C) (IX2,LZC,LZ1=NJ)
READ(C) (IX3,LZC,LZ1=NJ)
DO 501 JJ=V1M,V1L
IF(IN2(JJ).EQ.0) GO TO 501
NALIS=IN2(JJ-1)+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1
IF(NALIS.LE.0) GO TO 501
IF(NALIS.GE.6) GO TO 501
GO TO (502,503,504,505), NALIS
503 I12=I12+1
IF(I12.GT.20) GO TO 501
AI1=I1
C11=AI1+1000
E15(I12)=C11+JJ
GO TO 501
504 I14=I14+1
IF(I14.GT.50) GO TO 501
AI1=I1
C11=AI1+1000
E14(I14)=C11+JJ
GO TO 501
502 I12=I12+1
IF(I12.GT.200) GO TO 501
AI1=I1
C11=AI1+1000
B10(I12)=C11+JJ
GO TO 501
503 I12=I12+1
IF(I12.GT.1000) GO TO 501
AI1=I1
C11=AI1+1000
B12(I12)=C11+JJ
501 CONTINUE
C*******FIND BOUNDARY FOR THE OBJECT, IN FIRST VIEW.
DO 701 II=1:UL:V1M
    CALL FSEEK(2,II)
    READ(2) (IK1(I),LLZ=1,LNJ)
    DO 702 JJ=1:UL:V1M
    IF(IN2(JJ).EQ.0) GO TO 702
    I1M=II-1
    GO TO 703
702 CONTINUE
701 CONTINUE
702 CONTINUE
DO 706 II=1,255
    IC=W1M-II+1
    CALL FSEEK(2,IC)
    READ(2) (IK1(I),LLZ=1,LNJ)
    DO 704 JJ=1:UL:V1M
    IF(IN2(JJ).EQ.0) GO TO 704
    I1M=IC-1
    GO TO 705
704 CONTINUE
706 CONTINUE
705 CONTINUE
CLOSE 2

C*******START PROCESSING SECOND VIEW.
OPEN 4, FILE=NAME2, LEN=2-MMLEN, REC=MMREC
OPEN 4, FILE=NAME4, LEN=2-MMLEN, REC=MMREC

C*******APPLY LAPLACIAN OPERATOR TO SECOND VIEW, THRESHOLD
C*******IT AND STORE IT IN ANOTHER FILE.
DO 401 J=1:LNJ
401 IN4(J)=0
    WRITE(4) (IK4(I),J=1,MMLEN)
    DO 402 I=2,N11
        IN41=I-1
        CALL FSEEK(2,IN41)
        READ(2) (IK1(I),LLZ=1,MMLEN)
        READ(2) (IK4(I),LLZ=1,MMLEN)
        READ(2) (IK2(I),LLZ=1,MMLEN)
        DO 403 J=2,LNJ
            IN4(J)=ABS((IK4(I)+2*IK4(J-1)+IK4(J)+1)
                      -2*(IK4(J)+IK4(J+1)))/4
            IF(C14(J).GT.12THRS) GO TO 405
        IN4(J)=0
        GO TO 403
405 IN4(J)=1
402 CONTINUE
IN4(1)=0
IN4(NJ)=0
    WRITE(4) (IK4(I),J=1,MMLEN)
402 CONTINUE
DO 404 I=1,L,MLEN
    WRITE(4) (IK4(I),J=1,MLEN)
    CLOSE 4
    CLOSE 2
    CALL DRAKNVIEW
    N11=N11-1
    NJ1=NJ1-1
    DO 427 J=1,NJ1
        NFAST(J)=0
427 CONTINUE
C*******APPLY THINNING AND CLEANING ALGORITHM TO THRESHOLDED
C*******SECOND VIEW.
OPEN 4, FILE=NAME4, LEN=2-MMLEN, REC=MMREC
429 KOUNT=0
    NJ2=NJ1-1
    DO 432 I=2,N11
        CAL3=I-1
        IM1M=I-1
        GO TO 432
432 CONTINUE
CALL FREE(14), MIN1
RETURN(3) INTO(LZZ), LZZ=1, NJ
RETURN(4) INTO(LZZ), LZZ=1, NJ
DO 421 J=2, NJ
IF(INX(J), EQ.0) GO TO 421
P(J)=INX(J)
Q(J)=INX(J+1)
P(J)=INX(J+1)
P(J)=INX(J+1)
P(J)=INX(J+1)
P(J)=INX(J+1)
P(J)=INX(J+1)
Q(J)=P(J)
G(2)=INX(J+2)
G(3)=INX(J+2)
G(4)=INX(J+2)
G(5)=P(5)
G(6)=INX(J)
G(7)=P(7)
N=0
NN=0
GO TO 422 K=1
IF(KEQQ, EQ.0) AND (Q(K+1), EQ.1) NN=NN+1
422 IF(KP(K), EQ.0) AND (P(K+1), EQ.1) N=N+1
IF(KP(K), EQ.0) AND (P(K+1), EQ.1) NN=NN+1
IF(KP(K), EQ.0) AND (P(K+1), EQ.1) N=N+1
IF(KM, LE.0) GO TO 417
IF(KM, LE.0) GO TO 421
GO TO 421
427 IF(KM, EQ.0) GO TO 417
GO TO 441
429 IF(KP(1)+P(2)+P(3), EQ.0) OR (MNP(J), NE.1) GO TO 442
GO TO 421
442 IF(KP(1)+P(2)+P(3), EQ.0) OR (NN, NE.1) GO TO 445
GO TO 421
445 CONTINUE
MNP(J)=N
INX(J)=0
KOUNT=KOUNT+1
GO TO 421
441 BABOR=0.
GO TO 447 K=1, 2
447 BABOR=BABOR+10. +PK
IF(BABOR, EQ. 101100.0) GO TO 425
IF(BABOR, EQ. 10100010.0) GO TO 425
IF(BABOR, EQ. 1010110.0) GO TO 425
IF(BABOR, EQ. 10100010.0) GO TO 425
IF(BABOR, EQ. 101100.0) GO TO 425
IF(BABOR, EQ. 10100010.0) GO TO 425
IF(BABOR, EQ. 1010110.0) GO TO 425
IF(BABOR, EQ. 101100.0) GO TO 425
IF(BABOR, EQ. 10100010.0) GO TO 425
IF(BABOR, EQ. 101100.0) GO TO 425
IF(BABOR, EQ. 10100010.0) GO TO 425
GO TO 421
425 INX(J)=0
KOUNT=KOUNT+1
421 CONTINUE
INX(J)=0
CALL FSEEK(4, 1)
WRITE(4, 1)(X2, L2, L2=1, NJ)
CONTINUE
IF(COUNT.GT.0) GO TO 429
DO 440 J=1, NJ
CALL FSEEK(4, 1)
WRITE(4, 1)(X2, L2, L2=1, NJ)
DO 450 J=1, NJ
CLOSE 4
CALL FREADVIEW
110 CONTINUE
C********* FIND BOUNDARY FOR SECOND VIEW
MA=0
DO 531 I=1, 6
IF(X2(I).GT.MAX(X2(I))) MA=MA+X2(I)
531 CONTINUE
Y2=MA
TYPE "VALUE OF Y2 = ".Y2
MIN=NN
DO 532 I=7, 12
IF(Y2(I).LT.MIN) MIN=Y2(I)
532 CONTINUE
V2=MIN
TYPE "VALUE OF V2 = ".V2
MIN=NN
DO 535 I=1, 6
IF(U2(I).LT.MIN) MIN=U2(I)
IF(U2(I)+6).LT.MIN) MIN=U2(I)+6
535 CONTINUE
U2=MIN
TYPE "VALUE OF U2 = ".U2
MAX=0
DO 537 I=4, 6
IF(U2(I).GT.MAX) MAX=U2(I)
IF(U2(I)+6).GT.MAX) MAX=U2(I)+6
537 CONTINUE
U2=MAX
TYPE "VALUE OF U2 = ".U2
C********* DETERMINE CORNER POINTS IN SECOND VIEW.
OPEN 4, NAME=LEN=2*MMLEN, REC=MMREC
122=0
123=0
124=0
125=0
IF(S1=U2) GO TO 511
IMIN=I1-I
CALL FSEEK(4, IMIN)
READ(4) (X1, L1, L1=1, MMLEN)
READ(4) (X1, L1, L1=1, MMLEN)
READ(4) (X1, L1, L1=1, MMLEN)
IF(J1=U2) GO TO 511
IF(J1.LE.0) GO TO 511
IF(J1.GE.6) GO TO 511
GO TO 511
S1=S1+1
IF(S1.GT.30) GO TO 511
ALL=11
C11=C11+1000
S1=S1+1
GO TO 511
511 I2=I2+1
IF(I2.GT.30) GO TO 511
ALL=11
C11=C11+1000
S1=S1+1
GO TO 511
SEARCH: (INHOM<LL>,LZZ=U,11)
732 CONTINUE
730 CONTINUE
CLOSE 1
TYPE"VALUE OF LL = " LL
OPEN 2, NAME=2, LEN=2, MLEN, REC=MMREC
CALL FAST<2, 1>
DO 734 J=1,LL, LDCO
LL=LL+1
CALL PEOAP<2, LL, IEF>
IF: (IER .GE. 0) GO TO 9000
DO 736 J=1, LLDCO
II=II+1
IF: (II .GE. LDCO) GO TO 8000
PEOA<2, (INHOM<LL>,LZZ=JJ, 11)
736 CONTINUE
734 CONTINUE
CLOSE 2
TYPE"VALUE OF LL = " LL
NL=6
C+++++++ FIND MATCHED POINT PAIR FOR CORNERS FORMED BY 5 EDGES.
IF: (JH=115, EH=0, OR < 115, EH=0) GO TO 541
DO 543 INNOM=1, 115
L11=INNOM-1100
L12=INNOM+1100
ML=INNOM-L11+1000
CALL NACOT<1, L11, L12, COMAK>
COMAK=COMAK+1, 91
IF: (NL=JH) GO TO 546
DO 545 INNOM=1, 115
L12=INNOM+1100
L11=INNOM-1100
ML=INNOM-L11+1000
CALL CORRE<1, L11, L12, ML, CORGM>
L11=L11
ML=ML
CORL=INNOM-CORGM
IF: (CORL < CORGM) GO TO 547
545 CONTINUE
COMAK=COMAK+1, 93
COMAK=0
DO 549 JH=1, 115
IF: (COMAK<COMAJ) GO TO 549
NINOM=1H
COMAK=COMAK+1
549 CONTINUE
L12=655<INNOM-1000
L11=655<INNOM-12+1000
CALL MCL<1, L11, L12, ML, L11, L12, CORGM>
IF: (CORGM<COMAK) GO TO 547
GO TO 542
548 CONTINUE
COMAK=COMAK+1, 92
L11=INNOM+1100
ML=INNOM-1100
CALL MCL<1, L11, L12, ML, L11, L12, CORGM>
IF: (CORGM<COMAK) GO TO 547
GO TO 542
547 CALL COCOR<1, L11, L12, ML, RES, LLK>
NL=NL=1
LLK=LLK+1
IF: (NL=NLK) GO TO 547
LL1=L11
LL2=L11-111
DIF1=INLAK-LL1-111
DIF2=INLAK-ML-MLK
IF: (NL=GE. 4) GO TO 600
542 CONTINUE
541 CONTINUE
C+++++++ FIND MATCHED POINT PAIR FOR CORNERS FORMED BY 4 EDGES.
IF: (JH=114, EH=0, OR < 114, EH=0) GO TO 561
DO 562 INNOM=1, 114
LII=I14.INNOM.L1000
MLI=I14.INNOM.-L1111000
CALL MACOTO.LII.MLI.COMAX
COMAX=COMAX+L.02
IF(NLAK.GE.1) GO TO 583
DO 555 INNOM=1.L124
LII=I24.INNOM.L1000
MLI=I24.INNOM.-L1111000
CALL CORREL.LII.MLI.LII.MLI.CORGM
LII=LII
MLI=MLI
CORLM.INNOM.CORGM
IF(CORLM.INNOM.GE.COMAX) GO TO 587
555 CONTINUE
COMAX=COMAX+L.02
COMAX=0
DO 559 JH=1.L124
IF(COMAX.GT.CORLM.INNOM) GO TO 569
INNOM=JH
COMAX=COMAX
569 CONTINUE
LII=I24.INNOM.L1000
MLI=I24.INNOM.-L1111000
CALL HILL.LII.MLI.LII.MLI.LII.MLI.CORGM
IF(CORGM.GE.COMAX) GO TO 587
GO TO 563
563 CONTINUE
COMAX=COMAX+L.02
LII=LII+IINDIF.I10
MLI=MLI+IINDIF.I10
CALL HILL.LII.MLI.LII.MLI.LII.MLI.LII.CORGM
IF(CORGM.GE.COMAX) GO TO 587
GO TO 563
587 CALL COCORR.LII.MLI.LII.MLI.RES.LLKK
NLAK=NLAK+1
INNOM=NLAK=LL1
IYZY=NLAK=MLI
IINDIF=NLAK=LL1+LL1
IINDIF=NLAK=MS1+MS1
LLKM=NLAK=NLAK+1
IF(NLAK.GE.1) GO TO 566
566 CONTINUE
561 CONTINUE
INI=0
IINDIF=0
CALL FIND MATCHED POINT PAIR FOR CORNERS FORMED BY 3 EDGES.

IF((LL1.EQ.0).OR.(LL2.EQ.0)) GO TO 581
DO 555 INNOM=1.L111
LII=I24.INNOM.L1000
MLI=I24.INNOM.-L1111000
CALL MACOTO.LII.MLI.COMAX
COMAX=COMAX+L.02
IF(NLAK.GE.1) GO TO 584
DO 555 INNOM=1.L122
LII=I24.INNOM.L1000
MLI=I24.INNOM.-L1111000
CALL CORREL.LII.MLI.LII.MLI.MLI.CORGM
LII=LII
MLI=MLI
CORLM.INNOM.CORGM
IF(CORLM.INNOM.GE.COMAX) GO TO 587
585 CONTINUE
COMAX=COMAX+L.02
COMAX=0
DO 559 JH=1.L122
IF(COMAX.GT.COMAX.JH) GO TO 569
INNOM=JH
COMAX=COMAX.JH
CONTINUE
L12=622632.<NMM=1.1000
M12=622632.<NMM=-L12+1000
IF(CORRM.GE.COMAR) GO TO 587
GO TO 582.
CONTINUE
COMAX=CORRM1.01
L12=L11+111.IDIF1.
M12=M11+YIDIF1.
CALL 6176.(L11,M11,L12,M12,L11,M11,CORRM:)
IF(CORRM.GE.COMAR) GO TO 587
GO TO 582.
CONV=NMM11NMM11+1
IND1=IND11(L11-L11)
YDIF=YDIF+M11-M11
CALL 6176.(L11,M11,L12,M12,RES.LLKN)
L11=NMM11NMM11+1
YVS=MKLAK=M11
IND1=IND11(L11-L11)
YDIF=YDIF1(M11-M11)
L11=NMM11
IF(NMM11.GE.4) GO TO 665
CONTINUE
CONTINUE
C1*****FIND AVERAGE DIFFERENCE OF I AND J COORDINATES
C1*****FROM MATCHED POINT PAIRS.
IVND1=IND11NMM11
IVDIF=YDIF1NMM11
WRITE(12,582) (IVDIF1)
WRITE(12,583) (IVND1)
FORMAT(0,LE: AVERAGE DIFFERENCE OF N = '-15)
FORMAT(0,LE: AVERAGE DIFFERENCE OF Y = '-15)
GO TO 685
665 CONTINUE
C1*****FIND AVERAGE DIFFERENCE OF I AND J COORDINATES
C1*****FROM 4 MATCHED POINT PAIRS.
DO 625 JLL=1,4
NNOPTX(JLL)=0
625 CONTINUE
IF(1<IND11(L11).EQ.11IDIF(JLL)) NNOPTX(JLL)=NNOPTX(JLL)+1
DO 627 JLL=1,4
IF(1<IND11(L11).EQ.11IDIF(JLL)) NNOPTY(JLL)=NNOPTY(JLL)+1
NMARK=0
DO 627 JLL=1,4
IF(NMARK.GE.NNOPTX(JLL)) GO TO 629
NMARK=JLL
NMARK=NNOPTX(JLL)
629 CONTINUE
IF(NMARK.GE.NNOPTY(JLL)) GO TO 632
NMARK=JLL
NMARK=NNOPTY(JLL)
632 CONTINUE
IVDIF=IVDIF1NMM11
IVND1=IVND11NMM11
WRITE(12,582) (IVDIF)
WRITE(12,583) (IVND1)
C1*****FIND MATCHED POINT PAIR FOR CORNER POINTS.
GO TO 635,635,635,635,635,LLKN
635 CONTINUE
DO 651 INNM=1,115
L11=BS5(INNM11000
M11=BS5(INNM+L111000
DO 626 JLL=1,4
IF(1<IND11(L11).EQ.11IDIF(JLL)) AND (1<IND11(L11).EQ.11IDIF(JLL)) GO TO 651
626 CONTINUE
CALL 6506.(L11,M11,COMAR)
L12=L11+YDIF1
M12=M11+IVYDI.
CALL HILL(L11, M11, L12, M12, L1L, M1L, CORGM).
IF(CORGM .GE. COMAX) GO TO 667
GO TO 651
657 CALL COCOR(L11, M11, L1L, M1L, RES, LLKK)
651 CONTINUE
655 CONTINUE
GO 651 INNOM=L11
L11=L11+I4
M11=M11+IVYDI.
CALL HILL(L1L, M1L, L12, M12, L1L, M1L, CORGM).
IF(CORGM .GE. COMAX) GO TO 667
GO TO 651
657 CALL COCOR(L11, M11, L1L, M1L, RES, LLKK)
671 CONTINUE
675 CONTINUE
GO 691 INNOM=L11
L11=L11+I4
M11=M11+IVYDI.
CALL HILL(L11, M11, L12, M12, L1L, M1L, CORGM).
IF(CORGM .GE. COMAX) GO TO 667
GO TO 691
691 CONTINUE
695 CONTINUE
C*****FIND MATCHED POINT PAIR FOR ALL PIXELS OF EDGED
C*****VERSION OF PICTURE.
DO 665 INNOM=1, L11
L11=L11+I4
M11=M11+IVYDI.
CALL HILL(L11, M11, L12, M12, L1L, M1L, CORGM).
IF(CORGM .GE. COMAX) GO TO 667
GO TO 665
665 CONTINUE
C*****FIND MATCHED POINT PAIR FOR ALL PIXELS OF OBJECT.
DO 777 L1L=1, L1M
DO 777 M1L=J1, J1M
L1L=I1
M1L=J1
L12=L11+IVYDI.
M12=M11+IVYDI.
CALL HAKOTO(L11, M11, COMAX).
CALL HILL(L11, M11, L12, M12, L1L, M1L, CORGM).
IF(CORGM .GE. COMAX) GO TO 725
WRITE(12, 218) (L1L, M1L).
GO TO 777
725 CALL COCOR(L11, M11, L1L, M1L, RES, LLKK)
777 CONTINUE
218 FORMAT('0', 20X, 15X, 2X), 'ERROR IN FINDING MATCH. CO-ORDINATES CANNOT BE FOUND OUT Mildly.
GO TO 7000
4000 TYPE 'ERROR IN ERG6 , ERROR # = ', IER.
GO TO 7000
SUBROUTINE TRANSF(IIND1, IIND2, IJIND, XY, Z, NA)
C******
C******* A SUBROUTINE TO CALCULATE CAMERA TRANSFER MATRIX
C******* FOR TWO VIEWS
C*******
D DIMENSION IIND1(12), IIND2(12), JIND1(12), JIND2(12),
C Y(12), X(12), Z(12),
D DIMENSION X(12), Y(12), X(12), Z(12), A(24,11), ATR(24,12),
D CAPR(11,11), B(24,24), IR(11), IC(11), DC(11,11), C(11,2),
C T(4,4), T(4,3)
D INTEGER CCOLM, COLM
D COMMON X(11,12), T(4,3)
D COMMON A(24,11)
C COMMON B(24,24)
D DOUBLE PRECISION DAPR(11,11), DET
N=2
DO 11 I=1,N
IND1=IIND1(I)
IND2=IIND2(I)
XC1(I)=A(IND1+1,IND1-NA)
XC2(I)=A(IND2+1,IND2-NA)
JC1(I)=IND1-1
JC2(I)=IND2-1
11 CONTINUE
C******* CALCULATE CAMERA TRANSFER MATRIX FOR FIRST VIEW
C*******
C*******
DO 22 I=1,N
A(I,1)=X(I)
A(I,2)=Y(I)
A(I,3)=Z(I)
A(I,4)=1.0
A(I,5)=0.0
A(I,6)=0.0
A(I,7)=0.0
A(I,8)=0.0
A(I,9)=(XC1(I)^2+Y(I)^2+Z(I)^2)^(-1/2)
A(I,10)=X(I)*X1(I)+Y(I)*Y1(I)+Z(I)*Z1(I)
A(I,11)=Y(I)*X1(I)+Z(I)*Y1(I)
B(I,1)=XC1(I)
22 CONTINUE
NL=NL+1
DO 11 I=NL,N2
IMM=IM+1
A(1,1)=0.0
A(1,2)=0.0
A(1,3)=0.0
A(1,4)=0.0
A(1,5)=X(1)
A(1,6)=Y(1)
A(1,7)=Z(1)
A(1,8)=0.0
A(1,9)=X(1)*X(1)+(-1.0)
A(1,10)=X(1)*Y(1)+(-1.0)
A(1,11)=X(1)*Z(1)+(-1.0)
B(1,1)=X(1)

31 CONTINUE

C*******CALCULATE TRANSFER MATRIX T1 FOR FIRST VIEW
C*******FROM MATRIX A AND B.
DO 1 I=1,N2
   DO 1 J=1,4
      1 ATR(J,1)=A(K,1)
      NCOLM=11
      NCOLM=24
      NROWM=2
      DCOLM=1
      CCOLM=2
      CALL FRMA(A, ATR, NCOLM, NCOLM, NROWM, DCOLM, CCOLM, CCOLM)
      CALL FRMB(A, ATR, NCOLM, NCOLM, NROWM, DCOLM, CCOLM, CCOLM)
      CALL DOUBLE(A, ATR, NCOLM, NCOLM, NCOLM, NCOLM)
      CALL MINRD(A, ATR, NCOLM, NCOLM, DET, TERR, ERR, IC)
      CALL SINGLE(A, ATR, NCOLM, NCOLM, NCOLM, NCOLM)
      CALL FRMA(A, ATR, NCOLM, NCOLM, DCOLM, CCOLM)
      T1(1,1)=C(1,1)
      T1(1,2)=C(2,1)
      T1(2,1)=C(1,2)
      T1(2,2)=C(2,2)
      T1(3,1)=C(4,1)
      T1(3,2)=C(5,1)
      T1(4,1)=C(6,1)
      T1(4,2)=C(7,1)
      T1(5,1)=C(9,1)
      T1(5,2)=C(10,1)
      T1(6,1)=C(11,1)
      T1(6,2)=C(12,1)
      WRITE(12,38)
      38 FORMAT('0.28E', 'TRANSFER MATRIX FOR VIEW #1 I.E. LEFT HAND', C 'SIDE VIEW', )
      WRITE(12,38)
      DO 27 I=1,4
         J=I
         WRITE(12,39) (T1(I,J), JJ=1,3)
         39 FORMAT('0.', '28.3E', 'F15.4, 10E')
      C*******CALCULATE TRANSFER MATRIX T2 FOR SECOND VIEW.
      C*******CALCULATE MATRIX A AND B.
      DO 41 I=1,N2
         A(1,1)=X(I)
         A(1,2)=Y(I)
         A(1,3)=Z(I)
         A(1,4)=1.0
         A(1,5)=0.0
         A(1,6)=0.0
         A(1,7)=0.0
         A(1,8)=0.0
         A(1,9)=X(I)*X(I)+(-1.0)
         A(1,10)=X(I)*Y(I)+(-1.0)
         A(1,11)=X(I)*Z(I)+(-1.0)
      B(1,1)=X(I)
      41 CONTINUE
      NI=NI+1
      DO 42 I=NI,N2
         IMN=1
         WRITE(12,38)
      42 CONTINUE
A(I,1)=0.0
A(I,2)=0.0
A(I,3)=0.0
A(I,4)=0.0
A(I,5)=<INN>
A(I,6)=<INN>
A(I,7)=<INN>
A(I,8)=1.0
A(I,9)=(Y2C2<INN>-<INN>)=(-1.0)
A(I,10)=(Y2C2<INN>-<INN>)=(-1.0)
A(I,11)=(Y2C2<INN>-<INN>)=(-1.0)
B(I,1)=Y2C2<INN>

42 CONTINUE
C**CALCULATE TRANSFER MATRIX FOR SECOND VIEW
C**FROM MATRIX A AND B.
DO 44 JJ=1,N2
DO 44 II=1,N1
44 ATR(JJ,II)=A(II,II)
CALL PRAP(A(II,II),NAPR,NCOLM,NCOLM,NCOLM,NCCOL,NCOLM,NCCOL)
CALL PRAP(A(II,II),B(II,II),NAPR,NCOLM,NCOLM,NCOLM,NCCOL,NCOLM,NCOLM,NCOLM)
DO 55 II=1,11
DO 55 JJ=1,11
55 DAPR(JJ,II)=DAPR(I,II)
CALL MIXARD(DAPR(NAPR,NCOLM,NCOLM,DET,IER,IR,IC)
DO 66 II=1,11
DO 66 JJ=1,11
66 DAPR(JJ,II)=DAPR(I,II)
CALL PRAP(DAPR(DAPR(I,II),NAPR,NCOLM,NCOLM,NCOLM,NCCOL,NCOLM,NCOLM,NCOLM)
T2(L,1)=C(Y1,L)
T2(L,2)=C(Y2,L)
T2(L,3)=C(Y3,L)
T2(L,4)=C(Y4,L)
T2(L,5)=C(Y5,L)
T2(L,6)=C(Y6,L)
T2(L,7)=C(Y7,L)
T2(L,8)=C(Y8,L)
T2(L,9)=C(Y9,L)
T2(L,10)=C(Y10,L)
T2(L,11)=<L1,L>
T2(L,12)=1.0
WRITE(12,40)
40 FORMAT('TRANSFER MATRIX FOR VIEW #2 I.E. RIGHT HAND',
' SIDE VIEW',/26.59(1H4),/)
DO 45 II=1,4
45 WRITE(12,30) (T2(I,I),I=1,4)
RETURN
END

SUBROUTINE CCGCR(INCL,INCY,INC2,INCY2,RES,LLK)
C**SUBROUTINE TO CALCULATE 2-D INFORMATION FROM MATCHED POINT PAIR. 2-D CO-ORDINATES ARE FOUND BY MEAN SQUARE FIT.
DIMENSION T1(4,2), T2(4,2), CD(4,2), COEFF(2,4), COEFF(2,3),
CGOF(I2,2), DJFR(2,2), COS(2,2), RES(2,2)
DOUBLE PRECISION DAPR(2,2),DET
DIMENSION IR(2,2), IC(2,2)
DIMENSION INCL(256), dripping NAME(15), NAME(5)
COMMON /CC/ T1, T2
COMMON /DC/ול NAME(15), NAME(5)
COMMON /PP/ ALL, NAME, NAME, NAME
COMMON /CC/ ALL, ALL, ALL

C**DIMENSION T1(4,2), T2(4,2), CD(4,2), COEFF(2,4), COEFF(2,3),
CGOF(I2,2), DJFR(2,2), COS(2,2), RES(2,2)
IXVC1=I*ICL-NACCD
JXVC1=IXVC1-NACCD
IXGC2=INC2-NACCD
JXVC2=IXGC2-NACCD
BNC1=JXVC1
BVC1=I*ICL
BVC2=JXVC2
BVC3=I*ICL
AXC1=BNC1+CCBCD
AXC2=BVC1+CCBCD
AXC3=BVC2+CCBCD

C*******FIND ELEMENTS OF MATRIX C.
COEFF(J, J)=T(I,J)*T(I,J)+T(I,J)*T(J,I)

C*******FIND ELEMENTS OF MATRIX D.
DD(J,J)=T(I,J)*T(I+4,J)+AXC1*T(I+4,J)

C*******CALCULATE X; Y; Z CO-ORDINATES.
CALL FPRX,COEF,C,COEF,C,COEF,D,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL
CALL FPRX,COEF,C,COEF,C,COEF,D,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL
CALL FPRX,COEF,C,COEF,C,COEF,D,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL
DO 8 JJ=1,3
DO 8 II=1,3
8 DAPR(JJ,JJ)=DAPR(JJ,JJ)
CALL MINPRO,MINPRO,NCOL,NCOL,DETEIER,IER,IER,IER
DO 9 II=1,2
DO 9 JJ=1,2
9 COFIN(JJ,JJ)=COFIN(JJ,JJ)
CALL PRFX,COFIN,D,DDP,RES,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL,NCOL

C*******FIND INTENSITY BY TAKING AVERAGE INTENSITY OF MATCHED POINT PAIR.
OPEN 1,NAM=LEN=2,MMLEN,REC=MMREC
CALL FSEEK(1,IXCL)
READ(1),IXCL(LSL),LLSL,MMLEN
CLOSE 1
J=IXCL(JVC1)
OPEN 2,NAM=LEN=2,MMLEN,REC=MMREC
CALL FSEEK(2,IXCL)
READ(2),IXCL(LSL),LLSL,MMLEN
CLOSE 2
J=IXCL(JVC1)
J=IXCL(JVC1)
WRITE(12,99)(1,IXCL,JVC1,IXCL,JVC2,RES(J,J),RES(J,J),CRES(J,J),LLSL)
99 FORMAT(*3,15x,4('15e5.2x'),10x,5('F10.4,5x'),10x,5x)
RETURN
END
SUBROUTINE CORREL(I, J, IV, JY, CORR)

**** A SUBROUTINE TO CALCULATE CORRELATION FOR GIVEN POINT PAIR. ****

DIMENSION NAME(5), NAME2(5)
DIMENSION IVINC(100)
DIMENSION INX(256, 4)

EQUIVALENCE (IVINC, INX)

COMMON +SL, IV, LLMN, NAME, NAME2, MNLEN, MNREC
COMMON +NN, LLP

**** DETERMINE PROPER SIZE OF WINDOW. ****

IN=IX-LLMN
IJ=JX-LLMN
IIY=JY-LLMN
IJY=JY-LLMN
IF(IX .LT. I) GO TO 2
IL=LLMN
GO TO 3
2 IL=IN-1
3 CONTINUE
IF(IJX, LT, 1) GO TO 4
JL=LLMN
GO TO 5
4 JL=IN-1
5 IF(IJY, LT, 1) GO TO 6
J2=LLMN
GO TO 7
6 J2=IV-1
7 IF(IJY, LT, 1) GO TO 8
JL=LLMN
GO TO 9
8 J2=IV-1
9 CONTINUE
IF(I2 .LE. I2) IL=I2
IF(I2 .LE. I2) JL=I2
IF(I2 .LE. I2) IL=I2
IF(I2 .LE. I2) JL=I2
ILL=1
ILL2=1
I2L=IX+JL
I2JL=X+JL
I2YL=JY+IL
I2JYL=Y+IL
IF(I2LL .GT. NI) GO TO 12
IL=LLMN
GO TO 13
12 I2L=NI-1
13 IF(I2JL, GT, NJ) GO TO 14
JL=LLMN
GO TO 15
14 J2L=NI-JK
15 IF(I2YL, GT, NJ) GO TO 16
I2L=LLMN
GO TO 17
16 I2L=NI-JY
17 IF(I2YL, GT, NJ) GO TO 18
JL=LLMN
GO TO 19
18 J2L=NI-JY
19 IF(IlL .LE. I2L) IL=JL
IF(I2L .LE. I2L) JL=JL
IF(I2L .LE. I2L) IL=JL
IF(I2L .LE. I2L) JL=JL
ILL=1
ILL2=1
I2L=IL
I2L2=JL
IF(I2LL .LE. I2L) ILL=ILL
IF(I2LL .LE. I2L) ILL=ILL

END
IF(ILL2.LT.ILL1) ILL1=ILL2
IF(ILL1.LT.ILL2) ILL1=ILL2
I2LM=I2LM+2
J2LM=J2LM+2
IF(I2LM.EQ.I2NM+1) I2LM=1
IF(J2LM.EQ.J2NM+1) J2LM=1
C******CALCULATE AVERAGE VALUE OF PIXELS IN TWO WINDOWS.
SUMX=0.0
SUMY=0.0
LDDO=1024/MJ
MLK=NI/LDDO
DO 27 I=1,I2LM
INX=IN2+I-1
IVC=IV2+I-1
IF(1.GE.I) GO TO 45
40 LNNN=IYVC-((IYVC-1)/LDDO)*LDDO
LNN=IYVC-LDDO
LNN=LNN+MLKK
LNNN=LNNN-1
45 IF(LNNN.GT.LDDO) GO TO 40
C*******ACCESS PROPER ELEMENT FROM EXTENDED MEMORY.
DO 27 J=1,J2LM
CALL REMAP(0.LNN,IER)
IF(IER.GE.2) GO TO 2000
J2X=JN+J-1
J2Y=JY+J-1
AP2=I2N(JYVC-LNNN)
AP1=I2N(I2X+J-1)*IL2J2+J2X
SUMX=SUMX+AP1
SUMY=SUMY+AP2
27 CONTINUE
AI2LM=I2LM
AJ2LM=J2LM
AVRGX=SUMX/(AI2LM+AJ2LM)
AVRGY=SUMY/(AI2LM+AJ2LM)
C******CALCULATE CORRELATION.
PROXY=0.0
PRODXY=0.0
PROD2Y=0.0
DO 29 I=1,I2LM
INX=IN2+I-1
IVC=IV2+I-1
IF(1.GE.I) GO TO 55
50 LNNN=IVYC-((IVYC-1)/LDDO)*LDDO
LNN=IVYC-LDDO
LNN=LNN+MLKK
LNNN=LNNN-1
55 IF(LNNN.GT.LDDO) GO TO 50
C*******ACCESS PROPER ELEMENT FROM EXTENDED MEMORY.
DO 29 J=1,J2LM
CALL REMAP(0.LNN,IER)
IF(IER.GE.2) GO TO 2000
J2X=JN+J-1
J2Y=JY+J-1
AP2=I2N(JYVC-LNNN)
AP1=I2N(I2X+J-1)*IL2J2+J2X
PROXY=PROXY+((AP1+APR2)**2)
PRODXY=PRODXY+((AP2+AP2)**2)
PRODXY=PRODXY+((AP2+AP2)**2)
29 CONTINUE
DEN2=PROD2Y*PROD2Y
IF(DEN2.EQ.0.0) GO TO 31
DEN=SQRT(DEN2)
CORR=PRODXY/DEN
GO TO 35
SUBROUTINE OTOCOR(IKL, JKL, IKL2, JKL2, AUTCOR)

C*----------------------------------------------------------------*
C* SUBROUTINE TO CALCULATE AUTOCORRELATION FOR A               *
C* GIVEN POINT PAIR.                                         *
C*----------------------------------------------------------------*

C*C----------------------------------------------------------------*
C* DIMENSION NAMEL(5), NAME2(5)                               *
C* DIMENSION IWIN(1024)                                       *
C* COMMON /SL, IWIN                                          *
C* COMMON /DD, NJ, NJLN, NAMEL, NAME2, NMLEN, NMREC           *
C* COMMON /PAR, LLNN                                         *
C*----------------------------------------------------------------*
C*C---------Determine proper size of window.                   *
C* IF(JKL*JKL LT 1) GO TO 2                                    *
C* I1=LLNN                                                   *
C* GO TO 2                                                  *
C* 2 I1=I1-1                                                *
C* 3 CONTINUE                                               *
C* IF(JKL*JKL LT 1) GO TO 4                                  *
C* I1=LLNN                                                   *
C* GO TO 5                                                  *
C* 4 I1=I1-1                                                *
C* 5 IF(JKL2*JKL2 LT 1) GO TO 6                              *
C* I2=LLNN                                                   *
C* GO TO 7                                                  *
C* 6 I2=I2-1                                                *
C* 7 IF(JKL2*JKL2 LT 1) GO TO 8                              *
C* J2=LLNN                                                   *
C* GO TO 9                                                  *
C* 8 J2=J2-1                                                *
C* 9 CONTINUE                                               *
C* IF(IKL*IKL LE 12) IL=IL                                  *
C* IF(IKL*IKL LT 12) IL=IL                                  *
C* IF(JKL*JKL LE 12) JL=JL                                  *
C* IF(JKL*JKL LT 12) JL=JL                                  *
C* ILL=IL                                                   *
C* ILL2=JL                                                  *
C* IIXL=IKL+LLNN                                            *
C* IJXL=IKL+LLNN                                            *
C* IIXL2=IKL2+LLNN                                          *
C* IJXL2=IKL2+LLNN                                          *
C* IF(IJK*IJKL GT NJ) GO TO 12                              *
C* I1L=LLNN                                                 *
C* GO TO 12                                                 *
C* 12 I1L=N1-IKL                                            *
C* 12 IF(IJK*IJKL GT NJ) GO TO 14                           *
C* J1L=LLNN                                                 *
C* GO TO 15                                                 *
C* 14 J1L=NJ-IKL                                            *
C* 15 IF(IJK*IJKL GT NJ) GO TO 16                           *
C* I2L=LLNN                                                 *
C* GO TO 17                                                 *
C* 16 J2L=N1-IKL                                            *
C* 17 IF(IJK*IJKL GT NJ) GO TO 18                           *
C* J2L=LLNN                                                 *
C* GO TO 19                                                 *
C* 18 J2L=NJ-IKL                                            *

GO TO 30
19 IF(ILL LE ILL) IL2=ILL
IF(ILL LT ILL) IL2=I2L
IF(JIL LE JIL) JL2=JIL
IF(J2L LT JIL) JL2=J2L
ILL2=I2L
ILL2=J2L
IF(ILL LE ILL) ILL=ILL1
IF(ILL2 LT ILL) ILL=ILL2
IF(ILL2 LE ILL2) ILL=ILL2
ILL2=ILL2+1
J2L=J2L+1
IF(ILL IM LE ILL) ILL=ILL1
IF(ILL IM LT ILL) ILL=ILL2
A2LM=I2LM
AJ2LM=J2LM
ILL1=ILL1+1
ILL2=ILL2+1
C*******CALCULATE AVERAGE VALUE OF PIXELS IN TWO WINDOWS.
SUMX=0.0
SUMY=0.0
DO 27 I=1, I2LM
IXC=I+1-I-ILL1
IVYC=I+2-I-ILL1
DO 27 J=1, J2LM
JXC=J+1-J-ILL2
JYVC=J+2-J-ILL2
C*******ACCESS PROPER ELEMENT FROM EXTENDED MEMORY.
AP1=IVF(I+I-1)*ILL+J*X2C
AP2=IVF(I+1)*JYVC
SUMX=SUMX+AP1
SUMY=SUMY+AP2
27 CONTINUE
A1LM=I2LM
AJ2LM=J2LM
AVRGX=SUMX/(AI2LM+AJ2LM)
AVRGY=SUMY/(AI2LM+AJ2LM)
C*******CALCULATE AUTOCORRELATION.
PROXY=0.0
PRODY=0.0
DO 29 I=1, I2LM
IXC=I+1-I-ILL1
IVYC=I+2-I-ILL1
DO 29 J=1, J2LM
JXC=J+1-J-ILL2
JYVC=J+2-J-ILL2
C*******ACCESS PROPER ELEMENT FROM EXTENDED MEMORY.
AP1=IVF(I+1+1)*ILL2+J*X2C
AP2=IVF(I+1)*JYVC
PRX=PROXY+<AP1*AVRGY+<AP2*AVRGY**2>
PRODXY=PRODXY+<AP1*AVRGY+<AP2*AVRGY**2>
29 CONTINUE
DEN2=PRODXY*PRODXY
IF(DEN2 LE 1.0) GO TO 21
DEN2=SQRT(DEN2)
AUTCOR=PROXY/DEN2
RETURN
31 AUTCOR=0.0
RETURN
END
SUBROUTINE MAKOTO(IN, JN, CORMAX)

C********
C*******A SUBROUTINE TO CALCULATE MAXIMUM VALUE OF AUTOCORRELATION
C*******IN THE REGION AROUND A GIVEN PIXEL.
C*******
   IMAX=IN-2
   JMAX=JN-2
   AMAX=0.0
   DO 20 I=1,J
      DO 20 J=1,L
         IF([[I, EQ. 2]. AND. [J, EQ. 2]]) GO TO 20
         IMIC=IMAX-I
         JMIC=JMAX-J
         CALL SUBROUTINE OTOCOR TO FIND AUTOCORRELATION.
         CALL OTOCOR(IN, JN, IMIC, JMIC, AUTCOR)
         IF(AUTCOR .LT. AMAX) GO TO 20
         AMAX=AUTCOR
   20 CONTINUE
   C*******MULTIPLY MAXIMUM VALUE OF AUTOCORRELATION BY A FACTOR TO FIND
   C*******VALUE TO APPLY CHECK TO MAXIMUM VALUE OF CORRELATION.
   CORMAX=AMAX*.92
   RETURN
END

SUBROUTINE HILL(IN, JN, IVL, JVL, IV2, JV2, CCOMP)

C*******
C*******A SUBROUTINE TO FIND POINT OF MAXIMUM CORRELATION
C*******WITH THE HELP OF HILL CLIMBING METHOD.
C*******
   DIMENSION IXCOR(9), IYCOR(9), CORR(9)
   DIMENSION CORRAP(9), CORRINC(9), CORMAX(9), CCOMP(9)
   CCOMP(9), JCOMP(9)
   IIND=IN
   JIND=JN
   1 IVL=IIND-2
   JVL=JIND-2
   KK=0
   JL=0
   C*******FIND CORRELATION FOR PIXELS AROUND A PIXEL.
   DO 5 I=1,I
      DO 5 J=1,J
         KK=KK+1
         IYC=IVL+I
         JYC=JVL+J
         IF(IYC .EQ. 0) GO TO 2
         IF(JYC .EQ. 0) GO TO 2
         IF(IYC .EQ. IVL+I) GO TO 125
         IF(JYC .EQ. JVL+J) GO TO 125
      5 CONTINUE
   125 CORR=CORRAP(JL)
      GO TO 7
   7 CALL CORREL(IN, JN, IYC, JYC, CORR)
   8 CORR=KK=CORR
      I=ICORP(KK)=IYC
      IYICP(KK)=JYC
   9 CONTINUE
   C*******STORE VALUE OF CORRELATION.
   CORR(COMP(KK)=CORR(KK))
   2 CONTINUE
   CORMAX=CORR(5)
   C*******SEARCH FOR A PIXEL WITH MAXIMUM CORRELATION.
   DO 9 I=1,9
      IF(I .EQ. 5) GO TO 2
      IF(I .EQ. 9) GO TO 2
   9 IF(CORR(I) .LT. CORR(S)) GO TO 2
   CORMAX=CORR(I)
2 CONTINUE
   IF(CORMX .GT. CORPAR(5)) GO TO 13
   NNO=0
   I1 = I+1
   IF(CORMX(I1) .LT. CORMAX) GO TO 11
   INNO=INNO+1
   NNO(INNO)=I1
11 CONTINUE
C********* CHECK WHETHER PIXEL OF MAXIMUM CORRELATION IS FOUND.
C********* MORE THAN ONE PIXEL HAS MAXIMUM VALUE OF CORRELATION.
15 INNO=0
   DO 17 I=1,9
   IF(CORMX(I) .LT. CORMAX) GO TO 17
   INNO=INNO+1
   NNO(INNO)=I
17 CONTINUE
   IF(INNO .NE. 1) GO TO 29
   XVAR(NOMX)=LY/V
   IVAR=NOMX/IVAR+1
   IINDX=INDX-1+IVAR
   JINDX=JINDX-2+IVAR
   LXX=1
   GO TO 1
29 INICT=1
   GO TO 31
31 INICT=0
   CONTINUE
   MM=INNO
   DO 28 KK=1,MM
      LINDX=0
      IVAR=1
      JVAR=1
      INDX(IVAR)=INDX(IVAR+2)
      JVAR=JVAR+1
      LXX=LXX+2
   28 JVAR=JVAR+1
   LXX=LXX+1
   GO TO 29
   CONTINUE
   CORRE=CORPAR(JL)
   GO TO 106
105 CALL CORREL(IN, JX, JY, JVC, JYCC, CORRE)
106 CORRMM(KK,LKK)=CORRE
   JYCC(KK,LKK)=JYCC
   JYCC(LKK,KK)=JYCC
   CORPAR(KK,LKK)=CORRMM(KK,LKK)
22 CONTINUE
   CORRMM(K)=CORRMM(KK,5)
   GO 29 I=1,9
   IF(C(K) .LE. 5) GO TO 29
   IF(CORRMM(KK,1) .LT. CORRMM(KK,5)) GO TO 29
   CORRMM(KK)=CORRMM(KK,1)
29 CONTINUE
   IF(CORRMM(KK), GT. CORRMM(KK,5)) GO TO 45
   INNO(KK)=0
   GO 41 JJ=1,9
   IF(CORRMM(KK, JJ) .LT. CORRMM(KK)) GO TO 41
   INNO(KK)=INNO(KK)+1
   NNO(KK, INNO(KK))=JJ
   CONTINUE
41 CONTINUE
IF(INDX <= 1) GO TO 51
GO TO 52
45 INDXI = I
DO 47 I = 1, IY
IF(CORAM(I, IY).LT. CORAM(INDXI)) GO TO 47
INDXI = INDXI + 1
NOMER(INDXI) = IJ
47 CONTINUE
IF(INDXI .NE. IY) GO TO 49
IVAR = INDXI - 1
JVAR = INDXI - IVAR + 1
INDXJ = INDXI - 2 + IVAR
LINE = 1
GO TO 21
51 IVY2(INDXI) = INDXI
JY2(INDXI) = INDXJ
GO TO 25
52 IVY2(INDXI) = INDXI
JY2(INDXI) = INDXJ
GO TO 25
49 IVY2(INDXI) = INDXI
JY2(INDXI) = INDXJ
GO TO 25
75 CONTINUE
CMAK = 0
DO 55 K = 1, NM
IF(CMAK.GT. CORAM(INDXI)) GO TO 55
CMAK = CORAM(INDXI)
JLL = K
55 CONTINUE
IF(CMAK.EQ. CORAR(5)) GO TO 115
NNOMBR = 0
ISUM = 0
JSUM = 0
DO 75 I = 1, NM
IF(CORAM(I).LT. CMAK) GO TO 75
NNOMBR = NNOMBR + 1
ISTOR(NNOMBR) = I
ISUM = ISUM + IVY2(I)
JSUM = JSUM + JY2(I)
75 CONTINUE
IF(NNOMBR .EQ. 1) GO TO 115
GO TO 121
C++RETURN THE VALUE OF MAXIMUM CORRELATION AND
C++ITS POSITION.
111 IV2 = INDXI
JY2 = JINDEX
CMAK = CORAR(5)
RETURN
115 IV2 = IV1
JY2 = JY1
CMAK = CORAR(5)
RETURN
119 IV2 = IVY2(ISTOR(NNOMBR))
JY2 = JY2(ISTOR(NNOMBR))
CMAK = CMAK
RETURN
121 IV2 = ISUM / NNOMBR
JY2 = JSUM / NNOMBR
CMAK = CMAK
RETURN
END
SUBROUTINE MINVR(A, IA, MA, DETA, IER, IR, IC)
C*********
C********** A SUBROUTINE TO CALCULATE DOUBLE PRECISION INVERSE OF
C********** A DOUBLE PRECISION MATRIX BY GAUSS-JORDAN ELIMINATION
C********** METHOD;
C**********
DOUBLE PRECISION A(IA, JA), DETA, PIV, PIVL, TEMP
DIMENSION IR(MA), IC(MA)
IER=0
DO 1, I=1, MA
  IR(I)=0
1  IC(I)=0
  DETA=1.0D0
DO 123, JNL=1, MA
  CALL SUBMKD(A, IA, JA, NA, IR, IC, I, J)
  PIV=A(I, J)
  DETA=DETA*DETA
  IF(PIV.EQ.0.0D0) GO TO 17
  IR(I)=J
  IC(J)=I
  PIV=1.0D0/PIV
  DO 5, K=1, MA
    A(I, K)=A(I, K)*PIV
    A(I, J)=PIV
 5  DO 2, K=1, MA
    IF(K.EQ.1) GO TO 9
    PIV=PIV/A(K, J)
 6  CONTINUE
  DO 9, L=1, MA
    A(K, L)=A(K, L)-PIVL*A(I, L)
    A(K, J)=PIVL
 9  CONTINUE
    PIVL=PIV/A(K, J)
 11  DO 11, K=1, MA
    A(I, J)=A(I, J)-PIVL*A(K, J)
10  CONTINUE
12  CONTINUE
13  CONTINUE
  DO 16, I=1, MA
    K=IC(I)
    M=IR(I)
    IF(K.EQ.1) GO TO 16
    DETA=DETA
    DO 14, L=1, MA
      TEMP=A(K, L)
      A(K, L)=A(CL, L)
      A(CL, L)=TEMP
14  DO 15, L=1, MA
      TEMP=A(L, M)
      A(L, M)=A(L, I)
15  A(L, I)=TEMP
16  CONTINUE
17  IER=1
RETURN
END

SUBROUTINE SUBMKD(A, IA, JA, MA, NA, IR, IC, I, J)
C*********
C********** A SUBROUTINE TO FIND PIVOT VALUE FOR GAUSS-JORDAN
C********** ELIMINATION METHOD;
C**********
DOUBLE PRECISION A(IA, JA), TEST, ICAB
DIMENSION IR(MA), IC(MA)
I=0
SUBROUTINE PRMA(A, B, C, N, M, L, NL, ML, N2, M2, N3, M3)
DIMENSION A(NL, ML), B(N2, M2), C(N3, M3)

C*******
C******* A SUBROUTINE TO MULTIPLY TWO MATRICES.
C*******
DO 25 I=1, N
DO 25 J=1, L
SUM=0.0
DO 27 K=1, M
27 SUM=SUM+A(I, K)*B(K, J)
C(I, J)=SUM
25 CONTINUE
RETURN
END

SUBROUTINE TRANP(A, B, N, M, NL, ML, N2, M2)
DIMENSION A(NL, ML), B(N2, M2)

C*******
C******* A SUBROUTINE TO FIND TRANSPOSE OF A MATRIX.
C*******
DO 27 I=1, N
DO 27 J=1, M
B(J, I)=A(I, J)
27 CONTINUE
RETURN
END

SUBROUTINE DOUBLE(A, B, N, M, NL, M1)
DOUBLE PRECISION B(NL, M1)
DIMENSION A(NL, M1)

C*******
C******* A SUBROUTINE TO CONVERT A SINGLE PRECISION MATRIX
C******* TO A DOUBLE PRECISION MATRIX.
C*******
DO 5 I=1, N
DO 5 J=1, M
B(I, J)=A(I, J)
5 CONTINUE
RETURN
END
SUBROUTINE SINGLE(A, B, M, N, L, M)
DOUBLE PRECISION A(NL, ML)
DIMENSION B(NL, ML)
C
C SUBROUTINE TO CONVERT A DOUBLE PRECISION MATRIX
C TO A SINGLE PRECISION MATRIX.
C
DO 5 I=1, N
  DO 5 J=1, M
  B(I, J)=A(I, J)
5 CONTINUE
RETURN
END

SUBROUTINE SUBSTR(A, B, C, N, M, N1, ML, MS, N2, MS, N3)
DIMENSION A(NL, ML), B(NL, MS), C(NL, MS)
C
C SUBROUTINE TO SUBTRACT ONE MATRIX FROM ANOTHER MATRIX.
C
DO 25 I=1, N
  DO 25 J=1, M
  C(I, J)=A(I, J)-B(I, J)
25 CONTINUE
RETURN
END

SUBROUTINE DRAW(NVIEW)
DIMENSION IN(1:256), NAME3(5), NAME4(5)
C
C SUBROUTINE TO PRINT DIGITIZED VIEW
C
INTEGER A(128)
COMMON 'EE/ N112, NJ1, NAME3, NAME4
DATA IDOT,'+','+, 'IBLANK,'+','+
IF(NVIEW.EQ.2) GO TO 21
C FOR PRINTING FIRST VIEW
WRITE(12, 1)
1 FORMAT('1',+','+, '26X, 'PICTURE OF VIEW 1',+','+)
C PRINT FIRST 128 COLUMNS
OPEN 2, NAME3, LEN=2+NJ1, REC=NI12 CALL FSEEK(2, 1)
DO 3 I=1, NI12
READ(2) (IX(I, J), J=1, NJ2)
3 CONTINUE
CLOSE 2
IF(NJ2.LE.128) GO TO 111
WRITE(12, 2)
2 FORMAT('1',+','+, '2X, 128A1')
C PRINT NEXT 128 COLUMNS OF PICTURE
OPEN 2, NAME3, LEN=2+NJ2, REC=NI12 CALL FSEEK(2, 1)
C
C
7 IF(IN(J,J).EQ.1) A(J,C)=IDOT
   WRITE(12,5) (A(J,J), J=1, JC)
6 CONTINUE
   CLOSE 2
   GO TO 111
C----------FOR PRINTING SECOND VIEW
21 WRITE(12,6)
   FORMAT(1X, 2X, 25X, 'PICTURE OF VIEW 2', 1X, 2X)
   OPEN 4, NAME=4, LEN=2*NJJ2, REC=NII2
   CALL FSEEK(4, 1)
C----------PRINT FIRST 128 COLUMNS
   DO 12 I=1, NII2
   READ(4) (IN(J,J), J=1, NJJ2)
   DO 14 J=1, 128
     IF(IN(J,J).EQ.0) A(J,J)=IBLANK
     IF(IN(J,J).EQ.1) A(J,J)=IDOT
14 WRITE(12,5) (A(J,J), J=1, 128)
12 CONTINUE
   CLOSE 4
   IF(NII2.LE.128) GO TO 111
   WRITE(12,25)
   OPEN 4, NAME=4, LEN=2*NJJ2, REC=NII2
   CALL FSEEK(4, 1)
C----------PRINT NEXT 128 COLUMNS
   DO 16 I=1, NII2
   READ(4) (IN(J,J), J=1, NJJ2)
   DO 17 J=129, NJJ2
     JC=J-128
     IF(IN(J,J).EQ.0) A(J,J)=IBLANK
     IF(IN(J,J).EQ.1) A(J,J)=IDOT
17 WRITE(12,5) (A(J,J), J=1, JC)
16 CONTINUE
   CLOSE 4
   GO TO 111
111 RETURN
END
REFERENCES


VITA AUCTORIS

VYOMESH F. SHAH

1954 Born on April 24th, in Ahmedabad, Gujarat, India.
1976 Graduated from Gujarat University, Ahmedabad, Gujarat, India, with the degree of Bachelor of Engineering in Electronics and Communications.
1978 Candidate for the degree of Master of Applied Science in Electrical Engineering at the University of Windsor, Windsor, Ontario, Canada.