

THESIS



This is to certify that the

dissertation entitled A SYMBOLICALLY-ASSISTED APPROACU TO DIGITAL IMAGE REGISTRATION WITH APPLICATION IN COMPUTER VISION

presented by

ARDESHIR GOSHTASBY

has been accepted towards fulfillment of the requirements for

PH.D. degree in COMPUTER SCIENCE

Call 1. Vag Major professor -e

Date 7-21-83

MSU is an Affirmative Action/Equal Opportunity Institution

0-12771



# RETURNING MATERIALS:

Place in book drop to remove this checkout from your record. FINES will be charged if book is returned after the date stamped below.

# ROOM USE OFTY

## A SYMBOLICALLY-ASSISTED APPROACH TO DIGITAL IMAGE REGISTRATION WITH APPLICATION IN COMPUTER VISION

いいいも

1

By

Ardeshir Goshtasby

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

.

Department of Computer Science

1983 ·

Copyright by ARDESHIR GOSHTASBY

•

#### ABSTRACT

#### A SYMBOLICALLY-ASSISTED APPROACH TO DIGITAL IMAGE REGISTRATION WITH APPLICATION IN COMPUTER VISION

By

#### Ardeshir Goshtasby

Analyzing a sequence of images from the same scene often requires image registration; that is, the process of determining the position of corresponding points in the images.

Two images taken from the same viewpoint from a two-dimensional scene can be registered if the position of at least two pairs of corresponding points in both images are known. However, images obtained at different viewpoints from a three-dimensional scene (such as stereo images) cannot be registered by knowing only the position of a few corresponding points in the two images, and a window search is often needed to individually determine the position of corresponding points.

Images from a two-dimensional scene that have translational, rotational, and scaling differences, are registered by first segmenting the images and determining

#### Ardeshir Goshtasby

corresponding regions in both images. Corresponding regions are refined to obtain optimally similar regions. The centroids of corresponding regions are then used as corresponding points to determine the registration parameters. These centroids will correspond to each other regardless of translationsl, rotational, and scaling differences between the corresponding regions.

When registering stereo images of three-dimensional scenes, the images are segmented and a search is carried out for the position of points in one image that correspond to points on the region boundaries of the other image. Windows centered at region boundaries are used for the search because their high variances make searches more reliable than those using windows from homogeneous areas. To reduce the effect of geometric difference between the images in the window search process, window shapes are taken such that only parts of objects (and not objects and background) are used in the search. The position of corresponding points in two stereo images are used to determine disparity between these points. Depth of points in the scene can then be determined by knowing the disparity measures and the camera parameters. To My Parents

•

#### ACKNOWLEDGMENTS

I would like to thank Dr. Carl V. Page for introducing me to the area of Computer Vision and carefully supervising my research. I would like to thank Dr. George C. Stockman for spending a generous amount of his time discussing subjects related to my research and giving me many new ideas. I also would like to thank Drs. Stuart H. Gage, Roy V. Erickson, and Hans Lee for serving on my committee.

My appreciations go to Dr. Jon F. Bartholic and Mr. William R. Enslin for providing data for this work and offering me a research assistantship for the past two and a half years. I am also grateful to the National Aeronautic and Space Administration and the National Science Foundation for partially supporting this work.

Very special thanks go to my wife for her encouragement and understanding throughout this research endeavor.

iii

#### TABLE OF CONTENTS

LIST OF TABLES, vi LIST OF FIGURES, vii 1. INTRODUCTION, 1 2. BACKGROUND, 7 2.1. Definitions, 7 2.2. Notations, 11 2.3. Statement of the Problem, 11 2.4. Data, 12 2.5. Computer System, 12 2.6. Past Works, 19 3. SHAPE DISCRIMINATION, 28 3.1. Introduction, 28 3.2. Shape Description, 32 3.3. Shape Similarity Measurement, 32 3.4. Results, 42 4. IMAGE MATCHING BY PROBABILISTIC RELAXATION, 51 4.1. Introduction, 51 4.2. Probabilistic Relaxation Labeling, 54 4.3. Initial Label Probability Estimation, 54 4.4. Neighbor Contribution Factors, 56 4.5. Convergence of the Label Probabilities, 61 4.6. Results, 56 5. REGISTRATION OF IMAGES FROM A TWO-DIMENSIONAL SCENE, 80 5.1. Introduction, 80 5.2. Image Segmentation, 82 5.2.1. Segmentation of Image 1, 85 5.2.2. Segmentation of Image 2, 89 5.2.3. Region Correspondence, 94 5.2.4. Region Refinements, 98 5.2.5. Segmentation Results, 108 5.3. Selection of Control Points, 115 5.4. Determination of Transformation Function Parameters, 120 5.5. Results, 124 5.5.1. Experiment 1, 124 5.5.2. Experiment 2, 131

5.5.3. Summary of Results, 139 6. REGISTRATION OF IMAGES FROM A THREE-DIMENSIONAL SCENE, 142 6.1. Introduction, 142 6.2. The Correspondence Problem, 143 6.3. Avoiding Mismatches, 151 6.3.1. Occluded points, 151 6.3.2. Geometric Differences, 153 6.3.3. Homogeneous Areas, 89 6.3.4. Window Size, 156 6.3.5. Similarity Measure, 157 6.4. Computation of Depth, 158 6.5. Results, 160 6.5.1. Disparity on Region Boundaries, 164 6.5.2. Distinction of Objects and Background, 171 6.5.3. The Mismatches, 174 6.5.4. Disparity Inside Regions, 184 6.5.5. Summary of Results, 184 6.6. Further Improvements, 186 6.6.1. Model Matching, 188 6.6.2. Shadows, 191 6.6.3. Reflections, 195 7. CONCLUSIONS, 199 7.1. Summary, 199 7.2. Contributions, 204 7.3. Future Research, 204 APPENDIX: THE PROJECTIVE TRANSFORMATION, 211 BIBLIOGRAPHY, 214

## LIST OF TABLES

- Table 3.1. Similarity measures for two sets of regions, 44
- Table 3.2. Roundness of two sets of regions, 44
- Table 3.3. Alphabet shape similarities of 3.7,b & c, 48
- Table 4.1. Initial Label Probabilities, 70
- Table 4.2. The knowledge matrix, 71
- Table 4.3. Measurements from image 2, 71
- Table 4.4. Label probabilities after iteration 1, 72
- Table 4.5. Label probabilities after iteration 2, 72
- Table 4.6. Label Probabilities after iteration 5, 73
- Table 4.7. Label Probabilities after iteration 10, 73
- Table 4.8. Label probabilities after iteration 30, 74
- Table 4.9. Label probabilities after iteration 100, 74
- Table 4.10. Label probabilities after iteration 2, s=5, 78
- Table 4.11. Label Probabilities after iteration 5, s=5, 78
- Table 4.12. Label probabilities after iteration 10, s=5, 79
- Table 5.1. Coordinates of centroid versus noise, 121
- Table 5.2. centroids in images 1 and 2, 127
- Table 5.3. Square-root errors for the 11 points, 130
- Table 5.4. Centroids in MSS and TMS images, 134
- Table 5.5. Square-root errors for the 14 points, 139
- Table 6.1. Speed and accuracy of stereo registration, 185

vi

## LIST OF FIGURES

- Figure 2.1. Point p and its image p' in a pin-hole camera, 8
- Figure 2.2. The HCMM Day-Vis and Night-IR images, 13
- Figure 2.3. The MSS and TMS images, 15
- Figure 2.4. Stereo images of objects on a table, 16
- Figure 2.5. Aerial stereo images of a residential area, 16
- Figure 2.6. Stereo images of the truck scene, 17
- Figure 2.7. Stereo images of the garden scene, 17
- Figure 2.8. Rotated rectangular windows, 22
- Figure 2.9. Rotated circular windows, 23
- Figure 3.1. A shape and its shape matrix, 33
- Figure 3.2. Reconstructed shape of Figure 3.1.a, 34
- Figure 3.3. A shape and the same shape with missing part, 41
- Figure 3.4. Shape matrices and defect-showing vector, 42
- Figure 3.5. Two sets of lake boundaries, 43

Figure 3.6. Measurement of perimeter and area of a shape, 45 Figure 3.7. Original alphabets, and digitized ones, 46 Figure 4.1. Two segmented images of the same scene, 69 Figure 4.2. Probabilities of object 14 having label 7, 75 Figure 4.3. Probabilities of object 14 having label \*, 76 Figure 4.4. Probabilities of Object 9 having label 2, 77

vii

Figure 5.1. Histogram of gradient image 1, 86 Figure 5.2. Three different Histograms of M(j), 87 Figure 5.3. Histogram of gradient image 2, 90 Figure 5.4. Estimation of the optimal threshold value, 102 Figure 5.5. Computation of the optimal threshold value, 104 Figure 5.6. Image slices from Algorithm 5.2, 106 Figure 5.7. Determination of the optimal boundary points, 106 Figure 5.8. Segmentation of Day-Vis and Night-IR images, 109 Figure 5.9. Segmentation of MSS and TMS images, 110 Figure 5.10. Image 1 and image 2 intensity relations, 111 Figure 5.11. Computation of the optimal centroid, 117 Figure 5.12. Windows with different noise levels, 119 Figure 5.13. Windows of Figure 2.14 thresholded at 84, 120 Figure 5.14. The HCMM Day-Vis and Night-IR images, 125 Figure 5.15. Closed boundary regions of images of 5.14, 126 Figure 5.16. Resampled and overlaid images, 129. Figure 5.17. The MSS and TMS images, 132 Figure 5.18. Closed-boundary regions of MSS and TMS images, 133 Figure 5.19. The TMS image resampled and overlaid, 135 Figure 5.20. Registration by polynomial transformation, 138 Figure 6.1. A balanced stereo camera model, 146 Figure 6.2. Locating right image boundaary in left image, 146 Figure 6.3. Stereo images of a three-dimensional scene, 148 Figure 6.4. Estimation of match points inside regions, 150 Figure 6.5. Image of an object in two stereo cameras, 152 Figure 6.6. Traditional and new window shapes, 154

viii

Figure 6.7. Stereo images of objects on a textured table, 161 Figure 6.8. Aerial stereo images of a residential area, 162 Figure 6.9. Stereo images of the truck scene, 163 Figure 6.10. Stereo images of the garden scene, 163 Figure 6.11. Segmentation of image of Figure 6.7.b, 166 Figure 6.12. Segmentation of image of Figure 6.8.b, 166 Figure 6.13. Segmentation of image of Figure 6.9.a, 167 Figure 6.14. Segmentation of image of Figure 6.10.a, 167 Figure 6.15. Disparities on region boundaries of 6.11, 169 Figure 6.16. Disparities on region boundaries of 6.12, 169 Figure 6.17. Disparities on region boundaries of 6.13, 170 Figure 6.18. Disparities on region boundaries of 6.14, 170 Figure 6.19. Same as 6.13 object & background interchanged, 172 Figure 6.20. Disparities on region boundaries of 6.19, 1172 Figure 6.21. Disparities of region boundaries of the truck, 173 Figure 6.22. Disparities on region boundaries of the bldg, 1173 Figure 6.23. Segmentation of image of Figure 6.10.a, 177 Figure 6.24. Computed disparity from both images, 178 Figure 6.25. Disparity before and after smoothing, 180 Figure 6.26. Image of box, segmented, its edges, 182 Figure 6.27. Disparity on edges and faces of the box, 183 Figure 6.28. Image of an object and its shadow, 192 Figure 6.29. Images of a point and its reflection, 197 Figure A.1. Straight lines under projective transformation, 213

ix

#### Chapter 1

## INTRODUCTION

This dissertation discusses the design of a system for automatic registration of digital images by a symbolically-assisted approach. Image registration is the process of determining the position of corresponding points in two images of the same scene. It is needed in many areas of image processing and computer vision, such as change detection, object tracking, motion analysis and stereo depth perception.

The proposed approach is symbolically-assisted in the sense that the images are first segmented into regions (symbolic features), and correspondences are made between regions in the two images. Knowing region correspondences, correspondence can be established between points in the two original images.

The study has been carried out on two types of images.

 Images that have been obtained from the same viewpoint from a scene in which distances of objects from the camera are much larger than the heights of objects in the

scene so that images can be approximated as having been obtained from a two-dimensional scene. It is assumed that the images do not have non-linear geometric distortions, although they may have translational, rotational, and scaling differences. Examples of these images are the satellite images.

2. Images that have been obtained from two different viewpoints from a scene where the distances of objects from the camera are not large compared to the heights of objects in the scene. These images may have non-linear geometric differences due to the fact that they have been obtained different viewpoints from from a three-dimensional scene. However, it is assumed that the viewpoint differences are small (only a few degrees) and the images have been obtained parallel to the scene and do not have rotational or scaling differences. Examples of these images are stereo images.

Since two-dimensional scene images have linear geometric registered by differences, they can be a single transformation function. The parameters of the transformation function can be determined by knowing the position of at least two pairs of corresponding points in the two images. But three-dimensional scene images may have undefined non-linear geometric differences. In such a

situation, a single transformation function is not capable of registering the images, and there is a need to carry out a search for the position of every point from one image in the other image.

To register two images obtained from the same viewpoint from a two-dimensional scene, the images are first segmented into regions. Then an attempt is made to determine the correspondence between closed boundary regions in the two images. Since centroids of corresponding regions correspond to each other no matter what the translational, rotational, and scaling differences between the images, the centroids of corresponding regions are selected as corresponding points to estimate the transformation parameters.

Image segmentation has been approached by using an iterative thresholding procedure to obtain optimally similar regions in the two images. The segmentation procedure involves first segmentation of one of the images using the gradient information in the image. The other image is then using its gradient information segmented and the segmentation result of the first image. At this point, corresponding regions in the two images are determined. For every region in the first image where is there а correspondence in the second image, the area around the region in the second image is iteratively thresholded until

a region is obtained that is most similar to the corresponding one in the first image.

A probabilistic relaxation approach is given for the determination of corresponding regions in the images. In this approach, regions in one image are assumed to be a set of objects and regions in the other image a set of labels. Then the initial probability of a given object having a given label is determined by the boundary shape similarity of the two regions refered to by the object and the label. Label probabilities are iteratively updated using the inter-region distances as a constraint until a unique labeling is found or computation time reaches its limit.

Region shape similarities are determined by the matix approach. In this approach, shapes are transformed into matrices by polar redigitization of the shapes, and similarity measurements are made on the matrices. This approach lends itself to measurement of shape similarities irrespective of rotational and scaling differences between the shapes.

To register stereo images, the images are first segmented and an attempt is made to establish correspondence between regions in the images (global correspondence). Then point correspondence (local correspondence) is achieved by assistance from the global correspondence. Since in stereo

images, the position of corresponding points differs by a horizontal value, to determine corresponding regions in the two images, a window centered at a point on the region boundary in one image is taken and is searched in a narrow band horizontally in the other image. Once correspondence is established between a pair of points belonging to corresponding regions by following the boundary of the region in one image, corresponding points in the other image are located by searching in small windows centered at the region boundaries.

The main operation in stereo image registration is window searching. To increase the accuracy of window seaches, the nature of three-dimensional scenes and camera geometries are studied carefully, and main sources causing mismatches are identified as occluded points, geometric difference between images, homogeneous areas, window size, and similarity measure. Then counter measures are introduced to avoid the mismatches.

The organization of this dissertation is as follows: In Chapter 2, technical terms that are used in the following chapters are defined, the statement of the problem is given, and past works related to this study are reviewed. In Chapter 3, a method for discrimination of two-dimensional shapes using shape matrices is given. This shape

discrimination method is then used in Chapter 4 to measure region shape similarities for the purpose of determining corresponding regions in the images. The result of Chapter 4 is then used in Chapter 5 to register two-dimensional scene images. which is given along with experimental results. Chapter 6 then discusses the problem of registering three-dimensional scene images and determination of depths of points in the scene. Experimental results are then given for stereo image registration and suggestions are made for further improvements. Finally in Chapter 7, the contributions of this work are reviewed and topics for further research are given.

## Chapter 2

## BACKGROUND

During the past two decades considerable effort has been put into research and development in the area of computer vision. Computer vision deals with analysis of the signal-based information in an image and transforming this information to descriptions that are understandable by humans.

Computer vision may involve analysis of one image or a number of images taken from a scene. When a sequence of images is involved often there is a need to determine the position of corresponding points in the images. In the following, we will be concerned with the point correspondence problem called "image registration."

# 2.1. Definitions

<u>Def. 2.1.</u> Image: The projection of points in the scene to a pin-hole camera's image plane. A pin-hole camera, as

shown in Figure 2.1, is an ideal camera with the image plane at distance f in front of the camera. The projection of a point to the image plane is obtained by the intersection of the line joining the point to the camera's lens center, with the image plane. Most real life cameras act approximately like pin-hole cameras.





Note that a straight line in space is projected to a straight line on the image plane. The image of the line is obtained by determining intersection of the image plane with the plane created by the straight line and the lens center. Since the property of straightness stays invariant under this transformation, we can relate points on the line to their images by the following projective

transformation formulas [Peet 75].

If the scene can be approximated by a plane, then transformation (2.1) actually gives the relation between points (x,y) on the plane and their images (x',y'). Images obtained by satellites can be approximated as images of planes. Using the coordinate system of Figure 2.1, since x' and y' are inversely proportional to z, points from a three-dimensional scene can be related to points in the image by,

$$x' = (ax+by+c)/z(dx+ey+1.0)$$
(2.2)
$$y' = (fx+qy+h)/z(dx+ey+1.0)$$

Knowing coordinates of a point (x,y,z) in the scene, transformation (2.2) gives coordinates of its image (x',y') on the image plane [Caelli 81, pp 79].

Def. 2.2. Digital Image: A two-dimensional array of digital numbers in which each number represents the average value over a small rectangular area in the image. The values can be properties of the scene at that area such as intensity, color, texture, etc. The images that are used in the following are arrays of intensity values.

The larger the value, the brighter the area in the scene. 0 represents no light or black and 255 represents full light or white.

- <u>Def. 2.3.</u> Pixel: A picture element, or one element of a digital image.
- <u>Def. 2.4.</u> Image Registration: The process of determining the position of corresponding points in two images of the same scene.
- <u>Def. 2.5.</u> Region: A connected set of pixels such that each pixel is reachable from another pixel in the set.
- <u>Def. 2.6.</u> Image Matching: Determination of correspondence between regions in two images of the same scene. Note that image registration is point correpondence while image matching is region correpondence.
- <u>Def. 2.7.</u> Centroid of a Region: An imaginary point with coordinate values equal to the average coordinate values of pixels on the boundary (not interior) of the region. Interior pixels are not used to avoid holes in a region to affect the position of the centroid.

Other definitions will be given in appropriate places as the discussion proceeds.

#### 2.2. Notations

- A square root sign will be shown by SQRT.
- A multiplication is shown by concatenation of two operands like 2.3m meaning 2.3 multiplied by m.
- A vector name is shown by the name of the vector underlined; for example, V means vector V.
- An image is shown in the form f(i,j) where f is the name of the image and i and j are two variables counting the rows and columns of the image.

### 2.3. Statement of the Problem

Given two images of the same scene, we want to determine the position of all corresponding points in the images automatically. The images might have been obtained 1) from the same viewpoint from a two-dimensional scene but at different orientations or distances to the scene (images with translational, rotational, and scaling differences). Or 2) from two different viewpoints from a three-dimensional scene but at the same orientation and distance to the scene (stereo images). In Chapter 5, registration of images obtained at the same viewpoint from a two-dimensional scene will be considered. In Chapter 6 registration of images obtained at slightly different viewpoints from a three-dimensional scene is given. These images are called stereo pairs. Registration of stereo images will make it possible to measure the depth of points in the scene.

#### 2.4. Data

Data for registration of two-dimensional scene images has been provided by the Center for Remote Sensing, Michigan State University. Data consists of a pair of satellite images from the Heat Capacity Mapping Mission (HCMM) with 500 meter resolution (see Figure 2.2). These images have translational, and rotational differences, but are of the same scale. The images are obtained at different spectral bands and at different times. The image of Figure 2.2.a is a day-visible (Day-Vis) image acquired on 9/26/79 while the imaqe of Figure 2.2.b is a night-infrared (Night-IR) image acquired on 7/4/78.

Another pair of images also used are as two-dimensional scene images. These are Landsat Multi-Spectral Scanner (MSS) and Thematic Mapper Simulation (TMS) images shown in Figure 2.3. These





(b) Figure 2.2. HCMM (a) Day-Vis and (b) Night-IR images.

images were obtained at the same spectral band but have translational, rotational, and scaling differences. The TMS image has been obtained by an aircraft and involves some geometric distortions due to the change in attitude of the aircraft during the flight.

sets of data are used for registration of Four three-dimensional scene images. These are shown in Figures 2.4, 2.5, 2.6, and 2.7. Stereo images of Figures 2.4 2.6, and 2.7 have been obtained by the author. Images of Figure 2.4 show a number of objects arranged on a textured table while stereo images of Figures 2.6 and 2.7 were obtained from two outdoor scenes on the campus of Michigan State University. Stereo images of Figure 2.5 were provided by the Center for Remote Sensing, State University. These images have been Michigan obtained by a low altitude aircraft from a residential Perception of these scenes in three-dimensions is area. possible if the the images are viewed by, a stereoscope. The objective in registering these stereo images was to determine the relative depth of points in the scene.

All these images are of size 240x240 with 256 different gray values.



(a)



(b) Figure 2.3. (a) MSS and (b) TMS images.



(a) (b) Figure 2.4. Stereo images of objects on a table.



(a) (b) Figure 2.5. Aerial stereo images of a residential area.



(a) (b) Figure 2.6. Stereo images of the truck scene.



(a) (b) Figure 2.7. Stereo images of the garden scene.

#### 2.5. Computer System

The programs have been developed on the PDP 11/34 computer of the Image Processing Laboratory, Computer Science Department, Michigan State University. This system requires 65 micro seconds when adding two real numbers and 85 micro seconds when multiplying two real numbers. It has a main memory of size 64K, of which 32K can be used to load user programs and data. Because of the small size of the main memory, an image larger than 180x180 cannot be stored at one time in the main memory.

Since digital images consume a large amount of main memory, management of memory for storage of images and intermidiate results becomes a difficult problem. For this reason implementation of the proposed image registration system is recommended on a machine with virtual memory like the Digital Equipment Corporation's VAX-750 or the Harris Corporation's H500.

To speed up the processing time, most of the given algorithms can be implemented on parallel machines such as an array processor (CRAY-1) or multi-processor (ILLIAC IV). For example, refinement of regions in the image segmentation step, updating of label probabilities in the symbolic matching step, or image resampling are tasks in which each can be processed in parallel using a series of

processor elements with the same control program. On the other hand, tasks like matrix Exclusive-OR operation for the measurement of similarity between shapes, or other matrix operations, can be processed in parallel by an array processor.

#### 2.6. Past Works

One of the first attempts to register images digitally was made by Anuta [Anuta 69]. He used cross-correlation as the similarity measure to search for corresponding windows in the two images. Then he took centers of corresponding windows in the two images as corresponding points to estimate the translational difference between the images. This technique was able to register only images that had translational differences. Image registration by correlation has proven to be very effective over a wide range of imagery and still is one of the best techniques in image registration.

To speed up the time-consuming process of computation of cross-correlation, Anuta later used the fast Fourier transform algorithm to compute the cross-correlation values [Anuta 70]. The sum of absolute differences has also been used as an alternative similarity measure to speed-up the window search process [Barnea 72, Bernstein 76]. Most of the work in image translational registration has been centered on speed-up techniques rather than the accuracy of the registration. Dewdney has proposed a steepest-descent algorithm to limit the window search domain and therefore achieve a higher speed [Dewdney 78].

A two-stage window search technique has been used in [Vanderburg 77]. In this technique, first a subwindow is used to locate the possible locations for a match. Then the whole window is used to locate the best match position among the possible ones. This technique was later extended to the coarse-fine searching technique, where a coarse window is used in the first stage and then the fine resolution window is used in the second stage to find the best match position [Rosenfeld 77a]. The two-stage window search process has been extended to the multi-stage search technique in which the lowest resolution window is used in the first stage, and a higher resolution window is used as we go up the stages in the search [Hall 76, Tanimoto 81].

A11 speed-up techniques these (except for the steepest-descent technique) work when the sum of absolute differences is used as the similarity measure. A technique to speed up the window search process for the more accurate similarity measure, namely the cross-correlation coefficient, is given in [Goshtasby 83a]. It has been

experimentally shown [Svedlow 76] that the cross-correlation search technique consistently produces better results than the sum of absolute differences on different types of images.

Other similarity measures have also been used in the window search process. In [Wong 78], invariant moments have been used as the similarity measure. Similarity between two windows are measured by determining the cross-correlation between the logarithms of 7 invariant moments of the windows. In [Chandra 82], Haar transform coefficients and in [Schutte 80] Walsh-Hadamard transform coefficients are used in the same manner to carry out the search.

When the images are rotated with respect to each other, the above window search techniques, will fail. This is because even though the centers of two windows may correspond, we may obtain a low similarity measure due to the fact that other points in the windows do not correspond to each other. In other words, the similarity measures that were discussed above can not truly measure the similarity between two windows which are rotated with respect to each other.

As a matter of fact, similarity measure is not the only problem. When two windows are rotated with respect to each other, it is impossible for them to contain the same parts
As a matter of fact, similarity measure is not the only problem. When two windows are rotated with respect to each other, it is impossible for them to contain the same parts of a rectangular scene (except when the two windows are rotated by a multiple of 90 degrees with respect to each other). Figure 2.8 shows two windows in which their centers correspond to each other but contain different parts of the scene.



Figure 2.8. Rotated rectangular windows.

If we make the windows circular, when the centers of two windows correspond to each other, the two windows will cover the same parts of the scene no matter what the rotational difference between the windows. Figure 2.9 shows two circular windows which are rotated with respect to each other, but since their centers correspond to each other, they contain the same parts of the scene.



Figure 2.9. Rotated circular windows.

Now if we use invariant moments as the similarity measure with circular windows, we will be able to carry out the window search as we did before [Goshtasby 83b]. Hu has derived a set of invariant moments in which two windows containing the same pattern will have similar invariant moments no matter what their rotational differences [Hu 62].

Registration of rotated images by a symbolic approach has been described by Stockman [Stockman 78,82]. In this approach, symbolic features like line segments and line intersections are extracted from the images. Then by matching features of the same kind, transformation parameters are derived for each possible pair of features. extended to registration of images that have unknown scaling differences.

Another technique which uses symbolic features to register images with rotational differences is Clark's technique [Clark 80]. With this technique, boundaries of dominant objects in the images are determined and object boundaries are approximated by chords. Then matching is carried out between three connected chords from the two images to determine the approximate transformation parameters. The original boundary lines are used to refine the parameter values. This technique can also be extended to registration of images with unknown scaling differences.

In practice, images that have rotational and scaling differences are registered by some interaction from the user. To find corresponding points in the images, shade prints of the images with pixel coordinates printed at the borders are prepared. Then corresponding points in the images are carefully selected in unique areas and coordinates of corresponding points are read from the image borders [Van Wie 77]. Once a set of corresponding points from the images are found, determination of transformation parameters is straightforward.

One of the first attempts to register stereo images digitally was made by Hannah [Hannah 74]. In Hannah's work windows were taken from one image and were searched in the other image. Centers of the matched windows were then taken as the corresponding points. Since searching in low variance areas results in more mismatches than searching in high variance areas of images, in [Hannah 74], areas of low variance in the images are marked before matching is carried out. By this, attempts at matching in low variance areas forewarned of the low information condition. can be However, if the area under search has high variance, it does not automatically imply that the matching is going to be absolutely reliable. High variance areas could result in mismatches if geometric difference between them is large.

Assuming that two windows from two stereo images are available in which their centers correspond to each other, then as we go away from the centers of the windows, usually the geometric difference between them becomes large. This tells us that we should use pixels away from the window centers less importantly than pixels near the window centers. In [Mori 73], the windows are weighted with Gaussian type weights to reduce the effect of distortions at points far away from the window centers. In [Nevatia 76] a sequence of images obtained at small viewing angle

increments have been used to reduce the effect of geometric difference between images.

A computational model for human stereo vision has been proposed by Marr and Poggio [Marr 79]. According to this model, first the left and right images are filtered with four different filter sizes at different orientations. The filters essentially determine the second directional derivatives of the images. Then zero-crossings (zeros of the second directional derivatives) in the filtered images are determined in the horizontal direction. To establish correspondence between the left and right images, matching is carried out first between zero-crossings obtained from the same filter size and with the same sign and orientation in the two images, starting with zero-crossings from the largest filter size. The reason for matching zero-crossings from the largest filter size first is to establish a global correspondence so that matching of zero-crossings from the smaller filter sizes, which are more detailed and difficult, become easier.

The work presented here is based on the finding of Julesz that the human visual system uses global correspondences to assist in determination of local correspondences [Julesz 71]. This idea has been the philosophical basis for this work. A registration system was designed to first find a

global correspondence and then use the result of the global correspondence to obtain local correspondence.

Past works in digital image registration that are based on the idea of obtaining local correspondence by assistance from global correspondence are the works of Vanderburg and Rosenfeld in two-stage template matching [Vanderburg 77, Rrosenfeld 77a], Hall and Tanimoto in multi-stage template matching [Hall 76, Tanimoto 81], and Marr and Poggio in computational theory of human stereo vision [Marr 79].

Also of some similarity to this work is the work of Price [Price 76] where changes between two images of the same scene are determined by first segmenting the images into regions. Corresponding regions in the two images are determined and changes are detected region by region. To determine correspondence to a region in one image, features such as area size, perimeter size, length to width ratio and color are measured from the region. These measurements are then compared to those of regions in the other image. The region producing the most similar measurements is taken as its corresponding region.

1

#### Chapter 3

#### SHAPE DISCRIMINATION

#### 3.1. Introduction

Shape refers to the geometry of an object's surface appearance. The discrimination of shapes of two-dimensional objects will be discussed in this chapter. To discriminate shapes, first a procedure to describe them is given, then by comparing shape descriptions, shape discrimination is achieved. The description of shape is to be invariant to translation, rotation, and scaling.

Shapes can be described using distances and angles in many ways [Zusne 70]. Shape descriptions using distance measures include the ratio of the length of the longest side and the perimeter of the shape, the standard deviation of the length of the sides of the shape, the length of the

Descriptions using angles are the ratio of the largest to the smallest angle, the ratio of the largest angle to the total degrees of angles, and the standard deviation of angles.

Moments of side lengths, radii, and angles can be used to describe shapes too. In [Brown 76], the first four moments of the angles given by

 $M_{k} = \sum_{i=1}^{n} (a_{i} - \underline{a})^{k} / n$ 

are used to describe shapes. In this formula, k is the power of the moment, n is the number of angles,  $a_i$  is the size of angle i, and <u>a</u> is the mean of the angles. Similar formula can be used to compute the moments of side lengths and radii.

A measure which describes the degree of compactness of a shape is  $P^2/A$ , where P and A are the perimeter and area of the shape, respectively. Given different shapes of a constant area, the one with the smallest perimeter is the most compact shape.

Other measures which are used for shape discrimination are measures of regularity [Zusne 70] and angular variability [Atteneave 57]. Regularity is defined as the ratio of the standard deviation of side lengths and standard deviation of all angles. Angular variability is the mean absolute difference of adjacent angles taken in overlapping absolute difference of adjacent angles taken in overlapping pairs about the boundary, where convex angles are given a positive sign and concave angles are given a negative sign.

The shape measures (descriptors) given above are not information preserving since it is not possible to reconstruct the original shapes using their descriptions. A wide variety of shapes may have the same description. However there are descriptors which are information preserving, such as the Fourier descriptors [Persoon 77, Zahn 72], invariant moments [Hu 62], and syntactic techniques [Fu 82]. A review of information preserving descriptors is given in [Pavlidis 78, 80].

Of the many ways to describe a shape, which is the most effective? Marr and Nishihara [Marr 78] have given three criteria for judging the effectiveness of a shape descriptor. These are accessability, scope and uniqueness, and stability and sensitivity.

Accessability tells how economically (in terms of computation time and memory usage) a description can be generated for a shape in a digital image. Scope shows the class of shapes that can be described by the descriptor, and uniqueness tells whether a description uniquely describes a shape or if it represents a wide range of shapes. Stability tells how stable the description is in representing the

general property of the shape, and sensitivity shows how sensitive the descriptor is to finer distinctions between shapes.

Since our purpose in shape description is shape discrimination, add one more criterion to the we effectiveness of a shape descriptor: discriminability. Discriminability is the ease of determining the similarity between two shapes using the description of the shapes. Α shape descriptor should be designed in such a way that determination of similarity between shapes is easy and fast.

In the following, only quantized shapes or shapes in a digital image are considered. This implies that knowing the position of pixels belonging to the shape, we will be able to reconstruct the exact original shape. In section 3.2, a shape descriptor is given in matrix form using the polar quantization of the shape. Then in section 3.3, using descriptions of two shapes, their similarity is measured independent of their rotational and scaling differences. Finally, in section 3.4 the proposed shape discrimination technique is tested on shapes of lakes extracted from satellite images, and shapes of 26 alphabet characters.

#### 3.2. Shape Description

Figure 3.1.a shows a shape with its centroid at O. Let the size of its maximum radius be L. We draw circles centering at O and radii,

L/m, 2L/m, 3L/m, . . mL/m. Let the circles intersect the maximum radius of the shape, line OA, at al, a2, . . am. Then starting from al, a2, . . am on the maximum radius OA and counter-clockwise, we divide each circle into n equal arcs, so each arc is  $d\theta$ =360/n degrees. We construct an nxm matrix for a given shape as below and will call it the shape matrix.

<u>Algorithm 3.1:</u> Construction of shape matrix of size nxm for a given shape.

- Create an nxm matrix and initially set all elements of the matrix to 0.
- 2. Set element (i,j) of the shape matrix to l if the point on the circle of radius j(L/m) and arc value i(360/n) lies inside the shape.

The shape matrix of Figure 3.1.a for n=6 and m=4 is shown in Figure 3.1.b.



Figure 3.1. (a) A shape and (b) its shape matrix.

There is no limit to the scope of the shapes that the shape matrix can represent. It can describe even shapes with holes. If m and n are selected large enough, the description becomes unique, and for any one shape, there exists only one shape matrix. As it will be shown by Algorithm 3.1, this shape descriptor is information preserving, and the original shape can be reconstructed.

- <u>Algorithm 3.2:</u> Reconstruction of a shape from its shape matrix.
- 1. Generate a grid with (2m+1)x(2m+1) cells, as shown by Figure 3.2. Initially mark all cells as not belonging to the shape.
- 2. Assuming a Cartesian coordinate system at the center of

the grid, as shown in Figure 3.2, mark cell (x,y) of the grid as belonging to the object if element (i,j) of the matrix is 1 and the following relations hold,

 $i=(tan^{-1}y/x)/d\theta$ ;  $j=SQRT(x^2+y^2)/d1$ 

where  $d\theta=360/n$  and dl is the length of a side of a cell in the grid.



Figure 3.2. Reconstructed shape of Figure 3.1.a.

Selection of the appropriate values for m and n is very important. If m and n are larger than a given value, no new information about the shape will be obtained. If m and n are smaller than a given value, it would not be possible to reconstruct the exact original shape from its shape matrix. What are the smallest values of m and n which still allow the exact original shape to be reconstructed? If the length of the maximum radius of a shape is L and m > L, more than one circle will be passed through a pixel, and repeated information will be collected. If d $\theta$  is the smallest angle between two neighboring pixels on the boundary of the shape when viewed from its centroid, n should also not be greater than  $360/d\theta$ .

For a shape with maximum radius L, n=360/d0 (where  $d\theta=360/2\pi L$ , so actually n=2 $\pi L$ ), and m=L guarantee reconstruction of the exact original shape.

<u>Proposition 3.1:</u> If the values of m and n are selected properly (m>L and n>2 $\pi$ L), then a shape matrix that has been obtained by Algorithm 3.1 can be used to reconstruct the exact original shape using Algorithm 3.2. In the following, an element of the original shape is referred to as a pixel, an element of the shape matrix is referred to as an element, and an element of the grid produced by Algorithm 3.2 is referred to as a cell. L is the length of the maximum radius of the shape.

Proof: A pixel which belongs to the shape at distance  $d \le L$ from the centroid of the shape can be viewed from the centroid of the shape by angle  $d\theta'=360/2\pi d \ge 360/2\pi L$  degrees. Now if we let  $m \ge L$  and  $n \ge 2\pi L$ , two neighboring elements in a column of the shape matrix differ in  $d\theta=360/n \le 360/2\pi L \le d\theta'$  degrees and two neighboring elements in a row of the shape matrix differ in  $L/m \le 1$  units. This shows that each pixel in the shape is mapped to at least one element in the shape matrix by Algorithm 3.1.

Also, since a cell at distance d' $\leq$ L from the center of the grid is viewed from the center of the grid by angle d0"=360/2 $\pi$ d' $\geq$ 360/2 $\pi$ L  $\geq$ 360/n=d0, then in Algorithm 3.2, a cell in the grid also corresponds to at least an element in the shape matrix. This shows that Algorithm 3.2 will mark each element of the grid at least once. If element (i,j) of the matrix is 1, the cell with coordinates (x,y) (such that the cell's distance to the center of the grid is SQRT(x<sup>2</sup>+y<sup>2</sup>) and its angle with respect to the x-axis is tan<sup>-1</sup>y/x) ) will be marked as belonging to the shape by Algorithm 3.2.

Now we know that element (i,j) of the shape matrix is set to 1 by Procedure 3.1 only if the pixel at distance j(L/m)from the centroid of the shape and angle id0 with the maximum radius of the shape, belongs to the shape. In this mapping, the relation of a pixel to the centroid and the maximum radius of the shape is equal to the relation of a cell to the origin and the x-axis of the coordinate system in the grid. The relation is the distance of the pixel (cell) to the centroid (coordinate system origin) and the angle with the maximum radius (x-axis). Therefore, the reconstructed shape will have the same geometry as the original shape. We conclude that if a shape with maximum radius L is available and a shape matrix of size  $m\geq L$  and  $n\geq 2\pi L$  is constructed using Algorithm 3.1, the original shape can be reconstructed from this shape matrix using Algorithm 3.2. #

Since m and n are the only parameters of a shape matrix, a shape can be described in this way regardless of its (maximum radius) size. This descriptor is invariant with respect to sizes of the shapes. Since the angles are measured with respect to the orientation of the maximum radius, it does not matter how the shape is oriented, and the descriptor is invariant under rotation of the shapes too. In the next section, the last criterion for effectiveness of the shape matrix, discriminability, will be examined.

#### 3.3. Shape Similarity Measurement

To determine the degree of similarity between two shapes, the matrices of the two shapes are compared. To compare two matrices, the dimensions of the matrices should be equal. So, for similarity measurement purposes, if mlxnl and m2xn2 are the sizes of the two shape matrices then we let m=min(ml,m2) and n=min(nl,n2) and construct shape matrices
of size mxn for both of the shapes.

To determine how similar two mxn matrices are, we count the number of corresponding elements in the two matrices that are equal. This can be done simply by Exclusive-ORing the two matrices. Equal elements will result in value 0 and non-equal elements in value 1. Total number of the 0's shows the similarity.

Let Sl and S2 be two shape matrices of size mxn, then the similarity between Sl and S2 is computed by the following Algorithm.

<u>Algorithm 3.3:</u> Determination of similarity between two shape matrices Sl and S2 of size nxm.

- 1. S=:0
- 2. For i=l to n
- 3. | For j=l to m
- 4. | | S:=S+S1(i,j).XOR.S2(i,j)
- 5. Similarity:=l-S/(n-2)m

Step 5 of Algorithm 3.3 is adjusted so that when two shapes are completely similar (S=0), we obtain Similarity=1. We have defined a circle and a line as two completely nonsimilar shapes. A line corresponds to 2 rows with values equal to 1 in the shape matrix, and a circle will produce a shape matrix with all elements equal to 1. Their difference is S=(n-2)m, and Similarity=0 is obtained for a circle and a line. Other shapes will produce values between 0 and 1.

<u>Proposition 3.2.</u> If a shape is rotated counter-clockwise by i(360/n)degrees, its shape matrix is shifted down cyclically by i rows. The proof is straightforward.

There exist shapes with more than one unique maximum radius which may produce different shape matrices depending on the maximum radii used. Proposition 3.2, however, assures that if a shape has two equal maximum radii and the angle between them is i(360/n), and if we create two shape matrices using the two radii, then the two shape matrices will be similar if one matrix is rotated cyclically by i rows with respect to the other.

Since existence of noise in one of the images may cause selection of the wrong maximum radius as the start point, we refine Algorithm 3.3 so that one of the shape matrices is rotated cyclically by 0, d $\theta$ , 2d $\theta$ , . . (n-1)d $\theta$  and each case is compared to the other shape matrix. Among the n obtained similarity measures, the one with the largest value is picked up to be the similarity between the two shapes.

<u>Algorithm 3.4:</u> Determination of similarity for shapes with more than one maximum radius.

- 1. For k=0 to n-1
- 2. { S(k+1):=0
- 3. | For i=l to n
- 4. | For j=1 to m
- 5. | | S(k+1):=S(k+1)+S1(i,j).XOR.S2((i+k)MODn,j)
- 6. Similarity:=l-max(S(i))/(n-2)m

Another property of the shape matrix is that, if a part of a shape is missing or a hole in the shape is misplaced, then the effect will be only on the few related columns in the shape matrix. This property is shown in Figures 3.3 and 3.4. If we determine the number of elements in each row of the two shape matrices that are different, as shown in Figure 3.4.c, we see that rows 6.7, and 8 have large values while other row values are small. This tells us the position of the defect and its magnitude. We will call the vector of Figure 3.4.c the "defect-showing vector."

The difference between two non-similar shapes and two similar shapes, one with a missing part, is that in the former, the difference between the two matrices is spread randomly to most of the rows, while in the latter, the

(a) (b)

Figure 3.3. (a) A shape and (b) the same shape with a missing part.

difference is concentrated on only the rows that correspond to the defect at the given angle. By using the information in the defect-showing vector, one can argue about the defectiveness of a shape.

Assume objects on a conveyor belt approach a camera, the camera views the objects from above, and is required to recognize the objects by comparing their boundary shapes to the models already stored in the computer. Imagine also that some of the objects might be defective, and there is a need to recognize them and set them aside. The shape discrimination technique which is proposed here can take care of this problem up to a point. Note that the defect should be small, otherwise the centroid of the shape may move far enough to produce a very different shape matrix.

11111111111111	111111111111111	0
1111111111100	1111111111100	Ő
1111111110000	1111111110000	Ő
11111110000000	1111110000000	0
11111110000000	111111000000	ŏ
1111111000000	1111100000000	Ť
1111111000000	1111000000000	Δ
11111111000000	11111100000000	
11111111100000	1111111100000	0
1111111110000	11111111111000	1
1111111111100	11111111111100	Т
1111111111100	1111111111100	ŏ
11111111111100	1111111111100	ŏ
1111111111000	11111111111000	а С
1111111111000	1111111110000	1
1111111111000	1111111111000	ن
1111111111100	11111111111000	1
1111111111000	1111111111000	, D
1111111100000	11111111000000	1
1111111000000	11111111000000	
1111111000000	1111111000000	
11111111000000	1111111000000	
11111111000000	1111111000000	
11111111000000	1111110000000	
1111111000000	1111111000000	ı.
1111111100000	1111111100000	
1111111110000	1111111110000	
1111111111100	1111111111100	

(a)

(b) (c)

Figure 3.4. Shape matrices and defect-showing vector.

# 3.4. Results

To evaluate the performance of the proposed shape discrimination technique, six lakes in Oakland County, Michigan were used. These are shown in Figures 3.5.a and 3.5.b. These shapes were obtained by segmenting a Landsat-2 band 7 and a Thematic Mapper Simulation band 3 of the same area, respectively. The two sets of shapes have scaling and rotational differences.



(a)(b)Figure 3.5. Two sets of lake boundaries.

The similarity of each pair of shapes from the two images are determined using Algorithm 3.4. The results are shown in Table 3.1. Entry (i,j) of the table shows the similarity between shape i of Figure 3.5.a and shape j of Figure 3.5.b. All the shapes were recognized correctly. Table 3.1. Similarity measures of regions from Figures 3.5.a and 3.5.b computed by Algorithm 3.4. The best matches are underlined.

			Regio	ns of	Figure	3.5.b	•
		1	2	3	4	6	6
Regions	1	0.82	0.72	0.80	0.70	0.77	0.57
3.5.a.	2	0.74	0.96	0.69	0.82	0.74	0.83
	3	0.79	0.71	0.82	0.68	0.74	0.52
	4	0.66	0.81	0.71	0.90	0.71	0.73
	5	0.74	0.72	0.72	0.72	<u>0.80</u>	0.66
	6	0.58	0.81	0.53	0.72	0.64	0.94





Table 3.2 shows the roundness  $(P^2/4\pi A)$ , where P and A are perimeter and area of the shape, respectively). Three out of the six matches were wrong. We computed the perimeter of a shape as follows.

While travelling along the boundary of the shape, if we are presently at (i,j) and the next location to go is one of (i-1,j), (i+1,j), (i,j-1), (i,j+1), then we add 1 to the count. But if the next location is one of (i-1,j-1), (i-1,j+1), (i+1,j-1), (i+1,j+1) then we add SQRT(2) to the count. For example, the triangle of Figure 3.6 has perimeter of size P=10+5SQRT(2), assuming the side of a pixel as the unit. This is actually the perimeter of the triangle shown by the broken lines in Figure 3.6.



Figure 3.6. Measurement of perimeter and area of a shape.

The area of this shape is measured as the number of pixels occupied by the shape - P/2. The area of the shape shown in Figure 3.6 is A=21-(10+5SQRT(2))/2 = 16

-5SQRT(2)/2. This is actually the area of the triangle shown by the broken lines in Figure 3.6 (not exactly, but very close).

As a tougher test, 26 upper-case alphabet characters, as shown in Figure 3.7.a, were used in the above shape discrimination procedure. The characters were digitized once, as shown in Figure 3.7.b, and then rotated and enlarged slightly and digitized again, as shown in Figure 3.7.c. The two sets of characters are in good shape but have rotational and scaling differences. We want to see how well the above shape discrimination technique can recognize them.



Figure 3.7. (a) Original characters, (b) and (c) digitized.

The computed alphabet shape similarities using Algorithm 3.4 are shown in Table 3.3. Taking the largest similarity measure for each row or each column as the correct match, three errors out of 26 were made in recognizing the characters. In this process, characters "C", "G", and "K" of Figure 3.7.b are confused with characters "G", "D", and "P" of Figure 3.7.c, respectively; while characters "G", "H", and "K" of Figure 3.7.c are confused with characters "D", "B", and "E" of Figure 3.7.b, respectively.

Since the elements of the shape matrices used by the proposed technique are binary numbers, each row of the matrix can be stored in a computer word (if m is larger than the number of bits in a computer word, each row can be stored in several words) and the Exclusive-OR can be carried out word by word. A shape which has a shape matrix of size 36x16 can be stored in 36 words of a 16-bit word computer. Computation of similarity between two shapes is fast because the operation involved is Exclusive-OR, which is one of the fastest operations in any computer. Computing the shape matrices for any pair of shapes shown in Figures 3.5 and 3.7 and determining their similarity takes about 2 seconds on a PDP 11/34 computer. The shapes in Figures 3.5 and 3.7 fit in 32x32 windows.

	•	8	υ	0	ш	u.	σ	I	H	<b>ح</b>	×	L	Σ	z	0	٩	0	2	s	⊢	5	>	3	×	≻	2	
Z	. 58	. 65	. 62	. 59	.70	.61	. 59	.66	.65	.64	.67	69.	.64	.68	. 52	. 56	. 59	. 59	.64	.67	. 58	. 56	80.	. 70	.60	.83	2
7	. 79	. 65	.63	.64	.68	. 75	60.	. 56	.66	. 75	. 71	. 75	.64	.66	.53	. 75	.61	69	.72	. 76	.74	. 83	.70	.74	8	.67	>
×	.77	. 77	. 62	. 65	.71	.67	.61	.42	. 59	. 63	.66	. 60	.63	. 62	. 53	.62	. 59	. 59	.64	.61	.62	.62	00.	. 89	5	. 70	×
3	81	.88	80.	. 85	. 84	. 79	84	. 58	44	.64	.61	52	. 78	83	. 76	84	82	58	83	54	. 77	69	94	65	57	51	3
>	. 79	.65	.68	.66	.67	. 10	67	64	59	68	68	99.	10	65	57	72	69	72	67	99.	<b>6</b>	88	62	62	72	59	>
þ	74	1	14	69	67	68	46	41	57	65	20	59	18	. 76	65	50	13	80	0.4	61	86	72	67	67	6	62	Þ
⊢	68	80	99	65	72	<b>6</b>	. 73	69	. 73	. 78	72	74	.63	.67	54	8	. 69	61	13	87	11	. 73	67	74	84	74	+
s	55	68	86	87	84	72	36	62	47	56	74	59.	81.	. 61	89	71.	87.	80	92.	51.	69	69	8		61	65 .	s
ď	84	87	87	85.	78	83.	85.	67 .	44	68 .	02	51.	86 .	89	. 61	83.	86 .	89	6	57	84	73 .	72 .	65 .	66	65 .	œ
0	82.	89.	88	92.	81.	75	42.	40 40	45	58.	09	44	82.	84.	Ee	78 .	97.	64	87	51.	71.	64 .	67 .	59.	59.	61.	0
٩	84.	. 17	75 .	77 .	77 .	89	80	63.	64 .	80	80.	62 .	71.	. 69	67 .	97 .	8.	69	81.	70.	78 .	77 .	72 .	71.	75 .	65 .	٩
0	69	. 61	81.	83.	72 .	60	88.	31.	34	44.	63 .	46.	73 .	74 .	. 96	61.	87 .	73 .	73 .	. 99	59.	58.	75 .	59.	<u>4</u>	53 .	0
z	80	81.	81.	82.	76 .	77 .	84.	. 69	54.	62 .	70.	49.	86.	. 69	80	80	84.	66 .	82.	55.	73 .	68.	71 .	. 99	61.	64 .	z
Ξ	75 .	88	83 .	82.	80.	72 .	. 61	63 .	47 .	62 .	64 .	54 .	. 68	68.	75 .	72 .	81.	63 .	78 .	59.	75 .	. 69	73 .	. 69	53 .	55 .	Σ
	64 .	59 .	56.	56 .	64 .	67 .	54 .	36 .	73 .	71 .	60	81.	80	 56	46 .	66 .	54 .	90	61.	74 .	. 99	. 69	61.	65 .	71 .	65 .	L
¥	78 .	79 .	76 .	75.	82.	79.	75 .	57 .	55 .	67 .	77 .	54 .	78 .	65 .	69.	80.	77 .	65 .	80.	59.	74 .	1.	67 .	73 .	59 .	56 .	¥
2	65 .	80	57 .	56 .	64 .	72 .	55 .	62.	75 .	88.	68	72 .	63 .	60	47 .	74 .	56 .	58 .	62 .	74 .	65 .	68 .	64 .	69 .	50.	68 .	5
	55.	50 .	50 . 50	51.	58 .	71 .	50.	54.	87.	72 .	63.7	72 .	53 .	56 .	43.	63 .	51.	58.	61.	. 69	55	58	55	. 99	65 .	. 69	-
I		85.	80.	79.	1.	75 .	. 61	78 .	99	. ۔ وی	78	56 .	о. О	81.	73 .	75 .	· 61	74 .	78		 	٠. 88	72 .	20	85	58 .	т
c		88	. 68	96.	73 .	55 .	87	53.	68 68	 48	. 99	•	78	63	87	57	84		•	ĩ. Đ	~. 69	57	67	99	5	55 .	
	°. Og	. 57	 E1	4	92	. 40	5	000	54		·. 20	51.		. 67	38 .	32 .	. 61	40	 0	36		15	·. 2	2		55 . !	
	32 .1	ະ. ເຄ	4	 98	 68	18 18	92 92	36 .	. 65	40	 60	52 .		22	32 .	5. 02	37	9.	88	9	82	5	2	88	4	. 01	
	<u>ה</u>	<del>.</del> و	37 .1	۲. 90	ר. מו	r : 2		 69	15 1	9. 99	2. 2	. 1	. 18	6	יי גע	т. е		9. Eg	3. 9		9	. 42	4	2	9.		
	3. 3.	2. 8	3. 5.	3. 61	2 12	 65	3. 5.	ະ. ຄ	4. ES	3. 95	65	54 . 4	3. 0	54 .6	5.		2. 5	9. 65	3.1.	3. 8	5	9. 82		Э. E	3. 45	Э. е	
Ĩ	Э. Е	20	ي. ع	4		ч. 6	۲. و	٩. 0	5. 5.	34 .6	1.6	5.6	11 .8	9.0	۲. و	2.6	<u>د</u>	14 .5	16 .7	8. 8.	г. е.	9. 9.	1.6	1.6	6.5	ί2 . Ε	5
Ξ	2 .8	<b>[</b> 20 [ 10]	4	8.8	5. 8.	1.6	5. 0	4.6	8.4	2.5	. 8	4.	A. 6	2.7	5.6	6.7	<b>6</b> .9	1.6	8.0	4	4.7	4.6	9.0	5. 6	8 5.	2.6	
<	6.	۴.		. 7	٢.	80.	œ.	9.	<u>ເ</u>			Ø.		2	9.	Ó	ي ب		Ő	Ó	Ċ	¢,	Ň.	Ö	2	9.	◀
	•	80	υ	۵	ш	LL.	σ	I	н	7	¥	_	Σ	z	ο	۵	0	α	S	⊨	∍	>	3	×	≻	N	

•

Table 3.3. Alphabet shape similarities of Figures 3.7.b,c.

The defect-showing vector which was introduced in section 3 contains information about the difference of two shape matrices, row by row. This corresponds to the difference of the original shapes at a given angle. Here it becomes possible to conclude whether the shape difference is due to defect or to the fact that the two shapes are different. This capability becomes especially useful in a vision system where objects are inspected for possible defects by comparing the boundary shape of the sensed object to boundary shapes of models already stored in the computer.

A comparison of the above described technique with the shape description of Peli [Peli 81] where distances of points on the boundary of a shape to its centroid are used for shape description, would be apropriate. In the Peli's technique, points on the boundary of the shape are sampled with equal angular steps, measured from the centroid of the shape. The distances of points on the boundary of the shape to its centroid are used as features to describe the shape. Similarity between two shapes is computed by correlating the distance features from the two shapes.

- The technique of Peli cannot handle shapes with holes, while the above technique can.
- 2. The shape description of Peli is not information preserving because at a given angle only one distance

value is measured, and if the shape intersects the radius at that angle at more than one point, the sum of the distances from the intersected points to the centroid is taken as the distance feature. Once distances are added together it is not possible to tell how many intersection points have existed or how much their distances to the centroid are. There are many different ways to add a number of distances together to obtain a given sum. Therefore, from the distance measures it is not possible to reconstruct the original shape. The above technique, however, is information preserving, and the original shape can be reconstructed from its shape matrix.

3. Computation of similarity between two shapes by Peli's technique requires computation of cross-correlation between two sets of distance features, while computation of similarity in the above technique is a matrix Exclusive-OR operation which is faster. Since a row of a shape matrix can be stored in a computer word (or several computer words) the Exclusive-OR can be carried out word by word rather than bit by bit, and the computation of similarity in the above technique is faster than in Peli's technique.

#### Chapter 4

#### IMAGE MATCHING BY PROBABILISTIC RELAXATION

### 4.1. Introduction

Image matching is the process of determining the correspondence between regions in two binary images (segmented images) of the same scene. In the following, a procedure will be given which determines the region correspondences by a probabilistic relaxation labeling.

Suppose an aerial image is available. From the measurements in the image we can assign labels to each pixel in the image to identify them. The labels can be vegetation, water, road, building, etc. Due to noise, pixel size, brightness conditions, or other factors, it may not be possible to obtain sufficient information to label each pixel unambiguously. From the measurements in the image alone, we may not be able to tell whether a pixel is water, vegetation, road, or building. We may then assign labels to pixels with given probabilities. If we are sure a pixel is

water, we may give probability 1 to water and probability 0 to other labels. If we are not sure about the label of a pixel, we may assign labels with probabilities based on the measurements from the image. This labeling, however, may be ambiguous, and relaxation is a tool to reduce the label ambiguities.

Relaxation is a cooperative process for reducing ambiguity between object labels by iteratively updating the probabilities of the labels using initial observation and world knowledge. Initial observation includes intensity or the color values of pixels, and world knowledge is some constraint among labels. A constraint may come from the fact that a road pixel cannot be (or has a very low probability of being) surrounded by water pixels.

Relaxation was first described by Rosenfeld [Rosenfeld 76] and has since been used in many computer vision applications [Zucker 76b,77, Rosenfeld 77, Davis 80,81]. These relaxation processes are designed to reduce local ambiguities, but there is no guarantee of reaching a globally consistent labeling. A process which reduces global ambiguity as well as local ambiguity is the hierarchical relaxation labeling process [Zucker 78a].

The traditional relaxation is a bottom-up process, and probabilities updated fixed label are based on neighborhoods. There are relaxation processes available, however, that enable top-down control in updating label probabilities. These processes are known as augmented relaxation labeling and dynamic relaxation labeling and have been described by Kuschel and Page [Kuschel 82]. These update label probabilities based on dynamic processes neighborhoods through a broadcast and receive mechanism.

The relaxation process which is used in the following for determination of corresponding regions in two images is based on the original relaxation labeling process of Rosenfeld. But the neighbor contribution factors are computed using information from all object labels, rather than the immediate neighbors, so that label probabilities are updated to achieve a globally consistent labeling.

In section 4.2, the probabilistic relaxation labeling is defined formally. Sections 4.3 and 4.4 determine the initial label probabilities and neighbor contribution factors, respectively. Then convergence of the label probabilities are discussed in section 4.5. Section 4.6 gives the results of the proposed technique on two sets of regions obtained by segmenting two satellite images of the same scene.

## 4.2. Probabilistic Relaxation Labeling

Let  $A=\{a1, a2, ..., am\}$  be a set of objects (the regions in image 2), and let  $B=\{b1, b2, ..., bn, b^*\}$  be a set of labels (the regions in image 1). From local measurements (shown in the next section), we can estimate the initial probability  $Pi^{(0)}(bj)$  that object ai has label bj. b\* is the "undefined object" label and  $Pi^{(0)}(b^*)$  is the probability that object ai is undefined (region ai is not present in image 1). The initial probabilities satisfy the condition  $\Sigma Pi^{(0)}(bj)=1$  for all ai  $\in A$ .

The label probabilities are updated according to the following iterative formula.

$$Pi^{(k+1)}(bj) = \frac{Pi^{(k)}(bj)[1+qi^{(k)}(bj)]}{\sum_{j} Pi^{(k)}(bj)[1+qi^{(k)}(bj)]}$$
(4.1)

where qi<sup>(k)</sup>(bj) is the neighbor contributions to Pi<sup>(k)</sup>(bj), in the kth iteration, and is in the range [-1,1]. The label probabilities at any iteration satisfy  $\sum_{j} Pi^{(k)}(bj) = 1$ . The relaxation process is iterated until the label probabilities converge or the computation time exceeds its limit.

There are two sources of information for this relaxation process, the initial probability estimates and the neighbor contribution factors. In the following, each is discussed

in detail.

## 4.3. Initial Label Probability Estimation

The initial probability that region ai has label bj is computed using the similarity between the two regions. Since image 1 and image 2 may have translational, rotational, and scaling differences, region bj and region ai may have translational, rotational, and scaling differences. We should therefore use the properties of the regions which are invariant under translation, rotation, and scaling.

Some of these properties are average intensity, color, and shape. We will be using the shapes of regions to determine the similarity between them. A shape similarity measurement technique has been designed in Chapter 3 which determines shape similarities and is invariant under translation, rotation, and scaling of the shapes. The similarity between two shapes is represented by a number between 0 and 1, 1 showing complete similarity and 0 showing complete dissimilarity (a line and a circle has been defined as two completely dissimilar shapes).

We define the weight associated with the label bj of region ai to be

Wi(bj) = the similarity between regions ai and bj.

Wi(bj) changes between 0 and 1, and the more similar the two regions ai and bj, the closer the value of Wi(bj) to 1. We also define Wi(b\*) = 1-max Wi(bj). This is natural because if all the label weights of a region are small, it shows that there is no region in image 1 similar to ai, and so Wi(b\*) (the weight of the undefined region label) should be large. If one of the labels has a high weight, it shows that a similar region exists in image 1, and so Wi(b\*) should be small. We use weight functions Wi(bj) to estimate the initial probability that region ai has label bj as below.

 $Pi^{(0)}(bj) = Wi(bj)/\Sigma Wi(bj)$ 

## 4.4. Neighbor Contribution Factors

Neighbor contribution factors are tools by which the label probabilities can be updated so that the label assignments become consistent with the world knowledge. World knowledge can be, for example, relative sizes of the regions, distances of the regions from each other, and relative positions of the regions in the first image. We represent world knowledge in matrix form, calling it a knowledge matrix, and denoting it by W. An element of a knowledge matrix W(b,b') may show,

- 1. the size (perimeter size or area size) ratio of regions b
  and b'.
- 2. the distance between regions b and b',
- 3. the position of region b relative to region b' (above, below, to the right, to the left),
- 4. the shape similarity of regions b and b',
- 5. etc.,
- in image 1.

To find how compatible label bj of object ai is with labels of other objects, we pursue as follows. Let W(bj) = $\{W(bj,bl), W(bj,b2), . . W(bj,bn)\}$  show the distances between region bj and the other regions in image 1. Then for region ai in image 2 we determine the distances of ai to every other region. Let  $D(ai) = \{D(ai,al), D(ai,a2), . . . D(ai,am)\}$  denote this. Every object in image 2 has a set of labels, each with a different probability. Assuming ci (ci  $\in$  B) is that label of ai which has the largest probability, then we construct another vector
$$U^{(k)}(bj) = \{W(bj,cl), W(bj,c2), . . W(bj,cm)\}$$
where  $W(bj,ci)$  shows the distance between regions bj and ci  
in the first image. Then a measure like
$$qi^{(k)}(bj) = \frac{E(U^{(k)}(bj) D(ai)) - E(U^{(k)}(bj))E(D(ai))}{\sigma(U^{(k)}(bj))\sigma(D(ai))}$$

Now  $qi^{(k)}(bj)$  is a measure in the range [-1,1], and if region ai truely corresponds to region bj, then regardless of some mistakes in the labels of other regions,  $qi^{(k)}(bj)$ will be high. On the other hand if region ai truely does not correspond to region bj, then even though all other regions in image 2 might be labeled correctly, the value of  $qi^{(k)}(bj)$  will be low. This is a very desirable property, and we use  $qi^{(k)}(bj)$  as the neighbor contribution factor in formula (4.1).

Neighbor contribution factors for the undefined region labels  $qi^{(k)}(b^*)$  cannot be determined in this manner because correlation of distances that do not exist does not make

(W(b\*) is undefined). To determine the neighbor sense contributions for object ai having label b\* (the neighbor contribution for region ai of image 2 having no correspondence in image 1), we observe the following. If object ai truely has label b\*, then assigning label b\* to ai should increase the true label probabilities for other objects, while if b\* is not the true label of ai, assigning b\* to ai should decrease the true label probabilities of other objects.

Assuming that most of the highest probability labels of objects show their true labels, we can 1) compute the largest label probabilities of objects when b\* is assigned to ai (except when the largest label probability is b\*), and 2) compute the same label probabilities this time by assigning the largest probability label of ai to ai (except when the largest probability label is b\*). If there are M label probabilities in case 1 that are larger than in case 2 and m' is the number of objects whose largest probability label is not b\*, we should,

- i) increase the label probability for object ai having label b\* if M>m'/2,
- ii) decrease the label probability for object ai having label b\* if M<m'/2, and</pre>

iii) not change the label probability for object ai having label b\* if M=m'/2.

A measure which can simulate this property is 2(M/m'-0.5)and we will use it as the neighbor contribution factor, qi<sup>(k)</sup>(b\*). This measure also shows, if by assigning label b\* to ai, the largest probability for every object increases, then qi<sup>(k)</sup>(b\*)=1, while if the probability of all largest probability labels decreases, then qi<sup>(k)</sup>(b\*)=-1. This measure well characterizes the neighbor contribution factor.

In the above, we used the distance between regions as the world knowledge to estimate the neighbor contribution factors. Other features such as relative sizes of the regions, relative positions of the regions, etc., could also be used for this purpose.

For images with translational, rotational, and scaling differences we should select a knowledge matrix which makes the neighbor contribution factors,  $qi^{(k)}(bj)$ , invariant with respect to translation, rotation, and scaling of the images. For example, the position feature (above, below, to the right, to the left) is not invariant under rotation, but the distance feature is.

# 4.5. Convergence of the Label Probabilities

The label probabilities in the kth iteration was given by equation (4.1). For this equation to converge, we should have

$$\lim_{\substack{k \neq i \\ k \neq i}} \{ \operatorname{Pi}^{(k+1)}(bj) - \operatorname{Pi}^{(k)}(bj) \} = 0$$
  
or 
$$\lim_{\substack{k \neq i \\ im}} [\operatorname{Pi}^{(k)}(bj)[1+qi^{(k)}(bj)]/T^{(k)} - \operatorname{Pi}^{(k)}(bj) \} = 0$$

where

$$T^{(k)} = \sum_{j} Pi^{(k)} (bj) [1+qi^{(k)} (bj)]$$
  
or lim Pi^{(k)} (bj) {[1+qi^{(k)} (bj)]/T^{(k)}-1} = 0  
This implies that either  
1. lim Pi^{(k)} (bj) = 0 or  
2. lim [1+qi^{(k)} (bj)]/T^{(k)}-1 = 0  
Case 2 implies that  
1+ qi^{(k)} (bj) = \sum\_{j} Pi^{(k)} (bj) [1+qi^{(k)} (bj)]  
or = \sum\_{j} Pi^{(k)} (bj) + \sum\_{j} Pi^{(k)} (bj) qi^{(k)} (bj)  
or = 1+  $\sum_{j} Pi^{(k)} (bj) qi^{(k)} (bj)$   
or qi^{(k)} (bj) =  $\sum_{j} Pi^{(k)} (bj) qi^{(k)} (bj)$ 

For the process to converge, we should have either

1. 
$$\lim_{k \to \infty} \operatorname{Pi}^{(k)}(b) = \begin{cases} 1 & \text{if } b = b \\ \\ \\ 0 & \text{otherwise} \end{cases}$$

This shows that a unique assignment will be found after a large enough number of iterations. Or,

2. lim Pi<sup>(k)</sup>(bj) = constants other than 0 for more than one label. In this case lim Pi<sup>(k)</sup>(bj) will converge but a unique labeling is not possible. If a unique labeling is not found, we may eliminate those objects with label probability 1 from the set of objects and try to carry out the relaxation process on the remaining objects with a new knowledge matrix.

For a k large enough that label probabilities do not change much, we can approximate  $qi^{(k)}(bj)$  by  $qi^{(k-1)}(bj)$ . So, we have

In other words we can estimate  $Pi^{(k+1)}(bj)$  without using the result of the kth iteration. This shows that we can leave behind a large number of iterations in this manner and still reach the same label probabilities. In general we have

Equation (4.2) shows the approximation of Pi<sup>(k+s)</sup>(bj) by jumping over s-l steps. In the same fashion we can guide the relaxation process so that it will execute one in every s steps and force the label probabilities to converge faster. In the following, we will call s the speed-up factor. The new iterative formula is of the form

$$Pi^{(k+1)}(bj) = Pi^{(k)}(bj)[1+qi^{(k)}(bj)]^{S} / \sum_{j} Pi^{(k)}(bj)[1+qi^{(k)}(bj)]^{S}$$
(4.3)

The larger the s the more computational saving is obtainable, but large values of s may cause the label probabilities to converge to false values. To detect the occurance of such a case, in [Zucker 78b], extrapolation has been used to predict label probabilities in the next iteration. Then comparing the predicted probabilities and the computed ones, those probabilities that change unexpectedly are detected. This may not work all the times because some label assignments may be wrong from the very beginning. Their probabilities may keep increasing so that no change in the trend is observed, and ultimately the probabilities converge to a false fixed point.<sup>\*</sup> This fact is demonstrated by an experiment in section 4.6.

To be able to use formula (4.3) with more security, the following alternative is proposed. Use formula (4.1) for the first s iterations (k=1, 2, ..., s). The label probabilities will stabilize considerably. Then use formula (4.3) for iterations k=s+1, s+2, ... By doing so, convergence to a false fixed point may not be eliminated completely, but the probability of its occurance will be reduced. This assertion is backed up experimentally in the next section.

\* Fixed point: The point where label probabilities no longer change. A relaxation process may have more than one fixed point which are characterized by the neighbor contribution factors [Haralick 80b].

#### 4.6. Results

To study the behavior of this relaxation process, two sets of regions were used as shown by Figures 4.1.a and 4.1.b. Figures 4.1.a and 4.1.b are prepared by segmenting two (Heat Capacity Mapping Mission) satellite images of the same scene and extracting closed boundary regions in the images. There are 14 regions in each set, but there are some regions in one set which are not present in the other. The correct region correspondences are, (1,9), (2,10), (3,11), (4,\*), (5,13), (6,14), (7,\*), (8,1), (9,2), (10,3), (11,4), (12,5), (13,6), (14,\*), where the first number shows the region in image of Figure 4.1.b and the second number shows the corresponding region in image of Figure 4.1.a.

Initial label probabilities were determined by the shape similarity measure of Chapter 3. The results are shown in Table 4.1. The distances between the centroids of regions in Figure 4.1.a were used as the knowledge matrix, Table 4.2. The distances between the centroids of regions in image 2 are also shown in Table 4.3. Performing the relaxation process described above on these data, label probabilities changed as shown in Tables 4.4, 4.5, 4.6, 4.7, 4.8, and 4.9, for iterations 1, 2, 5, 10, 30, and 100, respectively. s is the speed-up factor of section 4.5. The label probabilities seem to be converging, but even after 100 iterations, the fixed point is not reached. Applying the speed-up technique of section 4.5, we can reach the fixed point faster. Figures 4.2, 4.3, and 4.4, respectively, show the convergence speed of the label probabilities for object 14 having label 7, object 14 having label \*, and object 9 having label 2. The probabilities at the fixed point for object 14 having label 7 is 0.0, object 14 having label \* is 1.0, and object 9 having label 2 is 1.0. For all values of s experimented here, no false convergence was obtained. This is because for a given s, the first s iterations were computed using formula (4.1).

If in all the iterations, we use formula (4.3), we will be amplifying probabilities that have not stabilized yet, and this will force the label probabilities converge to false values. Tables, 4.10, 4.11, and 4.12 show the label probabilities for iterations 2, 5, and 10, respectively, when s=5 and formula (4.3) is used in all iterations. Note that labels of objects 1, 9, 12, and 14 are amplified wrongly from the very initial iterations. Once enough number of labels are wrongly amplified, the label probabilities may converge to a different fixed point. This is what has happened to the probabilities of objects 1, 9, 12, and 14, that are converging in a wrong direction.

Assuming given initial label probabilities and the knowledge matrix for two sets of objects with cardinality m and n (m=number of regions in image 2; n=number of regions in image 1) the computation time needed to determine the correspondence between objects from the two sets consists of,

- 1. Computation of neighbor contribution factors, which involve computation of cross-correlation coefficient between two sets of m numbers. This computation is of order  $m^2$ . The neighbor contribution factor should be computed for all the m objects and for n labels of an object. Overall, the neighbor contribution factor should be computed mn times. So, the time required for computation of neighbor contribution factors in each iteration is of order  $m^3n^2$ .
- 2. Computation of formula (4.1) or (4.3), which is negligible compared to the computation of the neighbor contribution factors.
- 3. The number of iterations required for convergence depends on the number of labels, n. Assuming an order of n iterations is needed to determine the region correspondence problem, then the overall computation time becomes of order  $m^3n^3$ .

It takes about 10 minutes to determine the correspondence between regions in Figures 4.1.a and 4.1.b, on a PDP 11/34 computer. The order of the computation time shows that the speed of this procedure falls sharply as m and/or n increase.

When m and n are large, we may selectively take subsets of the data and work on the subsets. By determining point correspondence in the subsets, we can determine the transformation parameters for mapping one set into another. The transformation parameters then can be used to determine the correspondence between the rest of the points. For example, using the transformation parameters, for every point in one set we determine its position in the other set. The closest point to it, with distance smaller than a threshold value, can be taken as its corresponding point. Subsets could be points falling on the convex-hull of the sets.

Note that by selecting subsets, we may lock into a wrong correspondence and miss the correct correspondence.



(a)



(b) Figure 4.1. Two segmented images of the same scene.

Table 4.1. Initial label probabilities.

THE INITIAL LABEL PROBABILITIES: 3 4 5 7 8 9 10 11 12 2 13 14 1 6 1 0.07 0.08 0.07 0.05 0.06 0.08 0.08 0.08 0.08 0.05 0.06 0.06 0.06 0.07 0.04 2 0.07 0.06 0.07 0.09 0.07 0.06 0.06 0.07 0.07 0.09 0.07 0.06 0.07 0.08 0.02 3 0.06 0.08 0.08 0.04 0.06 0.08 0.09 0.08 0.05 0.05 0.09 0.07 0.08 0.06 0.04 4 0.05 0.08 0.07 0.05 0.07 0.08 0.08 0.10 0.06 0.05 0.07 0.07 0.07 0.06 0.04 6 0.09 0.07 0.07 0.06 0.07 0.07 0.07 0.08 0.06 0.07 0.06 0.06 0.07 0.09 0.02 7 0.06 0.08 0.07 0.06 0.07 0.07 0.08 0.09 0.06 0.06 0.06 0.07 0.08 0.07 0.03 8 0.09 0.07 0.08 0.06 0.07 0.07 0.08 0.08 0.05 0.06 0.06 0.06 0.07 0.08 0.02 9 0.08 0.08 0.07 0.06 0.07 0.07 0.08 0.08 0.05 0.06 0.05 0.06 0.07 0.08 0.03 10 0.08 0.07 0.09 0.06 0.07 0.07 0.08 0.08 0.05 0.06 0.06 0.06 0.09 0 02 11 0.07 0.06 0.07 0.08 0.06 0.05 0.07 0.07 0.07 0.08 0.07 0.05 0.08 0.08 0 02 12 0.06 0.06 0.05 0.09 0.09 0.06 0.06 0.08 0.08 0.09 0.05 0.07 0.07 0.07 0.02 **13** 0.07 0.06 0.06 0.08 0.09 0.07 0.09 0.05 0.07 0.05 0.07 0.07 0.08 0.03 14 0.08 0.08 0.08 0.05 0.06 0.07 0.09 0.08 0.05 0.05 0.07 0.06 0.07 0.09 0.02

Tal The	5] 2	.e K	n	4 01	.2 w]	2. le	d	g	e	M	la	t	r	ix.
14	91.96	49.10	53.94	71.19	56.32	94.77	150.24	210.61	198.50	178.62	169 24	148.55	34.62	0.00
13	57.44	41.91	61.50	85.33	79.22	101.14	149.31	215.13	172.69	155.32	141.58	121.32	0.00	34.62
42	83.25	105.21	122.22	138.72	154.62	124.75	121.67	185.32	51.77	40.71	20.73	0.0	121.32	148.56
:	100.41	125.83	142.26	157.85	17.4.54	142.18	132.50	192.32	33.43	32.22	0.00	20.73	141.58	169.24
t 0	123.19	131.38	142.79	153.80	173.57	133.34	111.38	165.48	26.03	0.00	32.22	40.71	155.32	178.62
თ	133.76	152.77	156.16	178.53	197.55	158.93	136.78	187.94	0.00	26.03	33.43	51.77	172.69	198.50
¢	228.98	173.29	157.37	140.72	162.04	116.01	69.49	0.00	187.94	165.48	192.32	185.32	215.19	210.61
٢	159.50	107.69	96.43	86.85	110.88	58.27	0.00	69.49	136.78	111.38	132.50	121.57	149.31	150.24
9	128.44	60.22	41.39	28.72	52.90	0.00	58.27	116.01	158.93	133.34	142.18	124.75	101.14	94.77
ល	126.97	54.84	32.53	24.20	0.00	52.90	110.88	162.04	197.55	173.57	174.54	154.62	79.22	56.32
4	124.22	50.27	25.08	0.00	24.20	28.72	86.85	140.72	178.53	153.80	157.85	138.72	85.33	71.19
х: х	99.48	25.38	0.0 0	25.08	32.53	41.39	96.43	157.37	166.16	142.79	142.26	122.22	61.50	53.94
GE MATRI 2	74.10	0.00	25.38	50.27	54.84	60.22	107.69	173.29	152.77	131.38	125.83	105.21	41.91	49.10
KNOWLED	0.00	74.10	99.48	124.22	126.97	128.44	159.50	228.98	133.76	123.19	100.41	83.25	57.44	91.96
THE	-	2	ო	4	ຍ	9	٢	∞	თ	9	Ŧ	12	Ę	4

Ta Ma	a b e a	) I S	e u:	4 re	<b>4</b> . еп	. 3 1e	• n	t	5	f	r	01	n	i	m	age	•	2.
		14	147 32	121 70	138 04	160 74	139 90	136 49	207.87	156.37	95,83	80.57	66.91	90.57	39, 17	0.00		
		61	154.89	129 83	137.43	125 77	104 01	97.63	197.31	129.44	60.85	42.05	28.70	52 95	00	39.17		
		10	193.02	169.90	169.25	101.91	79.83	57.92	213.85	125.47	53.31	30.95	24.28	c	52.95	90.57		
		11	173.98	149.92	152.53	109.56	87.03	73.51	203.90	123.65	49.51	24.29	00.00	24.28	28.70	66.91		
		0	163.22	140.65	138.59	86.15	63.77	55,99	183.74	99.78	25.49	00.0	24.29	30.95	42.05	80.57		
		σι	150.29	125,88	122.67	65.15	44.09	50.22	160.80	74.29	0.00	25.49	49.51	53.31	60.85	95.83		
		80	134.91	125.73	101.77	48.92	56.29	90.25	101.25	0.00	74.29	99.78	123.65	125.47	129.44	156.37		
		7	96.21	108.59	78.11	149.47	156.79	190.05	0.00	101.25	160.80	183.74	203.90	213.85	197.31	207.87		
		9	196.80	178.09	166.95	50.93	34.10	00.00	190.05	90.25	50.22	55.99	73.51	57.92	97.63	136.49		
ŝ		ى ئ	172.18	156.16	140.67	22.55	0.00	34.10	156.79	56.29	44.09	63.77	87.03	79.83	104.01	139.90		
	2:	4	177.76	164.42	145.12	0.00	22.55	50.93	149.47	48.92	65.15	86.15	109.56	101.91	125.77	160.74		
	M IMAGE	ო	33.14	31.87	0.00	145.12	140.67	166.95	78.11	101.77	122.67	138.59	152.53	169.25	137.43	138.04		
	ENTS FRO	0	25.67	0.0 0	31.87	164.42	156.16	178.09	108.59	125.73	129.88	140.65	149.92	169.80	129.83	121.70		
	MEASUREM	-	0 <sup>.00</sup>	25.67	33.14	177.76	172.18	196.80	96.21	134.91	150.29	163.22	173.98	193.02	154.89	147.32		
	THE		-	2	e	4	ß	9	7	œ	<b>6</b>	<b>0</b>	11	12	13	4		

Table 4.4. Label probabilities after iteration 1.

#### Table 4.5. Label probabilities after iteration 2.

LA	BEL P	PROBABILITIES AFTER INTERATION #					# 2	WITH S= 1							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	٠
1	0.06	0.02	0.01	0.01	0.00	0.02	0.07	0.08	0.23	0.13	0.17	0.17	0.02	0.01	0.01
2	0.05	0.01	0.01	0.01	0.01	0.02	<b>0</b> .05	0.06	0.18	0.25	0.18	0.14	0.02	0.01	0.00
3	0.06	0.02	0.01	0.00	0.00	0.01	0.06	0.08	0.14	0.13	0.26	0.18	0.02	0.01	0.01
4	0.09	0.12	0.03	0.05	0.10	0.06	0.02	0.03	0.01	0.01	0.02	0.03	0.14	0.12	0 12
5	0.11	0.13	0.13	0.07	0.10	0.06	0.02	0.02	0.01	0.01	0.02	0.02	0.17	0.15	0.00
6	0.10	0.11	0.12	0.09	0.12	<b>0</b> .08	0.02	0.02	0.01	0.01	0.01	0.02	0.13	0.17	0.00
7	0.05	0.02	0.01	0.01	0.01	0.01	0.06	0.17	0.11	0.10	0.12	0.11	0.03	0.02	0.17
8	0.20	0.10	0.08	0.04	0.06	0.03	0.01	0.03	0.03	0.03	0.04	0.05	0.15	0.13	0.03
9	0.09	0.14	0.12	0.09	0.12	0.09	0.03	0.02	0.01	0.01	0.01	0.02	0.11	0.13	0.00
10	0.07	0.10	0.16	0.10	0.12	0.11	<b>0</b> .05	0.02	0.01	0.01	0.01	0.02	0.08	0.13	0.00
11	0.06	0.09	0.13	0.15	0.11	0.09	0.06	0.03	0.01	0.02	0.02	0.02	0.09	0.11	0.00
12	0.05	0.10	0.09	0.16	0.16	0.10	0.04	0.03	0.02	0.02	0.01	0.02	0.09	0.10	0 00
13	0.05	0.09	0.10	0.10	0.13	0.17	0.08	0.05	0.02	0.03	0.02	0.03	0.06	0.09	0 01
14	0.05	0.09	0.10	0.07	0.08	0.13	0.15	0.06	0.03	0.03	0.04	0.04	0.05	0.07	0.02

Table 4.6. Label probabilities after iteration 5.

LAE	BEL	PF	OBAB	LITI	S AFT	TER IN	TERA	TION 4	¥ 5	WITH	+ S=	1				
		1	2	З	4	5	6	7	8	9	10	11	12	13	14	*
1	Ο.	00	0.00	0.00	0.00	0.00	<b>0</b> .00	0.01	0.01	0.33	0.19	0.24	0.22	0.00	0.00	0.00
2	ο.	00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.24	0.34	0.23	0.17	0.00	0.00	0.00
3	ο.	01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.19	0.17	0.37	0.26	0.00	0.00	0.00
4	ο.	10	0.14	0.07	0.03	0.06	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.16	0 20
5	ο.	10	0.16	0.12	0.05	0.08	0.02	<b>0</b> .CO	0.00	0.00	0 00	0.00	0.00	0.27	0.21	0.00
6	Ο.	06	0.15	0.13	0.08	0.13	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.24	0.00
7	Ο.	03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.13	0.18	0.18	0.00	0.00	0.32
8	Ο.	45	0.08	0.03	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.20	0.12	0.06
9	Ο.	04	0.20	0.16	0.10	0.14	0.08	0.01	0.00	0.00	0.00	0.00	0.00	0.12	0.15	0.00
10	Ο.	01	0.12	0.22	0.13	0.16	0.13	0.02	0.01	0.00	0.00	0.00	0.00	0.05	0.13	0.00
11	ο.	01	0.09	0.18	0.24	0.16	0.14	0.04	0.02	0.00	0.00	0.00	0.00	0.05	0.09	0.00
12	ο.	01	0.11	0.12	0.22	0.23	0.12	0.02	0.01	0.00	0.00	0.00	0.00	0.06	0.10	0.00
13	Ο.	00	0.07	0.12	0.14	0.15	0.29	0.03	0.05	0.00	0.00	0.00	0.00	0.02	0.05	0.01
14	Ο.	00	0.03	0.08	0.08	0.06	0.22	0.33	0.12	0.00	0.01	0.00	0.00	0.00	0.02	0.04

# Table 4.7. Label probabilities after iteration 10.

LAE	BEL	PF	OBAB	LITI	ES AFT	TER IN	TERA	TION -	۲ <b>1</b> 0	WITH	+ S=	1				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	•
1	0.0	00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.35	0.20	0.24	0.21	0.00	0.00	0.00
2	0.0	00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.26	0.37	0.22	0.15	0.00	0.00	0.00
3	0.0	ю	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19	0.16	0.39	0 27	0.00	0.00	0.00
4	0.0	28	0.11	0.03	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.31	0.16	0.29
5	0.0	26	0.16	0.07	0.02	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.39	0.26	0.00
6	0.0	21	0.16	0.12	0.05	0.12	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.34	0 00
7	0.0	21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.09	0.18	0.19	0.00	0.00	0 42
8	0.7	70	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.04	0 09
9	0.0	21	0.28	0.19	0.09	0.14	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.16	0.00
10	0.0	ю	0.12	0.29	0.16	0.19	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.09	0.00
11	0.0	00	0.06	0.20	0.32	0.18	0.17	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.04	0.00
12	0.0	ю	0.08	0.14	0.27	0.29	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.07	0.00
13	0.0	00	0.03	0.10	0.16	0.13	0.44	0.07	0.04	0.00	0.00	0.00	0.00	0.00	0.01	0 02
14	0.0	00	0.00	0.02	0.03	0.02	0.19	0.49	0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.07

Table 4.8. Label probabilities after iteration 30.

LABEL PROBABILITIES AFTER INTERATION # 30 WITH S= 1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 14 0.00 0.00 0.00 0.00 0.00 0.03 0.63 0.21 0.00 0.00 0.00 0.00 0.00 0.00 0.12

Table 4.9. Label probabilities after iteration 100.

LABEL PROBABILITIES AFTER INTERATION # 100 WITH S= 1 2 3 4 5 6 7 8 9 10 11 12 13 14 







Figure 4.3. Probabilities of object 14 having label \*.



Table 4.10. Label probabilities after iteration 2, s=5.

LABEL PROBABILITIES AFTER INTERATION # 2 WITH S= 5 9 0.01 0.21 0.14 0.07 0.16 0.02 0.00 0.00 0.00 0.00 0.00 0.00 0.16 0.24 0.00 11 0.00 0.08 0.21 0.31 0.19 0.13 0.01 0.00 0.00 0.00 0.00 0.00 0.02 0.05 0.00 

Table 4.11. Label probabilities after iteration 5, s=5.

LABEL PROBABILITIES AFTER INTERATION # 5 WITH S= 5 1 2 4 5 6 9 10  $11 \ 0.00 \ 0.03 \ 0.24 \ 0.42 \ 0.25 \ 0.05 \ 0.00 \ 0.$  Table 4.12. Label probabilities after iteration 10, s=5.

LABEL PROBABILITIES AFTER INTERATION # 10 WITH S= 4 5  $13 \ 0.00 \ 0.00 \ 0.02 \ 0.21 \ 0.03 \ 0.74 \ 0.00 \ 0.$ 

:

#### Chapter 5

#### REGISTRATION OF IMAGES FROM A TWO-DIMENSIONAL SCENE

### 5.1. Introduction

In this chapter, a two-dimensional scene is approximated by a three-dimensional scene where the heights of objects in the scene are small compared to the distances of objects to the camera. "Small enough" means that we would get nearly the same image if the scene was painted on a flat plane. Satellite images are good approximations of two-dimensional scene images.

One of the characteristics of these images is that it is always possible to find a single linear transformation function which can map one image into another. Two images that have been obtained from the same viewpoint of a flat scene do not have non-linear geometric differences.

The aim of this chapter is to design a procedure for automatic registration of two images that may have translational, rotational, and scaling differences. The technique should not be sensitive to noise and intensity difference between the images. It should be able to work on a broad class of images and give subpixel accuracy.

How accurate is subpixel accuracy? When a scene is scanned by a camera, there is a positional error from zero to half a pixel involved in the digitization process. So, on the average we expect every point in the scene to have 1/4 of a pixel positional error in the image, and unless the digital image is interpolated by a continuous plane, we would not be able to attain accuracy of better than 1/4 of a pixel, on average, no matter how accurate the registration process.

Registration of images that have small geometric differences, such as images of a three-dimensional scene taken from two different viewpoints, or images that have slight non-linear geometric distortions due to scanner's movement, atmospheric effects, scanner's nonlinearity, etc., will be considered in Chapter 6.

The proposed technique for registration of two-dimensional scene images consists of the following steps:

- Segmentation of the images and isolation of closed boundary regions from the images.
- 2. Determination of corresponding regions in the images.
- 3. Using centroids of corresponding regions as corresponding points and determining the registration parameters.

In the following, each step is discussed in detail.

## 5.2. Image Segmentation

Image segmentation is the process of dividing an image into regions whose interior points have nearly the same property. This task can be accomplished in two ways. One is by determining boundaries between homogeneous regions of different properties. Since the boundaries are places where there are sharp changes in property values, region boundaries can be found by an edge operator like the Roberts, the Sobel, or the Heuckle operator. Davis gives a review of image segmentation by this approach [Davis 75].

Another way to segment an image is by determining regions of homogeneous property. This can be done by thresholding the image at different levels and letting each region include those neighboring points which have property values between two given threshold values. A survey of image segmentation techniques by thresholding can be found in [Weszka 78]. Another way to obtain regions of homogeneous property in an image is by region growing. In this approach, neighboring homogeneous regions of similar property are joined, or nonhomogeneous regions are split into homogeneous ones to segment the images. A survey of image segmentation by region growing is given by Zucker [Zucker 76a].

Usually image segmentation involves one image and the task is dividing it into regions so that the regions best represent objects in the scene. In this section, we will address the problem of segmenting two images of the same scene. Consideration of some of the purposes of segmentation such as image registration, object tracking, or image analysis, leads to the conclusion that it is important that the resulting corresponding regions in the images be similar. So, we take as a requirement for segmentation of two images segmenting the images so that corresponding regions in the images are most similar. For example, in

registering two satellite images of the same scene, the goal would be to segment the images such that corresponding regions (fields, lakes, etc.) are most similar. It may not be important at this point to find regions which correspond very well to the fields on the ground, but it is important that the corresponding regions in the images are similar.

In the following technique for segmentation of two images of the same scene, one of the images will be called image 1 and the other image 2. In section 5.2.1, segmentation of image 1 is considered, and in section 5.2.2, the result of segmentation of image 1 is used to guide the segmentation of image 2. In section 5.2.3, corresponding regions in the two images are determined, and in section 5.2.4 necessary refinements are made by re-examining the information in image 2. The result of the proposed segmentation technique on satellite images is given in section 5.2.5.

The proposed image segmentation technique is designed to work on image pairs that may have translational, rotational, and scaling differences, but no geometric distortion is allowed. 5.2.1.Segmentation of Image 1

In segmenting an image, usually the objective is to extract as many object boundaries, as possible and as well as possible. Since object boundaries usually have high gradient values, we will determine a threshold value for segmentation of image 1 using information about hiah gradient pixels in the image. A threshold value which slices image 1 at high gradient values is favorable, because changing the threshold value slightly will not change the segmented regions much, and 50 а small error in determination of the threshold value of image 2 will not result in very different regions. However, noise pixels also have high gradient values and thresholding the image using information from noise pixels is disasterous. Even without noise, requiring only high gradient values is not sufficient. It may happen that only a small area of the 'image has very high gradient pixels. Determining the threshold value using those pixels will result only in extraction of that small area. Therefore, we should be using those gradient values which are both high in value and also numerous in the image.

After obtaining the gradient of image 1 we construct a histogram of the gradient image, to tell us how many pixels of a certain gradient are in the image. Figure 5.1 shows such a histogram and G(j) shows the number of pixels with gradient j.



Figure 5.1. Histogram of gradient image 1.

Now we define variable M(j) as below.

M(j)=jG(j)

Those pixels with large j (high gradient) and large G(j) (showing that a large number of them are available) will result in a large value of M(j). This tells us that we should be looking at high values of M(j) to determine the threshold value. How about the maximum value of M(j)? Figure 5.2 shows three different cases of M(j) depending on the contents of image 1.



Figure 5.2. Three different histograms of M(j).

Figure 5.2.a shows the case where the maximum value of M(j) corresponds to the high gradient pixels in the image (j1 is high), and determination of the threshold value using gradient j1 is proper. Figure 5.2.b shows the case where the maximum of M(j) does not correspond to high gradient edges (j2 is not high). But M(j) has another peak value at j1 and j1 is high and is proper for threshold estimation. Figure 5.2.c shows the case where M(j) has one peak at j2 which is low and is not proper for threshold estimation. Using these facts, the following algorithm is proposed for threshold estimation.

<u>Algorithm 5.1.</u> Estimation of threshold value for segmentation of a single image (image 1).

- Step 1. Obtain the gradient of image 1 and construct its histogram, G(j).
- Step 2. Compute M(j)=jG(j) for all j, and smooth M(j) to avoid detection of noisy peaks (we have smoothed in 5 neighborhoods, i.e. smoothed M(j)=(1/5)  $\sum_{k=2}^{2} M(j+k)$ ).
- Step 3. Determine the rightmost peak of M(j). Let the peak be at j=jl.

Step 4. Compute SUMG=∑G(j). If SUMG ≤ 5% of total image (total image=∑G(j)) then Threshold value = average intensity of pixels with gradient jl. Otherwise (when SUMG > 5% of total image), Threshold value = average intensity of pixels with gradient jl where here jl satisfies ∑G(j)=5% of total image.

Step 3 requires that M(j) have at least one peak value. M(j) will in fact have at least one peak because it starts at zero (when j=0) and returns to zero (when G(j)=0). The constants 5 neighborhoods and 5% which were used in Steps 2 and 4, respectively, were determined heuristically to best fit our set of imagery. For other images, this might need modification for best results.

#### 5.2.2. Segmentation of Image 2

We could segment image 2 the same way we segmented image 1. This approach has been taken by some authors [Clark 80, Price 79]. The problem with doing so, however is that after segmentation, there is no guarantee that the obtained corresponding regions in the two images are maximally similar. Since the similarity of corresponding regions in the two images is what we need, it makes sense to lead segmentation of image 2 in such a way that the obtained regions in image 2 are most similar to their corresponding ones in image 1.

To determine the threshold value for segmentation of image 2, we determine the gradient of image 2 and construct the gradient histogram. Figure 5.3 shows a gradient histogram for image 2. G'(j') shows the number of pixels that have gradient j'. If jl was the gradient value at which we determined the threshold value of image 1, then we determine the threshold value of image 2 using the pixels with gradient jl', where jl' is determined such that  $\sum_{i \in I} G'(j') = \sum_{i \in I} G(j)$ 



Figure 5.3. Histogram of gradient image 2.

Then the threshold value of image 2 = average intensity of pixels with gradient jl'. It turns out that, if the images cover exactly the same scene, are of the same scale, have no noise, and the intensity relation between the images is linear, then the corresponding regions in the two images are exactly the same. This is proved below.

Proposition 5.1. If two images of the same scene are available and the intensity of a point in one image, I(X,Y), is a linear function of the intensity of the same point in another image,

I'(x',y') = f(I(x,y)) = aI(x,y) + b (5.1)
and if a>0 (5.2)
the following assertion holds.
If j(x1,y1) > j(x2,y2) then j'(x1',y1') > j'(x2',y2')

where j and j' are gradient values in image 1 and image 2, respectively, and (x1,y1), (x2,y2) are two points from image 1 and (x1',y1') and (x2',y2') are corresponding points in image 2.

Proof: In image 1, if for two points (x1,y1) and (x2,y2) the relation

I(x1,y1)-I(x1+dx,y1+dy) > I(x2,y2)-I(x2+dx,y2+dy) (5.3) holds then by multiplying both sides by parameter a we get

aI(x1,y1)-aI(x1+dx,y1+dy)>aI(x2,y2)-aI(x2+dx,y2+dy)

or aI(x1,y1)+b-[aI(x1+dx,y1+dy)+b]>

aI(x2,y2)+b-[aI(x2+dx,y2+dy)+b]

using relation (5.1) we get

I'(xl',yl')-I'(xl'+dx,yl'+dy) >

I'(x2',y2')-I'(x2'+dx,y2'+dy) (5.4)

where dx and dy are small distances like one pixel. So, if (5.3) holds in image 1 then (5.4) will hold in image 2. Now if we rewrite relation j(x1,y1) > j(x2,y2) in intensity forms, we get

 $\{ [I(x1,y1)-I(x1+dx,y1)]^2 + [I(x1,y1)-I(x1,y1+dy)]^2 \}^{0.5} > \\ \{ [I(x2,y2)-I(x2+dx,y2)]^2 + [I(x2,y2)-I(x2,y2+dy)]^2 \}^{0.5}$ 

Using relation (5.4) we get

{[I'(xl',yl')-I'(xl'+dx,yl')]<sup>2</sup> + [I'(xl',yl')-I'(xl',yl'+dy)]<sup>2</sup>}<sup>0.5</sup> >
{[I'(x2',y2')-I'(x2'+dx,y2')]<sup>2</sup> +  $[I'(x2',y2')-I'(x2',y2'+dy)]^2\}^{0.5}$ This means that j'(xl',yl') > j'(x2',y2'). #

Using the result of Proposition 5.1, we conclude that if

 $\sum_{j \neq i} \mathbf{G}(j) = \sum_{j \neq i} \mathbf{G}'(j')$ (5.5)

then all pixels in image 1 with gradient larger than j1 correspond to all pixels in image 2 with gradient larger than j1'. Now, slicing image 2 at the average intensity of pixels with gradient j1' will produce regions which cover the same areas of the scene as when slicing image 1 at the average intensity of pixels with gradient j1. This is proved below.

Proposition 5.2. When slicing image 1 at a level equal to the average intensity of pixels with gradient j1, and slicing image 2 at a level equal to the average intensity of pixels with gradient j1' such that  $\sum_{j>0} G(j) = \sum_{j>0} G'(j')$ , the regions which are obtained by segmenting the two images cover the same parts of the scene.

Proof: The threshold value for image 1 is equal to

 $h = \sum_{(ij)} I(x,y)/G(j1)$ 

where  $\sum_{i=1}^{n}$  means the sum over all pixels with gradient jl.

The average intensity of the same pixels in image 2 is equal to

 $\sum_{(1)} f(I(x,y))/G(j1)$  (5.6)

If image intensities could assume continuous values we would expect G(j1)=G'(j1'). However, due to digital rounding, all the pixels in image 1 with gradient j1 do not have gradient j1' in image 2. G'(j1') is the number of pixels in image 2 that have gradient j1'. The average intensity of these pixels is

$$\sum_{(j')} f(I(x,y))/G(jl') = \sum_{(j')} I'(x',y')/G(jl') = h' \quad (5.7)$$

(5.6) and (5.7) are close but may not be equal due to the digital rounding errors (in section 5.2.4 we will see how this kind of error can be compensated by region refinements).

Now, regions in image 1 are constituted of pixels which have intensity greater than h or all pixel such that  $I(x,y) > h = \sum_{(j1)} I(x,y)/G(j1)$  (5.8) all these pixels in image 2 will have intensity

 $f(I(x,y)) > f(h) = \sum_{(j)} f(I(x,y))/G(j1)$ Approximating this by (5.7) we get

I'(x',y') > h' (5.9)

So, we see that the same pixels which satisfy (5.8) in image 1, satisfy (5.9) in image 2. And so thresholding image 2 at h' will produce the same regions as when thresholding image 1 at h. #
What if the images do not cover exactly the same scene, there is noise in one or both of the images, or they are not of the same scale, or the intensity difference between the images is not linear? One or more of these conditions may hold for any pair of natural images. If so, the regions obtained from segmentation of the two images will have differences. How can we refine the regions in image 2 so that they become most similar to their corresponding ones in image 1?

In section 5.2.4, it will be shown how this goal can be reached by going back to the original image 2 and using more of the information that it contains.

### 5.2.3. Region Correspondence

When the two images are sequented into regions, the next task is to find corresponding regions in two images. In [Price 79], to find the region in image 1 which corresponds to a given region in image 2, the region of image 2 is compared to every region in image 1. The best match is considered to be the corresponding region.

The match rating procedure used for this purpose incorporates the weighted difference between all features of the region. Features include, area size. roundness (Perimeter<sup>2</sup>/Area), length to width ratio, and color. Note that in this technique, even though a region may not have a corresponding region in the other image, some region will be the corresponding region. selected as Only local information about regions are used to find region correspondence in this technique. The time needed to extract features from the segmented images and find region correspondence for two images with 10 regions each was 5.3 min on a PDP 10 computer [Price 77].

If a region is represented by a point, such as its centroid, Zahn has given a technique to determine point correspondences using interpoint distances (global information) [Zahn 74]. In this technique, the minimum spanning tree (MST) for each set of points is determined. Then the degree, the minimum angle, and the ratio of edges that make the angle are determined for each point. Starting with the point of highest degree and decreasing, a search is carried out in the other set for a node with similar minimum angle and associated length ratio. If the match rating of the best match is lower than a threshold value, it is concluded that the node in the first set does not exist in

the other set. Otherwise, best match nodes are marked as corresponding nodes. The time needed to find correspondence between two sets each with 10 points is reported to be 0.6 sec on a CDC 6400 computer [Zahn 74].

Another procedure which uses inter-point distances information) to match two sets of data points is (qlobal based on the notion of clustering and is given by Stockman [Stockman 83]. In this procedure, a graph for each data set is created by joining pairs of points selected by local information, then matching is carried out between all possible edges in the two graphs. While matching edges, the translational, rotational, and scaling difference between edges with similar endpoints are determined and a point is entered into the parameter space showing the parameter values. Correct matches tend to make a dense cluster while mismatches randomly fill the space. The parameter values corresponding to the densest cluster center are taken as the transformation parameters. Knowing the transformation into another. In this parameters, we can map one set mapping, points falling within a threshold value of the transformed points are taken as corresponding points. The points having no remaining points labeled as are correspondence.

The time required to match two sets with 10 points each takes about 20 seconds on a Harris model H500 computer. When the number of points in each set becomes large, this process becomes very time-consuming, but can be speeded up by selectively choosing subsets from each set and matching the subsets. For example, points falling on the convex hull of each set can be taken as the subset for matching.

The convex-hull of a set does not vary by translating, rotating, or scaling the set. This property can be used to characterize a point set. So, rather than taking all points in the set, we may take only points on the convex-hull for matching. Note that by doing so, however, depending on the data sets, we may miss the correct match. As the number of points in the set increases, the probability of locking onto a wrong match decreases, and using the points on the convex-hull becomes reliable and is preferable to an exhaustive search.

A procedure which uses global information as well as local information in the images to determine corresponding regions has been developed in Chapter 4. The procedure is based on the probabilistic relaxation labeling process of Rosenfeld [Rosenfeld 76]. In this process, shape similarities of regions (local information) are used to determine the initial label probabilities. Then distances

97

between the centroids of regions (global information), are used to update the label probabilities. The time required to find correspondence between two sets of 10 regions each is about 2 min on a PDP 11/34 computer. Since the computational complexity of this procedure is of order  $m^3n^3$  where m and n are a number of regions in image 2 and image 1, respectively, the speed of the process drops sharply as m and n increase. When m and n are large (larger than 30), we may select subsets of the data that, for example, fall on the convex-hull of the sets and work on the subsets. When correspondence between the subsets are found, we can determine the transformation parameters between the two sets. Knowing the transformation parameters, we can map one set into another and determine the rest of the correspondences.

In this chapter, the procedure of Chapter 4 will be used to determine the region correpondences.

#### 5.2.4. Region Refinements

Segmentation of image 2 might have produced regions which are not optimally similar by any measure to their corresponding ones in image 1 due to various differences between the two images. For example, it might be that one threshold value is not capable of isolating every object in image 2 satisfactorily. In this section we will refine the regions in image 2 such that they become optimally similar by an appropriate measure to their corresponding ones in image 1. We consider first a simple case and then a general case.

After determining corresponding regions in the two images using the method of Chapter 4, each region in image 2 is refined to make it optimally similar to its corresponding region in image 1, as shown below.

<u>A. The simple case:</u> The two images have only translational differences.

Suppose region A and region B are two corresponding regions from image 1 and image 2, respectively. Let WA and WB be two windows each with size mxn, which contain regions A and B individually and their centers coincide with the centroids of the regions. We let a pixel in WA or WB have value 1 if the pixel belongs to the region and have value 0 if the pixel is outside the region. We lay WA over WB such that their centers overlay and determine their Exclusive-OR. Suppose WC is the window such that WC=WA@WB, then WC can be used to show the similarity between region A and region B. If the two regions are similar, we will have zero or a small number of pixels with value 1 in WC. If the two regions are dissimilar we will have a large number of pixels with value 1 in WC. The similarity between regions A and B is defined below.

<u>Def. 5.1.</u> Similarity of regions A and B that belong to two images with only translational difference is measured by

 $r_{AB}=1-[\sum_{x}\sum_{y}WC(x,y)]/mn$ where mxn is the window size which can contain regions A and B, individually, and WC is the third window obtained by Exclusive-ORing the windows containing regions A and B. # This similarity measure is easy to implement, fast to compute, and fulfills our needs.

The following algorithm finds the optimal threshold value for image 2 so that when sliced at that value, region B is optimally similar to region A. Essentially this algorithm is a peak seeking algorithm specialized to the discrete values given by pixel count measures.

<u>Algorithm 5.2.</u> In the following, h and h', respectively show the present and the previously estimated threshold value for segmentation of image 2. Initially, h' is the threshold value estimated in section 5.2.2. r and r' show the similarity of regions A and B when image 2 is sliced at h and h', respectively. SLOPE shows increase when it is positive and decrease when it is negative for the threshold value, and k shows the number of iterations.

- Step 0. h<--h'; k<--0; SLOPE<--positive; Compute WC; Compute
  r<sub>AB</sub>; r'<sub>AB</sub><--r<sub>AB</sub>; Continue.
- Step 1. If SLOPE>0 then h<--h+1 otherwise h<--h-1; Continue.
- Step 2. k<--k+1; Slice image 2 at h; region B<--new region
  B; Compute WC; Compute r<sub>kB</sub>; Continue.
- Step 3. If  $r_{AB}>r'_{AB}$  then set  $r'_{AB}<-r_{AB}$  and go to step 1 otherwise continue.
- Step 4. If k=1 then go to Step 5 otherwise go to Step 6.
- Step 5. h<--h-1; SLOPE<--negative; r'AB<--rAB; Go to Step 1.
- Step 6. If SLOPE>0 then the optimal threshold value falls between h and h-1. The linear approximation to the optimal threshold value is  $H=h-r_{AB}/(r_{AB}+r'_{AB})$ ; Exit. Otherwise (when SLOPE<0) the optimal threshold value falls between h and h+1. The linear approximation to the optimal threshold value is  $H=h+r_{AB}/(r_{AB}+r'_{AB})$ ; Exit.

Step 0 initializes the parameters h, k, and SLOPE and computes  $r_{AB}$ , the similarity between regions A and B. At the beginning we let  $r'_{AB}=r_{AB}$ . If by increasing the threshold value,  $r_{AB}$  increases (showing the new B is more similar to A than the previous B), Steps 1, 2, and 3 are executed. This is continued until  $r_{AB}$  starts to decrease. This is detected by Step 3 and the optimal threshold value is estimated in Step 6 (see Figure 5.4.a).



Figure 5.4. Estimation of the optimal threshold value.

If the start point is of the form shown in Figure 5.4.b, then Steps 1, 2, and 3 are executed once, because by increasing the threshold value, a more dissimilar region is found ( $r_{AB}$  has decreased). Then in Step 5 the value of SLOPE is set to negative, showing the threshold value should be decreased. Steps 1, 2, and 3 are executed as long as  $r_{AB}$  keeps increasing. When  $r_{AB}$  starts to decrease, it shows that the optimal threshold value is passed. This is detected in Step 3 and the optimal threshold value is estimated in Step 6.

Assuming  $r_{AB}$  and  $r'_{AB}$  are similarity measures obtained for threshold values h and h-1, then the optimal threshold value is found by

 $H = h - r_{AB} / (r_{AB} + r'_{AB})$ 

This is because when  $r_{AB}>r'_{AB}$ , it means that thresholding image 2 at h creates a region B more similar to A than the region B obtained by thresholding image 2 at h-1. So, the optimal threshold value should be closer to h than to h-1, and vice versa when  $r_{AB}>r'_{AB}$ . Referring to Figure 5.5, by linear interpolation we can write

 $dr'_{AB} = (1-d)r_{AB}$  or  $d = r_{AB}/(r_{AB} + r'_{AB})$ where d is the distance of the optimal threshold value H from h.

So,  $H = h - d = h - r_{AB} / (r_{AB} + r'_{AB}).$ 



Figure 5.5. Computation of the optimal threshold value.

In the same fashion, when  $r_{AB}$  and  $r'_{AB}$  are similarity measures for threshold values h and h+1, the estimate to the optimal threshold value is found to be,

 $H = h + r_{AB} / (r_{AB} + r'_{AB})$ 

Proposition 5.3. Algorithm 5.2 always estimates the optimal threshold value in the sense that no other threshold value can segment image 2 so that the obtained region B is more similar to region A.

Proof: We know that by increasing the threshold value, the area of region B gets smaller, and by decreasing the threshold value, region B gets larger. If the initial threshold value is such that region B is larger than region A, by increasing the threshold value we will reach a point where region B becomes smaller than region A. And so there exists a point (and only one point) where A and B are optimally close (in other words, there exists a threshold value where  $r_{AB}$  is maximum). If the initial threshold value is such that region B is smaller than region A, by decreasing the threshold value, we will reach a point where region B becomes larger than region A. And again, there exists a threshold value such that regions A and B become optimally close.

The optimal threshold value results in the maximum of  $r_{AB}$ . This is detected in Steps 3 and 4, and there is no way to by-pass these steps. When the existence of the optimal threshold value is detected, its value is estimated in Step 6. The algorithm then terminates. So, Algorithm 5.1 estimates the optimal threshold value and halts, for all inputs. #

Figure 5.6 depicts the process of region refinement by Algorithm 5.2. For clarity, the window around region B of image 2 is shown in three dimensions where the third dimension shows the intensities. The intensities are sliced at different values and the slice most similar to that of region A is determined.

Note that by using Algorithm 5.2, it is possible to determine the location of boundary points of the optimal region up to subpixel accuracy. This is shown in Figure 5.7. Assuming c with coordinates (xc,yc) is a boundary



Figure 5.6. Image slices from Algorithm 5.5.

point which is obtained by slicing image 2 at h, and c' is the same point shifted to (xc',yc') when slicing image 2 at h-1, then, if the optimal threshold value H is between h and h-1, slicing image 2 at H would move point c to point c" (see Figure 5.7) with coordinates,



Figure 5.7. Determination of the optimal boundary points.

 $xc''=xc-(xc-xc')r_{AB}/(r_{AB}+r'_{AB})$  $yc''=yc-(yc-yc')r_{AB}/(r_{AB}+r'_{AB}).$ 

<u>B. The general case:</u> The two images have translational, rotational, and scaling differences.

The similarity measure which was used in Algorithm 5.2 can take care of images which only have translational differences. If the images have rotational and scaling differences also, we should use a similarity measure which can find the similarity of two regions irrespective of their rotational and scaling differences.

A procedure for this purpose has been developed in Chapter 3. In this procedure, similarity between two regions is measured irrespective of their rotational and scaling differences and is shown by a number between 0 and 1, with 1 showing complete similarity and 0 showing the similarity between a line and a circle. The similarity of other shapes falls between 0 and 1. If we use this similarity measure in Algorith 5.2 (as the r), we will be able to refine regions of image 2 even though the images have rotational and scaling differences.

### 5.2.5. Segmentation Results

To show the validity of the proposed segmentation technique, two pairs of images are used. Figure 5.8.a is a Heat Capacity Mapping Mission Day-Visible (HCMM Day-Vis) image acquired on 26 September 1979 from an area over Michigan. Figure 5.8.b is an HCMM Night-IR image of about the same area acquired on 4 July 78.

HCMM Day-Vis and Night-IR images were assumed to be images 1 and 2, respectively. Algorithm 5.1 was applied to the Day-Vis image and the segmentation result of Figure 5.8.c was obtained. The gradient operator used in the algorithm was the maximum difference operator because of its fast speed (see [Hall 79] pp 403). The Night-IR image was segmented as image 2 and the result is shown in Figure 5.8.d.

Although the images may seem to have been segmented very well, some of the extracted corresponding regions in the two images are not similar. This is because the two images are acquired by different sensors and the relation between the intensities of the two images is not linear. Figure 5.10.a shows the relation between the intensities of the two images.

108



(a)

(b)

(c)



(d)





(f)





Figure 5.8. Segmentation of Day-Vis and Night-IR images.







(a)

(ь)

(c)



(b)

(e)

(f)





Figure 5.9. Segmentation of MSS and TMS images.



(a) (b) Figure 5.10. Image 1 and image 2 intensity relations. (Specify the intensity of a point on the horizontal axis in one image and look up the (digitally rounded) intensity of the same point in the other image from the vertical axis)

Region correspondences were found using the procedure of Chapter 4. Using speed-up factor s=20, the process was terminated when the largest label probability for each object passed 0.95. Forty one iterations were required for doing so. All the probabilities passing 0.95 showed the correct labels with unique correspondence. For more detailed results see Chapter 4.

show how Algorithm 5.2 can refine То the region boundaries of the Night-IR image so that they become optimally similar to their corresponding ones in the Day-Vis image, first closed boundary regions are extracted from the images. These are shown in Figure 5.8.e and Figure 5.8.f. The extracted regions are limited to those which have (perimeter) sizes that are not too large or too small. Two threshold values were used for this purpose. The lower threshold value prevents very small regions prone to noise from being selected. The upper threshold value is used to avoid the selection of very large regions, which might be divided into two or more regions in the other image. The two threshold values that were used here are 10 pixels and 150 pixels.

Algorithm 5.2 with similarity measure of Chapter 3 was applied to regions shown in Figure 5.8.f. The resultant refined regions are shown in Figure 5.8.g. Note that there are some regions which are present in only one of the images because of scene differences or their small size. For these regions no refinement is possible.

As a second example, band 7 of a Landsat-2 Multi-Spectral Scanner (MSS) image acquired on 27 August 1980 from an area over Pontiac, Michigan and band 3 of a Thematic Mapper Simulation (TMS) image acquired on 19 August 1981 from the same area is used. The two images are in different scales. The MSS image has an 80x80 meter resolution while the TMS image has a 60x60 meter resolution. The original TMS image had 30x30 meter resolution, but we reduced the resolution of it to half, so that with the same image size it can cover about the same areas as the MSS image. Figures 5.9.a and 5.9.b show the MSS image and the TMS image, respectively. After applying the segmentation procedures of section 5.2.1 and 5.2.2 on the images, we get results of Figures 5.9.c and 5.9.d, respectively, for the MSS image and the TMS image. This pair of images had approximately linear intensity differences (see Figure 2.10.b) but were not of the same scale, which made the edge information in the two images different.

Region correspondences were determined using the procedure of Chapter 4 with speed-up factor s=20. Region correspondences were found after 57 iterations with no mismatches.

The closed boundary regions from the segmented TMS image were refined using Algorithm 5.2 with similarity measure of Chapter 3 to make them optimally similar to their corresponding regions in the MSS image. Figure 5.9.f and 5.9.g show the closed regions of the TMS image before refinement and after refinement, respectively. Our aim has been to segment the images for image registration purposes, but the same segmentation result might be needed for object detection [Selfridge 81] or in object tracking [Schalkoff 82]. An accurate segmentation result will enable better object boundaries for both shape recognition and object position computation.

Gradient operations, threshold estimation, image slicing and region refinements which are done by this technique are all operations of order N or less, where N is the number of pixels in the image, and so the technique is fast. Segmenting a pair of images of size 240x240 on a PDP 11/34 computer takes about 3 minutes. Of this time about 1 minute is used to segment image 1 and about 2 minutes is used to segment image 2 and refine it. This technique is capable of determining the boundaries of corresponding regions in the images up to subpixel accuracy which might be needed in image registration, object tracking or for other purposes.

As a final note, image segmentation was defined as a process which partitions an image into homogeneous regions. Does the above segmentation technique partition the images into homogeneous regions? As a matter of fact, it may not. This is because each image is sliced at one value, and if the image contains more than two types of regions, then we will obtain regions where each may contain several homogeneous subregions.

Ohlander [Ohlander 78] has used a recursive procedure to segment images with more than two types of regions. This idea can be applied here too. For each segmented region pair, we may compute some measure of variance over the regions. If the variance is above a threshold value, we process the region pair as being two new images and apply the above image segmentation technique on them to split the regions into homogeneous subregions. This process can be recursively applied until the variance of all regions falls below the threshold value. In the next chapter, an example which uses recursive image segmentation is given.

## 5.3. Selection of Control Points

<u>Def. 5.2.</u> Control Points: Points that are uniquely identifiable in an image. # Two images that have translational, rotational, and scaling differences can be registered if the coordinates of at least two pairs of corresponding control points from the two images are known.

115

In [Davis 78], a set of windows which contain a large number of high gradient, connected edges which are well dispersed over image 1 are selected. Then an attempt is made to search the location of these windows in image 2. The upper left hand window corners were selected as corresponding control points. Later in [Hall 80]. the criterion of uniqueness was added to the control point selection to make the determination of corresponding control points in image 2 more reliable. This control point selection technique, however, is limited to images that do not have rotational or scaling differences because window searches can be applied to images that only have translational differences. In [Kanal 81] line intersections in [Yam 81] high curvature points have been taken for and control points. These features can be selected in images have translational, rotational, and that scaling may differences. In the following, centroids of corresponding regions are used as corresponding control points. Once it is known that two regions in two images correspond to each other, their centroids will correspond to each other no matter what the translational, rotational, and scaling difference between the two images. Of course region similarity must be determined in a manner that is invariant to translation, rotation, and scaling. If regions are obtained by the image segmentation technique of section 5.2,

116

then it would be possible to determine the location of corresponding centroids up to subpixel accuracy.

If o with coordinates (xo,yo) and o' with coordinates (xo',yo') are centroids of a region when sliced at h and h-l, respectively, and if the optimal threshold value H is known to be between h and h-l (see Figure 5.11), then the coordinates of the optimal centroid (Ox,Oy) are determined by linear interpolation,

$$xO = xo - (xo-xo')r_{AB}/(r_{AB}+r'_{AB})$$
  
$$yO = yo - (yo-yo')r_{AB}/(r_{AB}+r'_{AB})$$

where  $r_{AB}$  and  $r'_{AB}$  are similarity measures of region A and region B when image 2 is sliced at h and h-l, respectively.



Figure 5.11. Computation of the optimal centroid.

A nice property of the centroid is its stability to random noise. Random noise will affect the region boundaries, but since the coordinates of a centroid are computed by averaging the coordinates of points on the boundary of the region, random noise should mostly cancel out and the effect on the centroid is considerably smaller.

To find out how realistic this assertion is, the following experiment was carried out. A 64x64 window was taken from the image of Figure 5.8.a. This is shown in Figure 5.12.a. This window was thresholded at (gray value 84) the level at which the image of Figure 5.8.a was thresholded. The segmented window is shown in Figure 5.13.a. Standard deviation of intensity values that fell above the threshold value (the threshold value that was used to segment the window) was found to be  $\sigma$ 1=22.0 and standard deviation of values below the threshold value  $\sigma$ 2=10.0.

Random noise was generated from a uniform distribution over (-0.5,0.5), was amplified by a factor of  $\sigma$ l=22.0 and was added to regions in the window with intensities above the threshold value. Noise was amplified by a factor of  $\sigma$ 2=10.0 and was added to regions with intensities equal to or below the threshold value. This is shown in Figure 5.12.b. The same amplitude noise is not generated for all areas of the window because in real life images, noise tends to vary with the signal level (intensities).



Figure 5.12. Windows with different noise levels.

Noise amplitude was increased by factors of 2, 3, 5, and 10, and was added to the window of 5.12.a. Windows of 5.12.c, d, e, and f were obtained, respectively. The windows were thresholded at gray value 84, the same level at which we thresholded the window of Figure 5.12.a. The segmentation results are shown in Figure 5.13.b, c, d, e, and f. Table 5.1 shows the coordinates of the centroid of the largest object in the window for each case. As we can see, the coordinates of the centroid are very stable under random noise, unless the amplitude of noise becomes so large



Figure 5.13. Windows of Figure 5.12 thresholded at 84.

that the object collapses. This is what has happened to the object of Figure 5.13.f.

# 5.4. Determination of Transformation Function Parameters

When two images of the same scene have translational, rotational, and scaling differences, corresponding points in the two images can be related to each other by the transformation of Cartesian coordinate systems, Table 5.1. Coordinates of centroids versus noise. Noise was added to points above and below the threshold value proportional to  $\sigma$ l and  $\sigma$ 2, respectively.

$$\sigma = [\sigma_1 \quad \sigma_2]^{t}$$

Figure	Noise Level	Coordinates
Figure 5.13.a	0	27.48, 18.88
Figure 5.13.b	σ	27.48, 18.88
Figure 5.13.c	20	27.53, 18.89
Figure 5.13.d	30	27.66, 18.86
Figure 5.13.e	5σ	27.99, 18.80
Figure 5.13.f	<b>1</b> 0 <i>σ</i>	29.85, 18.02

$$x' = m(x\cos\theta - y\sin\theta) + h$$
(5.10)
$$y' = m(x\sin\theta + y\cos\theta) + k$$

where (x,y), and (x',y') are coordinates of corresponding points in image 1 and image 2, respectively. m is the scaling factor,  $\theta$  is the rotational difference, and (h,k)are the translational differences of image 2 with respect to image 1. A minimum of 2 corresponding points are required to determine the parameters m,  $\theta$ , h, and k.

In section 5.2.4, a technique to estimate the optimal threshold value for each region of image 2 was given. The optimal threshold values guarantee that the best centroid for each region of image 2 is obtained. This doesn't mean that the centroids are determined without error. The optimal threshold values reduce the errors but they may not remove them completely. So, some of the errors may still exist with smaller magnitudes. The amount of error on the centroid of every region is not the same. Some of the centroids turn out to be more accurate than the others because the boundaries of some regions can be extracted more accurately than the others, depending on the region and its neighborhood.

Since we don't know which centroids are more accurate than the others, it may turn out that the two pairs of corresponding centroids that we choose for parameter estimation are among the less accurate ones. To downgrade the effect of inaccurate corresponding points on estimation averaging scheme like the of parameter values, an minimum-squared-error criterion seems appropriate. In this case more than two corresponding points are necessary to determine the transformation parameters. If n corresponding points from the two images are available, then m,  $\theta$ , h, and k can be estimated by minimizing the sum of squared errors,

$$E = \sum_{i=1}^{n} \{ [xi' - m(xicos\theta - yisin\theta) - h]^2 + [yi' - m(xisin\theta + yicos\theta) - k]^2 \}$$
(5.11)

To minimize E with respect to m,  $\theta$ , h, and k, we find partial derivatives of E with respect to m,  $\theta$ , h, and k, then set them equal to zero, and solve the system of linear equations for m,  $\theta$ , h, and k. To make this problem easier to solve, we replace mcos $\theta$  by a and msin $\theta$  by b. Then (5.11) becomes,

$$E = \sum_{i=1}^{n} \{ [xi' - (axi-byi+h)]^2 + [yi' - (bxi+ayi+k)]^2 \}$$

Now if we find partial derivatives of E with respect to a, b, h, and, k, and set them equal to zero this will lead to the following linear system of equations,

$$\begin{bmatrix} \Sigma(xi^{2}+yi^{2}) & 0 & \Sigma xi & \Sigma yi \\ 0 & \Sigma(xi^{2}+yi^{2}) - \Sigma yi & \Sigma xi \\ \Sigma xi & -\Sigma yi & n & 0 \\ \Sigma yi & \Sigma xi & 0 & n \end{bmatrix} \begin{bmatrix} a \\ b \\ h \\ k \end{bmatrix} = \begin{bmatrix} \sum_{i} (xi^{i}xi+yi^{i}yi) \\ \sum_{i} (yi^{i}xi-xi^{i}yi) \\ \sum_{i} (yi^{i}xi-xi^{i}yi) \\ \sum_{i} yi^{i} \end{bmatrix}$$
(5.12)

by which we can find the transformation function parameters.

### 5.5. Results

To measure the performance of the proposed image registration technique, two experiments were carried out with two sets of satellite images as follows.

## 5.5.1. Experiment 1

Images of Figures 5.8.a and 5.8.b have been used in experiment 1 and are shown in Figures 5.14.a and 5.14.b. These two images are our image 1 and image 2, respectively, and have been used for registration. Figures 5.15.a and 5.15.b show closed boundary regions that have been used in the registration process. The regions in each image are labeled so that we can refer to them by their labels. Table 5.2 shows the coordinates of centroids of regions in image 1, coordinates of centroids of regions in image 2 before refinement, and coordinates of centroids in image 2 after refinement. Note that only those regions in image 2 that have a correspondence in image 1 can be refined.

124



(a)



(b) Figure 5.14. The HCMM Day-Vis and Night-IR images.



(a)



(b) Figure 5.15. Closed boundary regions of images of 5.14.

Table 5.2. Centroids of HCMM Day-Vis and Night-IR images.

LabelImage 1Image 2Refined111.8,150.4104.6,10.2104.7,1284.2,164.9114.6,32.9117.3,33109.2,169.883.0,36.9484.2,354134.2,170.421.3,168.35130.6,194.343.5,172.045.3,176140.0,141.259.3,202.261.2, 27158.6,87.06.9,19.588223.1,61.323.4,119.425.2,11942.0,19.785.9,159.687.8,151061.6,35.9107.3,173.4109.2,11133.1,50.9130.0,182.1131.9,1	
111.8,150.4104.6,10.2104.7,1284.2,164.9114.6,32.9117.3,33109.2,169.883.0,36.9484.2,354134.2,170.421.3,168.35130.6,194.343.5,172.045.3,176140.0,141.259.3,202.261.2, 27158.6,87.06.9,19.588223.1,61.323.4,119.425.2,11942.0,19.785.9,159.687.8,151061.6,35.9107.3,173.4109.2,11133.1,50.9130.0,182.1131.9,1	
2       84.2,164.9       114.6,32.9       117.3,3         3       109.2,169.8       83.0,36.94       84.2,35         4       134.2,170.4       21.3,168.3         5       130.6,194.3       43.5,172.0       45.3,17         6       140.0,141.2       59.3,202.2       61.2, 2         7       158.6,87.0       6.9,19.5       8         8       223.1,61.3       23.4,119.4       25.2,11         9       42.0,19.7       85.9,159.6       87.8,15         10       61.6,35.9       107.3,173.4       109.2,1         11       33.1,50.9       130.0,182.1       131.9,1	0.1
3       109.2,169.8       83.0,36.94       84.2,35         4       134.2,170.4       21.3,168.3       45.3,17         5       130.6,194.3       43.5,172.0       45.3,17         6       140.0,141.2       59.3,202.2       61.2, 2         7       158.6,87.0       6.9,19.5       8         8       223.1,61.3       23.4,119.4       25.2,11         9       42.0,19.7       85.9,159.6       87.8,15         10       61.6,35.9       107.3,173.4       109.2,1         11       33.1,50.9       130.0,182.1       131.9,1	1.8
4       134.2,170.4       21.3,168.3         5       130.6,194.3       43.5,172.0       45.3,17         6       140.0,141.2       59.3,202.2       61.2,2         7       158.6,87.0       6.9,19.5       23.4,119.4       25.2,11         9       42.0,19.7       85.9,159.6       87.8,15         10       61.6,35.9       107.3,173.4       109.2,1         11       33.1,50.9       130.0,182.1       131.9,1	.4
5       130.6,194.3       43.5,172.0       45.3,17         6       140.0,141.2       59.3,202.2       61.2,2         7       158.6,87.0       6.9,19.5       61.2,2         8       223.1,61.3       23.4,119.4       25.2,11         9       42.0,19.7       85.9,159.6       87.8,15         10       61.6,35.9       107.3,173.4       109.2,1         11       33.1,50.9       130.0,182.1       131.9,1	
6140.0,141.259.3,202.261.2, 27158.6,87.06.9,19.58223.1,61.323.4,119.425.2,11942.0,19.785.9,159.687.8,151061.6,35.9107.3,173.4109.2,11133.1,50.9130.0,182.1131.9,1	2.0
7158.6,87.06.9,19.58223.1,61.323.4,119.425.2,11942.0,19.785.9,159.687.8,151061.6,35.9107.3,173.4109.2,11133.1,50.9130.0,182.1131.9,1	02.0
8223.1,61.323.4,119.425.2,11942.0,19.785.9,159.687.8,151061.6,35.9107.3,173.4109.2,11133.1,50.9130.0,182.1131.9,1	
942.0,19.785.9,159.687.8,151061.6,35.9107.3,173.4109.2,11133.1,50.9130.0,182.1131.9,1	9.5
1061.6,35.9107.3,173.4109.2,11133.1,50.9130.0,182.1131.9,1	9.6
11 33.1,50.9 130.0,182.1 131.9,1	72.8
	81.9
12 38.0.70.4 117.2.202.7 118.8.2	03.0
13 50.4.192.0 146.7.158.7 148.9.1	58.4
14 75.7,214.6 178.8,136.3	

There are 11 regions which exist in both of the images. The centroids of these 11 regions from the two images are used in the linear system of (5.12) and the parameters are determined to be a=0.925, b=0.381, h=-58.0, k=50.9. From a=mcos0 and b=msin0, we can also find the scaling factor m=1.000 and the rotational difference  $\theta$ =5.892 radians between the images. Applying these parameter values to equations of (5.10) we get

$$x' = 0.925x + 0.381y - 58.0$$
 (5.13)  
 $y' = -0.381x + 0.925y + 50.9$ 

where (x,y) and (x',y') are coordinates of corresponding

points in image 1 and image 2, respectively. These equations provide the mapping from one image to another. So, given the coordinates of a point in one image, we can determine the coordinates of the same point in another image using this transformation function.

Using the transformation formulas of (5.13) we resampled the image of Figure 5.14.b to register with the image of Figure 5.14.a (see Figure 5.16.a). Nearest neighbor resampling technique was used because of its fast speed and simplicity of programming. The resampled image 2 was negated and overlaid with image 1 for clarity. This is shown in Figure 5.16.b.

To measure the accuracy of this technique, the square-root-error for each of the ll pairs of corresponding points in the two images were determined using,

$$Ei = \{ [xi' - (0.925xi+0.381yi-58.0)]^2 + [yi' - (-0.381xi+0.925yi+50.9)]^2 \}^{0.5}$$

The square-root-errors are shown in the rightmost column of Table 5.3. Note that most of the errors are less than 1 pixel and the rest are between 1 and 2 pixels.





(b) Figure 5.16. Resampled and overlaid images.
To find out how much improvement is made by refining the regions, the computation of square-root-errors was repeated, this time using coordinates of centroids before the refinement. In this process, transformation parameters m=1.001,  $\theta=5.897$ , h=-55.6, and k=49.4 were obtained giving square-root-errors shown in the third column of Table 5.3. Note that when the regions are not refined, most of the points have square-root-errors more than one pixel, while when the regions are refined, most of the points have errors less than a pixel.

Table 5.3. Square-root-errors for the ll points of data set.

Corresponding points		Square-root-errors		
Image 1	Image 2	Not Refined	Refined	
1	8	0.96	1.71	
2	9	1.29	0.92	
3	10	0.45	0.73	
4	11	1.77	1.72	
5	12	2.10	1.61	
6	13	0.24	0.58	
9	1	3.10	0.99	
10	2	1.51	1.06	
11	3	2.51	0.74	
13	5	1.98	1.18	
14	6	0.58	0.57	

### 5.5.2. Experiment 2

In another experiment, images of Figure 5.17a and 5.17b were used as image 1 and image 2 for registration. These images are the same as images of Figures 5.9.a and 5.9b. The images were segmented with the segmentation technique of section 5.2. Then the closed-boundary regions were isolated as shown in Figures 5.18.a and 5.18.b. The coordinates of centroids of regions in image 1 and image 2 are shown in Table 5.4. The coordinates of centroids of regions of image 2 after refinement are also shown in Table 5.4.

Using the centroids of corresponding regions as corresponding points, and using the min-squared-error criterion of section 5.4, transformation parameters are computed to be,

a=-0.0896; b=0.914; h=22.9; k=256.6; and m=0.918;  $\theta=4.615$ .

Using these transformation parameters, the image of Figure 5.17.b was resampled by the nearest neighbor technique to register with image of Figure 5.17.a. This is shown in Figure 5.19.a. The resampled image was overlaid with the image of Figure 5.17.a, as shown by Figure 5.19.b.



(a)



(b) Figure 5.17. The MSS and the TMS images.



(a)



(b) Figure 5.18. Closed-boundary regions of MSS and TMS images.

Table 5.	4.	Centroids	in	MSS	and	TMS	images.
----------	----	-----------	----	-----	-----	-----	---------

LabelImage 1Image 2Refined1 $17.6,44.9$ $44.6,23.9$ $44.4,23.7$ 2 $39.5,22.7$ $30.6,156.7$ $30.8,156.5$ 3 $18.4,127.0$ $18.2,212.4$ $4$ 4 $34.9,111.0$ $62.8,200.6$ $5$ 5 $57.0,104.4$ $81.1,215.0$ $6$ 6 $61.9,82.9$ $103.6,209.9$ $7$ 7 $105.6,120.2$ $81.5,183.4$ $81.5,183.2$ 8 $100.6,142.3$ $98.0,141.8$ $98.44,142.1$ 9 $117.2,62.7$ $116.0,91.2$ $116.4,91.2$ 10 $34.3,169.9$ $168.8,107.1$ $168.1,107.5$ 11 $44.6,205.3$ $197.1,148.2$ $197.1,148.2$ 12 $195.6,200.8$ $219.2,151.3$ $13$ 13 $212.0,147.2$ $167.9,199.4$ $168.9,199.2$ 14 $164.6,134.9$ $140.6,177.7$ $153.4,56.4$ $153.1,56.9$ 16 $162.0,30.2$ $134.1,49.6$ $134.5,49.7$ 17 $129.9,29.6$ $130.0,29.4$ $18$ 18 $141.5,25.6$ $19$ $116.6,12.4$ 20 $116.7,12.8$ $116.6,12.4$ 21 $103.9,17.9$ $211.6,37.5$ 23 $190.2,13.7$ $193.2,17.7$				Image 2
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	Label	Image 1	Image 2	Refined
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	1	17.6,44.9	44.6,23.9	44.4,23.7
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	2	39.5,22.7	30.6,156.7	30.8,156.5
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	3	18.4,127.0	18.2,212.4	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	4	34.9,111.0	62.8,200.6	
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	5	57.0.104.4	81.1.215.0	
7 $105.6,120.2$ $81.5,183.4$ $81.5,183.2$ 8 $100.6,142.3$ $98.0,141.8$ $98.44,142.1$ 9 $117.2,62.7$ $116.0,91.2$ $116.4,91.2$ 10 $34.3,169.9$ $168.8,107.1$ $168.1,107.5$ 11 $44.6,205.3$ $197.1,148.2$ $197.1,148.2$ 12 $195.6,200.8$ $219.2,151.3$ $168.9,199.2$ 13 $212.0,147.2$ $167.9,199.4$ $168.9,199.2$ 14 $164.6,134.9$ $140.6,177.7$ $153.1,56.9$ 15 $218.6,56.7$ $153.4,56.4$ $153.1,56.9$ 16 $162.0,30.2$ $134.1,49.6$ $134.5,49.7$ 17 $129.9,29.6$ $130.0,29.4$ 18 $141.5,25.6$ $19$ 19 $129.1,14.1$ $116.6,12.4$ 20 $116.7,12.8$ $116.6,12.4$ 21 $210.8,37.8$ $211.6,37.5$ 23 $190.2,13.7$ $193.2,17.7$	6	61.9.82.9	103.6.209.9	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	7	105.6.120.2	81.5.183.4	81.5.183.2
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	8	100.6.142.3	98.0.141.8	98.44.142.1
10 $34.3,169.9$ $168.8,107.1$ $168.1,107.5$ $11$ $44.6,205.3$ $197.1,148.2$ $197.1,148.2$ $12$ $195.6,200.8$ $219.2,151.3$ $13$ $212.0,147.2$ $167.9,199.4$ $168.9,199.2$ $14$ $164.6,134.9$ $140.6,177.7$ $15$ $218.6,56.7$ $153.4,56.4$ $153.1,56.9$ $16$ $162.0,30.2$ $134.1,49.6$ $134.5,49.7$ $17$ $129.9,29.6$ $130.0,29.4$ $18$ $141.5,25.6$ $19$ $19$ $129.1,14.1$ $116.6,12.4$ $20$ $116.7,12.8$ $116.6,12.4$ $21$ $210.8,37.8$ $211.6,37.5$ $23$ $190.2,13.7$ $193.2,17.7$	ğ	117 2 62 7	116 0 91 2	116 4 91 2
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	in	34 3 169 9	168 8 107 1	168 1 107 5
11 $14.6,205.3$ $197.1,146.2$ $197.1,146.2$ $12$ $195.6,200.8$ $219.2,151.3$ $13$ $212.0,147.2$ $167.9,199.4$ $168.9,199.2$ $14$ $164.6,134.9$ $140.6,177.7$ $15$ $218.6,56.7$ $153.4,56.4$ $153.1,56.9$ $16$ $162.0,30.2$ $134.1,49.6$ $134.5,49.7$ $17$ $129.9,29.6$ $130.0,29.4$ $18$ $141.5,25.6$ $19$ $129.1,14.1$ $20$ $116.7,12.8$ $116.6,12.4$ $21$ $103.9,17.9$ $22$ $210.8,37.8$ $211.6,37.5$ $23$ $190.2,13.7$ $193.2,17.7$ $111.2,210.6$ $111.2,210.6$	11		100.0,10/.1	100.1,10/.5
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	12	105 < 200.0	17/01/14002 210 2 151 2	19/01,140.2
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	12	195.6,200.6	219.2,151.3	160 0 100 0
14 $164.6, 134.9$ $140.6, 177.7$ $15$ $218.6, 56.7$ $153.4, 56.4$ $153.1, 56.9$ $16$ $162.0, 30.2$ $134.1, 49.6$ $134.5, 49.7$ $17$ $129.9, 29.6$ $130.0, 29.4$ $18$ $141.5, 25.6$ $19$ $19$ $129.1, 14.1$ $116.6, 12.4$ $21$ $103.9, 17.9$ $22$ $22$ $210.8, 37.8$ $211.6, 37.5$ $23$ $190.2, 13.7$ $193.2, 17.7$	13	212.0,14/.2	167.9,199.4	108.9,199.2
15       218.6,56.7       153.4,56.4       153.1,56.9         16       162.0,30.2       134.1,49.6       134.5,49.7         17       129.9,29.6       130.0,29.4         18       141.5,25.6       19         19       129.1,14.1       116.6,12.4         20       210.8,37.8       211.6,37.5         23       190.2,13.7       193.2,17.7         111.2,210.6       111.2,210.6	14	164.6,134.9	140.6,177.7	
16       162.0,30.2       134.1,49.6       134.5,49.7         17       129.9,29.6       130.0,29.4         18       141.5,25.6       19         19       129.1,14.1       116.6,12.4         20       116.7,12.8       116.6,12.4         21       103.9,17.9       211.6,37.5         23       190.2,13.7       193.2,17.7         111.2,210.6       111.2,210.6       111.2,210.6	15	218.6,56.7	153.4,56.4	153.1,56.9
17129.9,29.6130.0,29.418141.5,25.619129.1,14.120116.7,12.821103.9,17.922210.8,37.823190.2,13.7193.2,17.724111.2,210.6	16	162.0,30.2	134.1,49.6	134.5,49.7
18       141.5,25.6         19       129.1,14.1         20       116.7,12.8       116.6,12.4         21       103.9,17.9       22         22       210.8,37.8       211.6,37.5         23       190.2,13.7       193.2,17.7         24       111.2,210.6       111.2,210.6	17		129.9,29.6	130.0,29.4
19129.1,14.120116.7,12.8116.6,12.421103.9,17.92222210.8,37.8211.6,37.523190.2,13.7193.2,17.724111.2,210.6	. 18		141.5,25.6	
20116.7,12.8116.6,12.421103.9,17.922210.8,37.823190.2,13.724111.2,210.6	19		129.1,14.1	
21       103.9,17.9         22       210.8,37.8       211.6,37.5         23       190.2,13.7       193.2,17.7         24       111.2,210.6       111.2,210.6	20		116.7.12.8	116.6.12.4
22       210.8,37.8       211.6,37.5         23       190.2,13.7       193.2,17.7         24       111.2,210.6	21		103.9.17.9	
23 190.2,13.7 193.2,17.7	22		210.8.37.8	211.6.37.5
24 111.2.210.6	23		190.2.13.7	193.2.17.7
	24		111.2.210.6	

The images do not register especially well in the upper right part.

The reason for poor registration is that the TMS image (image 2) is geometrically distorted and a linear transformation function is not able to register the two images. The TMS image is a simulated Landsat-D image which actually has been obtained by an aircraft. The altitude of



(a)



(b) Figure 5.19. The TMS image resampled and overlaid.

the aircraft was 2750 meters with off-nadir look angles in the range of  $\pm$  29 degrees [Richard 78, Teillet 81]. If the aircraft changed its attitude even by 1 degree, pixels in the image borders would be shifted by 78 meters or 2.5 pixels.

The following polynomials of order 2 were used as transformation functions,

$$x'=a0+a1x+a2y+a3xy+a4x^{2}+a5y^{2}$$
  
y'=b0+b1x+b2y+b3xy+b4x<sup>2</sup>+b5y<sup>2</sup> (5.14)

12 unknown parameters are involved here which were estimated using the 14 known corresponding control points from the two images. Again by the min-squared-error criterion, the parameter values were estimated.

a0=229.9; a1=-0.028; a2=-0.887; a3=-0.00034; a4=0.00001; a5=0.0002; b0=6.7; b1=0.845; b2=0.025; b3=0.00000; b4=0.00020; b5=-0.00070.

The image of Figure 5.17.b was resampled using the polynomial transformation function of (5.14) with the above parameter values. In Figure 5.20, the resampled image is overlaid with the image of Figure 5.17.a. As can be seen, the registration result is improved. Table 5.5 shows the

square-root-error for the known 14 corresponding points in the two images, using the polynomial mapping function of (5.14) for both before and after refinement. The parameter values that were computed above were used for the case in which the regions are refined. Using the coordinates of region centroids before the refinements, the following parameter values were obtained.

a0=231.2; a1=-0.032; a2=-0.882; a3=-0.00038; a4=0.00005; a5=-0.00003; b0=6.1; b1=0.874; b2=0.027; b3=-0.00016; b4=0.00001; b5=0.00066;

Note the reduced square-root-errors for the case where the regions are refined in Table 5.5. The reason for obtaining larger errors for points near the borders of the image is that areas near image borders have larger geometric distortions. The registration accuracy can be improved further if the aircrafts altitude and attitude is known for the period the image is aquired. Then the TMS image can be corrected before the registration is attempted.

This example shows that the proposed region matching technique can be applied to registration of images that have small non-linear geometric distortions too. However, the subpixel accuracy can not be guaranteed any more because,



Figure 5.20. Registration by polymonial transformation.

- due to geometric distortions, the centroids of corresponding regions may not correspond to each other any more,
- depending on the amount of distortion, it might not be possible to obtain the large number of corresponding control points needed for accurate parameter estimation, and
- selection of the correct polynomial transformation function becomes very important and cannot be found easily. In [Leckie 80], a number of polynomials are

Table 5.5. Square-root-errors for the 14 points of data set.

Corresponding Points		Square-root-errors		
Image 1	Image 2	Not Refined	Refined	
1	23	7.57	5.33	
2	22	3.26	4.43	
3	20	1.73	1.56	
4	17	1.21	1.17	
5	16	1.02	0.97	
6	15	0.75	0.71	
7	9	1.16	0.47	
9	10	0.61	0.22	
11	1	3.04	3.41	
12	2	7.27	3.59	
13	7	7.35	2.16	
14	8	3.97	0.40	
15	13	4.51	1.22	
16	11	1.25	0.69	

tried experimentally and the one giving the smallest root-mean-square error is selected for registration. For images that have non-linear geometric distortions but don't have rotational or scaling differences, a more suitable registration technique is described in Chapter 6.

## 5.5.3. Summary of Results

Two sets of data were used in the experiments. In the first experiment, two images with translational, rotational, and intensity differences were used. The registration accuracy is summarized in Table 5.3. For most areas, subpixel accuracy was obtained. There are some areas, however, that have errors between 1 and 2 pixels. The errors could be due to the fact that the images have been obtained from slightly different viewpoints, causing small geometric distortion between the images.

In the second experiment, two images with translational, rotational, scaling, and small geometric distortions were used. The registration accuracy is shown in Table 5.5. Although subpixel accuracy was obtained in some areas of the image, in most parts, the errors were larger than a pixel. This shows that if the proposed technique is applied to registration of images with unknown geometric distortions, the subpixel accuracy cannot be guaranteed any more. The technique of the next chapter is more appropriate for registration of images with geometric distortions.

The computation time involved in registration of two images of size 240x240, (all images in this chapter are of size 240x240) includes times for,

- 1. Segmentation of the images, about 3 minutes,
- 2. Determination of region correspondences, about 10 minutes,

- Determination of transformation function parameters, about 0.2 minutes, and
- 4. Resampling of image 2, about 7 minutes, on a PDP 11/34 computer.

The resampling is time-consuming because equations of (5.10) or (5.14) must be evaluated N times, where N is the Determination of region number of pixels in image 1. correspondences also takes a great deal of time because mn label probabilities must be estimated in each iteration, where m and n are the number of regions in image 2 and image 1, respectively. Depending the initial label on probabilities and neighbor contribution factors, label probabilities will take a different amount of time to converge. To save some computation time, the relaxation process was stopped when the largest label probability for every object passed 0.95. At this point it is very unlikely that labels change. For regions in Figures 5.15.a and 5.15.b, 41 iterations and for regions in Figures 5.18.a and 5.18.b, 57 iterations with speed-up factor s=20 were needed to obtain the largest label probability for each object to pass 0.95.

## Chapter 6

# REGISTRATION OF IMAGES FROM A THREE-DIMENSIONAL SCENE

## 6.1. Introduction

In Chapter 5, registration of images from two-dimensional scenes was discussed. It was assumed that the distances of objects to the camera are much larger than the heights of objects in the scene, it could be assumed that the images do not have geometric differences. If the distances of some objects to the camera are not large compared to their sizes, then two images obtained at different viewpoints from the scene will have geometric differences.

In this chapter, registration of images with geometric differences will be discussed. It is assumed that the images are obtained at small viewpoint differences from the scene so that the amount of geometric difference between them is small. Also it is assumed that the images do not have rotational and scaling differences.

In this chapter, in section 6.2, the problem of determining the position of corresponding points in stereo images are discussed. In section 6.3, sources causing the mismatches in the window search process are identified and counter measures are given to avoid them. In section 6.4, the three-dimensional position of points in a scene are computed using the position of corresponding points in the images and the camera parameters. The results of the proposed approach on in-door and out-door scenes are shown in section 6.5. Finally in section 6.6 the validity of the results is discussed and some further suggestions for improvement are given.

## 6.2. The Correspondence Problem

Determination of corresponding points in stereo images has been studied by many groups. Marr-Poggio-Grimson have selected zero-crossings that are oriented non-horizontally elements for matching [Marr 79, Grimson 80]. as Zero-crossings are used to locate intensity changes in an image and are defined as the zeros of second directional derivatives of intensities in an image. Zero-crossings have been defined for masks of four sizes in [Marr 79]. Zero-crossings from larger masks are used to determine

global correspondences while zero-crossings from smaller masks are used to find local correspondences. Global correspondences, have been used to resolve ambiguities among local correspondences.

In other studies, Baker-Binford-Arnold have taken high gradient edges as elements for matching [Baker 81, Arnold 78], while Hannah-Moravec-Nevatia have taken windows in high information areas of images for matching [Hannah 74, Moravec 81, Nevatia 76]. The shortcoming of window searching is the need to identify low variance areas of an image before searching is carried out, since windows in low variance areas of an image do not provide reliable matching.

In the following, a technique is given which segments the images and uses windows centered at region boundaries as elements for matching. Using the image segmentation technique of section 5.2, the images are first segmented. Since the segmentation technique is based on edge information in the images, and since edges in stereo images mostly stay unchanged from one image to another, the obtained region boundaries in the two segmented images will mostly stay unchanged.

<u>Def. 6.1.</u> Baseline: A line connecting the lens center of camera 1 to the lens center of camera 2 (see Figure 6.1).

#

<u>Def. 6.2.</u> Match line: Any line in the images parallel to the baseline. If the images are obtained by a well balanced camera system, any row in the images is a match line (see Figure 6.2). #

If two balanced, equal focal length cameras are arranged with axes parallel (see Figure 6.1), then it can be assumed that they share a single common image plane. Any point in the scene will be projected into two points on the joint image plane. If we connect these two points, the obtained line will be parallel to the cameras' baseline. This shows that given a point in one of the images, we can determine its corresponding point in the other image by searching along the line passing through the given point and parallel to the baseline.

While following the same region boundary in the two images on a given match line, we first search for the position of the point on the boundary of the left image in the right image, search for the position of the point on the boundary of the right image in the left image. Among the two matches, the one with the higher match rating is taken as the true match. Points on region boundaries are searched twice, once in the left image and once in the right image. This is needed to avoid selection of occluded points.



image 2

Figure 6.1. A balanced stereo camera model.



Figure 6.2. Locating right image boundary in left image.

Figure 6.2 shows the search process for determining the position of points on the region boundary of the right image in the left image. Note that when following the boundary line, a search is needed only in a small neighborhood around the previous corresponding point (the search domain in Figure 6.2).

If a-priori knowledge about geometries of the scene and cameras are known, we can determine how large a search domain should be taken by simple triangulation. Figure 6.3 shows images of an object in two stereo cameras. Assuming point A on the object projects to points p and p' in the right and the left images, respectively, then if q is a neighbor of p in the right image, we should determine the domain of search in the left image so that the point corresponding to q is not missed.

Referencing Figure 6.3 we can write,

AB/BC=AF'/F'F=h/d

or BC=ABd/h=(h-h')d/h

Also BH/DG=HF/GF=h'/f

or DG=BHf/h'=df/h'

Also CH/EG=HF/GF=h'/f

and since CH=d-BC=d-(h-h')d/h

we can write EG=CHf/h'=[d-d(h-h')/h]f/h'



Figure 6.3. Stereo images of a three-dimensional scene.

Then 
$$DE=DG=EG=df/h'-[d-d(h-h')/h]f/h'$$
  
=(f/h')[d(h-h')/h] (6.1)

DE is the distance between points p' and q'. f and d are the parameters of the camera system, and h and h' depend on the characteristics of the scene such as depth, size, and shape of objects in the scene.

If the images are recorded on 35 mm films and then digitized with each row scan digitized to 512 values, we get

DE=0.185x512/35=2.7 pixels This shows that the point corresponding to q will be within 3 pixels from the point corresponding to p. So, if point p' in the left image corresponds to point p in the right image, and if q is a neighbor of p in the right image, then the point which corresponds to q can be determined by searching along the match line of q and 3 pixels from either side (of the column number) of point p'. Note that in this example we need to carry out only 7 window searches for determination of each corresponding point.

This determines the position of corresponding points on region boundaries. If objects in the scene are known to have planar surfaces, then the position of corresponding inside regions obtained points can be by linear interpolation of the position of points on the region boundaries. Assuming points A', B', C', and D' correspond to points A, B, C, and D, respectively, on the boundaries of two corresponding planes, then point E, which is at the intersection of lines AC and BD, has its correspondence in the other image at the intersection of lines A'C' and B'D' (see the Appendix for proof). So, to estimate the position of E', which corresponds to E, we first select A, B, C, and Then using their corresponding points A', B', C', and D' D. the other image, we determine the intersection of lines in A'C' and B'D', see Figure 6.4.

Note that by doing so, a great deal of window searches are



Figure 6.4. Estimation of match points inside regions.

avoided. This will result in considerable computational savings. Also, since window searching in homogeneous areas is unreliable, a number of possible mismatches are avoided.

If objects in the scene do not have planar surfaces but their geometric models are known, then by fitting appropriate geometric models to points on region boundaries, the position of the rest of points on an object can be recovered.

#### 6.3 Avoiding Mismatches

Window search is the key operation in determination of corresponding points in two stereo images. There are many sources causing mismatches during the search. The main sources are identified as:

1. Occluded points,

2. Geometric differences,

3. Homogeneous areas,

4. Window size, and

5. Similarity measure.

In the following, each source is discussed in detail and counter measures are proposed for avoiding it.

## 6.3.1. Occluded Points

<u>Def. 6.3.</u> Occluded points: Points which can be seen from only one of the cameras. #

If a window centered at an occluded point is taken from one image and is searched in the other image, a mismatch will be obtained, because the point does not exist there. Figure 6.5.b depicts this situation. Point S on the object can be seen from the left camera but it cannot be seen from the right camera. So, if a window centered at point S in the left image is taken and a search is carried out in the right image a mismatch will be obtained.



(a)
(b)
Figure 6.5. Image of an object formed by stereo cameras.
(a) Case where all feature points are visible in both images.
(b) Case where some feature points are occluded.

To avoid mismatches caused by occluded points, we use the following fact. Referencing Figure 6.5.b, if point S on the boundary of a region in the left image does not exist in the right image, then on the same match line, the point on the corresponding region in the right image (image of point T) can be seen in both of the images. So, instead of selecting all points from one image for searching, we select points from both images. On each match line, and for each corresponding region, we first take the window centered at the left image boundary and search it in the right image, then take the window centered at the right image boundary and search it in the left image. Among the two matches, the one with the higher match rating is taken to be the true one. This reduces the number of mismatches due to occluded points.

## 6.3.2. Geometric Differences

Windows centered at region boundaries contain more information than windows centered at points inside regions. This, however, does not necessarily imply that matching on region boundaries will be absolutely error free. Geometric difference between the images is a major source of mismatches when searching along region boundaries. Figure 6.5 depicts this situation. Note that the background to the left of the object when viewed from the left camera corresponds to area A', while this same area viewed from the right camera corresponds to area A. Now if A and A' are considerably different, windows centered at region boundaries that should match may not correspond to each other, because they contain different parts from the

background causing a low similarity measure for the windows. To avoid this, window shapes should be taken in such a way that they cover only parts from the object and not from both the object and the background. Figure 6.6 compares the traditional and the new window shapes.





Implementation of the new window shape is easy. Once the boundaries of regions are determined, the background can be replaced by zeros. When cross-correlation is computed, only non-zero pixels are used in the computation. Note that the regions should not contain zero-valued pixels (if they do, replace the background by a negative value and take non-negative values for determination of cross-correlation). 6.3.3. Homogeneous Areas

If a selected window belongs to a homogeneous area, it may not contain enough information to distinguish it from other windows, and a mismatch may occur. Fortunately, by taking windows centered at region boundaries, the occurrence of such cases is very rare. However, if selection of a homogeneous window is inevitable, there are two ways to recover a possible mismatch.

- 1) If geometric models of objects in the scene are known and the objects in the images can be recognized, then by fitting the appropriate geometric model to those boundary points that have high match ratings, the rest of the points on the object can be recovered.
- 2) Since depth on an object varies smoothly almost everywhere [Marr 76], if two neighboring points on the boundary in one image fall apart by more than a threshold value in the other image, then it is likely that a mismatch has occured, and the position of the point can be approximated by linear interpolation of the position of its neighboring points.

## 6.3.4. Window Size

Window size is a factor which can also affect the accuracy of the search. If the images do not have geometric differences, then the larger the window size the more accurate the search. But in images with geometric differences, larger windows may not necessarily imply more accurate matching. On the other hand if the windows are too small, they will not contain enough information to carry out a reliable search.

As far as the speed is concerned, the smaller the window the faster the search. The question then becomes, what is the smallest size window which can hold enough information to carry out a satisfactory search? It is clear that depending on the area at which the search is carried out, we may need different window sizes. For example, a low variance area requires a larger window than a high variance area.

Since an image may contain both low variance and high variance areas, a dynamically varying window size seems to be the best. In [Hannah 74, Yakimovsky 78], dynamically varying window sizes are used. To determine window size for a given location (x,y), correlations of the window centered

at (x,y) with windows centered at its neighboring points are determined (this process is called auto-correlation). A good window size produces a local correlation peak at (x,y). Otherwise, window size is increased until the window at (x,y) extends to a high variance area where it would have low correlation values with its neighboring windows, and so a local peak is detected. Determination of window size in each step, however, brings overhead to the already slow search process.

## 6.3.5. Similarity Measure

It has been experimentally shown that cross-correlation provides a higher accuracy than the sum of absolute differences as the similarity measure [Svedlow 76]. Is cross-correlation the best similarity measure in window searching? There are other similarity measures such as invariant moments and image transform coefficients that could be used as well. Since invariant moments or image transform coefficients are information preserving, if a sufficient number of them are taken, the original windows can be reconstructed. It could be that invariant moments and image transform coefficients provide a better similarity measure than the cross-correlation in the window search process. The performance of cross-correlation versus invariant moments and image transform coefficients is not known. No attempt has been made to investigate them here either. However, the author feels that these similarity measures may provide a more accurate tool for window matching and will be studied in future research.

## 6.4. Computation of Depth

Assume that the camera model of Figure 6.1 is available. Ideally, there exist two numbers, a and b, such that  $a\underline{u}l=\underline{d}+\underline{b}\underline{u}^2$ , where  $\underline{u}l$  and  $\underline{u}^2$  are unit vectors pointing from the lens centers of camera 1 and camera 2 to point  $\underline{V}$ , respectively.  $\underline{d}$  is a vector of length d pointing from the lens center of camera 1 to the lens center of camera 2. If the two rays  $\underline{u}l$  and  $\underline{u}^2$  intersect, we can obtain the coordinates of point  $\underline{V}$  by computing  $\underline{c}l+\underline{a}\underline{u}l$  or  $\underline{c}2+\underline{b}\underline{u}2$  where  $\underline{c}l$  and  $\underline{c}^2$  are coordinates of the lens centers of camera 1 and camera 2, respectively. Because of various errors, the two rays may not intersect, and a reasonable way to estimate  $\underline{V}$  is to place it midway between  $\underline{c}l+\underline{a}\underline{u}l$  and  $\underline{c}2+\underline{b}\underline{u}^2$ , or

 $\underline{\mathbf{V}} = (\underline{\mathbf{c}} + \underline{\mathbf{u}} + \underline{\mathbf{c}} + \underline{\mathbf{b}} \underline{\mathbf{u}}^2)/2 \tag{6.2}$ 

When the coordinates of two corresponding points  $\underline{v}l$  and  $\underline{v}2$  are known, then we can compute  $\underline{u}l=(\underline{L}l+\underline{v}l)/||\underline{L}l+\underline{v}l||$  and  $\underline{u}2=(\underline{L}2+\underline{v}2)/||\underline{L}2+\underline{v}2||$ , where  $\underline{L}l$  and  $\underline{L}2$  are vectors on the optical axes of camera 1 and camera 2, respectively, as shown in Figure 6.1. So, if we know  $\underline{u}l$ ,  $\underline{u}2$ , and the coordinates of the lens centers, then the coordinates of points in three dimensions can be obtained, but only if we know the values of a and b.

a and b can be estimated by using corresponding points in the two images and minimize

 $E = ||c|+au|-(c2+bu2)||^2$ 

Values of a and b which minimizing E are as below [Duda 73],

 $a = [\underline{u} | \underline{d} - (\underline{u} | \underline{u}^{2}) (\underline{u}^{2} | \underline{d})] / [1 - (\underline{u} | \underline{u}^{2})^{2}]$   $b = [(\underline{u} | \underline{u}^{2}) (\underline{u} | \underline{d}) - (\underline{u}^{2} | \underline{d})] / [1 - (\underline{u} | \underline{u}^{2})^{2}]$ where d=c2-c1.

In summary, if the coordinates of the lens centers,  $\underline{cl}$ and  $\underline{c2}$  and the orientation and the focal length of the cameras,  $\underline{Ll}$  and  $\underline{L2}$  are given, to determine the position of points in three dimensions we

1. Determine the corresponding points  $\underline{v}1$  and  $\underline{v}2$ , in the two images using the procedure of section 6.2.

2. Then determine  $\underline{u}i = (\underline{L}i + \underline{v}i) / ||\underline{L}i + \underline{v}i||$ ; i = 1, 2.

6. Then estimate a and b from (6.3).

4. Finally compute the coordinates of point  $\underline{V}$  (whose projections in the two images are  $\underline{v}$ l and  $\underline{v}$ 2) in three dimensions from (6.2).

If the camera parameters are not given but the actual position of some points in the scene are known, the camera parameters can be estimated [Gennery 79, Haralick 80a].

6.5. Results

To evaluate the performance of the proposed stereo image registration technique, four sets of stereo images as shown in Figures 6.7, 6.8, 6.9, and 6.10 were used. Figure 6.7 shows a number of objects (a box, a reel of tape, a belt, a wad of paper, a piece of chalk, a roll of film, and a pipe cap) arranged on a textured table. The lighting of the scene was carefully adjusted to avoid any shadows from the objects.

Since a stereo camera system was not available, the images were obtained by a single camera but with displacement of about 7 cm horizontally and parallel to the background. The distances of objects to the baseline were on average of 1 m. The background was marked with small circles so that the two images could be aligned by registering the background. Alignment of the images is necessary before disparity measurement is done, because a stereo camera system was not used to obtain the images.



(b)

Figure 6.7. Stereo images of objects on a textured table.

Figure 6.8 is a pair of aerial images taken from a residential area by a low altitude aircraft. The displacement of the aircraft between the time the images were obtained and the distance of the aircraft to the scene are not known. The images were taken on a sunny day, and as a result shadows from buildings, trees, and bushes are present in the images. The shadows differ significantly in the two images.





Figure 6.9 shows a scene with a truck present and Figure 6.10 is a garden scene. Images of Figures 6.9, and 6.10 were taken on a cloudy day so that no shadows are present. The displacement of the camera for both images was about 1 m. The above four sets of data are used for stereo image registration. In the following, after corresponding points in a pair of stereo images are determined, their horizontal difference, known as the disparities, are computed for the purpose of depth perception.



(a) (b) Figure 6.9. Stereo images of the truck scene.



(a) (b) Figure 6.10. Stereo images of the garden scene.

## 6.5.1. Disparity on Region Boundaries

Since the images were not obtained by a stereo camera system, there was a need to align them to make them look as if they were. To do this, some points that are very far to the viewer were selected by hand and were registered. Bv registering the selected points, the disparity at those points are set to zero, making the disparities of other points in the scene appear smaller than what they should be and the computed disparities are relative to the disparities of the registered points. So, it should be noted that the disparity values computed below are relative to the disparity of the registered points.

In the following, one image in each set is segmented according to the procedure of section 5.2.1. Then an attempt is made to locate points on the obtained region boundaries in the other image. The original design requires segmentation of both images and locating region boundaries of the left image in the right image, and locating region boundaries of the right image in the left image. This scheme, however, has been very difficult to implement on the available PDP 11/34 computer with the very limited internal So here, only one image is segmented and a search memory. is carried out in the other image for positions of points on region boundaries.

After an image is segmented, regions with perimeter size smaller than 30 pixels are eliminated from the image, because small regions will not contain enough information to carry out a reliable search. Regions belonging to the background are removed from the image also (background is replaced by zeros) so that windows centered at region boundaries will contain only parts from the objects. Figures 6.11, 6.12, 6.13, and 6.14 show images of Figures 6.7.b, 6.8.b, 6.9.a, and 6.10.a after segmentation, removal of regions with boundaries smaller than 30 pixels, and removal of the background. It does not matter which one of the two images (the left image or the right image) is segmented.

Window size 16x16 and search domain size 3x7 were selected. Three rows rather than 1 row were selected for the search because a stereo camera has not been used to obtain the images. When putting the pictures down for digitization, a slight rotational difference between the images appears, as if the camera's displacement hasn't been horizontal. Because two corresponding points in the two images might not fall on the same row, the neighboring rows also need to be searched. It is assumed that most neighboring points on the region boundaries do not have


Figure 6.11. Segmentation of image of Figure 6.7.b.



Figure 6.12. Segmentation of image of Figure 6.8.b.



Figure 6.13. Segmentation of image of Figure 6.9.a.



Figure 6.14. Segmentation of image of Figure 6.10.a.

disparity differences of more than 3 pixels so 7 column places have been taken for the search (3 pixels at either side of the previous corresponding point's column number).

For the initial point on the boundary of a region, a 5x33 search domain was selected. It has been assumed that maximum disparity in the images is 16 pixels. By this, a pair of corresponding points in the two images becomes known. Then the search domain is reduced to 3x7 areas for the rest of the points on the region boundary.

Since a sharp change in depth in the scene causes the  $\cdot$ disparity of two neighboring points in the images to differ by more than 3 pixels, a search in a 3x7 area will result in a mismatch. A mismatch usually has a low cross-correlation value. Therefore, if in the window search process the cross-correlation value for the matched windows fall below 0.6, the search is repeated, this time in a 3x17 area. After the position of corresponding points in two stereo images are determined, the horizontal differences between them are computed and plotted in a map called the disparity map. The larger the disparity, the closer the point to the Disparity on region boundaries of Figures 6.11, viewer. 6.12, 6.13, and 6.14 are shown in Figures 6.15, 6.16, 6.17, and 6.18, respectively. The larger the disparity, the darker the point.



Figure 6.15. Disparities on region boundaries of 6.11. Arrows show the mismatches.



Figure 6.16. Disparities on region boundaries of 6.12. Arrows show the mismatches.



Figure 6.17. Disparities on region boundaries of 6.13. Arrows show the mismatches.



Figure 6.18. Disparities on region boundaries of 6.14. Arrows and points in window A show the mismatches.

# 6.5.2. Distinction of Objects and Background

The computed disparities of Figures 3.15, 3.16, 3.17, and 3.18 belong to points on region boundaries on the object side and not on the background side. The notion of object and background, however, is arbitrary. After segmenting an image, we can assign points falling above the threshold value to the background and points below the threshold value to the objects, or vice versa. In Figure 6.13, the background is assigned to values above the threshold value. We can assign points below the threshold value to the background also. This is shown in Figure 6.19. Disparities computed using Figures 6.9.b, and 6.19 are shown in Figure 6.20.

The disparity on the boundary of the truck in Figure 6.17 shows the disparity of the trees behind the truck, while the disparity on the boundary of the truck in Figure 6.20 shows the disparity on the body of the truck. Since the truck and the trees behind it are at different depths from the viewer, it is natural to obtain different disparities on different sides of the boundary. The disparity around the boundary of the truck for Figures 6.17 and 6.20 are enlarged and shown in Figures 6.21.a and 6.21.b.



Figure 6.19. Same as Figure 6.13 but the role of objects and the background is interchanged.



Figure 6.20. Disparities on region boundaries of image of Figure 6.19. Arrows show the mismatches.

(a)																														
		น กอออกอยี่อะขออยออรี่ออออออรอออออออออออออออออออออออออ	() 	• • • • • • • • • • • • • • • • • • • •	21		00000000000000000000000000000000000000					20202020202020202020202020202020202020			<ol> <li>coccessessessessessessessessessessessesses</li></ol>		00000000000000000000000000000000000000					<b>n</b>	000000000000000000000000000000000000				<ul> <li>0000800000000000000000088888888888888</li></ul>	00008000000000000000000000000000000000	00000000000000000000000000000000000000	
(	a )	01	n '	th	е	tr	'ee	9	51d	e	an	d	<b>D)</b>	0	n	th	le	t	ruo	: K	S	ið	le.	•						

**ດວວວວວຽວຈາວຊິດວິຊິວຊິວດາກ**້ວວວວວວວວວວວວວວວວວວວວວວວວວວວວວວ . ၀၀၀၀၀၀ အို၀၀၀၀ ပူ၀၀ပ္၀၀၀၀၀၀၀၀၀၀၀ ပူ၀၀၀၀၀ ပူ၀၀၀၀၀ ပူ၀၀၀၀၀ ပူ၀၀၀၀၀၀ ပူ၀၀၀၀၀၀ ပူ၀၀၀၀၀၀ ပူ၀၀၀၀၀၀ ပူ၀၀၀၀၀၀ ပူ၀၀၀၀၀၀ ပ 

This experiment shows the necessity for separating the object and the background in the window search process. То show how much improvement is obtained by applying this idea, another experiment was carried out. The region belonging to the building roof-top at the middle left side of the scene in Figure 6.12 is used. First, a search is carried out for points on region boundaries with windows containing areas both from the roof-top and its background. The disparities are shown in Figure 6.22.a. Then disparities were computed for the same roof-top boundary with the background removed. In this The disparities are shown in Figure 6.22.b. experiment, when the background was removed, all computed disparities were consistent with the roof-top's shape, while when the background was present, most of the disparities were computed wrongly.

# 6.5.3. The Mismatches

The mismatches obtained in the window search are indicated by arrows in Figures 6.15, 6.16, 6.17, and 6.18. Mismatches in these images are due to 1) occluded points, 2) homogeneous areas, and 3) scene differences.

(Ъ) Disparity on region boundary of the building Figure 6.22. (a) with background present, (b) with background removed.

ο

Õ

Ō

0

0

0

0

0

0

0

0

0

0

0

Ō

Ó

0

0

0

Ō

Ó

õ

Ō

Ô 

0

ο

0

õ

ο

Ó

õ

It is possible to avoid most mismatches due to occluded points by selecting points from both images for searching, as mentioned in section 6.3.1. For example, the mismatches shown by A in Figure 6.18 belong to an area which does not exist in the right image. Since windows were taken from the left image (Figure 6.14) and the search was carried out in the right image (Figure 6.10.b) where area A does not exist, mismatches were obtained. Mismatches due to occluded points can be avoided if we segment the right image (See Figure 6.23) and take windows centered at the obtained region boundaries and search them in the left image. Figure 6.24.a shows the disparities in area A when windows were selected from the left image and the search was carried out in the right image, and Figure 6.24.b shows the disparities when windows were selected from the right image and searched in the left The cross-correlation values when the image. points were selected from the right image and searched in the left image were consistently higher than when points were selected from the left image and searched in the right image, showing that disparities in Figure 6.24.b are more reliable than disparities in Figure 3.24.a. Actually points in Figure 3.24.a are occluded points, and they should be avoided in window searching. Selecting points from both images for searching is a mechanism for discovering this.



Figure 6.23. Segmentation of image of Figure 6.10.b.

Mismatches due to homogeneous areas are something that cannot be avoided by the proposed approach unless a test for homogeneity of each window is carried out, which makes the search process very slow. If the model for an object in the scene is known, by identifying the position of a number of points on the object, we can register it with points on its model and recover the position of the rest of points on the object. For example, area A in Figure 6.15 is due to the homogeneity of the surface of the box. This error can be recovered if the box can be matched to its model (see section 6.6.1 for discussion about model matching).

() () () ٥٥٥٥٩٩٥٥٥٥٥٥٥٥٩٩٩ ٥٥٥٥٥٥٥٥٥٥٥٥٥٥٥٥٥٥٥٩٥٩ Figure 6.24. Computed disparity when points are (a) from the left and (b) from the right image selected 6.10. of

00 (a)

Mismatches that occur in Figure 6.16 are mostly due to image differences. Trees with shadows look very different from one viewpoint than another. Errors due to shadows are very difficult to recover unless their properties are well understood and predicted (see section 6.6.2 for a discussion of shadows).

Noisy mismatches can be corrected by smoothing the disparity values. For example, if disparity on a boundary changes sharply and then returns to the previous value, a possible mismatch is indicated, and its disparity value can be replaced by the median of the disparity of its n neighboring points. For example, Figure 6.25.a shows a window around area B in image of Figure 6.15. The disparity on the boundary of the wad of paper changes from 9 pixels (the values are multiplied by 10 for display purposes) to 1 pixel and then returns to 9 pixels again. This is a mismatch which can be eliminated by smoothing the disparities with a window of size 5 pixels (the disparity at each point is replaced by the median disparity of 5 pixels, 2 to its left and 2 to its right). After smoothing disparities of Figure 6.25.a with a window of size 5, disparities of Figure 6.25.b are obtained.

(Ъ) Figure 6.25. Disparity (a) before and (b) after smoothing.

0

90 

(a)

## 6.5.4. Disparity Inside Regions

the preceding sections, disparity on the region In boundaries were determined, leaving the disparity inside regions. Sometimes it is necessary to determine the exact three-dimensional position of a point on a homogeneous surface, for example, for drilling a hole. If the surface is known to be planar, then the position of points on the surface can be approximated by interpolating the positions of points on the boundary of the surface, as shown in Figure For example, the box in Figure 6.7 has planar 6.6. To obtain faces of the box, we must segment the surfaces. image of the box more finely. Images that contain more than two types of regions cannot be segmented satisfactorily by one threshold value. An image, however, can be segmented recursively by thresholding until a satisfactory result is obtained.

When the image of Figure 6.7.b was segmented, regions of Figure 6.ll were obtained. Each region in the segmented image can be tested to see if it contains more than one type of region by looking at its intensity variance. If the variance of a region is above a threshold value, the region can be assumed a new image for segmentation. This process can be continued until variances of all regions fall below a threshold value. For example, the region corresponding to

the box is shown in Figure 6.26.a. This is assumed to be a new image. It is segmented according to the procedure of section 5.2.1, giving results shown in Figure 6.26.b. Again, regions smaller than 30 pixels perimeter are eliminated, and the boundary of the remaining region is combined with the boundary of the box previously obtained (Figure 6.26.c).



Figure 6.26. (a) Image of box (b) segmented (c) its edges.

The disparities on edges of the box are computed and shown in Figure 6.27.a. The disparities on faces of the box are computed by linear interpolation of the disparity of four points that are to the left, to the right, above, and below the point and on the boundary of the box. The disparity of points on faces of the box are shown in Figure 6.27.b.

(a)

(Ъ)

Figure 6.27. Disparity (a) on edges, (b) on faces of box.

If the objects in the scene do not have planar surfaces, the position of points inside regions cannot be determined by linear interpolation. However, if the models for an object in the scene are known a-priori, then by matching the model to the points from the object, the position of points on faces of the object can be recovered (see model matching in section 6.6.1).

### 6.5.5. Summary of Results

Four sets of data were used for registration of stereo images. Disparities on region boundaries were determined, and improvements over the traditional window searching were demonstrated. The computed disparities, however, were not error free. Most of the errors that were obtained were in the image of Figure 6.16 due to image differences from shadows. Table 6.1 summarizes the accuracy of the proposed registration technique on the four sets of data.

The computation time for a search domain of size 5x33 (for establishing the first corresponding point) was 18 secs, for a search domain of size 3x17 (this is when the cross-correlation value is smaller than 0.6) was 6 secs, and for a search domain of size 3x7 (when cross-correlation value is larger than or equal to 0.6) was 2 secs. Total computation time needed to determine disparity on region boundaries of images of Figures 6.11, 6.12, 6.13, and 6.14 on a PDP 11/34 computer are shown in Table 6.1.

The disparity maps of Figures 6.15, 6.16, 6.17, and 6.18 show the relative depth of points on the boundaries of objects in the scene. The depth values determined in this manner carry valuable information about positional

Table 6.1. Speed and accuracy of stereo registration.

Image from	Computation Time	% of Mismatches
Figure 6.ll	l hr ll mins	2%
Figure 6.12	43 mins	28%
Figure 6.13.	l hr 16 mins	5%
Figure 6.14	56 mins	8%

relationships between objects in the scene, such as in front of, behind, nearer than, and farther than. Positional relations are constructs by which constraint among objects in an image can be defined to utilize contextual information for interpretation and description of the images [Baird 74].

In the scene of Figure 6.7, if we look at one of the images, we can conclude that the roll of film is behind the belt because the film is occluded by the belt. But we can't tell whether the roll of film or the reel of tape is farther from the viewer. By looking at the disparity map of Figure 6.15 we find out that the roll of film is farther from the viewer than the reel of tape. If we observe images of Figure 6.7, stereoscopically, we can perceive depths of objects in the scene and verify the correctness of the disparity map of Figure 6.15. In the aerial image of Figure 6.8, it is not evident which building in the scene is the tallest. We can find the tallest building, however, by referring to the disparity map of Figure 6.16. The two swimming pools that could not be distinguished from the buildings can easily be recognized as swimming pools by referring to the depth map of Figure 6.16.

These are some of the examples in which disparity maps can help understand the images better. Image understanding is the process of interpreting intensity values of images to descriptions that are understandable by humans. One of the factors that can greatly help understand images better is the positional relations between objects in the scene, and the depth map is a tool providing this information.

### 6.6. Further Improvements

The disparity maps that were obtained in Figures 6.15, 6.16, 6.17, and 6.18 involve some errors (areas shown by the arrows). Some further improvement of the technique is required before it can be used reliably.

For indoor scenes where the lighting and arrangement of objects in the scene can be planned beforehand, a stereo image registration technique assisted by model matching seems to be appropriate. Here, if models of objects in the scene are known, the task would be to recognize the objects and then to locate a number of prespecified points, such as corners, marked points, etc. on the object. By registering the object with the model, the position of the rest of the points on the object can be determined. Model matching is discussed in section 6.6.1.

Most mismatches obtained in Figure 6.16 are due to shadows of trees and buildings. These shadows appear different from one view to another and consequently cause mismatches. Model matching might be able to recover some of these errors, but since modeling any outdoor scene is not possible presently, understanding shadows and their relations to objects could help us carry out a more reliable search. Shadows are discussed in section 6.6.2.

Reflection is another factor which should be considered in stereo image registration. Reflection of an object on a shiny table, reflection of a house on a quiet pond, reflection of an object in a mirror or on a shiny metal surface, are examples where we need to interprete the computed disparities. Does a computed disparity correspond to the disparity of a point on the table, pond, mirror, or to points on the objects themselves, or correspond to neither of them? Section 6.6.3 discusses the reflections.

#### 6.6.1. Model Matching

In this chapter no a-priori knowledge about objects in the scene has been used. But if some knowledge about the scene, such as object arrangements or object geometries are known, most often we can register stereo images more accurately, and sometimes even faster.

If the geometry is known of objects present in the scene in the form of descriptions or models, then the problem would be to recognize objects in the scene and determine the position of a few points on each object. Positions of the rest of points on the object can be determined by fitting the right model to the available points.

In Figure 6.15, error A can be overcome by matching the image of the box to its model. This is conditioned upon the capability of recognizing the box correctly. Recognition of three-dimensional objects is by itself a major problem in computer vision. If objects in the scene are limited to polyhedrons, and are well separated in the scene, techniques are already available to recognize them [Shirai 71, Nagao 73, Shapira 77, Oshima 79].

One way to recognize an object is to compare it to the models already stored in the computer and determine the model most similar to it. To be able to do so, we need the capability for generating models of objects. A model 1) should be easy to generate, 2) should provide sufficient information about the object, 3) should work on a large class of objects, and 4) should be position-invariant.

Underwood has described a technique to generate models of objects that have planar surfaces [Underwood 75]. In this technique, a model is generated for a given object by first obtaining images from all around the object. Then planar surfaces of the object are determined by following high gradient edges in the image and finding closed boundaries. The model is then represented by a linked list in such a way that a node in the list shows a surface on the object, with information about the surface stored in the node. Two nodes in the list are linked only if the two surfaces that represent the two nodes are adjacent.

In [Agin 76, Nevatia 77] a technique to represent objects with conical parts in a computer is given. In this technique, the three-dimensional position of points on the object is determined by a laser range finder. The object is then segmented into conical parts. Each part is represented by its axis and planar cross-section normal to the axis. Finally a model is created by joining touching cones together. Model generation in this manner has been implemented in the ACRONYM vision system at AI Laboratory of Stanford University [Brooks 81].

The above techniques can be used to generate models with planar surfaces or conical parts. If we look around us, we do not see many objects with these properties. How can we represent objects without planar surfaces or conical parts inside the computer? If the precise geometrical structure of an object is known, then a model for the object can be created. Roth discusses modeling of mechanical parts with known geometrical properties by "ray casting" [Roth 82].

In ray casting, to generate a model of an object, the photographic process is simulated in reverse. For each pixel on the screen, a ray is passed through it to identify the visible surface by determining the intersection of the ray with the first surface. At the ray-surface intersection point, the surface normal is computed, and using the position of light source, the brightness of the pixel on the screen is determined. By this way, a gray value image of the object is generated as its model.

If object to model registration is accomplishable, considerable speed-up in depth perception is possible, since a large number of window searches are avoided by this approach. And if the position of those points used for object to model registration is accurate, overall accurate depth values on the object will be obtained.

For a complex scene containing multiple objects, the object recognition problem becomes very difficult, because parts of an object might be occluded by another object. But if techniques could be developed for recognition of three-dimensional objects in a complex scene, then there is only a need to determine the position of a number of points on each object.

With the object to model registration approach, if a three-dimensional model of a scene (such as a terrain or an apartment complex) is known, it is possible to register the whole scene to its model by identifying a few objects in the scene and determining the position of a number of corresponding points on the model scene and the observed scene.

### 6.6.2. Shadows

In section 6.5, we saw that shadows in a scene create image differences and when the differences, become large enough, wrong matches might be obtained. Here, however, we will see that if shadows are well understood, they can be used to our benefit.

.



Figure 6.28. Image of an object and its shadow.

Figure 6.28 shows the geometry of an object (in a plane) and its shadow. Since the shadow of an object in a plane is equivalent to the projection of the object to the plane, coordinates of points on the boundary of the shadow are related to points on the object by a projective transformation,

$$x'=(ax+by+c)/(dx+ey+1.0) = f(x,y)$$
  
y'=(fx+gy+h)/(dx+ey+1.0) = g(x,y)  
(6.4)

where (x,y) and (x',y') are points from the object and its shadow, respectively.

From the definition of an image (see Chapter 2), we know that points on a plane (x,y) and their images  $(x^{"},y^{"})$  are related to each other by a projective transformation.

$$x''=f_1(x,y)$$
  
 $y''=g_1(x,y)$  (6.5)

Points from a shadow (x',y') and their images (x'',y'') also relate to each other by the projective transformation.

$$x^{"}=f_{2}(x',y')$$
  
 $y^{"}=g_{2}(x',y')$  (6.6)

using (6.4), we get

$$x''=f_{2}(f(x,y),g(x,y))$$
(6.7)  
$$y''=g_{2}(f(x,y),g(x,y))$$

Formula (6.7) shows the coordinates of points on the boundary of the shadow on the image plane as a function of points on the object. Since the projective transformation of a projective transformation is equivalent to another projective transformation (because the property of straightness stays invariant ), then formula (6.7) can be written as

$$x''=f_{3}(x,y)$$
 (6.8)  
 $y''=g_{3}(x,y)$ 

(6.8) shows that the image of a shadow relates to the original object by a projective transformation.  $f_3$  and  $g_3$  can be determined by knowing the position of at least four pairs of corresponding points on the object and the image of their shadows. Or vice versa, if  $f_3$  and  $g_3$  are known we can determine the position of points on the object by knowing the position of the image of the shadows of the points.

Formula (6.5) shows the relation between points on an object and points in the image of the object, while formula (6.8) shows the relation between points on an object and points in the image of the shadow of the object. So, position of points on (the boundary) of an object can be determined by knowing either the position of points on its image or on the image of its shadow. In this process, of course, first we have to determine the transformation functions  $f_1$ ,  $g_1$  or  $f_3$ ,  $g_3$  by locating a number of corresponding points on the object and in its image, or corresponding points on the object and the image of its The position of points on an object, as was shadow. described in section 6.4, can be determined if the position of corresponding points in stereo images and the camera parameters is known.

We see how valuable information about the shadow of an object can be in locating and describing the object. In a scene where a part of an object is occluded by some other object, we can recover the boundary of the occluded part if the shadow of the occluded part is visible in the image.

## 6.6.3. Reflections

Reflection of an object on a shiny surface is a phenomenon that cannot be avoided in some scenes that contain for example polished metal parts. In Figure 6.7 we can clearly see the reflection of the wad of paper, the piece of chalk, the belt, and the reel of tape on the shiny table. When the image of Figure 6.7.b is segmented (see Figure 6.11), along with the objects in the scene, the reflections of these objects are also being extracted from the background. Now the question is, do the computed disparities on the region boundaries of the reflections belong to the points on the table or to points on the reflected objects?

In the image of scene 6.7, since the table was textured and not homogeneous and shiny enough, the computed disparities seem to belong to points on the table, because the disparity values of the reflections are larger than the disparity values of the objects. If the computed disparities belonged to the reflections, they should have been smaller than the disparities of corresponding points on the objects, because points from reflections are farther from the viewer than their corresponding points on the objects.

Although a reflection falls in front of the object in an image, disparity of a point on reflections is smaller than the disparity of its corresponding point on the object simply because it is farther from the viewer, and this may confuse the image understanding process. Object reflections are difficult to characterize because the image of a reflection is not symmetric to the image of the object. There are cases, however, where reflections help see occluded points on an object, such as points underneath an object that cannot be seen in the image of the object.

If the reflection is from a mirror, the image of points from the object and the image of points from the reflection of the object are equivalently important in reconstructing the object.

Figure 6.29 shows a point and its reflection from a mirror-like surface. The image of point p on a plane characterized by the XY-coordinate system can be determined in the image plane by the projective transformation,



Figure 6.29. Images of a points and its reflection.

where (x,y) and  $(x'_1,y'_1)$  are corresponding points on the XY-plane and the image plane (X'Y'-plane), respectively. The reflection of point p will be at point p' with coordinates (-x,y), and the image of point p' on the image plane will be at

$$x'_{2} = (-ax+by+c)/(-dx+ey+1.0)$$

$$y'_{2} = (-fx+gy+h)/(-dx+ey+1.0)$$
(6.10)

Formula (6.9) shows the relation between points on an object (in a plane) and their images, while formula (6.10) shows the relation between points on the object and the image of their reflections. (6.10) shows that we can determine the position of points on the object by knowing only the position of points on the image of the reflection of the object.

#### Chapter 7

#### CONCLUSIONS

#### 7.1. Summary

Image registration was approached as a two-stage process. In the first stage, global correspondence was achieved by segmenting the images and determining corresponding regions in the two images. Then an attempt was made to establish local correspondence by determining corresponding points belonging to corresponding regions.

Registration of images was studied, beginning with registration of images that have translational, rotational, and scaling differences while allowing no non-linear geometric difference between the images. The study was then extended to registration of images that are taken from a three-dimensional scene by a stereo camera system and have small non-linear geometric differences.

The following procedure was given for registration of images with no non-linear geometric differences.

- Segment the two images and isolate closed boundary regions.
- 2. Determine corresponding regions in the two images.
- 3. Take centroids of corresponding regions as corresponding points to determine the registration parameters.

Image segmentation was carried out with the objective of obtaining optimally similar regions, by an appropriate measure, from the two images. The segmentation process first assumes that the images cover exactly the same scene, are of the same scale, and that the intensity difference between them is linear. After segmentation, regions of image 2 are compared to their corresponding ones in image 1, one by one, to determine whether any error has been made in. obtaining the region boundaries because the assumptions for the images did not hold. Each region of image 2 is refined by re-examining the information in and around the region in The technique adjusts itself to intensity and image 2. linear geometric differences between the images. This technique has the capability of determining the position of region boundaries up to subpixel accuracy.

Region correspondence was achieved by a probabilistic relaxation labeling process. In this process, regions in image 2 are called objects, and regions in image 1 are called labels. Then the problem has become labeling the objects in such a way that labeling is consistent with world knowledge. World knowledge has been expressed as the inter-region distances.

To determine the initial label probability for an object, first the shape similarity of the regions corresponding to the object and the label is determined. This similarity is assumed to be the weight for the given object having the given label. The weights are then normalized to obtain initial label probabilities. The shape measure which is used can determine similarity of shapes irrespective of their rotational and scaling differences.

After the images are segmented and corresponding regions in the two images are determined, centroids of corresponding regions are taken as corresponding control points from the two images. Since the images are assumed to have only translational, rotational, and scaling differences, by knowing a minimum of two pairs of corresponding control points from the two images, it is possible to estimate the registration parameters. If more than two pairs of corresponding control points from the two images are
available, the registration parameters are estimated by obtaining the least-squares fit to the overdetermined system.

If the images have non-linear geometric differences, centroids of corresponding regions may not correspond to each other, and it is not possible to register them by knowing only a few corresponding points from the two images. Local window searches are required to determine the position of corresponding points in the two images. Images that have non-linear geometric differences were limited to those that do not have any rotational or scaling differences. Examples of these kinds of images are those obtained by a balanced stereo camera system from three-dimensional scenes.

Since corresponding points in images from a well-balanced stereo camera system differ only in column values (and maybe small row values due to some errors), a search can be carried out in a thin band for determination of match points.

First an attempt is made to determine points in image 2 that correspond to points on the region boundaries of image 1. If a point <u>on</u> a region boundary of image 1 exists in both images but is not <u>on</u> the corresponding region boundary in image 2, then it is somewhere <u>near</u> the corresponding region boundary. So, for each point on a region boundary of image 1 there is a need to search only in a small area in a thin band in image 2 near the corresponding region boundary. By following the region boundaries in this fashion, corresponding points on or near corresponding region boundaries are determined.

Since some of the points on the region boundaries in image 2 may not exist in image 1 due to occlusion, for these points the search result in mismatches. To avoid mismatches from occlusion, rather than taking all points from one image and searching in the other image, points are taken from both images, and searched in both images. Among the two matches, the one with the higher match rating is taken as the true match.

After corresponding points on or near the corresponding region boundaries are found, an attempt is made to locate the position of corresponding points inside the regions. Since areas inside regions are almost homogeneous and window searching has a low accuracy, corresponding points inside regions are determined by interpolation.

A by-product of registration of stereo images is perception of relative depth, which is obtained by measuring the disparity of corresponding points in the two images. A procedure for determination of depth of points in three-dimensions was given, knowing the camera parameters and the disparity values.

All the procedures that were given above were implemented and tested on real data.

### 7.2. Contributions

An automatic digital image registration algorithm has been designed which is capable of registering images that have translational, rotational, scaling, and intensity differences with subpixel accuracy. To reach this goal, most of the steps in the algorithm were designed anew.

Although there were many approaches available for image segmentation, none of them could fulfill our need. So, a new image segmentation technique was designed which works on multiple images (two at a time) and segments them in such a way that the obtained corresponding regions in the two images are optimally similar by an appropriate measure.

Previous techniques for determination of corresponding regions in two images have used either local or global information in the images. A new approach for determination of corresponding regions in the two images was given by the probabilistic relaxation labeling process which uses both local and global information in the images. The technique

204

becomes slow when the number of objects and labels become large (>30), but the number of regions in an image can be controlled by considering only regions with perimeter sizes between two threshold values. If the number of regions become inevitably large (>30), we can selectively choose a subset of the regions for determination of the transformation parameters between the images. Correspondence between the rest of the regions can then be established using the transformation parameters.

A shape similarity measure was needed both in image segmentation and in determination of region correspondences. Other shape similarity measure techniques were available, but a new technique using shape matrices was designed. The new technique is easy to implement, can represent any shape with no restrictions, can compute the similarity between two of regardless their rotational or shapes scaling differences, is information preserving, and has the capability of detecting shape defects.

Region centroids have been used as control points. It was shown that region boundaries change due to random noise, but coordinates of the centroids stay stable under random noise. Note that registration of images that do not have non-linear geometric differences depends on determination of two or more pairs of corresponding control points in the two images. Once corresponding control points in the two images are found, the rest of the process is straightforward.

In registration of stereo images, the nature of three-dimensional scenes and camera models were studied carefully, the following observations were made, and new strategies were introduced to the window search process to increase the accuracy of the point correspondence process.

- 1. If a point on the boundary of an object in a three-dimensional scene is visible from only one of the cameras (say the left camera), then there exists a point on the object boundary visible by the right camera on the same match line, which is visible from both of the cameras. This tells us that, rather than taking all the points from one image, we should select points from both images to reduce the number of impossible matches.
- 2. If a window lies on an area with depth discontinuity, since the area covered by the window has geometric differences in the two images, the window search may end up in a wrong match. To reduce the effect of geometric difference between windows from the left and the right images, the window shapes are taken in such a way that they cover only parts of the object. Since we are working on segmented images, we know where the region boundaries are. Then windows are centered at region

boundaries and their shapes are adjusted to the shapes of these boundaries so that they cover only parts of the regions.

3. Window search is more accurate in high variance areas than in the low variance areas of the image. Since region boundaries create high variance in an image, a window search on or near the region boundaries has high accuracy. To avoid search in low variance areas of the image and to save computer time, the positions of corresponding points inside regions are determined by interpolation.

Experiments have shown that introducing the above measures into the traditional window search process brings higher registration accuracy.

# 7.3. Future Research

Window search has been the key operation in determination of corresponding points in stereo images. The main similarity measure that has been used in the literature is the cross-correlation coefficient. The sum of absolute differences is sometimes used too but it is not as accurate as the cross-correlation coefficient [Svedlow 76]. There are other similarity measures that could be used to measure window similarities. These are the moments and the image transform coefficients, such as Fourier transform coefficients, Walsh-Hadamard transform coefficients, Haar transform coefficients, etc.

The performance of the moments and the image transform coefficients against the cross-correlation or sum of absolute differences is not known. But what is certain is that since most of the energy of a window is transformed to the very first few of its moments or transform coefficients, only the first few terms should be sufficient to measure the similarity between windows effectively.

Another property of the moments or the transform coefficients is that they characterize the structure of the windows, and if enough of them are known, the original windows reconstructed. In the can be case of cross-correlation coefficient, it really does not matter how the intensities are distributed in the windows. We may rearrange the intensities in a window and still obtain the same cross-correlation coefficient, while the transform coefficients will change by rearranging the intensities in the window. It seems that if the moments or the transform coefficients are used properly, higher accuracy over the cross-correlation should be obtainable. This is an area

which has not been explored, and deserves future research.

In this work, perception of depth as a result of stereo image registration was studied. Depth can be perceived by monocular vision too. We can close one of our eyes and still be able to walk around obstacles. One of many cues used in perceiving depth by monocular vision is focusing. From edge information in an image, we can determine the best focused image for a given object and then compute the depth of the object in the scene. This is an area in which very little research results are available, but it seems to be a very promising area for giving the capability of depth perception to machines.

The topic on reflections which was mentioned in section 6.6.3 was originated by Carl Page. In section 6.6.3, the topic has been scratched only on the surface. Reflections seem to contain a great amount of information about objects in the scene and could be used to our benefit. For example, know that if two images are obtained at two different we viewpoints from an object with known camera parameters, then locate position the can the of object in we three-dimensions. Now, rather than having two images at two different viewpoints from an object, what if we have one image containing the object and its reflection on а mirror-like surface? Is it possible to use one image to

209

locate the object in three-dimensions? If yes, how? The answer to this question and many others related to reflections are left for future research.

Image registration by window searching is known to be reliable on images with a lot of details and unreliable on images with smooth surfaces. In many natural or man-made scenes, existence of smooth surfaces cannot be avoided. For images with smooth surfaces we can use the theory of shape from shading [Horn 75] to obtain the surface normals at each point on object surfaces if the reflectivity function and position of the light source are known. Then windows with values showing the orientation or magnitude of the surface normals can be taken from one image and searched in another image in the same manner which was done for the intensity values. This idea, originated by George Stockman, seems to improve the accuracy of stereo image registration of scenes with smooth surfaces, and is considered for future research.

APPENDIX

.

.

.

# Appendix

## THE PROJECTIVE TRANSFORMATION

Two images that have been obtained at different viewing angles from a flat surface can be mapped to each other by the projective transformation [Peet 75],

$$x' = (clx+c2y+c3)/(c4x+c5y+1)$$
(1)
$$y' = (c6x+c7y+c8)/(c4x+c5y+1)$$

where  $c4x+c5y+1 \neq 0$  and (x,y) and (x',y') are coordinates of corresponding points in the two images.

<u>Proposition 1:</u> The property of straightness remains invariant under the projective transformation.

Proof: Let Ax'+By'+C=0 be the equation of a straight line in the original coordinate system. After applying transformation (1) to this line we get, A(clx+c2y+c3)/(c4x+c5y+1)+B(c6x+c7y+c8)/(c4x+c5y+1)+C=0or A(clx+c2y+c3) + B(c6x+c7y+c8) + C(c4x+c5y+1) = 0or (Acl+Bc6+Cc4)x + (Ac2+Bc7+Cc5)y + (Ac3+Bc8+C) = 0

211

or Ux + Vy + W = 0 (2)
where U=Acl+Bc6+Cc4; V=Ac2+Bc7+Cc4; W=Ac3+Bc8+C.
Equation (2) shows the equation of a straight line in the
transformed coordinate system. So, a straight line remains
straight under the projective transformation. #

Two images that have been obtained at different viewing angles from a flat surface can be mapped to each other by the projective transfromation. Proposition 1 insures that under this transformation, a straight line remains straight from one image to another.

<u>Proposition 2:</u> Assuming lines AC and BD are mapped to lines A'C' and B'D', respectively, by projective transformation (1), and if E is the point at the intersection of lines AC and BD and E' is the point at the intersection of lines A'C' and B'D', then E and E' are related to each other by the projective transformation (1) (see Figure 1).

Proof: Since point E is on line AC, its corresponding point in the transformed domain lies on line A'C'. Also since point E is on line BD, then its corresponding point should lie on line B'D'. So, the point corresponding to E should lie on both lines A'C' and B'D'. It can not be anywhere but at the intersection of lines A'C' and B'D'. So, points E and E' correspond to each other. Now since corresponding points on lines AC and A'C' are related to each other by



Figure 1. Straight lines under projective transformation.

transformation (1) and since E and E' on AC and A'C' correspond to each other, we conclude that E and E' are related to each other by the projective transformation (1).

Proposition 2 implies that if two images are obtained at different viewpoints from a flat surface, and if points A', B', C', and D' in image 1 correspond to points A, B, C, and D in image 2, respectively, then the point at the intersection of lines AC and BD in image 1 corresponds to the point at the intersection of lines A'C' and B'D' in image 2. BIBLIOGRAPHY

.

.

·

#### BIBLIOGRAPHY

- [Agin 76] "Computer Description of Curved Objects," Gerald J. Agin and Thomas O. Binford, IEEE Trans. on Computers, April 1976, pp 439-449.
- [Anuta 69] "Digital Registration of Multi-Spectral Video Imagery", Paul E. Anuta, Soc. Photo-Optical Instrum. Engs. J., Vol. 7, Sept. 1969, pp 168-175.
- [Anuta 70] "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transform Techniques," Paul E. Anuta, IEEE Trans. on Geoscience Electronics, Vol. GE-8, No. 4, Oct. 1970, pp 353-368.
- [Arnold 78] "Local Context in Matching Edges for Stereo Vision," R. D. Arnold, Proc. Image Understanding Workshop, May 1978, pp 65-72.
- [Atteneave 57] "Physical Determinations of the Judged Complexity of Shapes," F. Atteneave, Jounal of Experimental Psychology, Vol.53, No. 4, April 1957, pp 221-227.
- [Baird 74] "A Paradigm for Semantic Picture Recognition," Michael L. Baird and Michael D. Kelly, Pattern Recognition, Vol. 6, 1974, pp 61-74.
- [Baker 81] "Depth from Edge and Intensity Based Stereo", H. Harlyn Baker and Thomas O. Binford, 7th Int. Joint Conference on Artificial Intelligence, 1981, pp 631-636.
- [Barnard 80] "Disparity Analysis of Images", Stephen T. Barnard and William B. Thompson, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 4, July 1980, pp 333-340.

[Barnard 82] "Computational Stereo," Stephen T. Barnard and Martin A. Fischler, Computing Surveys, Vol. 14, No.4, Dec. 1982, pp 553-572.

- [Barnea 72] "A Class of Algorithms for Fast Digital Image Registartion" Daniel I. Barnea and Harvey F. Silverman, IEEE Trans. on Computers, Vol. C-21, No. 2, Feb. 1972, pp 179-186.
- [Bernstein 76] "Digital Image Processing of Earth Observation Sensor Data", R. Bernstein, IBM Journal of Research and Development, Jan. 1976, pp 40-67.
- [Brigham 69] "The Fast Fourier Transform", E. O. Brigham and R. E. Morrow, IEEE Spectrum, Dec. 1967, pp 63-70.
- [Brooks 81] "Symbolic Reasoning Among 3-D Models and 2-D Images," Rodney A. Brooks, Artificial Intelligence 17, 1981, pp 285-348.
- [Brown 76] "The Metrics of Visual Form: Methodological Dyspepsia," D. R. Brown and D. H. Owen, Psychological Bulletin, 1976, Vol. 68, No. 4, pp 243-250.
- [Caelli 81] Visual Perception, T. Caelli, University of New castle, 1981.
- [Chandra 82] "Feature Matching Multiple Image Registration Using Haar Coefficients," D. V. Satish Chandra, J. S. Boland, W. W. Malcolm, and H. S. Ranganath, Southcon, April 5-7, 1982, pp 549-552.
- [Cheng 82] "Image Registration by Matching Relational Structures", J. K. Cheng and T. S. Huang, 6th Int. Joint Conf. on Pattern Recognition, 1982, pp 354-356.
- [Clark 80] "Matching of Natural Terrain Scenes," C. S. Clark, D. K. Conti, W. O. Eckhardt, T. A. McCulloh, R. Nevatia, and D. Y. Tseng, 5th Int. Joint Conf. on Pattern Recognition, 1980, pp 217-222.
- [Croughwell 82] "Using Image Motion to Refine Stereopsis," W. J. Croughwell, W. E. Snyder, and S. A. Rajala, 6th Int. Sym. on Pattern Recognition, 1982, pp 760-763.

- [Davis 75] "A Survey of Edge Detection Techniques," L. S. Davis, Computer Graphics and Image Processing 4, 1975, pp 248-270.
- [Davis 78] "Automatic Selection of Control Points for the Registration of Digital Images", W. A. Davis and S. K. Kenue, 4th Int. Joint. Conf. on Pattern Recognition, 1978, pp 936-938.
- [Davis 80] "Cooperative Processes," Larry Davis, in Issues in Digital Image Processing, Robert M. Haralick and J. C. Simon, Eds., 1980, pp 171-193.
- [Davis 81] "Cooperating Processes for Low-level Vision: A Survey," Larry S. Davis and Azreil Rosenfeld, Artificial Intelligence 17, 1981, pp 245-263.
- [Dewdney 78] "Analysis of a Steepest-Descent Image-Matching Algorithm," A. K. Dewdney, Pattern Recognition, 1978, Vol. 10, pp 31-39.
- [Duda 73] Pattern Classification and Scene Analysis, Richard O. Duda and Peter E. Hart, Wiley-Interscience Publications, 1973, pp 379-404.
- [Fischler 81] "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," Martin A. Fischler and Robert C. Bolles, Comm. ACM, June 1981, Vol. 24, No. 6, pp 381-395.
- [Fu 82] Syntactic Pattern Recognition and Application, K. S. Fu, Prenctice-Hall, 1982, pp 79-128.
- [Gennery 77] "A Stereo Vision System for an Autonomous Vehicle," Donal B. Gennery, 5th Int. Joint Conf. on Artificial Intelligence, 1977, pp 576-582.
- [Gennery 79] "Stereo-camera Calibration," D. Gennery, Proc. Image Understanding Workshop, 1979, pp 101-107.
- [Goshtasby 83a] "A Two-Stage Cross-Correlation Approach to Template Matching," A. Goshtasby, S. H. Gage, and J. F. Bartholic, to appear in IEEE Trans. on Pattern Analysis and Machine Intelligence.
- [Goshtasby 83b] "Registration of Rotated Images by Invariant Moments," A. Goshtasby and W. R. Enslin, 17th Int. Sym. on Remote Sensing of Environment, May 1983.

- [Grimson 80] "Aspects of a Computational Theory of Human Stereo Vision," W. E. L. Grimson, Proc. Image Understanding Workshop, 1980, pp 128-149.
- [Hall 76] "Hierarchical Search for Image Matching," E. L. Hall, R. Y. Wong, and Lt. J. Rouge, IEEE Decision and Control Conf., 1976, pp 791-796.
- [Hall 79] Computer Image Processing and Recognition, Ernest L. Hall, Academic Press, 1979.
- [Hall 80] "The Selection of Critical Subsets for Signal Image, and Scene Matching," Ernest L. Hall, David L. Davies, and Michael E. Casey, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 4, July 1980, pp 313-322.
- [Hannah 74] "Computer Matching of Areas in Stereo Images" Marsha Jo Hannah, Ph.D. Thesis, July 1974, Computer Science Dept. Stanford University.
- [Haralick 80a] "Using Perspective Transformation in Scene Analysis," Computer Graphics and Image Processing 13, 1980, pp 191-221.
- [Haralick 80b] "Compatibilities and the Fixed Point of Arithmetic Relaxation Process," Robert M. Haralick, John L. Mohammed, and Steven W. Zucker, Computer Graphics and Image Processing 13, 1980, pp 242-256.
- [Horn 75] "Obtaining Shape from Shading Information," B. K. P. Horn, The Psychology of Computer Vision, P. H. Winston, Ed., McGraw-Hill, New York, 1975, pp 115-155.
- [Horn 77] "Understanding Image Intensities," Berthold K. P. Horn, Artificial Intelligence 8, 1977, pp 201-231.
- [Hu 62] "Visual Pattern Recognition by Moment Invariants," M-K Hu, IRE Trans. on Information Theory, 1962, pp 179-187.
- [Julesz 71] Foundations of Cyclopean Perception, Bela Julesz, 1971, The University of Chicago Press, pp 142-185.
- [Kanal 81] "Digital Registration of Images from Similar and Dissimilar Sensors," Lavean N. Kanal, Barbara A. Lambird, David Lavine, and George C. Stockman, Proc. of the Int. Conf. on Cyb. and Society, 1981, pp

347-351.

- [Kuschel 82] "Augmented Relaxation Labeling and Dynamic Relaxation Labeling," Stephen A. Kuschel and Carl V. Page, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, No. 6, Nov. 1982, pp 677-682.
- [Leckie 80] "Use of Polynomial Transformations for Registration of Airborne Digital Line Scan Images," D. G. Leckie, 6th Canadian Symposium on Remote Sensing, 1980, pp 635-641.
- [Levine 73] "Computer Determination of Depth Maps", Martin D. Levine, Douglas A. O'Handley, and Gary M. Yagi, Computer Graphics and Image Processing 1973, Vol. 2, pp 131-150.
- [Marr 76] "Cooperative Computation of Stereo Disparity", D. Marr and T. Poggio, Science, Vol. 194, Oct. 1976, pp 283-287.
- [Marr 78] "Representation and Recognition of the Spatial Organization of Three-dimensional Shapes," D. Marr and H. K. Nishihara, Proc. Royal Society of London, B. 200, 1978, pp 269-294.
- [Marr 79] "A Computational Theory of Human Stereo Vision," D. Marr and T. Poggio, Proc. R. Soc. Lond., 1979, B. 204, pp 301-328.
- [Moravec 81] "Rover Visual Obstacle Avoidance," Hans P. Moravec, 7th Int. Joint Conf. on Artificial Intelligence, 1981, pp 785-790.
- [Mori 73] "An Interactive Prediction and Correction Method for Automatic Stereocomparison," Ken-ichi Mori, Masatsugu Kidode, and Haruo Asada, Computer Graphics and Image Processing 2, 1973, pp 393-401.
- [Nagao 73] "Automatic Model Generation and Recognition of Simple Three-Dimensional Bodies," Makoto Nagao, Shigeo Hashimoto, and Toshiyuki Sakai, Computer Graphics and Image processing 2, 1973, pp 272-280.
- [Nevatia 76] "Depth Measurement by Motion Stereo", Ramakant Nevatia, Computer Graphics and Image Processing 5, 1976, pp 203-214.

- [Nevatia 77] "Description and Recognition of Curved Objects," Ramakant Nevatia and Thomas O. Binford, Artificial Intelligence 8, 1977, pp 77-98.
- [Ohlander 78] "Picture Segmentation Using a Recursive Region Splitting Method," Ron Ohlander, Keith Price, and Raj Reddy, Computer Graphics and Image Processing 8, 1978, pp 313-333.
- [Oshima 79] "A Scene Description Method Using Three-Dimensional Information," M. Oshima and Y. Shirai, Pattern Recognition, Vol. 11, 1979, pp 9-11.
- [Panton 78] "A Flexible Approach to Digital Stereo Mapping," Dale J. Panton, Photogrammetric Engineering and Remote Sensing, Vol. 44, No. 12, Dec. 78, pp 1499-1512.
- [Pavlidis 78] "A Review of Algorithms for Shape Analysis," T. Pavlidis, Computer Graphics and Image Processing, Vol. 7, 1978, pp 243-258.
- [Pavlidis 80] "Algorithms for Shape Analysis of Contours and Waveforms," T. Pavlidis, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, NO. 4, July 1980, pp 301-312.
- [Peet 75] "The Use of Invariants in the Transformation and Registration of Images," F. G. Peet, The Canadian Surveyer, Vol. 29, No. 5, Dec. 75, pp 501-506.
- [Peli 81] "An Algorithm for Recognition and Localization of Rotated and Scaled Objects", Tamar Peli, Proceedings of the IEEE, Vol. 69, No. 4, April 1981, pp 483-485.
- [Persoon 77] "Shape Description Using Fouries Descriptors," E. Persoon and K. S. Fu, IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-7, No. 3, March 1977, pp 170-179.
- [Pingle 75] "A Fast, Feature-Driven Stereo Depth Program", Karl K. Pingle and Arthur J. Thomas, Stanford Artificial Intelligence Laboratory, Computer Science Department, May 1975, Technical Report No. STAN-CS-74-462.
- [Price 76] "Change Detection and Analysis in Multi-spectral Images", Keith E. Price, Ph.D. Thesis, 1976, Dept. of Computer Science, Carnegie-Mellon University.

- [Price 79] "Matching Segments of Images," Keith Price and Raj Reddy, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, No. 1, Jan. 1979, pp 110-116.
- [Richard 78] "NS001MS-Landsat-D Thematic Mapper Band Aircraft Scanner," R. R. Richard, R. F. Merkel, and G. R. Meeks, Proc. 12th Int. Sym. on Remote Sensing of Environment, 1978, pp 719-727.
- [Rosenfeld 76] "Scene Labeling by Relaxation Operations," Azriel Rosenfeld, Robert A. Hummel, and Steven W. Zucker, IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-6, No. 6, June 1976.
- [Rosenfeld 77a] "Coarse-Fine Template Matching", Azreil Rosenfeld and Gordon J. Varderburg, IEEE Trans. on Sys., Man, and Cybern., Feb. 1977, pp 104-107.
- [Rosenfeld 77b] "Iteractive Methods in Image Analysis," Azriel Rosenfeld, Proc. Pattern Recognition and Image Processing, 1977, pp 14-18.
- [Schutte 80] "Scene Matching with Translation Invariant Transforms," Holger Schcutte, Stephen Frydrychowicz, and Johannes Schroder, Proc. 5th Int. Conf. on Pattern Recognition, 1980, pp 195-198.
- [Selfridge 82] "Reasoning About Success and Failure in Aerial Image Understanding," Peter G. Selfridge, Ph.D. Dissertation, Dept. of Computer Science, Univ. of Rochester, May 1982.
- [Shapira 77] "Reconstruction of Curved-Surface Bodies from a Set of Imperfect Projections," Ruth Shapira, 5th Int. Joint Conf. on Artificial Intelligence, 1977, pp 628-634.
- [Shirai 71] "Recognition of Polyhedrons with a Range Finder," Yoshiaki Shirai, Motoi Suwa, Proc. 2nd Int. Joint Conf. on Artificial Intelligence, 1971, pp 80-87.
- [Siegel 82] "Parallel Processing Approach to Image Correlation," Leah J. Siegel, Howard Jay Siegel, and Arthur E. Feather, IEEE Trans. on Computer, Vol. C-31, No. 3, March 1982, pp 208-218.

[Stockman 78] "Image Registration from Edge Content", George Stockman and Steve Kopstein, 8th Annual Automatic Imagery Pattern Recognition, April 1978, pp 139-157.

[Stockman 82] "Matching Images to Models for Registration and Object Detection via Clustering", George Stockman, Steve Kopstein, and S. Benett IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, No. 3, May 1982.

- [Stockman 83] "Point Pattern Matching by Clustering," George Stockman, Private Communication, March 1983, Dept. of Computer Science, Michigan state University.
- [Svedlow 76] "Experimantal Examination of Similarity Measures and Preprocessing Methods Used for Image Registration", M. Svedlow, C. D. McGillem, and P. E. Anuta, Symp. Machine Processing of Remotely Sensed Data 1976.
- [Tanimoto 81] "Template Matching in Pyramids," Steven L. Tanimoto, Computer Graphics and Image Processing 16, 1981, pp 356-369.
- [Teillet 81] "Forest Classification Using Simulated Landsat-D Thematic Mapper Data," P. M. Teillet, B. Guindon, and D. G. Goodenough, Canadian Journal of Remote Sensing, Vol. 7, No. 1, July 1981, pp 51-60.
- [Underwood 75] "Visual Learning from Multiple Views," Stephen A. Underwood and Clarence L. Coates Jr., IEEE Trans. on Computers, Vol. C-24, No. 6, June 1975, pp 651-661.
- [Van Wie 77] "A Landsat Digital Image Rectification System," Peter Van Wie and Maurice Stein, IEEE Trans. on Geoscience Electronics, Vol. GE-15, No. 3, July 1977, pp 130-137.
- [Vanderburg 77] "Two-Stage Template Matching," Gordon J. Vanderburg and Azriel Rosenfeld, IEEE Trans. on Computers, Vol. C-26, No. 4, April 1977, pp 384-393.
- [Weszka 79] "Histogram Modification for Threshold Selection", Joan S. Weszka and Azreil Rosenfeld, IEEE Trans. on Sys., Man, Cybern., Vol. SMC-9, Jan. 1979.

- [Wong 78] "Scene Matching with Invariant Moments," Robert Y. Wong and Ernest L. Hall, Computer Graphics and Image Processing 8, 1978, pp 16-24.
- [Yakimovsky 78] "A System for Extracting Three-dimensional Measurements from a Stereo Pair of TV Cameras," Y. Yakimovsky and R. Cunningham, Computer Graphics and Image Processing 7, 1978, pp 195-210.
- [Yam 81] "Image Registration Using Generalized Hough Transform", Simon Yam and Larry S. Davis, Proc. Pattern Recognition and Image Processing, 1981, pp 526-533.
- [Zahn 72] "Fourier Descriptors for Plane Closed Curves", Charles T. Zahn and Ralph Z. Roskies, IEEE Trans. on Computers, Vol. C-21, No.3, March 1972, pp 269-281.
- [Zahn 74] "An Algorithm for Noisy Template Matching," Charles T. Zahn, Jr., IFIP Congress, Aug. 5-10, 1974, Stockholm, pp 698-701.
- [Zucker 76a] "Region Growing: Childhood and Adolescence," S. W. Zucker, Computer Graphics and Image Processing 5, 1976, pp 382-399.
- [Zucker 76b] "Relaxation Labeling and the Reduction of Local Ambiguity", Steven W. Zucker, Int. Joint Conf. on Pattern Recognition, 1976, pp 852-861.
- [Zucker 77] "An Application of Relaxation Labeling to Line and Curve Enhancement," Steven W. Zucker, Robert A. Hummel, and Azriel Rosenfeld, IEEE Trans. on Computers, Vol. C-26, No.4, April 1977, pp 394-403.
- [Zucker 78a] "A Hierarchical Relaxation System for Line Labeling and Grouping," Steven W. Zucker and John L. Mohammed, Proc. Pattern Recognition and Image Processing, 1978, pp 410-415.
- [Zucker 78b] "Relaxation Processes for Scene Labeling: Convergence, Speed, and Stability," Steven W. Zucker, E. V. Krishnammarthy, and Robert L. Harr, IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-8, No. 1, Jan. 1978, pp 41-48.
- [Zusne 70] Visual Perception of Form, C. T. Zusne, Academic Press, 1970, pp 206-211.