1-1-1992

Geospecific Databases: Final Report

Brian S. Blau

INSTITUTE FOR SIMULATION AND TRAINING

Contract Number N61339-90-C-0042
PM TRADE

February 12, 1992

Geospecific Databases

Final Report
Visual Systems Laboratory

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando FL 32826

University of Central Florida
Division of Sponsored Research

IST-TR-92-13

INSTITUTE FOR SIMULATION AND TRAINING

Geospecific Databases
Final Report

The body of this report is VSL Document VSLM92.6. This document is the Final Report of a two year project under Contract N61339-90-C-0042, sponsored by the Army Project Manager for Training Devices (PM TRADE).
All opinions herein expressed are solely those of the authors.

Contract N61339-90-C0042
PM TRADE

February 12, 1992
Visual Systems Laboratory
IST-TR-92-13

Prepared by

Brian Blau, Project Manager _____

Kevin Bowyer, Mubarak Shah, J. Michael Moshell

Reviewed by

Brian Goldiez _____

Institute for Simulation and Training • 12424 Research Parkway, Suite 300 • Orlando, Florida 32826

University of Central Florida • Division of Sponsored Research

FINAL REPORT

Geospecific Databases

Brian Blau, Kevin Bowyer, Mubarak Shah, J. Michael Moshell

(The body of this Report is VSL Document 92.6)

Visual Systems Laboratory
Institute for Simulation and Training
University of Central Florida
Orlando, FL 32816

12 February 1992

Abstract

This document is the Final Report of a two year project at the Institute for Simulation and Training supported by PM-TRADE under contract #N61339-90-C-0042.

The purpose of this project were

1) to construct a "geospecific data center" consisting of software tools for the construction of visual databases; and

2) to evaluate and extend techniques for the extraction of information about buildings from aerial photography, in support of the rapid production of visual simulation databases.

These databases are required for time-critical mission rehearsal. Two techniques were selected for study: stereo pair analysis, and shadow analysis. Two teams of researchers in the computer science departments of the University of Central Florida and the University of South Florida undertook parallel projects. They constructed prototypes and evaluated various techniques. The results were brough back to the Institute for Simulation and Training and used to construct a demonstration database to show the feasibility of the methods used.

Outline

1. Purpose

The overall purpose of this project was to investigate techniques which are potentially useful in service of the goal of rapidly producing visual databases of specific geographic regions. These so-called "geospecific databases" are necessary for simulation-based mission rehearsal. They contrast with "generic databases", which are the most common form of simulator database for training. A generic database conveys the overall flavor of a particular kind of terrain (e. g. desert; temperate zone forest, etc.) without referring to (or requiring) the integration of timely real-world data.

In order to work in this area, a substantial collection of tools and resources had to be gathered. The construction of the Geospecific Data Processing Center is described in the following section.

2. The Geospecific Data Processing Center

Overall Activities

The activities of this project can be described in four broad areas: the acquisition of skilled personnel in key technical areas to support the Geospecific Data Processing Center (GeoData Center); the acquisition of hardware and software to support the complex activities in rapid production of databases; automatic production of terrain databases and technology transfers to industry.

Skilled Personnel. The Visual Systems Laboratory (VSL), in support of the Geospecific Data Processing Center has staffed itself with highly qualified personnel which are trained in the speciality areas of computer graphics, computer vision and image processing. Dr. Michael Moshell, who leads VSL has 20 years experience in the computer and physical sciences. Brian Blau, Curt Lisle and Ron Klasky have their Masters of Science in Computer science and each has experience in computer graphics. Klasky has additional experience in image processing and astronomy and Lisle has additional experience in low-level design of computer image generators.

The quality of the personnel listed above is enhanced by a pool of students from the Computer Science Department at the University of Central Florida. This department has a nationally recognized program with excellence in parallel computation and VLSI design. Most recently, their student computer programming team placed fifth in an internationally recognized programming contest. One of the programing team members is now employed at VSL.

Hardware and Software at the GeoData Center. During 1990, the VSL acquired software and hardware necessary to support the complex tasks involved in the production of databases. The following list summarizes the capabilities of the GeoData Center :

Image Generators

These machines have the capability to display geographic databases in three dimensions at real-time update rates.

VSL has the **ESIG 500** as its highest powered rendering engine. Databases from multiple sources can be built on support machines and then downloaded to the ESIG for display. Additionally, **SIMNET's Image Generator** from Delta Graphics is available for viewing more customized models and databases

The **Silicon Graphics Power Series** is a new addition to VSL which will enhance the capability to rapidly see low fidelity simulations. This is an excellent platform for visualization of low detail and low band width databases. Because this platform is a workstation, it is easily reconfigurable and will host many different software systems.

The **Sense8 WorldToolkit** uses **Intel DVI** technology to provide phototextured imagery in low-cost (PC) computing equipment. This cost/performance level was unanticipated at the outset of this project; its arrival allowed some nice demonstrations.

Image Processing Software

The **GRASS** image processing system, in conjunction with the **KURTA** digitizing tables, is being used to generate IG databases. Sources include DMA DTED/DFAD, satellite images and maps. These sources are combined in GRASS and written to a script file for use in the **ANIM** object animation system. Additional routines exist for exporting this data to the **Multigen** system, which is a tool for building image generator databases.

Modeling Software

The software packages **MultiGen, ElectroGig, AutoSolid, Alias, S1000** and **Geometric Modeling System** are useful when building custom image generator database models. Specifically, S1000 and Multigen has been used to build models for various industrial partners and projects tasks. These packages give the GeoData Center an ability to quickly build databases to customer specifications. These databases can then be transformed into the appropriate formats to be displayed.

Additional Software and Hardware

The VSL has two Sun Sparc workstations for use in the GeoData Center. These machines are general purpose UNIX computers that run at very high clock speed. This gives the GeoData Center the ability to perform complex tasks required for database construction.

Additionally, VSL has the ability to read and display **DMA DTED/DFAD** databases. Many of the software packages mentioned above have this feature which gives the GeoData Center additional capabilities to build geospecific databases.

Automation of Terrain Database Production. One of the main goals of this project is to determine unique and efficient methods for automatic and semi-automatic constructions of databases for image generators in support of some standard database interchange format. The GeoData Center has the capability to build databases in several formats and continues to add flexibility to this process.

Using the Multigen and GRASS systems mentioned above, the GeoData Center can take raw input data, such as DMA DTED/DFAD, remote sensing, maps and photos and create individual databases which are at first specific to the hardware and software platform. Additional support comes from human modelers, which is now the only way to actually integrate all of the data.

Additionally, VSL has developed a series of software packages which can extract useful 3D information from images. One package can recover depth from stereo images and the other package can recover building information from shadows. Both of these packages have been documented in the monthly reports for this project.

Technology Transfer

GeoData Center Technology Transfer: The GeoData Center is a toolset which creates computer representations of regions of geospecific terrain. Digitized terrain technology is attractive to a variety of industrial companies and government organizations. VSL has relationships which will make this technology available to them.

VSLhas established a relationship with the Army Topographic Engineering Center (TEC) where we will be cooperating with them in the development of geospecific database technology for simulators. VSL has already been funded by Martin Marietta to produce software prototypes which realistically display geospecific terrain in a simulator using **Continuous Levels of Detail** to minimize the number of polygons needed while preserving terrain features like ridge lines, contours, etc. VSL will be serving as a demonstration site for Project 2851 standardized databases

as they become available over the next eighteen months -- providing the simulator industry with an independent organization with expertise in this future database format. Project 2851 format databases are required for all new military training contracts involving Computer Image Generators.

3. Data Integration and Management System

Overall Activities. The primary goal of this project is to provide a way to increase the productivity of human database modelers, especially those working with low-cost image generators. Accomplishing this goal involves the introduction of several new technologies which are usually not associated with simulator database construction. These technologies include Geographical Information Systems (GIS) and computer vision. These new technologies will be used to support the Geospecific Data Processing Center, an organization within IST which will include skilled personnel, software and hardware necessary for the rapid integration of digital geographic data, photographs, including stereo pairs, maps and charts.

Over the course of the project, there have been a number of specific sub-projects that were designed to get a better understanding of the simulator construction process. This included analysis of off-the-shelf software and hardware that already has taken this technology to the forefront. Implementations of stereo processing and shadow analysis helped in the evaluation of the main goal of this project. Additionally, there were consultations with industry officials who have insight into this arena of work. Their help along with the work done by Dr. Mubarak Shah at the University of Central Florida and Dr. Kevin Boyer at the University of South Florida has contributed to the success IST understanding of the simulator construction process.

Accomplishments

During the first six months of this project, there was a concentrated effort to define the overall scope of this project, as well as define important goals for the coming year. The following is a list of accomplishments which took place during the first year :

- **SIMNET Database** : Programs were written to read the SIMNET database. This is the plan view database which is located on the Mascomp computer. This effort was very successful, the data gathered from this experiment was valuable in many different aspects of this project. The code written was also used in other spin-off projects (Virtual Reality Demo @ I/ITSC 90, VSL's bulldozer and car on SIMNET terrain demo, Blau's masters thesis).

  Demonstrations of the bulldozer on SIMNET terrain done at VSL during first quarter, 1991, used dynamic and

microterrain techniques developed at VSL. Demonstration of another derivative of SIMENT software was done at the I/ITSC 91 Conference in fourth quarter 1991.

Problems in this area came about because VSL was unable to have access to the SIMNET CIG database. Dynamic terrain was to be put on SIMNET, but because of restricted access to the CIG database, work stopped in this area.

- **Univ. of Waterloo** : An image processing tool kit was acquired at nominal cost. Uses include general image processing functions and display functions. Documentation and source code is available.

- **Stereo Extraction** : Dr. Mubarak Shah and graduate students from the Computer Science department of the University of Central Florida started work on extracting image depth from stereo images. He specifically concentrated on using off-the-shelf algorithms as a working introduction to this field. The results are reported in Appendix A.

- **Shadow Extraction** : Dr. Kevin Boyer and graduate students from the Computer Science department of the University of South Florida started work on extracting building height from shadows. Buildings cast shadows in an image, and through some geometry and computer vision techniques, the building height can be calculated. The results of this work are reported in Appendix B.

- **Literature Review** : Annotated bibliographies were constructed in the areas of stereo extraction and shadows analysis. These documents were submitted with the Quarterly report in June 1991.

- **Industrial Contacts and Logistics** : Information was gathered from the following sources: DMA data and the hardware necessary to read tapes, David McKeown at Carnegie Mellon for information about shadow extraction and images

The work done in the first six months defined clear goals that meet the expectations of this project. Specifically, the process of taking multiple inputs (ie: digital geographic data, single photographs, stereo images, maps, charts) can be integrated together for form accurate simulator terrain databases.

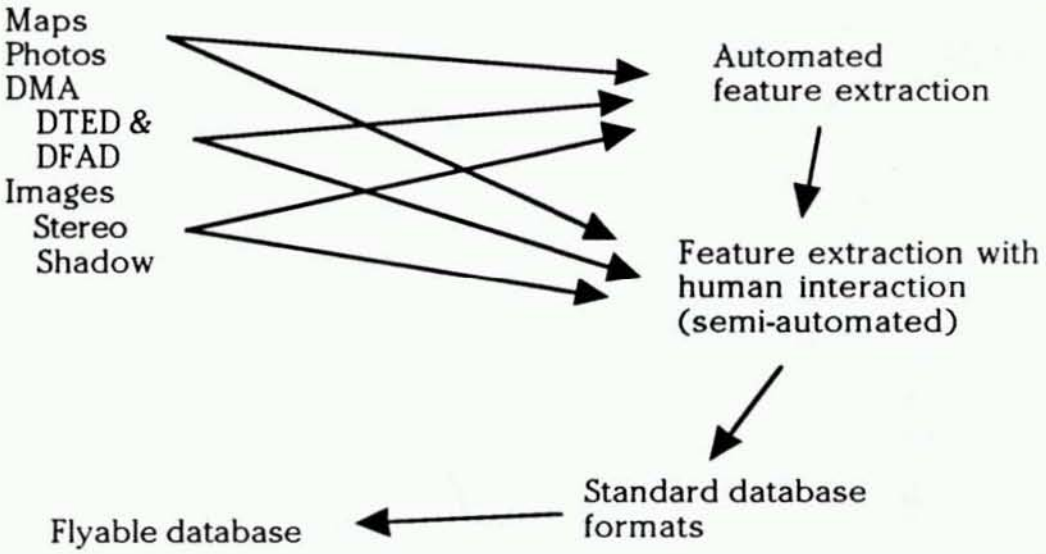


Figure 1. Path from raw data to flyable database

During the last six months of the first year of the project, efforts continued in the areas described above. The implementation was completed for the stereo extraction and shadow extraction algorithms. Additionally, industrial contacts were made during this period. The following is a specific list of work accomplished during the last half of the first year :

- **Stereo Extraction** : Dr, Shah and his students implemented three separate stereo extraction techniques. The output from all three is an image of depth values, where at an (x, y) location there is a corresponding depth. The problem here is that the (x, y) values are image coordinated, not actual physical values.

  Normalized correlation method by Cochran and Medioni matches a point in the left image and right images and computes the disparity, which is then translated into height.

  Sum of absolute difference method by Kayalap uses the difference of the correlation of points in the left and right images to compute the disparity. Again depth can be obtained from disparity.

  Prazdny's method computes the correlation only at edge points in the image and it is based on a smoothness criterion. The disparity is computed at the edge points and then translated to depth values.

- **Shadow Analysis** : Dr. Kevin Boyer and his students have completed implementation of an interactive software package which lets the user choose a building to analyze. First the user is presented with an image which contains buildings, they are

then instructed to trace a building and then its shadow. The height of the building is then computed and stored in a database file.

- **CAD Package for use with Stereo algorithms** : Chuck Campbell and Brian Blau have developed an interactive CAD style software package to interpret the output from Shah's algorithms. This software provides a view of the depth fields as a grid of posts, where the height of the posts correspond to the disparity between the left and right images. The user is then able to construct polygons using the tops and bottoms of the posts. The output of this program is a database of polygons.

- **Animated Fly Through** : The output from the interactive CAD software and the shadow extraction are inputs to Curt Lisle's ANIM Object Animation package. This software can be viewed as a generic object rendering engine. It is used to fly through the data that was created by both the stereo extraction and shadow analysis.

  The stereo depth extraction, shadow analysis, CAD interactive software and animated fly through were documented and demonstrated for the sponsors in January 1991.

- **Industrial Contacts and Logistics** : There has been contact with General Electric about their TARGET database construction toolkit. At the present time, IST is attempting to establish a working relationship which will benefit both organizations.

  Images in both digital and picture form are being collected by VSL to support the GeoData Center. Along with this, continued research is being done in the areas of cartography and satellite imaging.

The following is a diagram of how the different aspects of database production fit together. It shows how stereo extraction combined with shadow analysis are combined through software and human interaction into a simulator database.

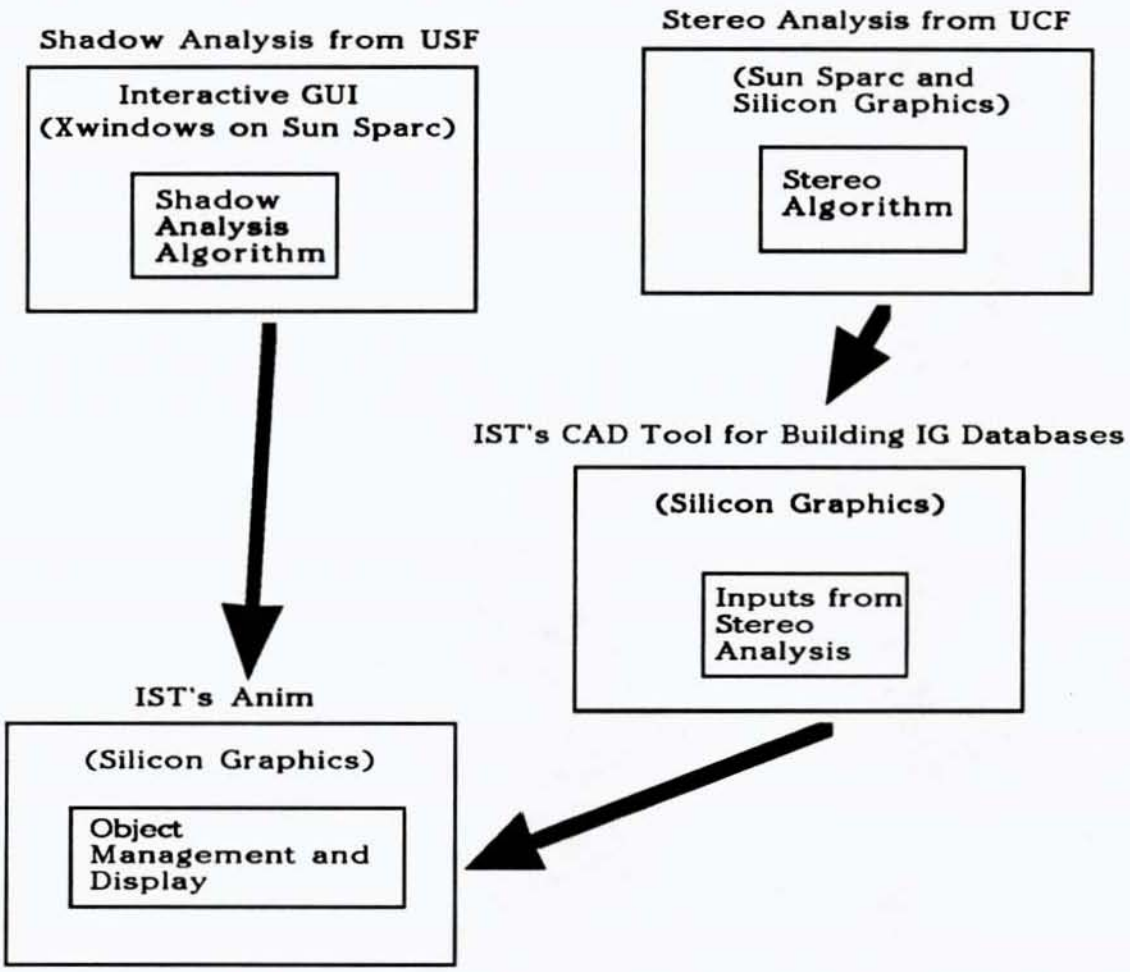


Figure 2. Stereo analysis, shadow analysis and database flythrough

Integration Management System. The original design for the Geospecific Data Center included a semiautomatic integration management system which would track database components and bring them together. This proved infeasible to construct, because of the large number of decisions that had to be made during each database's construction. The components provided by the sub-teams were "stand-alone" software systems, and were best used as individual tools.

Lessons learned in this project would be useful in the construction of a semiautomatic integration management system. However, the magnitude of such an effort far exceeds the scope of this small project .

4. Sample Database Production

Overview. During the first year of the project, there were two significant software developments which enabled the Visual Systems Laboratory (VSL)

to build a standard database. These two software packages were built by Dr. M. Shah and Dr. K. Boyer of UCF and USF respectively.

Stereo and Shadow Analysis. Dr. Shah and his students have built a software system which when given a stereo pair, can determine the depth of the scene. Additionally, Dr. Boyer and his students have built a software system which when given an image of buildings and shadows can determine the height of those buildings. Both of these software packages have been documented in the monthly reports for this project.

Demonstration. This software was used, along with the commercial packages Multigen and S1000 to build the sample database. On January 25 1991, VSL presented a demonstration of the capabilities which were developed during the first year of the project. Using both commercial software and applications developed for this project, the data flow from raw images to image generator database was shown.

During the second year, an additional database was built according to the guidelines proposed in this project. This database will be displayed during the final Project Review in March, 1992.

5. Conclusions

1. The two techniques (stereo and shadows) were both capable of providing information about the height of a building. However, neither of the subcontracting teams were willing to hypothesize a mechanism which would reliably find buildings by automatic means. Instead, both teams provided prototypes with a substantial manually operated user interface.

2. Shadow information was generally easier to use than stereo information, in actual hands-on efforts to build databases of simple rectangular buildings. This was because the shadow measurement techniques provided an unambiguous reference for the height of the significant roof-line, whereas the stereo-pair algorithms produced highly noisy height fields.

3. The two techniques can be used to complement one another, but will require the construction of numerous experimental user interfaces before really useful tools will result.

4. For the foreseeable future, the construction of urban databases from aerial photographs will remain a highly manual operation, with at most low-level assistance from software tools.

Appendix A:

Using Shadows to Estimate Properties of Buildings in Aerial Photographs

Maha Sallam, Kevin Bowyer
University of South Florida

USING SHADOWS TO ESTIMATE PROPERTIES OF BUILDINGS IN AERIAL PHOTOGRAPHS [1]

Maha Sallam and Kevin Bowyer
Department of Computer Science and Engineering
University of South Florida
Tampa. Florida 33620
sallam or kwb@csee.usf.edu

[1]This project was supported in part by the Army's Project Manager for Training Devices (PM-TRADE). However, the opinions expressed herein are solely those of the authors.

Abstract

The science of aerial photography has advanced greatly in the past few years and much of the equipment used in taking the actual photographs has become highly automated. However, the art of aerial photograph interpretation is still highly manual and primitive in comparison. As a step towards automating the interpretation process, this research work is aimed at investigating the task of automatically finding and estimating the properties of buildings which typically exist in aerial photographs. In monocular aerial photographs, shadows comprise the major source of information about the properties, and in some cases also the existence, of raised structures in the scene. Because of this, a significant part of this work has been devoted to finding shadow regions in images. The detected shadow regions and other edge information extracted from the image are then used to estimate the building properties.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Image Interpretation

Finding a way of automatically extracting the information available in two dimensional images can save an enormous amount of time and effort. The most common and least expensive type of digitized images used to provide a computer with a real world scene are intensity images. These are images which encode the intensity of the light reflected off surfaces in the scene at each point in a finite set of sample surface points into binary strings. This form of scene representation is easy to store and manipulate by the machine. However, it is also usually more difficult to analyze than other forms, such as range images, which contain explicit depth information [3].

Automatic gray-scale image interpretation has traditionally been an important and complicated problem in the field of computer vision. The main difficulties in performing this task lie in isolating the the various regions of interest from the background in the image, and in identifying the items in the actual scene which the isolated areas represent. Typically, the human eye and brain have the ability to identify the items in a two dimensional image despite the fact that some depth information is missing. This goal is what many researchers have been trying to make computers achieve in image interpretation. A two dimensional image contains depth information in the form of shading, texture or shadows. This information can be used to give a more precise description of a particular item in the image, which will increase the probability of identifying that item correctly.

1.2 Aerial Images

Aerial images are typically gray-scale images which are obtained by means of photographing a geographical area of interest from an airplane. Such images have been used and analyzed for many years to serve a variety of purposes. The most traditional uses for these images are in military surveillance, vegetation monitoring, soil and water testing and map drafting [1, 8]. Aerial images are also used in building models for flight and land maneuvering simulators. In this area of modeling and simulation, aerial images often comprise the only source of information about a real world location for which a model is needed.

The problem of automating the process of aerial photograph interpretation, though difficult, is not impossible to solve. Indeed, numerous researchers have already explored many different aspects of the problem [7, 9, 10, 11, 13]. Many of the basic image processing and interpretation techniques used in Computer Vision can be applied to aerial photographs, especially photographs of urban and suburban areas. However, there are certain aspects which are unique to aerial images. One of these aspects is the fact that all items in the scene are represented by their top view in the image. Another important characteristic is the shadow regions which exist in most aerial photographs due to fact that they are usually obtained during good weather conditions.

1.3 Use of Shadow Information

Shadows play an important role in the process of interpreting two dimensional images. They have been used to serve several purposes, from detecting the existence of three dimensional objects to estimating fairly accurate qualitative and quantitative features of objects in an image [9, 10, 11, 13].

The difficulty of interpreting two dimensional images of three dimensional objects lies mostly in losing the third dimension information. Shadows can be considered to be the projection of this third dimension onto the plane of the two dimensional view, and can be used to extract object features that are not explicitly available in the two dimensional image. For shadows to be of value in calculating the properties of the three dimensional objects that generate them, information must be available about the source of illumination causing these shadows. In aerial images, the source of the illumination causing shadows is the sun. The geometry of shadows cast by sun illumination is well understood. By knowing the date, time and location of an aerial photograph, the elevation of the sun can be accurately calculated. The sun elevation can then be used to define the relationship between objects and the shadows they cast, as we will explain in the next chapter.

Shadows play an important role in aerial image interpretation. In single aerial images, shadows comprise the most useful source of information about the three dimensional properties of raised structures. A raised structure in a two dimensional image can be distinguished from markings on the ground by the fact that it casts a shadow that also appears in the image. Shadows are also useful in providing information pertaining to the shape of such structures. When the direction of the light source is known, we can use the shape of the shadow cast by a particular building to reconstruct the shape of the building itself. Typically, only about half of the top edges of the building cast corresponding shadow edges. This means that we may have to extrapolate to find a possible complete building shape.

One other important use of shadows in image interpretation is the use of shadow length in an image to estimate the height of the structure which casts the shadow. This can be done only if image information such as the elevation and direction of the illumination source is available.

1.4 Problem Statement and Overview

This thesis presents a research project which attempts to provide a part of the solution to the problem of aerial image interpretation; namely identifying buildings and their shape and height in such images. Shadows and edge information play the main role in achieving this task. Chapter 2 provides an overview of the basic theory and a discussion of previous work on how shadows can provide information about the objects in an aerial image. We also explore work that used shadows to calculate features of buildings that cannot be directly measured in an aerial image, such as the height of a building. In chapter 3, the stage of shadow extraction is given in detail as well as the idea of establishing "ground truth". Chapter 4 explores a method of finding the actual buildings in the image using edge and shadow information. Finally, Chapter 5 provides a general discussion and summary.

CHAPTER 2
USE OF SHADOWS IN IMAGE UNDERSTANDING

2.1 Theoretical Studies

As general background, this section provides a review of previous theoretical work dealing with the relationship between objects and their shadows in images. The work reviewed in this section is considered "theoretical" because no real images are analyzed. Nevertheless, the ideas on which most of this research was built are the same ones which can be useful in analysis of real images. The input to the algorithms described here is assumed to be either a complete extracted line drawing, or a labeled line drawing that represents qualitative understanding of the scene in terms of the basic shape and position of objects.

In chapter two of the book titled "*The Psychology of Computer Vision*" [17], David Waltz provides a discussion of extracting shadows and reconstructing three dimensional models from their two dimensional line drawings. In this study, a catalog of labels for the possible types of lines and vertices that can exist in a line drawing of polyhedral objects is provided. A set of working programs that attempt to assign a unique label to each line and vertex in a line drawing is also explored. For this particular set of programs, the labeling of the various lines and vertices is based mainly on geometric constraints. Limiting the number of labels that a particular vertex can have leads to imposing constraints on adjacent vertices. By increasing the constraints on the type of label that a vertex can have, the possibility of producing a unique label for this vertex is increased. Shadows in this case present an important

source of additional constraints that can be imposed on lines in the image. Without shadows, the identification of parts of the objects relative to their background becomes impossible.

Another researcher who explored the theme of geometric constraints between objects and their shadows is Shafer in his book "*Shadows and Silhouettes in Computer Vision*" [15]. In this book, the relationship between illumination source positions, objects and the shadows they cast is explored. Shafer started with solving what he called the "Basic Shadow Problem" and went on to provide solutions for progressively more complex scenes. He studied scenes of multiple curved objects and illumination sources. The simplest case, or the "Basic Shadow Problem", occurs when a single polygon casts its shadow on the background due to a single source of illumination. The solution to this problem is the base on which most analysis of the more complicated situations in the book was built. Although this particular work did not deal with real images, it provided several equations and geometric relations that can be valuable when applied to shadow information extracted from real images.

Recently, Singh and Ramakrishna [16] expanded the work done by Shafer and combined shadows and texture to find the shape of three dimensional objects from their two dimensional line drawings. They used the solution to the "Basic Shadow Problem" proposed by Shafer to find the shape and orientation of the various surfaces of a polyhedra based on shadow geometry. They also introduced a method for finding the shape of curved objects by calculating the gradients of points on the surfaces of these objects. Their calculations were based on shadow geometry constraints as well as texture.

2.2 Shadow Analysis in Real Images

Shadow detection should not be viewed as an independent step which precedes object identification in the process of image interpretation. In fact, potential object

structure areas in an aerial photograph are important to verify the existence of shadows in potential shadow areas and vice versa. Shadow extraction is a complicated process which is affected by many variables. To reduce the complexity of the process, researchers have imposed certain restrictions on the types of structures and the intensity of the shadow areas relative to other components in an aerial photograph. Huertas and Nevatia [9, 10] for example, assumed that structures to be detected in an aerial photograph are constructed from rectangular components. This assumption allowed them to find shadows by detecting pairs of perpendicular line segments obtained by some low level edge detection algorithm. Once such corners have been detected, further analysis is performed based on the direction of the sun illumination and the relational positions of pairs of detected corners. The boundaries of the rectangular areas in the image are then found by looking for the groups of corners that could be connected to form a completed area boundary. In this work, shadow areas are not segmented out in the actual image, but rather, shadow information is stored as a part of each corner description. Using the shadow information for each corner, they decide which corners are likely to belong to a structure boundary and which are likely to belong to a shadow boundary.

Irvin and McKeown [11] introduced a more comprehensive approach for using shadows in aerial photographs. They implemented separate algorithms for each of the various stages of the aerial image interpretation process. The reason for dividing the process into modules is to make it possible to combine more than one method for each stage of the interpretation process. For the stage of shadow extraction, they use an algorithm that finds shadow areas using simple thresholding and region growing techniques. An average shadow intensity value is calculated based on the values of the pixels adjacent to all the potential structure areas in the image. Only the pixels that are on the opposite side of the illumination source are considered since this is the side where shadows have to be if they exist. The calculated threshold value is

then used in a region growing process to extract the shadow areas in the image. So far, this method does not impose any restrictions on the shape of structures nor their shadows in the image. Once shadow areas have been extracted, the the shadow boundary segments which are shared with the structure boundary are determined using knowledge of the direction of illumination. These segments are then extended into parallelograms which approximate the shapes of the structures in the image. This step restricts the structures that can be detected in an aerial image to parallelogram shaped structures. Irvin and McKeown also use shadows to group fragments that belong to the same structure in an image but are separated because of segmentation errors. The idea is that there is usually more consistency in color intensity within a shadow area that belongs to a certain building than there is in the building area itself. This reduces the chance for producing false area boundaries within the shadow area during segmentation. Based on this assumption, all the separate segments that are adjacent to the illuminated side of a shadow area can be grouped into one large segment that represents the building which cast the shadow.

Sometimes shadow areas can be used as a source of information without being explicitly extracted in an image. Some researchers found it useful to use shadow information at the level of segmentation and edge detection to produce more reliable structure boundaries in the image. Liow and Pavlidis [13] introduced two methods that used shadow/building edges as an initial source of reliable information for performing edge detection and region growing. The results of these methods are far better than any simple segmentation scheme because they take advantage of the sharp intensity of edges caused by the high contrast between structure areas and adjacent shadows. In their region growing algorithm, they use the pixels adjacent to a sharp edge which separates a structure area from its shadow area. These pixels are used as a representative sample from which they calculate the average intensity and the standard deviation among all pixels that belong to this structure area.

Thompson, Checky and Kaemmerer [18] used "shadow-stereo" to extract object boundaries in two dimensional images. In their analysis of a scene, they use two sets of images. Each set is taken from a different camera position and within each set, images are captured under different sources of illumination. Edges that are raised from the ground, such as those which represent object boundaries, are distinguished from edges that are on the ground using the stereo images. Images taken under various sources of illumination are used to identify true boundary edges which would remain in the same position regardless of the direction of illumination. Other edges that shift in position depending on the direction of illumination are considered to be shadow boundaries. Although this work did not use aerial photographs as a source of input images, it can certainly be applied to aerial images. Of course, we cannot change the direction of the source of illumination (the sun), but we can take several stereo images at different times or even on different dates when the scene location changes its position with respect to the sun.

As we mentioned earlier, shadow analysis cannot be used as an independent source of information in the interpretation of aerial photographs. The use of shadows should be an integrated part of a full aerial image interpretation system. Nagao, Matsuyama and Ikeda introduced such a system in [14]. In this system, all the various regions in the image are extracted and numbered. For each region found, they calculate all of the properties associated with that region, such as average intensity, location, shape, etc. Once these properties have been calculated, they are stored along with the region number in a table which they call "the basic property table". These calculated properties are used to identify the various objects in the actual real world location that the extracted regions in the image represent. In this system, shadows play their traditional role of aiding in identifying raised structures in an aerial image. To extract the shadow regions, they calculate some threshold value of brightness based on the average intensity among all pixels in the image. All regions in the image which have

brightness lower than the calculated threshold are considered to be shadow regions. Once shadow regions have been identified, the regions which are adjacent to them in the direction of the sun and which have common borders with them are considered to be regions that correspond to raised structures in the image.

Most methods which have been implemented so far are dependent on various assumptions and restrictions that limit their performance under various conditions. Due to the differences in contrast, illumination and quality of various aerial images, many systems may consider some dark non-shadow areas to be shadow areas. This problem may cause a false detection of objects or structures.

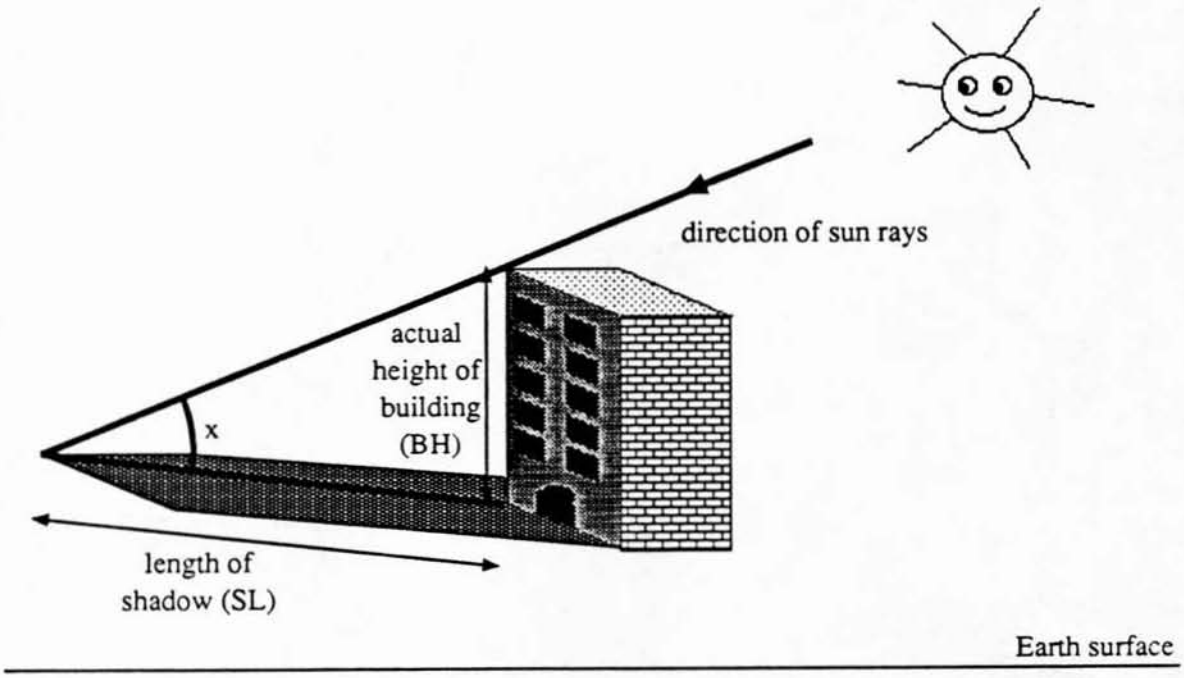
2.3 Determining the Height of a Structure

Scientists have used shadows to accurately determine the height of buildings in aerial images [1, 8, 9, 10, 11]. Figure 1 shows the basic idea of how the height of a building (BH) can be calculated if the angle of the sun elevation with respect to Earth (x) and the length of the shadow (SL) are known. The following equation describes the important relation:

$$BH = SL \times \tan x \tag{2.1}$$

The length of the shadow can be measured directly from the image. In general, finding the angle of the sun elevation requires a somewhat more complicated process. x can be found using one of the following methods:

1. If the height of an object in the image is already known, then by measuring the height of the shadow which this object casts on the background, we can use equation 1 to calculate the angle x. Once the value of x is known, it can be used to calculate other unknown object heights. The problem with this method is that it requires some prior knowledge about the image which is not always available. Also the chance for error in the final result of the calculated height is



If the angle of the sun elevation "x" is known then the actual height of a building can be found using the length of its shadow using the equation:

$$BH = SL \tan(x)$$

Figure 1: A structure height can be determined using shadow length.

rather large since it depends on measuring two shadow height values from the image rather than one.

2. This second method allows the calculation of the sun elevation angle x to be performed independently from the image. If the time, date and location of the photograph are known then x can be calculated, using astronomical tables, from the following relation:

$$\sin x = \cos a \times \cos b \times \cos c \pm \sin a \times \sin b \qquad (2.2)$$

where: a is the latitude of the photograph location, b is the latitude of the sun relative to Earth when the photograph was taken, and c is the difference in longitude between the sun and the photograph location.

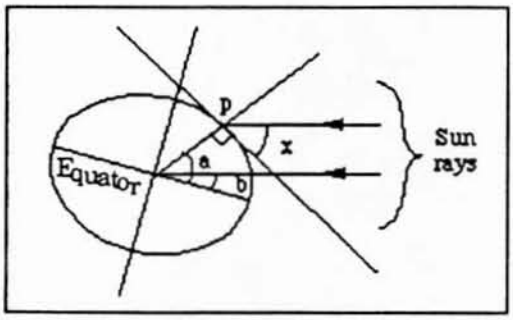
The sign in the equation is a " + " if both sun latitude and the photograph location are on the same side of the Equator, and it is a " − " if they are on opposite sides of the Equator. See Figure 2. This is because the sun is located to the south of the Equator between September 23 and March 21, and it is located to the north of the Equator between March 21 and September 23.

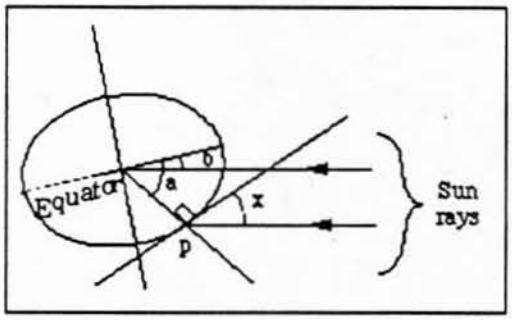
These two methods give accurate results only if the following conditions hold:

1. The heights of the objects for which shadow heights are measured should be vertical with respect the image plane.

2. The entire surface of where the shadows are cast is in the plane of the image.

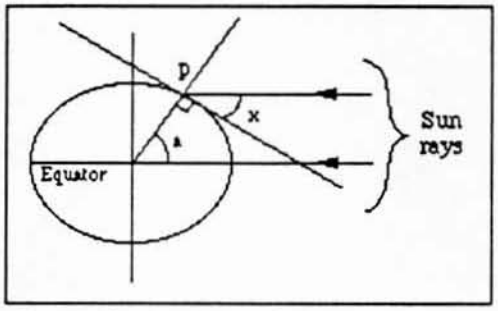
Figure 3 illustrates the problems that arise when one of these two conditions is not met.





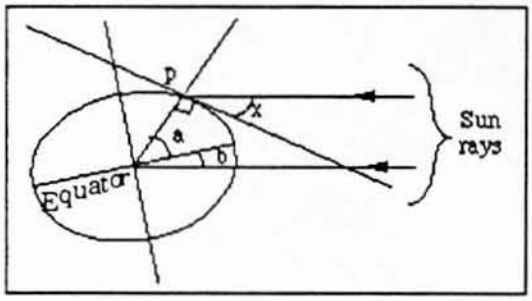
If the sun has some latitude b above the equator (between March 21 and September 23) and point p is north of the equator, or the sun has a latitude b below the equator (between Sptember 23 and March 21) and point p is south of the equator then:

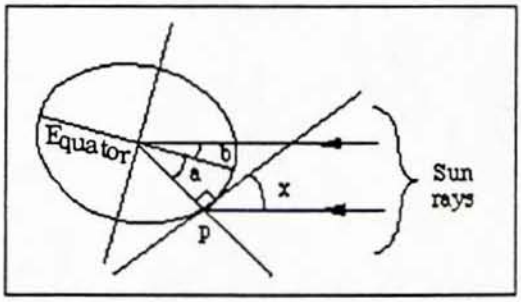
$$\sin(x) = \cos(a-b) = \cos(a)\cos(b) + \sin(a)\sin(b)$$



If the sun has 0 latitude (on March 21 and on September 23) then:

$$\sin(x) = \cos(a)$$





If the sun has some latitude b below the equator (between September 23 and March 21) and point p is north of the equator, or the sun has a latitude b above the equator (between March 21 and September 23) and point p is south of the equator then:

$$\sin(x) = \cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$$

Figure 2: Calculating the angle of the sun elevation "x" at point "p" which has a latitude "a" depends on the time of year.

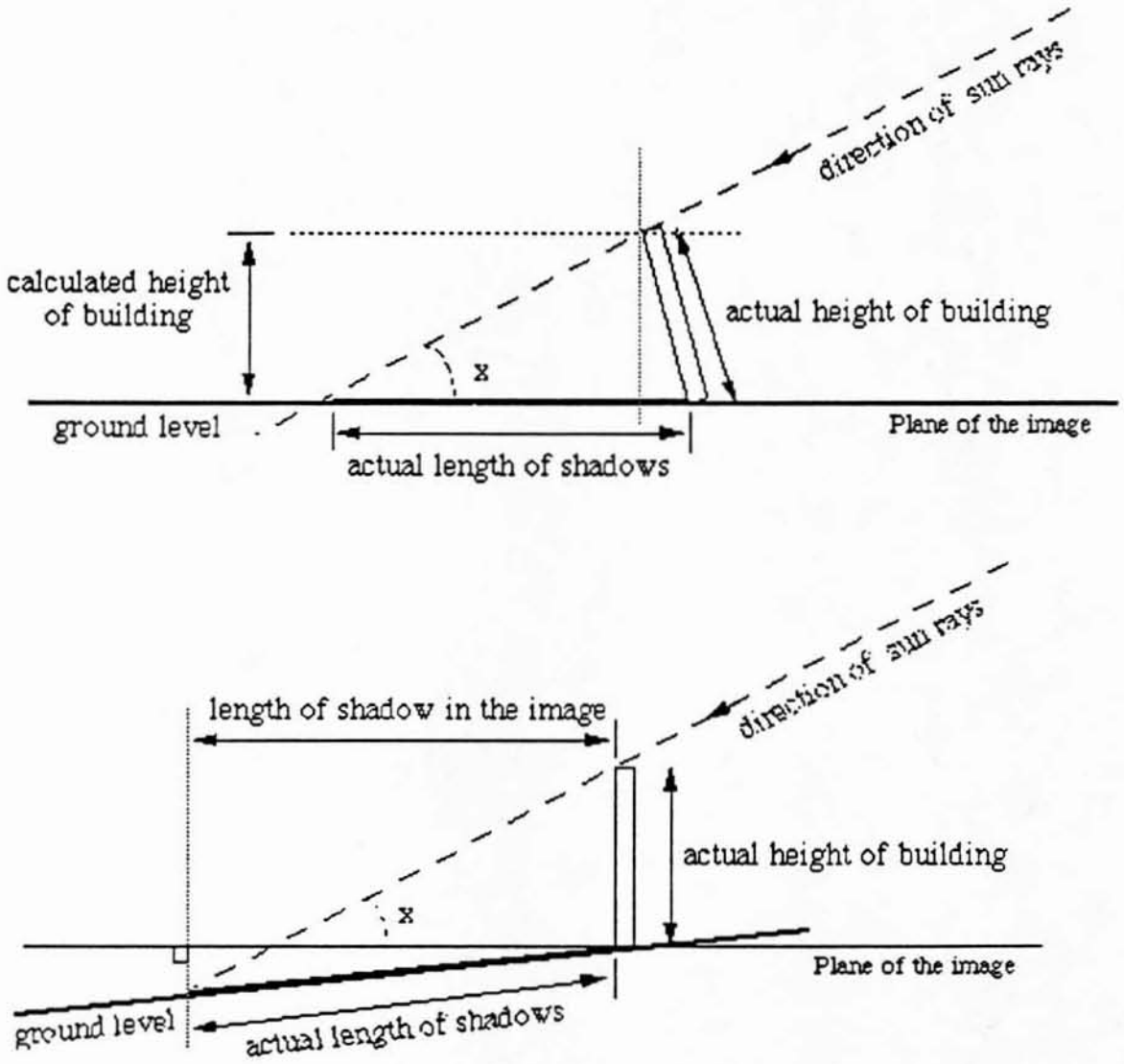


Figure 3: If the shadow is not cast on the plane of the image or the building height is not perpendicular to the plane of the image then the calculated building height will be different from its actual height.

CHAPTER 3
EXTRACTING SHADOW REGIONS

In order to use shadows in height and property estimation of buildings in aerial images, we first need to be able to extract the shadows in the image. Once an appropriate method of shadow extraction is found, the shadows can be used in later stages to calculate structure properties.

The most obvious property of shadow regions in an image is that they are generally darker than other regions. This suggests that one method of extracting shadows is to use thresholding techniques to extract the darker regions in the image. Using a global threshold to determine which pixels belong to dark regions is a relatively simple process. The difficult part of employing such a technique lies in automatically finding a threshold value which will produce good results. It is not very clear how to measure the performance of a region extraction technique. The ideal region extraction algorithm is one which extracts only regions of the required type and all regions of the required type. When a global threshold value is used to extract shadow regions, it seems reasonable to consider a good threshold value to be one which extracts all actual shadow regions in the image while extracting the minimum number of dark non-shadow regions possible. There is generally a trade off between extracting only, but not all, actual shadow regions, and extracting regions which include all shadow regions as well as other regions that happen to be dark. We have chosen to concentrate on obtaining all shadow regions at this point, based on the assumption that it is easier to eliminate non-shadow regions at later stages than it is to find the missing shadow regions.

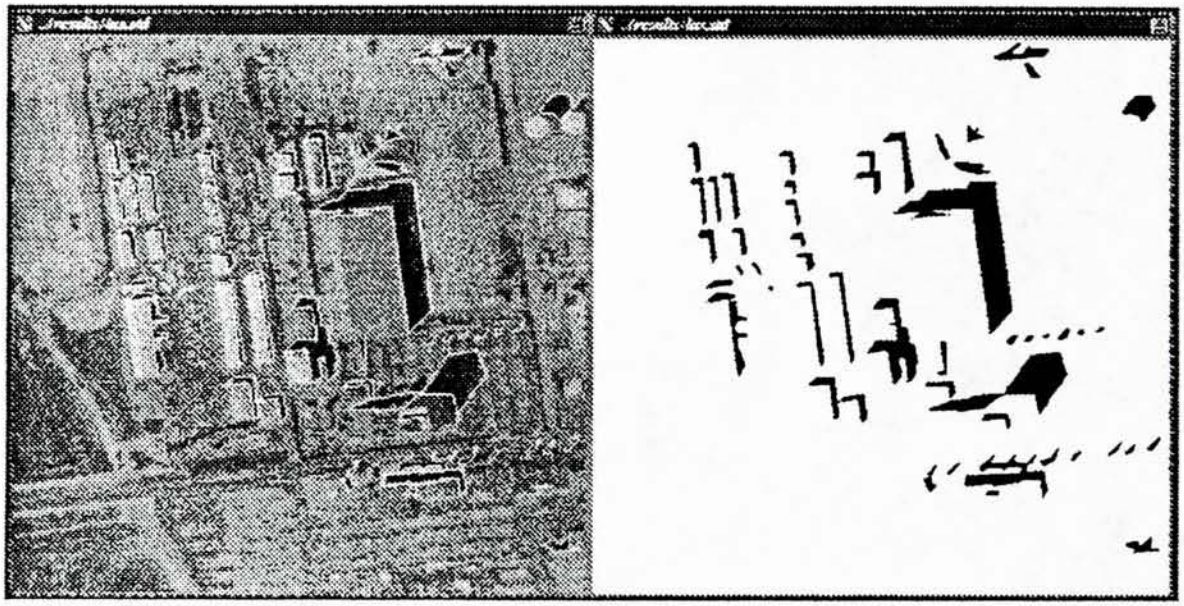
3.1 "Ground Truth" and Evaluating Shadow Extraction Techniques

Any rigorous evaluation of an automated region extraction technique should include some sort of comparison to the ideal performance. Therefore, we first construct a "ground truth" data base for a group of test images. This data base is produced using a high degree of human intervention. Humans have established their ability to interpret two dimensional scenes and they seem to be capable of adequately identifying shadow regions in aerial photographs. In order to represent this human interpretation in a measurable way, an algorithm has been implemented which allows for manual tracing of shadow regions in the image. This algorithm was used to perform manual shadow extraction on 5 images. Figure 4 contains the manually outlined test images and the shadow regions found using the outlines. [1]

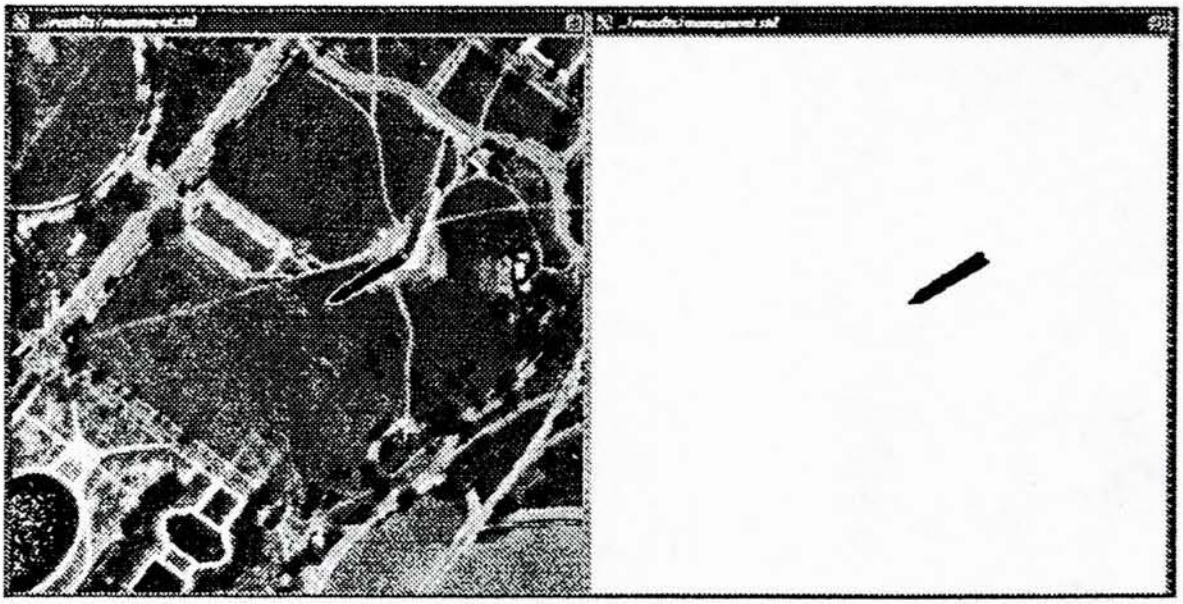
This manual tracing is then automatically interpreted into chains of point coordinates in the image plane that correspond to outlines of the traced shadow regions. Using these boundaries, we produce a data base for each image that contains an entry for each shadow region. This entry describes its corresponding region by the following information:

1. A region id number which is unique within each image.
2. The coordinates of the top-left and bottom-right corners of a bounding box for the region specified.
3. A list of the points which define the actual boundary of the region.
4. The number of pixels which make up the area of the shadow region.
5. The center of mass of the region.

[1]The image used in Figure 4-a was obtained from Andres Huertas at USC Institute for Robotics and Intelligent Systems while images used in Figures 4-b through 4-e are parts of larger images which were obtained from Joseph Sanjour at the Center for Automation Research, University of Maryland.

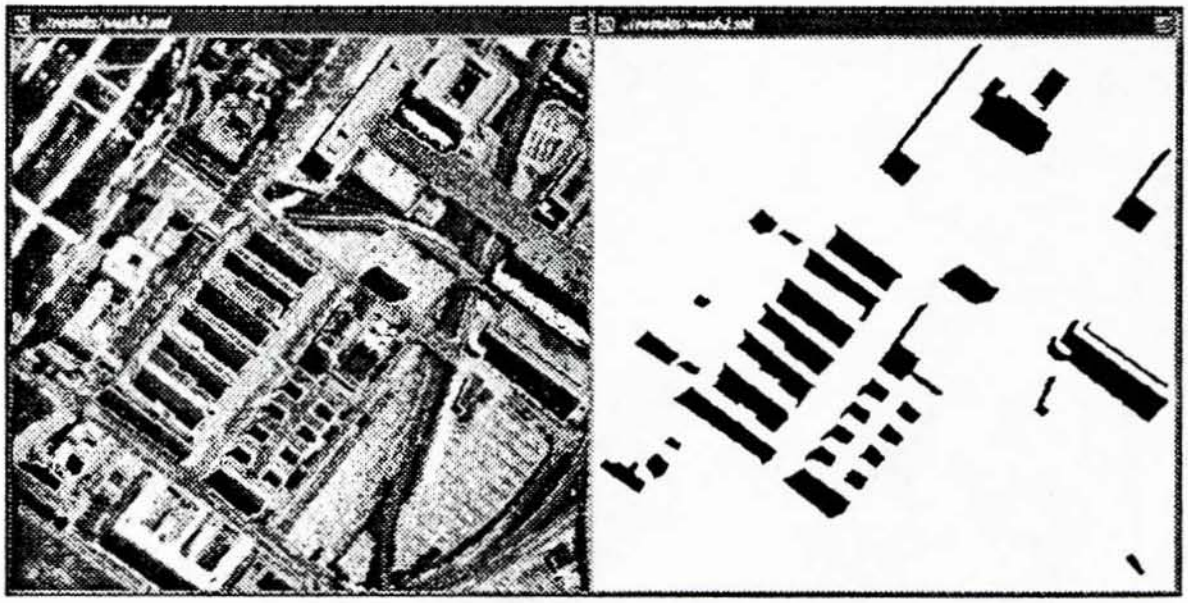


(a)A 512X512 image of a hangar and group of buildings at the Los Angeles Airport. Resolution information for this image is not available to us.

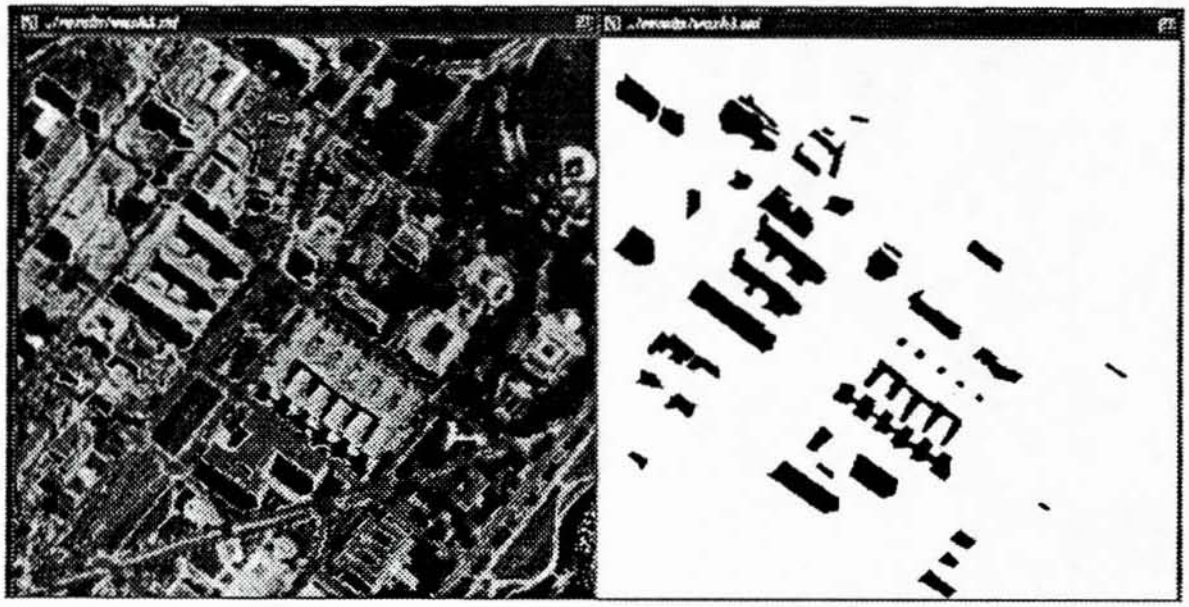


(b)A 512X512 image of the Washington monument. Approximate resolution is 1.2 meters/pixel.

(c) A 600X600 image of a group of buildings in Washington DC. Approximate resolution is 1.2 meters/pixel.



(d) A 512X512 image of a group of buildings in Washington DC. Approximate resolution is 1.4 meters/pixel.



(e) A 512X512 image of a group of buildings in Washington DC. Approximate resolution is 1.4 meters/pixel.

Figure 4: Manually outlined test images and their shadow regions.

Tables 1 through 5 list all shadow regions in each of the test images used in Figure 4. The regions are listed in order of their id numbers. For each region listed, the bounding box, area in number of pixels and centroid location are given.

To evaluate the performance of a particular shadow extraction technique, we compare the shadow regions found using each automated technique to the ground truth shadow regions of each test image. By making this comparison we can calculate performance histograms for the results of each method as applied to each image. These histograms describe the distribution of the number of the actual shadow regions given in the image data base over each of the following:

1. the number of segments into which each hand-traced shadow region is (erroneously) segmented using the automated method. This histogram provides a measure of how likely the method is to oversegment a shadow region.

2. the percentage of each hand-traced shadow area found using the automated method. This histogram is a measure of how completely the particular method segments out a shadow region.

3. the ratio of the area of regions erroneously extracted as part of the actual shadow area, using the automated method, to the area of the actual shadow region. This histogram is a measure of how likely the particular method is to merge a shadow region in the image with other shadow or non-shadow regions.

In addition to these three histograms, we calculate other quantities which provide important performance evaluation measures that are either not reflected at all, or are implicit in the performance histograms. These quantities are given as follows:

1. The percentage of regions found in a single segment using the automated method.

2. The percentage of the total hand-traced shadow area found using the automated method.

3. The ratio of the total area of regions erroneously extracted as part of the actual shadow regions, using the automated method, to the total area of the actual shadow regions.

4. The ratio of total false shadow area extracted using the automated method to the actual shadow area.

Obviously, a perfect shadow extraction technique is one which produces the maximum number possible of completely detected shadow regions but no distorted, oversegmented or false shadow regions. For practical purposes however, we can consider a certain shadow extraction technique to be better than another if it produces a larger number of completely detected shadow regions and a smaller number of distorted, oversegmented or false shadow regions.

Table 1: Ground truth data base for image used in Figure 4-a.

Region id	Bounding box				Number of pixels	Centroid		Region id	Bounding box				Number of pixels	Centroid	
	xmin	ymin	xmax	ymax		x	y		xmin	ymin	xmax	ymax		x	y
1	83	94	96	123	148	91.16	107.27	2	85	126	101	171	184	95.91	143.96
3	92	174	110	209	207	103.47	186.59	4	100	221	125	230	117	113.32	225.16
5	100	234	138	312	585	124.39	264.11	6	99	125	114	169	184	109.35	143.71
7	113	122	130	166	223	124.11	140.33	8	122	173	137	200	143	131.96	184.30
9	156	227	160	233	12	157.83	230.00	10	127	208	134	218	26	130.27	212.73
11	141	205	150	217	35	145.37	210.89	12	217	327	244	355	221	235.92	337.10
13	191	312	219	352	289	210.04	325.44	14	180	223	206	303	307	197.85	261.29
15	208	214	232	299	321	223.86	252.19	16	177	196	197	216	126	190.71	203.14
17	174	177	189	192	88	183.34	182.99	18	170	146	184	172	104	178.62	156.03
19	169	130	180	143	60	175.57	133.50	20	164	102	180	125	112	174.71	110.79
21	242	238	290	318	1364	270.54	281.47	22	232	120	256	143	212	247.86	127.82
23	233	102	251	120	124	245.31	108.77	24	265	130	373	273	4235	341.62	184.24
25	330	397	405	424	769	369.01	405.92	26	293	403	308	419	89	299.35	410.27
27	472	458	505	475	208	488.13	468.04	28	364	288	422	347	1695	396.81	312.74
29	299	326	365	344	575	333.57	335.78	30	295	315	322	332	154	311.89	320.25
31	306	279	314	311	120	311.21	293.29	32	257	84	285	142	432	275.84	108.49
33	319	112	354	124	220	335.20	117.43	34	304	86	317	111	112	310.12	99.00
35	332	77	348	95	117	337.68	86.99	36	382	19	398	35	100	388.68	27.72
37	356	2	405	18	376	380.76	11.62	38	469	49	500	75	461	486.06	60.44
39	298	389	308	403	28	301.64	398.29	40	312	390	327	402	37	319.68	396.68
41	344	381	395	399	307	367.49	392.15	42	397	377	414	394	80	405.61	387.35
43	424	380	434	390	38	428.61	384.16	44	458	369	471	386	51	463.37	378.20
45	473	370	483	384	43	477.21	378.58	46	491	366	504	380	54	497.93	373.98
47	367	273	377	282	47	371.47	277.60	48	382	273	391	279	29	386.76	276.07
49	397	270	406	276	14	402.07	273.57	50	412	267	423	277	59	418.15	271.97
51	430	267	436	274	29	432.41	269.93	52	443	263	454	271	20	449.20	267.75
53	345	345	371	360	144	359.49	349.67	54	348	416	361	423	45	355.22	418.71

Table 2: Ground truth data base for image used in Figure 4-b.

Region id	Bounding box				Number of pixels	Centroid	
	xmin	ymin	xmax	ymax		x	y
1	281	194	356	243	988	319.25	218.86

3.2 Shadow Extraction Using Global Thresholding

One way of finding a threshold value for an image is to calculate a histogram which describes the distribution of the various gray-levels in the image [2, 12]. Typically, this histogram will have peaks and valleys. Each peak in the histogram corresponds to a group of regions in the image which share a common average intensity. The peak which corresponds to the darkest regions in the image can generally be assumed to represent the shadow regions. The valley which separates this peak from peaks at lighter gray levels corresponds to the threshold value used to extract the dark regions in the image. We first consider the performance of a method which uses a global threshold chosen manually through iterative experimentation. Then we consider the performance of another method which uses an automatically calculated global threshold.

3.2.1 Using an operator selected global threshold

A simple algorithm has been implemented to confirm that shadows can be extracted using a global threshold value for the whole image. This algorithm accepts a single threshold value as input and uses it to extract regions in the image which have pixel values darker than the threshold. Using this program, we have been able to estimate reasonable threshold values for the 5 test images used in Figure 4.

The test images used have a wide range of threshold values and they contain shadow regions of various sizes. Determining a reasonable threshold value for each image is based on extracting all regions that a human would consider to be shadow

Table 3: Ground truth data base for image used in Figure 4-c.

Region	Bounding box				Number	Centroid		Region	Bounding box				Number	Centroid	
id	xmin	ymin	xmax	ymax	of pixels	x	y	id	xmin	ymin	xmax	ymax	of pixels	x	y
1	83	96	153	175	1554	115.27	139.88	2	23	119	74	186	948	46.03	158.57
3	51	217	87	286	1041	71.13	252.91	4	33	206	69	261	278	49.11	236.23
5	81	168	129	234	884	102.51	208.13	6	148	169	194	230	797	168.68	204.60
7	100	243	142	303	914	120.51	276.81	8	196	193	250	254	1117	220.89	227.93
9	141	238	207	334	1391	168.07	290.48	10	246	246	320	314	1479	284.39	281.70
11	247	294	282	336	425	264.29	313.82	12	194	313	233	381	974	214.41	350.43
13	181	303	204	322	198	190.57	313.18	14	300	306	342	358	819	319.72	334.14
15	236	351	253	364	83	244.84	356.75	16	229	361	273	416	940	250.73	392.44
17	396	374	457	442	1241	424.04	411.54	18	285	375	363	464	1453	321.10	421.20
19	354	403	399	457	976	375.81	430.57	20	395	433	418	455	231	405.69	445.06
21	451	439	533	531	1899	486.80	490.65	22	389	454	430	506	842	411.35	481.43
23	320	481	382	541	1137	347.97	513.08	24	427	485	494	559	1238	457.73	525.27
25	389	513	457	587	1649	419.64	552.98	26	276	523	298	554	119	286.08	539.97
27	247	490	275	540	121	258.69	517.83	28	100	375	157	434	678	128.11	407.03
29	147	428	187	464	207	167.00	449.34	30	38	452	62	473	254	50.11	465.35
31	51	486	90	543	1072	70.25	515.23	32	13	477	38	506	389	25.43	492.69
33	12	283	58	333	202	34.74	311.76	34	365	337	399	403	299	379.88	366.56
35	317	438	345	463	260	330.37	452.13								

Table 4: Ground truth data base for image used in Figure 4-d.

Region	Bounding box				Number	Centroid		Region	Bounding box				Number	Centroid	
id	xmin	ymin	xmax	ymax	of pixels	x	y	id	xmin	ymin	xmax	ymax	of pixels	x	y
1	457	101	509	181	1086	480.53	151.47	2	410	258	508	355	2809	457.32	311.62
3	430	258	508	323	357	468.89	293.97	4	399	273	418	298	195	406.92	287.85
5	470	476	486	498	156	477.15	486.21	6	331	34	404	108	2522	367.48	75.30
7	385	26	419	67	649	401.22	44.90	8	251	3	342	137	1283	284.46	89.05
9	304	208	358	246	1047	330.75	227.90	10	388	310	407	349	245	396.92	330.74
11	251	243	308	328	1156	275.21	293.98	12	232	311	262	336	317	248.41	324.18
13	266	333	290	356	259	277.65	344.78	14	251	354	275	376	255	262.46	365.52
15	216	334	245	356	269	233.04	344.50	16	235	374	259	397	267	246.72	385.18
17	200	356	232	375	282	217.59	365.01	18	219	396	244	417	255	231.78	406.62
19	186	375	216	395	296	201.98	385.11	20	167	396	229	443	1274	198.26	419.77
21	203	171	274	234	1531	236.75	204.25	22	184	192	253	262	1272	217.77	224.72
23	160	220	232	286	1715	194.74	254.36	24	145	244	214	311	1521	179.13	279.94
25	127	267	198	341	1416	161.88	303.93	26	105	296	173	364	1642	138.92	331.99
27	71	321	164	393	1670	117.14	355.89	28	135	157	163	181	370	148.70	170.26
29	85	236	103	249	108	94.34	242.84	30	35	270	78	302	571	55.80	285.77
31	162	175	184	191	136	172.72	183.44	32	73	296	95	311	142	84.69	303.06
33	3	387	46	421	607	26.78	401.97	34	44	382	68	407	312	55.76	394.13
35	62	368	78	380	101	69.98	373.42								

Table 5: Ground truth data base for image used in Figure 4-e.

Region id	Bounding box				Number of pixels	Centroid		Region id	Bounding box				Number of pixels	Centroid	
	xmin	ymin	xmax	ymax		x	y		xmin	ymin	xmax	ymax		x	y
1	12	31	53	76	859	32.42	53.85	2	50	62	74	89	426	62.92	75.15
3	12	169	49	205	831	32.72	188.93	4	104	54	158	101	1204	129.09	78.53
5	121	50	156	81	208	138.28	64.14	6	113	118	134	136	220	124.46	128.12
7	76	135	89	163	233	82.30	147.38	8	220	69	237	76	57	228.89	72.30
9	195	79	207	88	46	200.57	83.59	10	169	93	195	124	276	181.53	110.34
11	184	81	208	104	181	197.12	92.65	12	201	92	216	125	200	208.26	110.04
13	140	134	189	181	975	166.11	157.89	14	155	132	170	143	51	162.06	137.29
15	197	141	224	160	317	211.30	150.21	16	136	158	200	224	1428	167.86	195.58
17	115	189	179	253	1352	147.08	225.56	18	77	218	157	285	1808	117.77	251.91
19	78	286	107	310	287	92.53	297.46	20	58	322	86	346	301	72.87	334.09
21	43	267	86	297	617	64.86	281.35	22	34	301	58	319	214	44.64	310.38
23	26	377	43	393	114	34.91	384.70	24	234	185	264	219	549	248.74	205.20
25	251	183	268	198	65	258.92	191.03	26	325	181	356	211	369	340.70	195.62
27	272	227	320	274	758	296.07	252.49	28	331	280	374	313	514	351.63	298.02
29	263	271	272	280	41	266.90	275.54	30	279	282	288	291	42	283.36	286.79
31	299	298	307	305	30	302.33	301.30	32	314	311	323	318	24	319.25	314.08
33	332	323	340	331	30	336.03	327.07	34	233	291	321	373	1361	278.60	329.40
35	206	313	313	404	1861	262.26	358.18	36	221	378	265	420	935	242.94	400.50
37	183	352	207	377	252	194.67	365.43	38	191	387	208	402	86	199.83	393.93
39	150	384	213	434	1214	183.17	410.46	40	447	294	468	309	57	457.30	301.54
41	389	424	399	431	26	393.85	427.08	42	311	449	334	467	209	322.47	457.70
43	300	468	320	484	148	310.09	476.30	44	283	485	318	511	342	299.50	497.90

regions while extracting the minimum possible number of non-shadow regions. The ground truth data bases were not used in this case to determine the threshold value which will produce the best shadow extraction. The reason is that the process requires only comparison between the results which correspond to the various threshold values for each single image rather than absolute measurement of performance for each case. This comparison can be easily done by a human operator. Once the operator selects the threshold value which he or she considers to be the best, the performance of the algorithm using that particular threshold value can be measured against the ground truth data base. The results of these measurements represent the best possible shadow extraction result using a single global threshold. These results can be used to judge the performance of an algorithm which automatically calculates a global threshold value for the test images. They can also be used to compare the performance of shadow extraction using global thresholding methods versus other methods.

Figures 5 through 9 show the results of shadow extraction and the performance histograms using the estimated thresholds for the 5 test images in Figure 4. By examining the performance histograms, we find that in general this method performs well as far as extracting most of the area for most of the shadow regions. It also finds most of the shadow regions in one segment. The amount of distortion in each extracted shadow region is represented by the ratio of shadow area falsely found to be connected to the shadow region to the area of the actual shadow region. This measure is found to be low for most of the regions in the test images. The total area of what is falsely extracted as shadow regions is also reasonable for all images except for the image of Figure 6. In this image the shadow region is small and is surrounded by several dark non-shadow regions. Because of these dark regions it is not reasonable to expect an algorithm to extract fewer false shadow regions while extracting the actual shadow region based on intensity alone. As we indicated earlier, we are not very concerned about extracting false shadow regions at this stage, based on the

assumption that other information in the image can be used later on to eliminate such regions. The only time such false regions become of great concern is when they distort the true shadow regions, as we will see in the next section.

3.2.2 Using an automatically calculated global threshold

An algorithm which uses a histogram to calculate a global threshold value has been implemented and used to extract shadows from the test images. The main difficulty in using this technique is finding peaks and valleys in the histogram. This problem is inherent to the method of thresholding using histograms. Finding peaks and valleys in a histogram requires setting criteria which define what causes a certain intensity to be identified as corresponding to a peak, a valley or neither. These criteria may have to be determined experimentally using certain images and are therefore dependent on the image being used. The reason for this is that images differ in the amount of noise, average intensity, and the type and size of regions which they contain. All of these factors have a direct effect on the shape of the histogram and the amount of smoothing needed to eliminate meaningless peaks and valleys. In this implementation, these criteria have the form of a single parameter which defines how to smooth the histogram in a way that will make all of its peaks and valleys meaningful. In other words, smoothing the histogram will eliminate the peaks and valleys which are too small or too narrow to be representative of a region type in the image.

Figures 10 through 14 show the results and performance histograms of using automatically calculated global threshold values to extract shadows in the 5 test images used earlier. The performance of this method is considerably worse than using the operator selected threshold method. This is particularly obvious in Figures 10 and 11. In Figure 10 it seems that the performance is very good in terms of the percentage of total shadow area found and the percentage of the single segment regions found. However, if we examine the ratio of area falsely found to be part of actual shadow

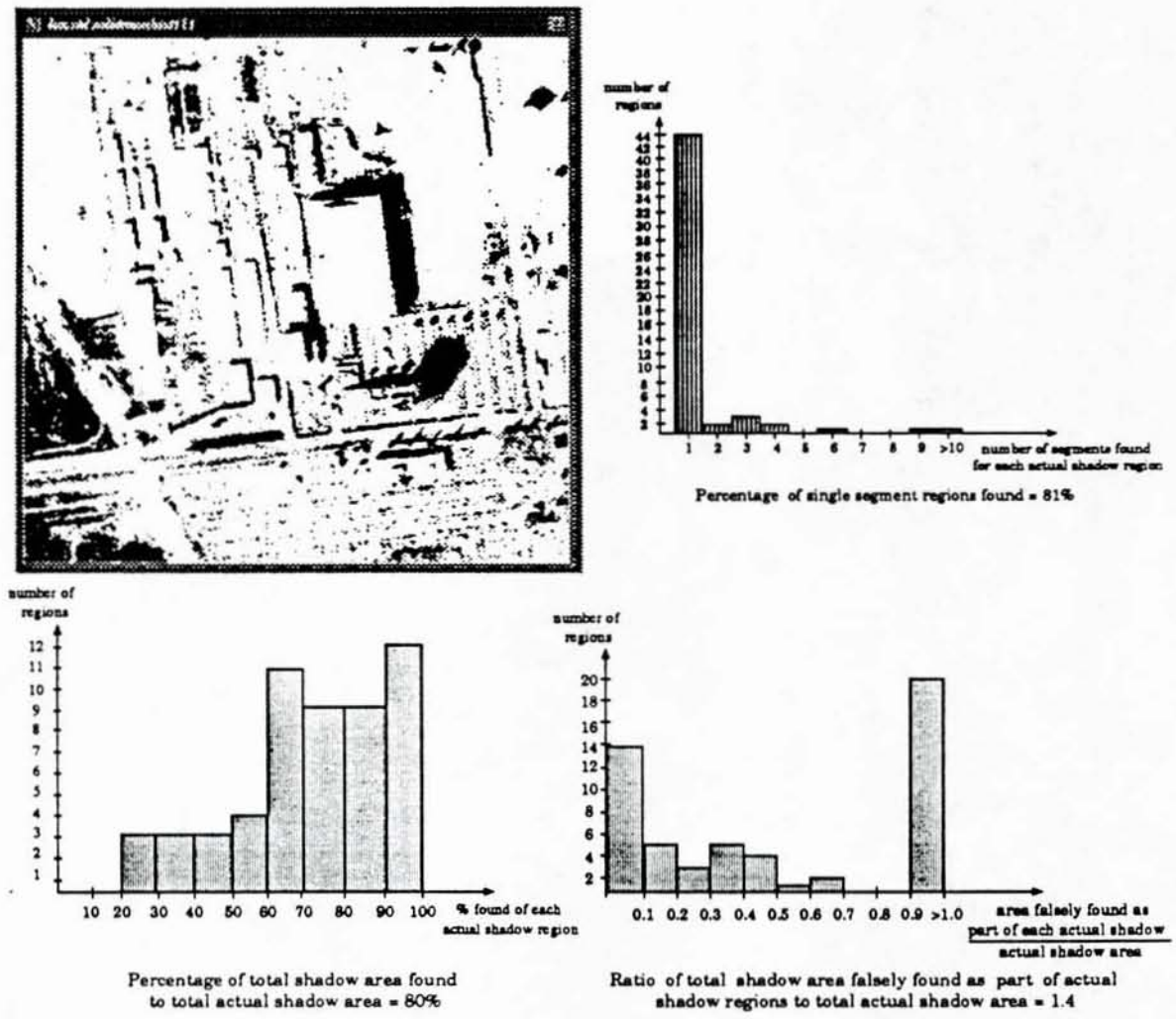


Figure 5: Extracted shadows and performance histograms for the image used in Figure 4.a using an operator selected threshold.

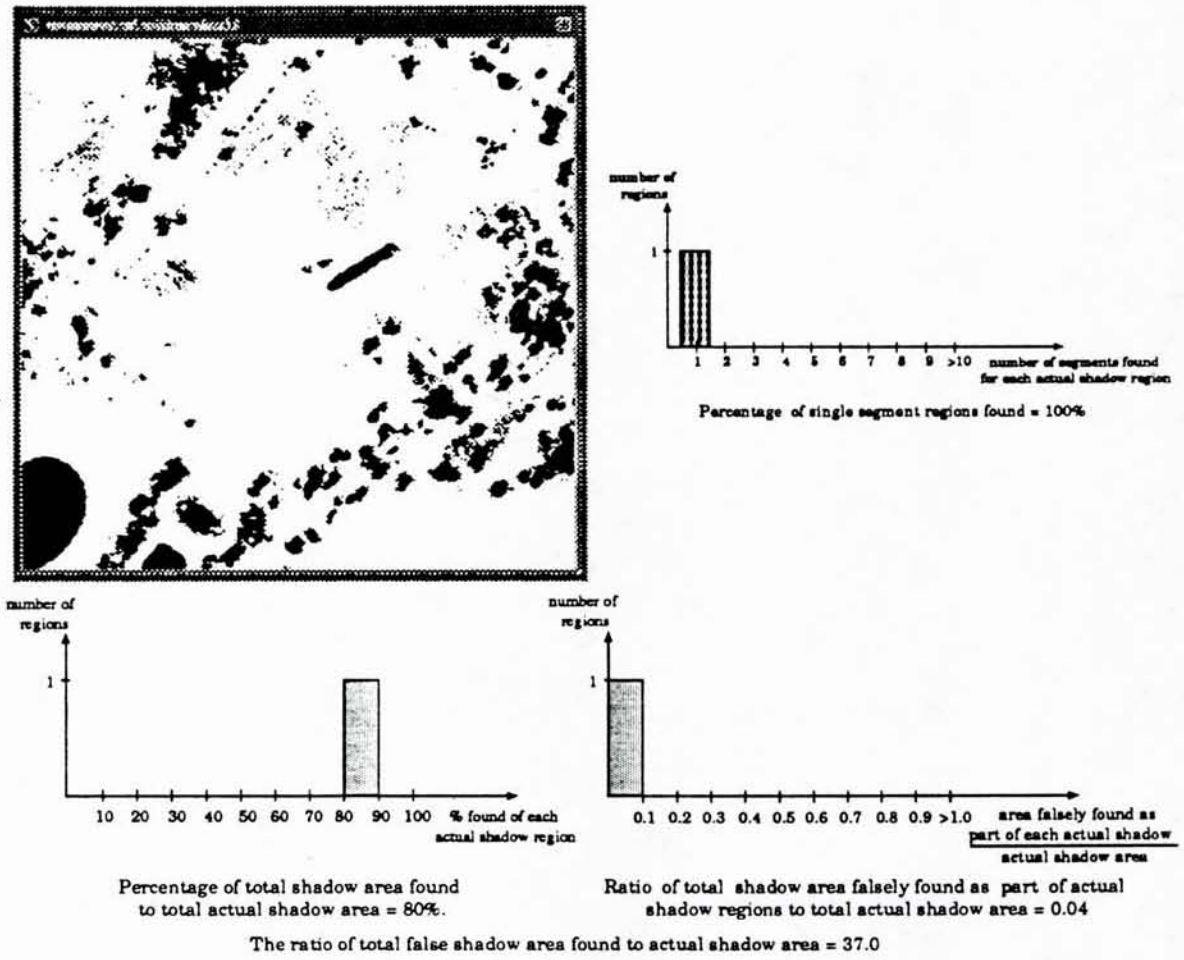


Figure 6: Extracted shadows and performance histograms for the image used in Figure 4.b using an operator selected threshold.

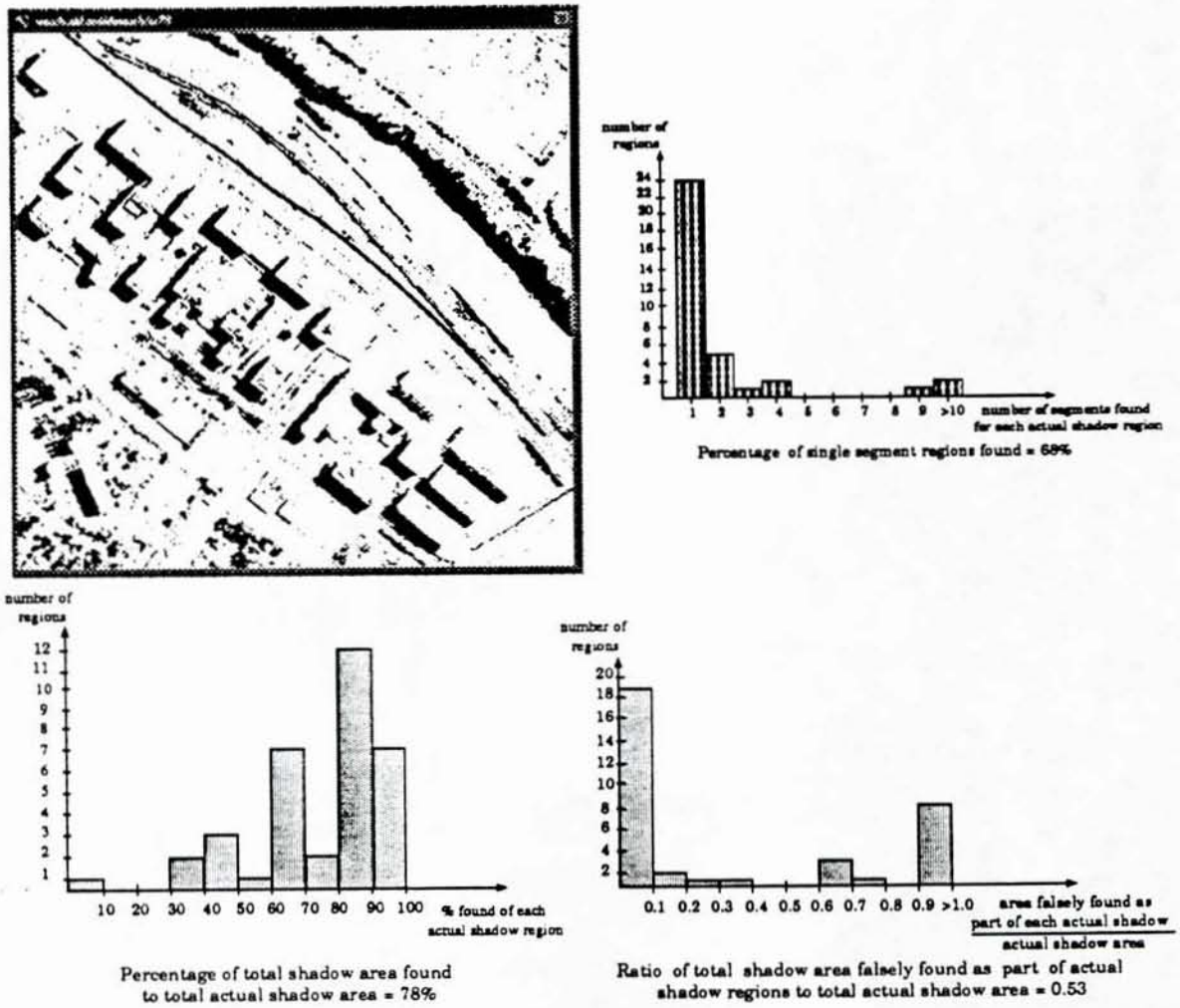


Figure 7: Extracted shadows and performance histograms for the image used in Figure 4.c using an operator selected threshold.

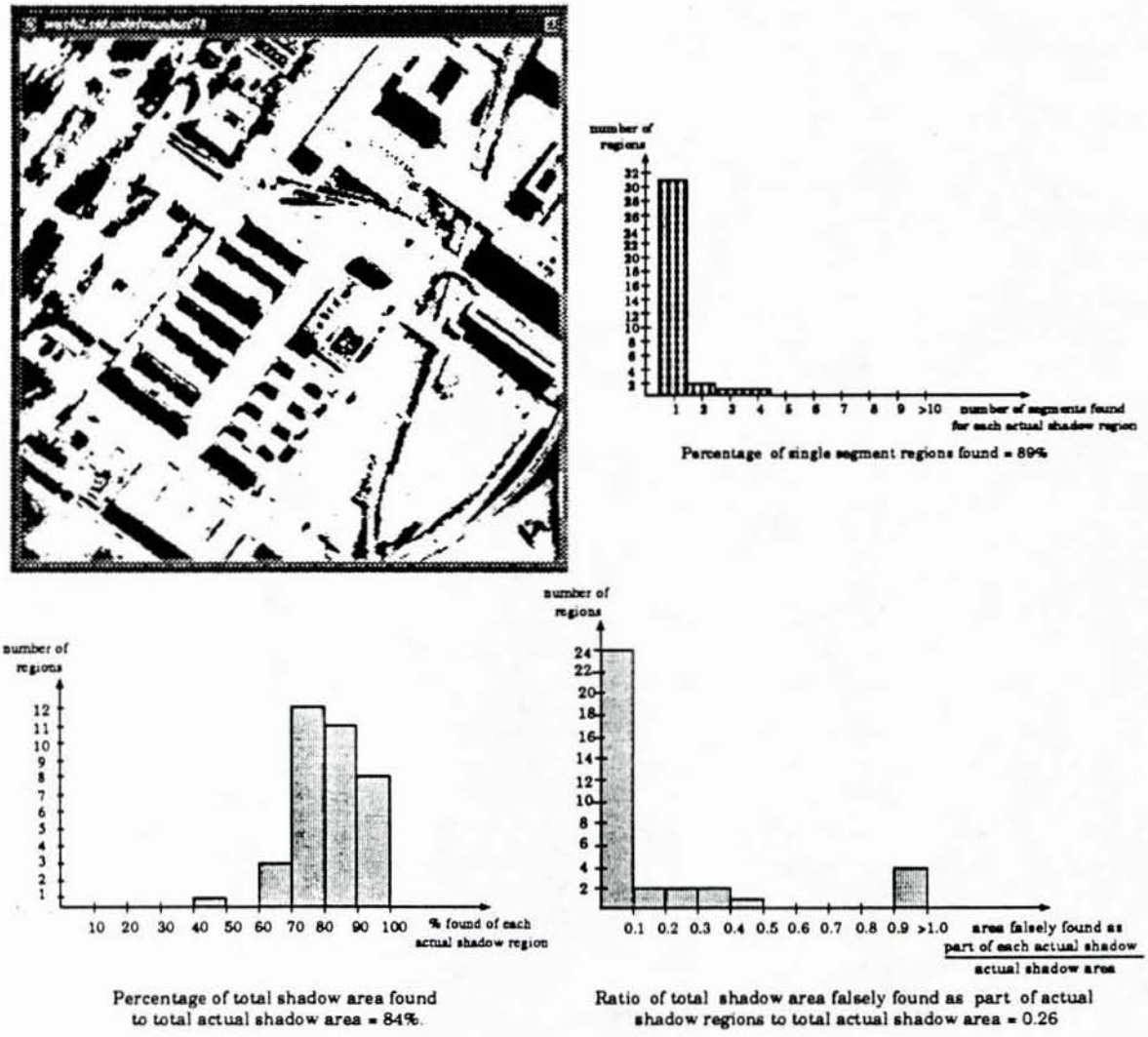


Figure 8: Extracted shadows and performance histograms for the image used in Figure 4.d using an operator selected threshold.

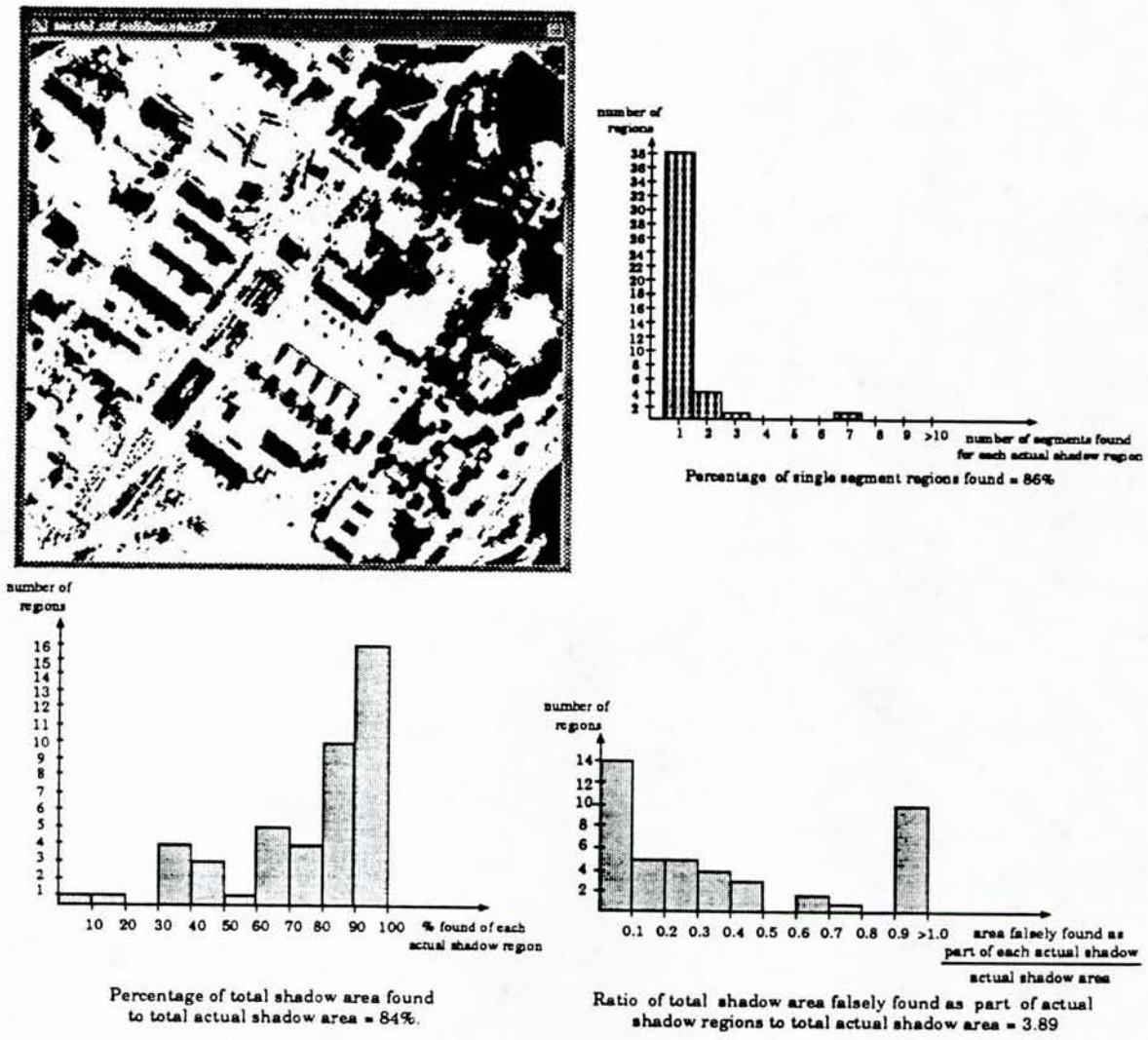


Figure 9: Extracted shadows and performance histograms for the image used in Figure 4.e using an operator selected threshold.

regions to the actual shadow area, we find it to be extremely high for all regions. This is obvious from the image in which most of the extracted shadow regions are blended with the background and are almost impossible to identify. The reason for such degraded performance is that the image used in Figure 10 contains a small number of shadow pixels which fail to make a significant peak in the dark area of the histogram. This causes the algorithm to pick a threshold that classifies all but very light regions in the image as shadow regions. In Figure 11, the algorithm fails to find most of the shadow region. This is caused by regions in the image that happened to be darker than the shadow region and which happened to form a significant peak in its histogram. The performance histograms given in Figures 12 through 14 indicate a far smaller degradation in performance for this method compared to the one explored in the previous section.

The method of using a single histogram cannot be used as a reliable automatic shadow extraction technique in its simple form. The reason for this is the need to set the parameter based on the image being used and the potential for not detecting the shadows in the image if the total shadow area is too small.

3.3 Shadow Extraction Using Automated Local Thresholding

To overcome some of the problems associated with using a single histogram for the whole image, local histogramming can be used [4, 6]. Based on this method, an image is divided into smaller parts and a local histogram is calculated for each part. The main advantage of this division is that smaller shadow regions have a better chance of forming peaks in the local histograms. Also the global variations in the average intensities of the various shadow regions will have a smaller affect on calculating the thresholds for the shadow pixels. The algorithm is composed of the following steps.

1. Divide the image into a reasonable number of overlapping rectangular areas. (For the given image test set, with its particular image sizes, division into 100

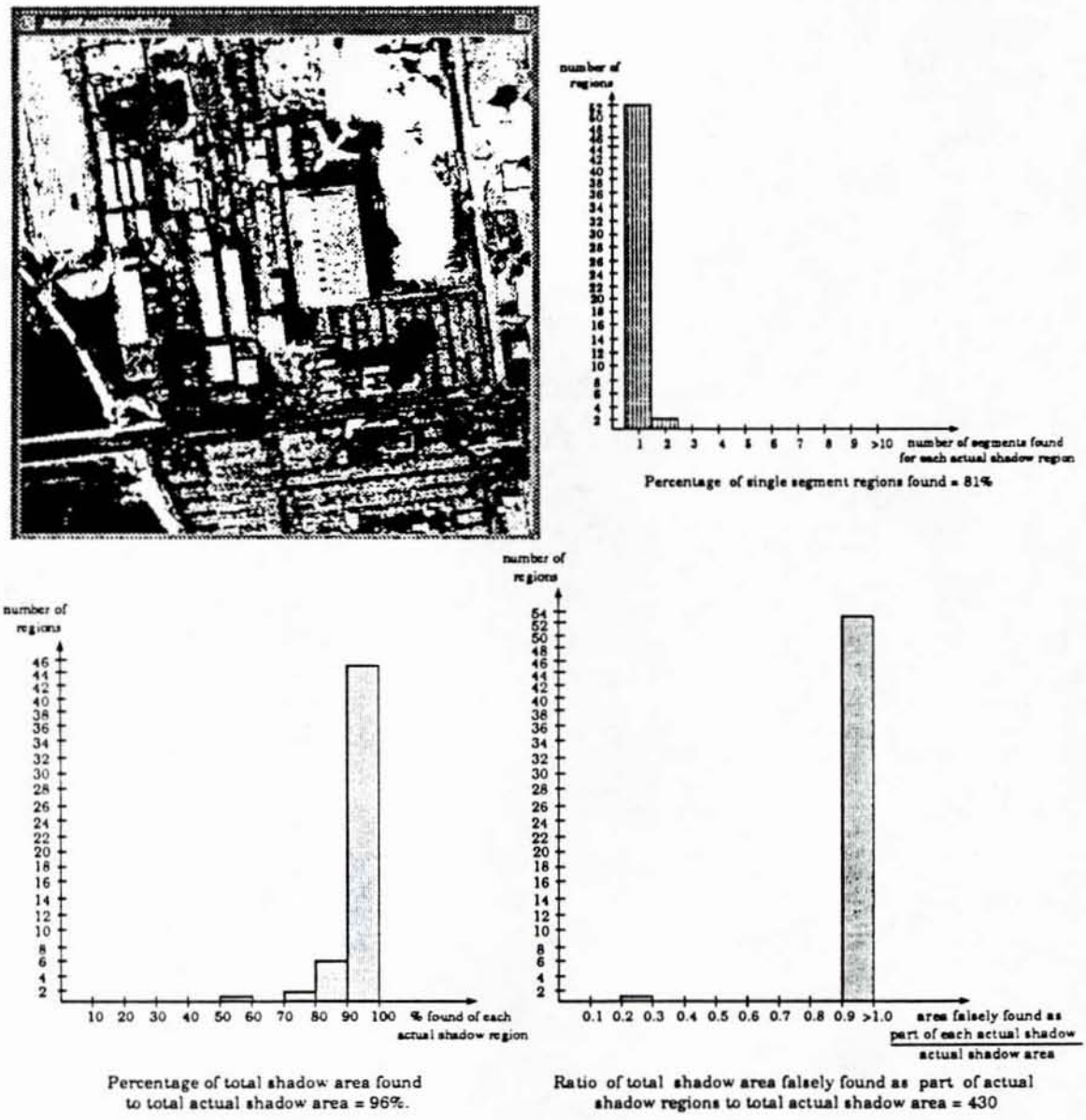


Figure 10: Extracted shadows and performance histograms for the image used in Figure 4.a using an automatically calculated global threshold.

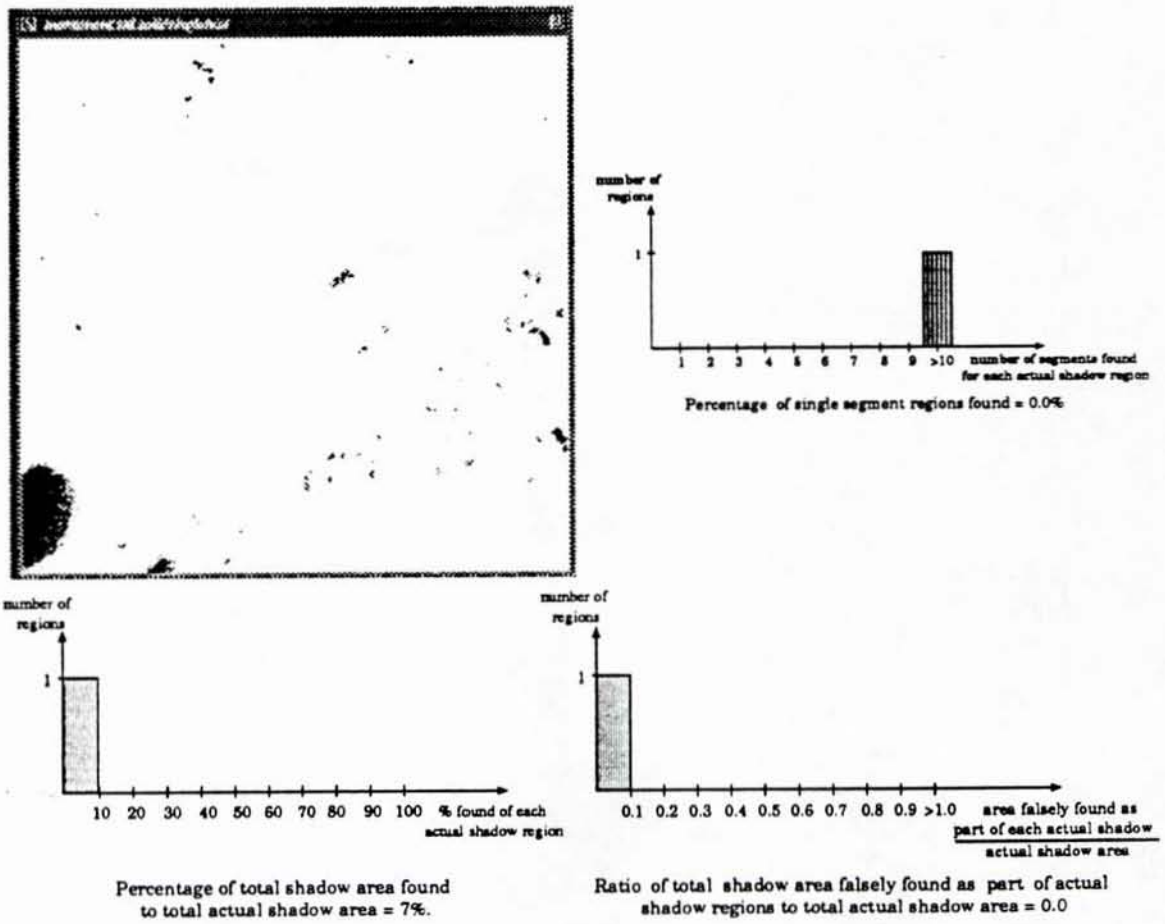


Figure 11: Extracted shadows and performance histograms for the image used in Figure 4.b using an automatically calculated global threshold.

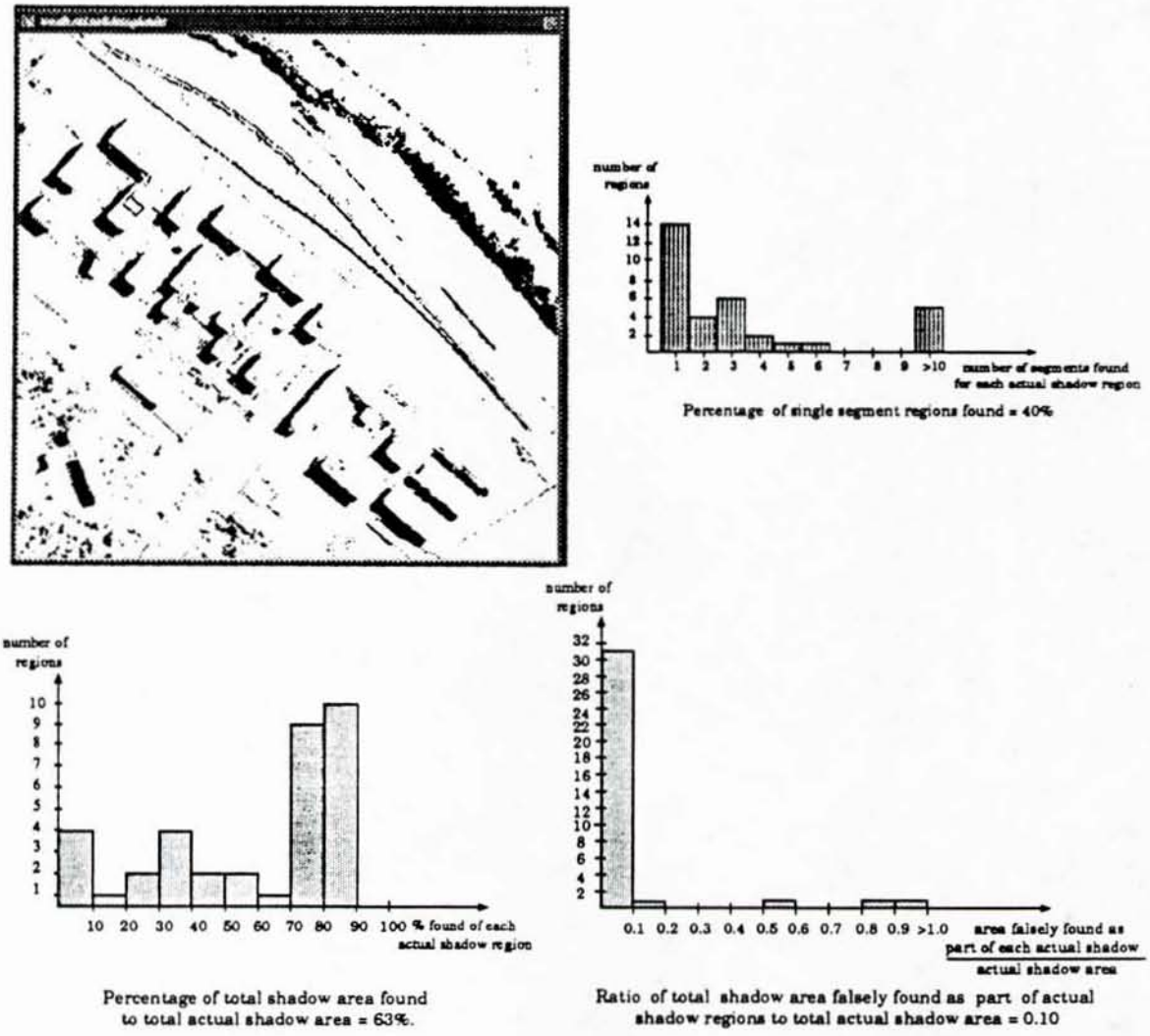


Figure 12: Extracted shadows and performance histograms for the image used in Figure 4.c using an automatically calculated global threshold.

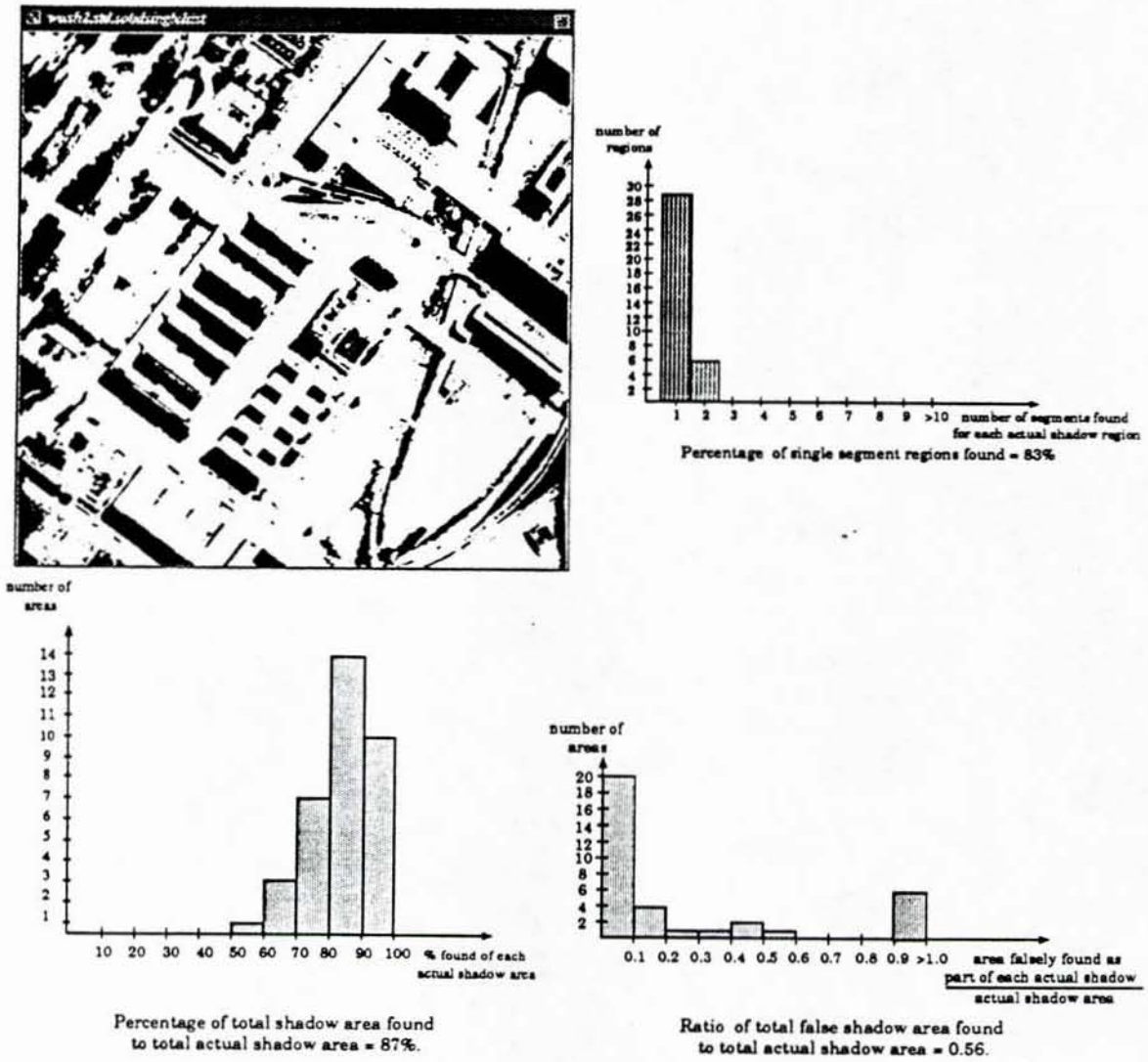


Figure 13: Extracted shadows and performance histograms for the image used in Figure 4.d using an automatically calculated global threshold.

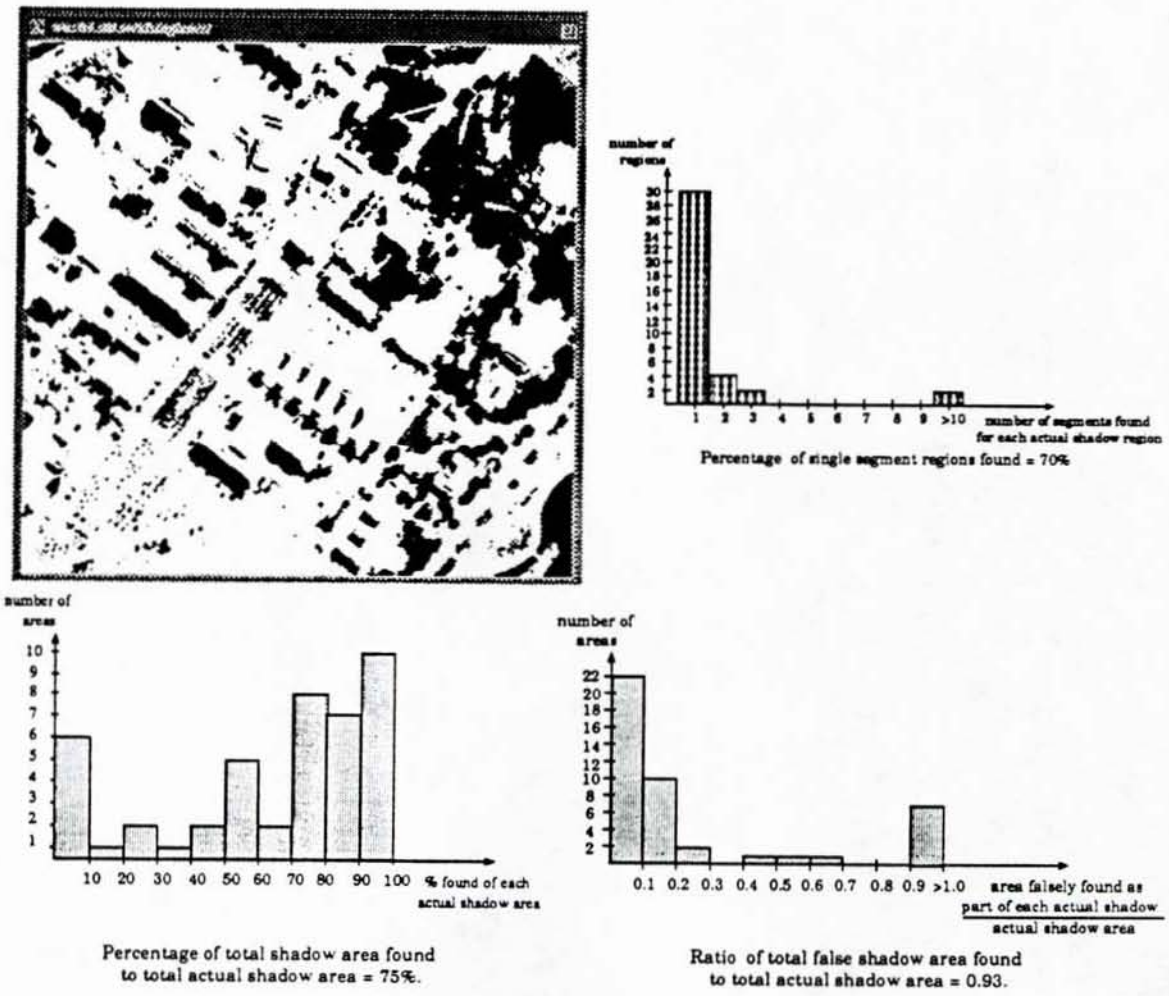


Figure 14: Extracted shadows and performance histograms for the image used in Figure 4.e using an automatically calculated global threshold.

areas produces areas which are just large enough to produce reliable histograms and small enough not to contain too many types of areas). For each rectangular subimage, a local histogram is calculated, then the histogram is used to find a threshold value for dark regions in each of the subimages.

2. Calculate a threshold value for each pixel in the image which belongs to at least one subimage for which a local threshold value was found. If such a pixel belongs to only one subimage, it is given the same local threshold value calculated for the subimage. If the pixel belongs to more than one overlapping subimage, it is given the average value of the local threshold values of these subimages.

3. Although some of the subimages used in the first step may not have any shadow regions at all, a threshold may be found for such subimages. The threshold calculated in this case represents the darkest regions in the particular subimage, but it may represent very light regions in the whole image. In order to avoid using such erroneous local threshold values to find shadow regions in the image, we have to somehow find a way of identifying these local thresholds. This can be achieved by establishing yet another threshold to determine whether a local threshold value corresponds to regions that are too light to be shadow regions. Such a threshold can be found by creating a histogram which describes the distribution of local threshold values calculated for all subimages over the range of grey-scale values allowed in the image. This histogram is then used to calculate a threshold which separates local thresholds that correspond to light regions.

4. At this point there are three types of pixels in the image; pixels which have threshold values that correspond to dark regions, pixels which have threshold values that correspond to light regions and pixels which have no calculated threshold values at all. In this step, the last two types of pixels are given

threshold values based on the average threshold values of the closest pixels of the first type.

5. The last step in this process finds the edge pixels which bound shadow regions in the image. This is performed using the threshold calculated for each pixel. The condition for a pixel to belong to a shadow boundary is that its east and west neighbors, or its north and south neighbors, have opposite relationships to the threshold values; i.e. one is less than the threshold value and the other is greater than the threshold value.

Figures 15 through 19 show the results and performance histograms of shadow extraction using the method described above on the images in Figure 4. This method is significantly better than using an automatically calculated single global threshold value, as can be concluded form examining its performance histograms. This method is completely automated and does not require adjusting any parameters based on the image. This method also seems to extract most shadow regions for all of the test images used and it seems to extract shadow regions which are small compared to the size of the image.

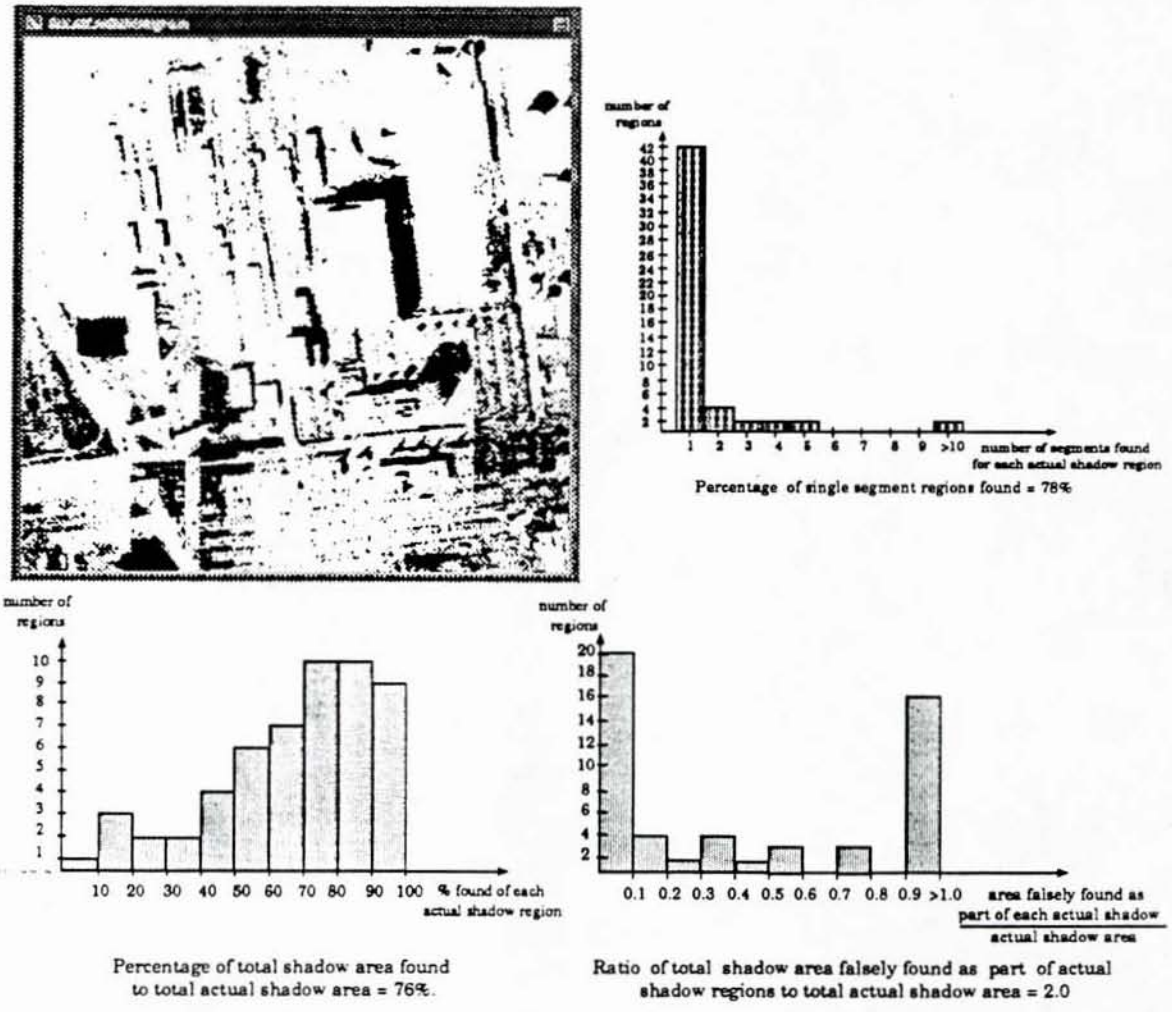


Figure 15: Extracted shadows and performance histograms for the image used in Figure 4.a using automated local thresholding.

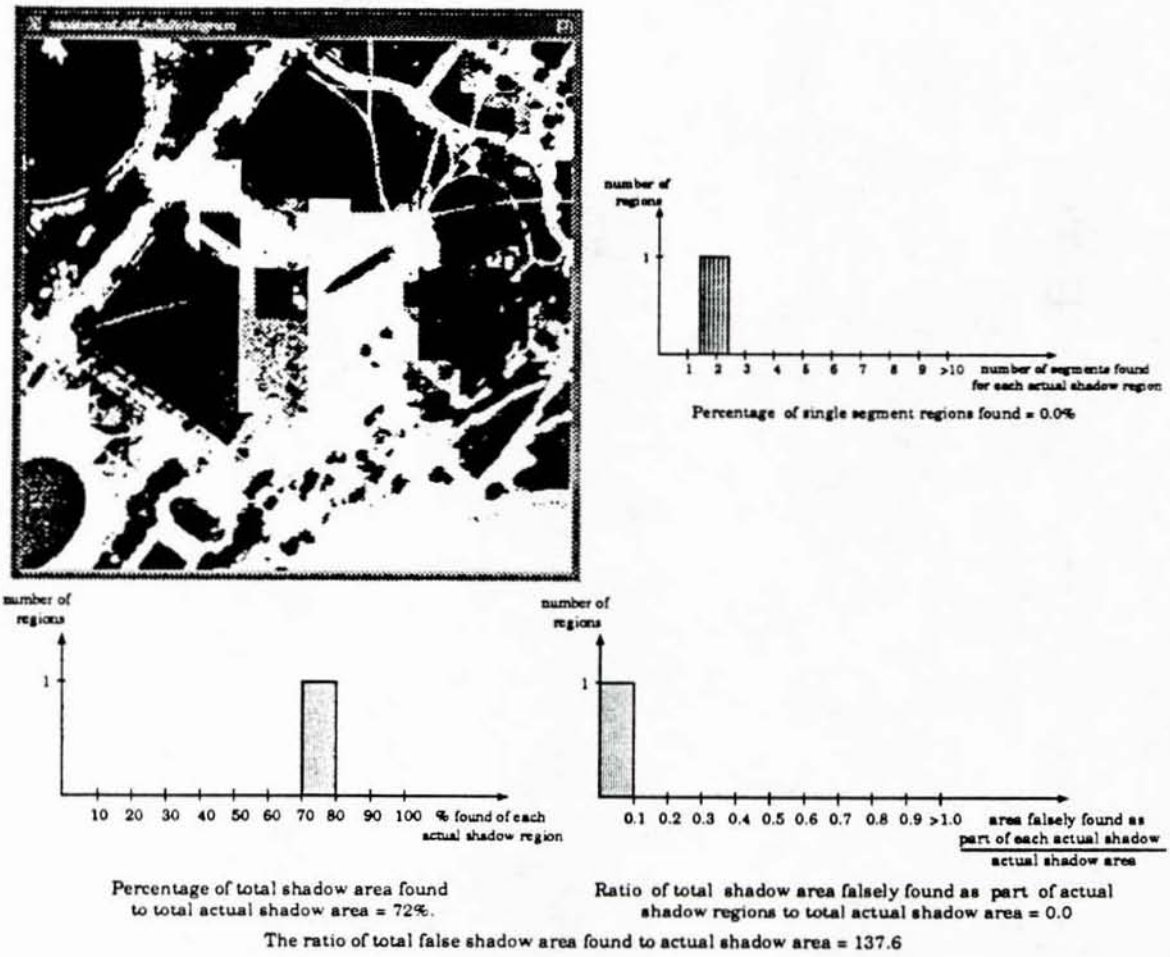


Figure 16: Extracted shadows and performance histograms for the image used in Figure 4.b using automated local thresholding.

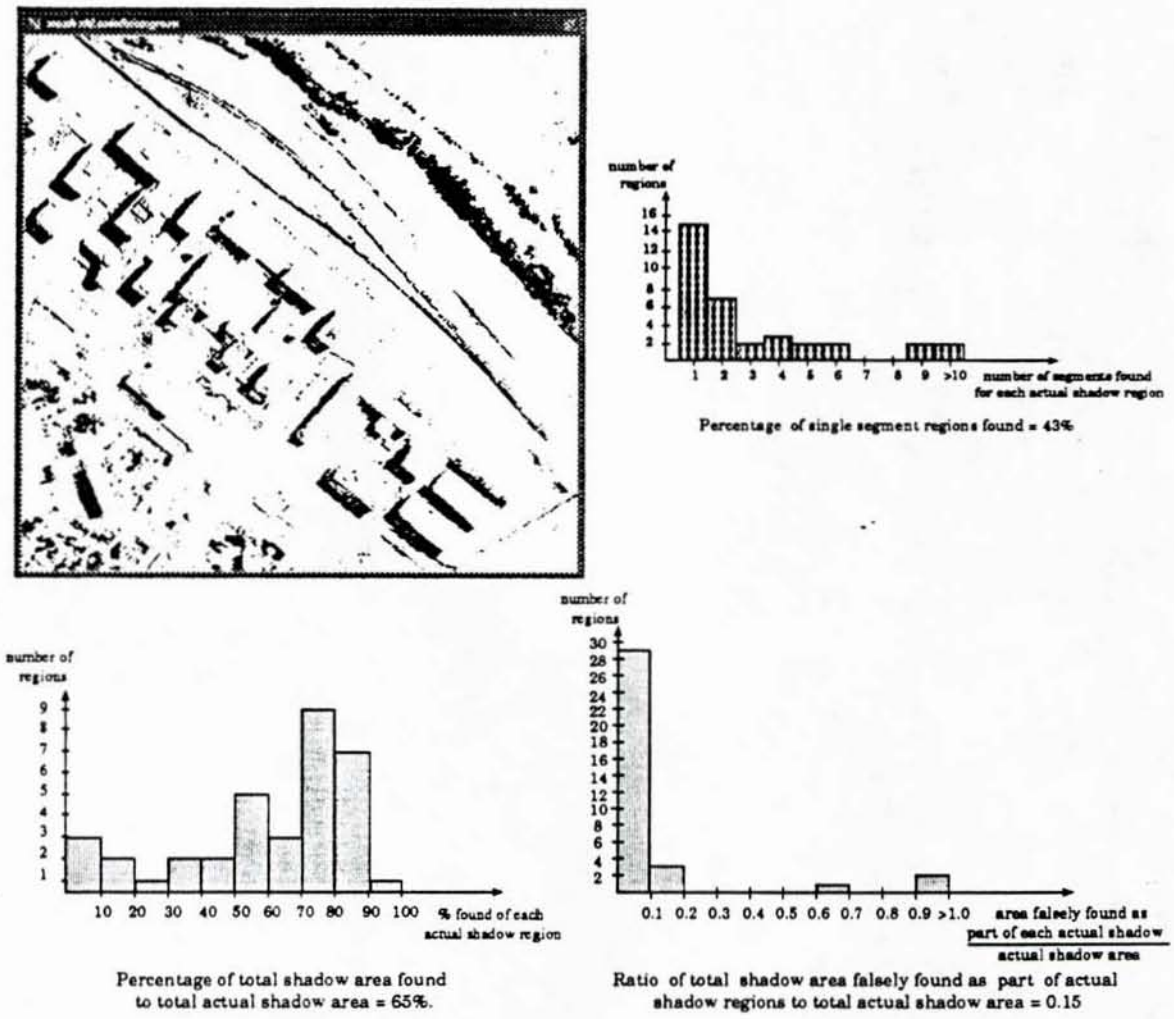


Figure 17: Extracted shadows and performance histograms for the image used in Figure 4.c using automated local thresholding.

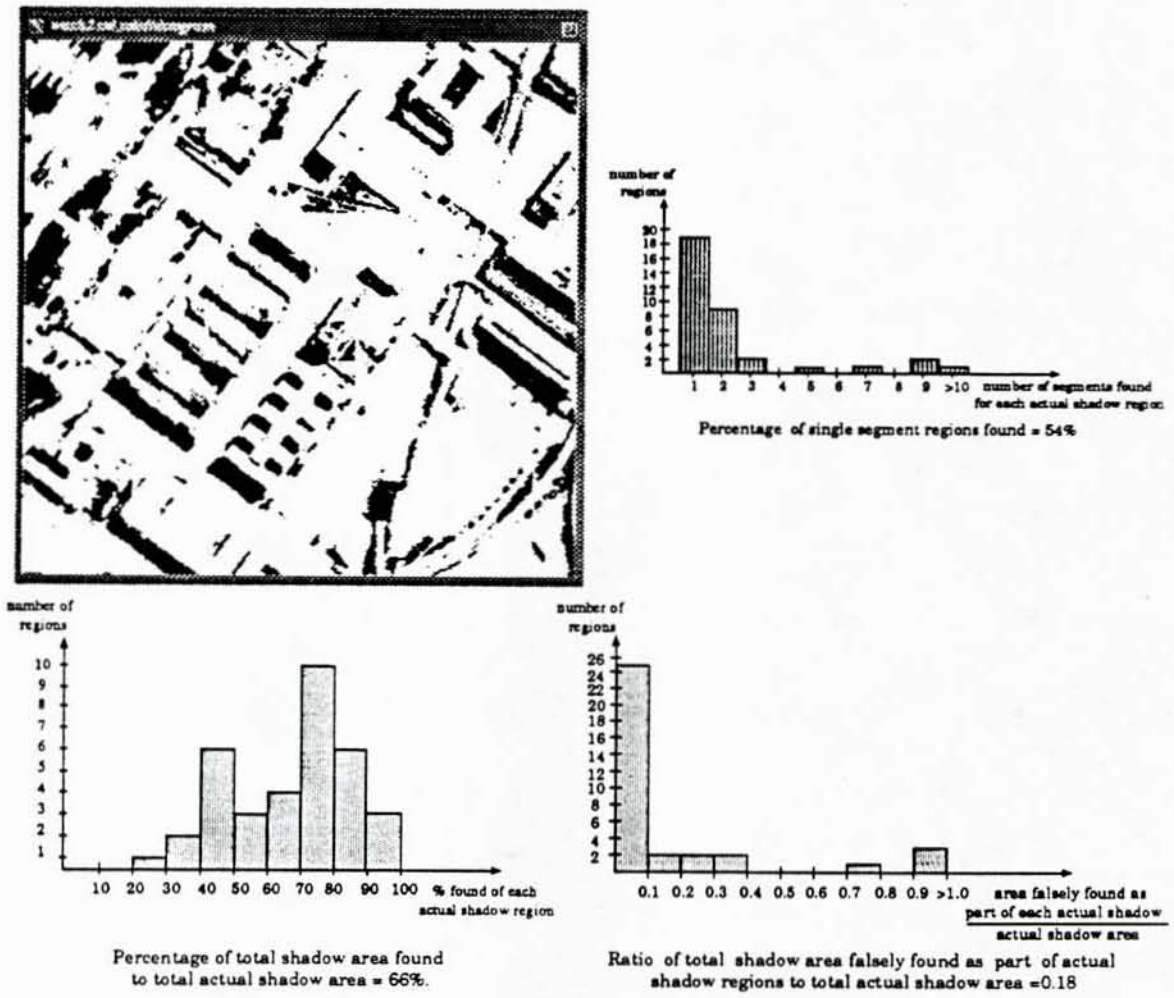


Figure 18: Extracted shadows and performance histograms for the image used in Figure 4.d using automated local thresholding.

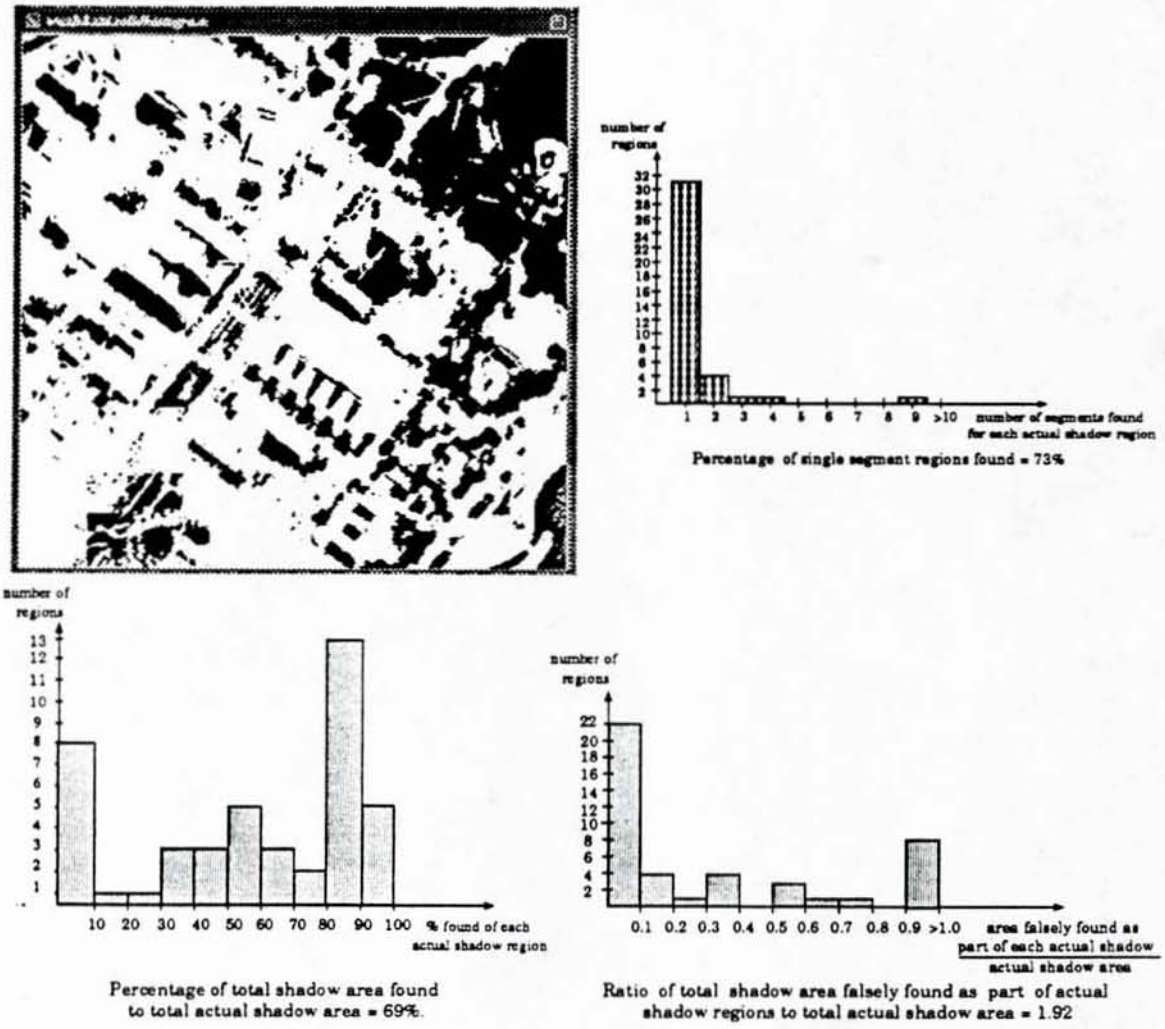


Figure 19: Extracted shadows and performance histograms for the image used in Figure 4.e using automated local thresholding.

CHAPTER 4
ESTIMATING BUILDING PROPERTIES

As indicated earlier, shadows provide very important information about buildings in aerial images. However, they are not the only source of such information. Extracted edges in the image are also very important for our purpose. It may be possible to detect edges that correspond to the boundaries of the top views of the buildings as well as the boundaries of the shadows which these buildings cast.

This chapter expands on the idea of combining shadow and edge information. This is done in the context of describing the details of an algorithm which finds the shape of the top view and the height of each building.

4.1 Overview of the Algorithm

The general algorithm can be outlined in the following stages.

1. Extract the shadow regions in the image.
2. Perform edge detection on the original image and obtain a labeled outline for each shadow region.
3. Estimate the sun direction using the shadow region boundaries.
4. Find the part of each shadow boundary which is shared by the top view of its corresponding building and extrapolate to find the complete boundary of the top view of each building.

5. Find the building height using the shadow length.

Figure 20 depicts a flowchart for this algorithm and the following sections describe stages 2 through 5 in detail (the shadow extraction stage has been explained in the previous chapter).

4.2 Obtaining Labeled Shadow Boundaries

Once the shadow regions are found in the image, we can easily label the individual regions by using a connected component algorithm. We also find the boundary chain code for each region [12]. Dealing with a chain code boundary can be difficult in determining the location of corners and corner angles in the boundary. The importance of finding corners will become more apparent in the next sections. To overcome this difficulty we convert the boundary description from the chain code form to a list of ordered line segments. Fitting the line segments to the chain code boundary is performed using a part of a software package called Perceptual Grouping. This package was developed by A. Etemadi at the University of Surrey in Guildford, United Kingdom. Perceptual Grouping is mainly concerned with extracting and grouping line and arc segments in an edge image. We only use this package to find a list of line segments which best represent the boundary of each shadow region. Each single list of line segments is labeled as belonging to a certain shadow region.

In our shadow region extraction technique, each of the pixels in a shadow region including boundary pixels is chosen based on a threshold value. This means that our boundary pixels are chosen without any regard to how well they contribute to a smooth boundary of some shape. This also means that our shadow region boundaries are very sensitive to noise. Fitting line segments to such noisy boundaries will result in producing somewhat distorted and oversegmented boundary approximations. At this point we use edges obtained by applying the edge detection technique developed by J. Canny [5] to the original image. The edges found have the advantage of being

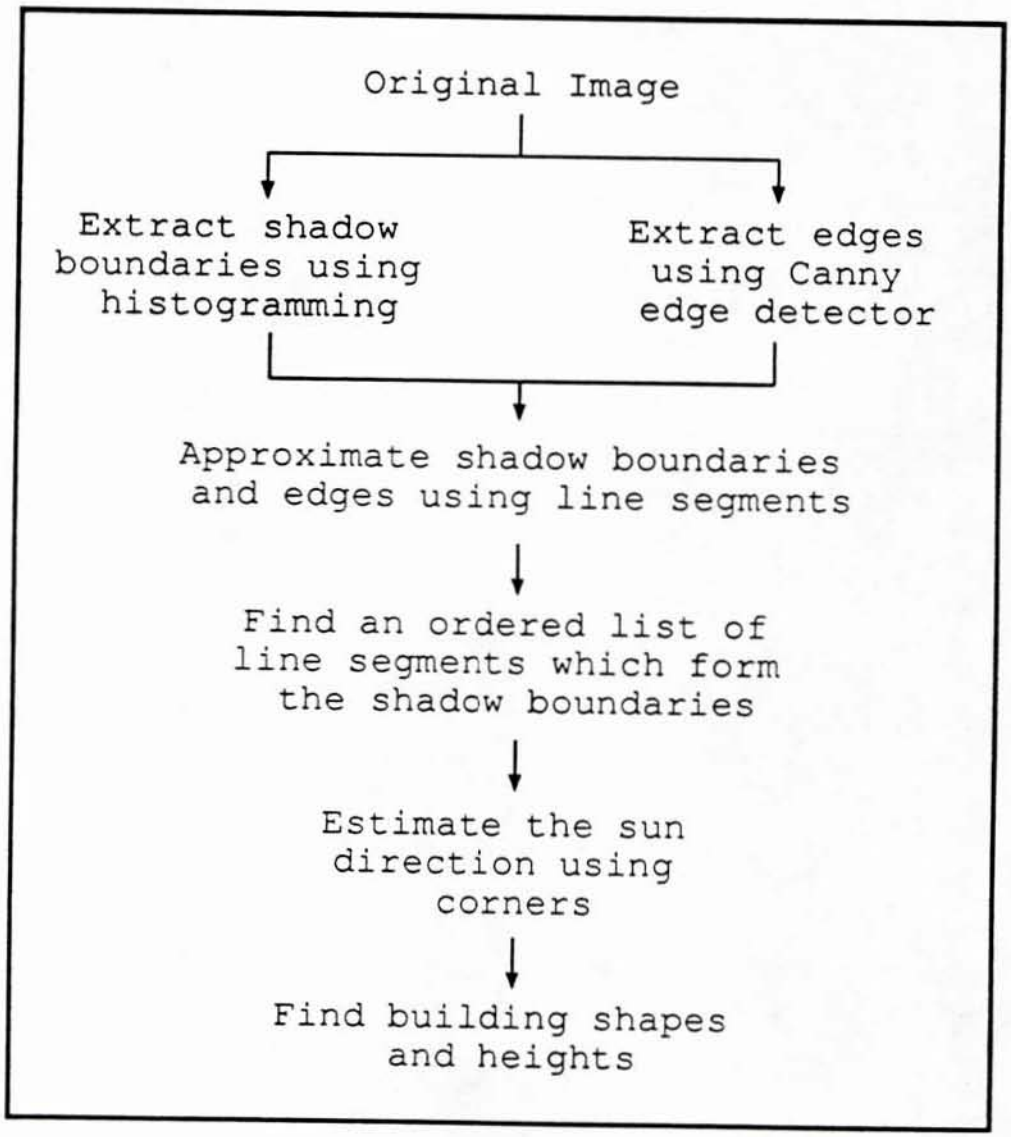


Figure 20: A flowchart for the general algorithm.

smoother than the extracted shadow boundary in general. However the extracted shadow boundary is more complete, especially around the corners. We approximate these edges using line segments in the same way we did the shadow region boundaries.

The line segments obtained by edge detection represent parts of various region boundaries in the image including shadow boundaries. Our next step is to identify the group of segments which belongs to each shadow boundary. This is done by calculating the proximity of each of the line segments obtained by edge detection to each of the line segments in an extracted shadow boundary. Once this step is completed, we have a group of line segments for each shadow region such that the lines in the group have been obtained from the two different sources. Out of all the lines in each group we form a final line segment approximation of the boundary. This final approximation uses mainly the more reliable and generally longer line segments from the edge detected image and completes the missing parts of the boundary with the remaining segments.

The information provided by edge detection is also useful in verifying which of the previously extracted regions is truly a shadow region. In our implementation we consider an extracted shadow region to be true if over 40 percent of the length of its boundary is supported by edge information.

4.3 Estimating the Sun Direction

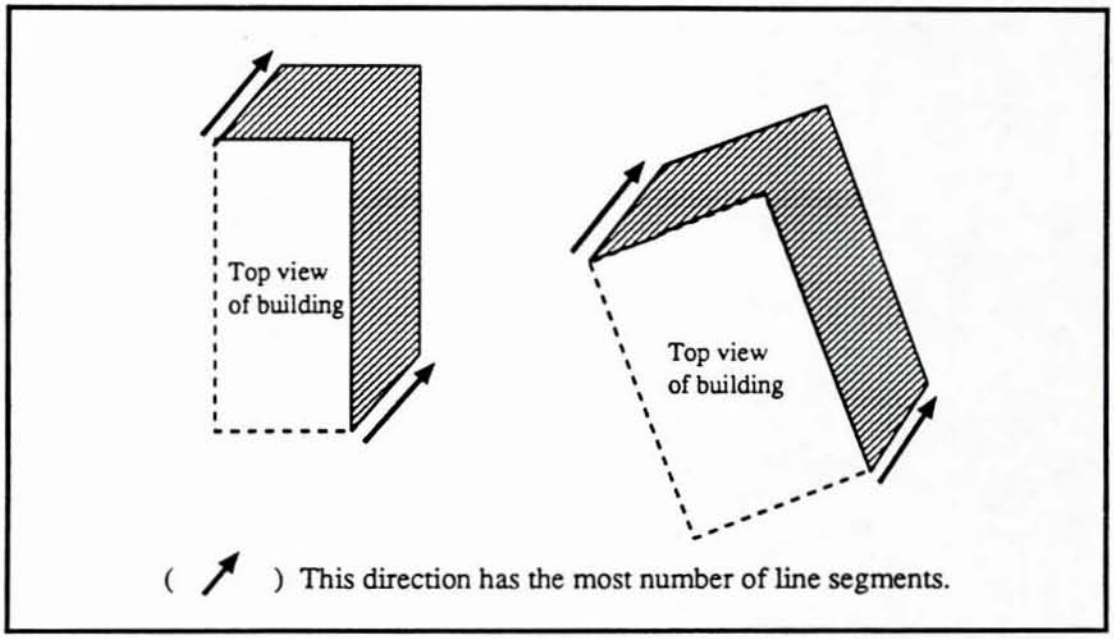
A corresponding shadow region boundary shares the part of the building region boundary which is closest along the sun direction. At this point, the only information that we have about the building region boundary is embedded in the shadow region boundary, and the only way to extract such information is by finding the sun direction in the image.

All building shadows in an aerial image are cast in the direction of the sun. This leads us to conclude that we should be able to use the shadow boundary line segments

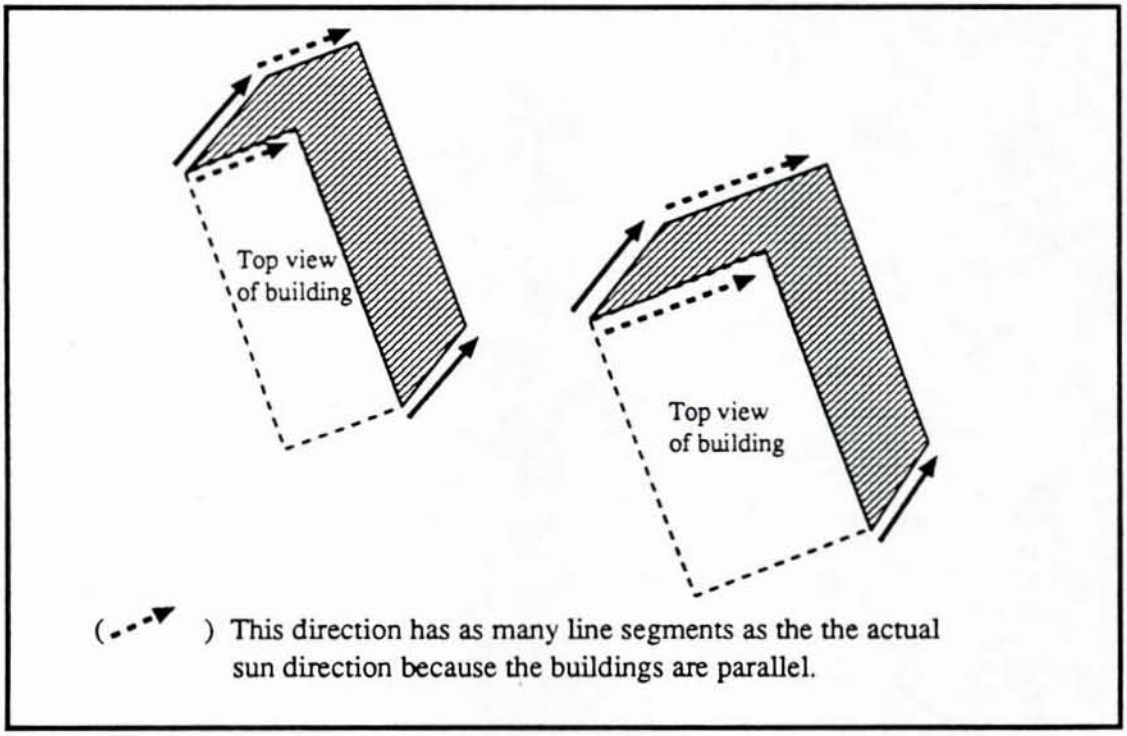
to estimate the sun direction. Our initial thought was to form a frequency histogram which represents the number of shadow boundary line segments in each possible direction or range of directions. Then we can choose the direction which has the most number of line segments to be the sun direction. Figure 21.a illustrates how this method would work in an idealized situation. This method did not work in practice however. One reason for this is that the line segments in the sun direction for each region are often distorted and sometimes too short to provide a reliable direction in a digital image. The second reason is concerned with the typical layout of buildings. In many city and suburban scenes, buildings tend to be parallel to each other. This means that the direction of the line segments shared between buildings and their shadows, as well as the projections of these line segments on the other side of the shadow region, will be parallel for many buildings, see Figure 21.b. This will cause certain directions other than the sun direction to have equal or even higher support than the actual sun direction.

A more reliable way of estimating the sun direction involves finding building corners of the part of the building which caused the shadow and their corresponding shadow corners. The direction of the vector which starts at a corner point and ends at its shadow point is the same as the direction of the sun. This idea is illustrated in Figure 22. In order to find the corresponding corners, we sort the line segments which describe each shadow boundary and we assign a direction to each segment. The sorting and direction assignment are done such that if the end and start of each pair of consecutive line segments is connected, the result is a closed polygon. Once we sort the line segments we can look for pairs of two consecutive line segments which have corresponding parallel line segments.

Since at this time we have no information about which part of the shadow boundary is shared between the shadow and its building, we can find each of the building corners and its corresponding shadow corner only as a pair, without actual knowledge



(a)



(b)

Figure 21: Estimating the sun direction using shadow line segments.

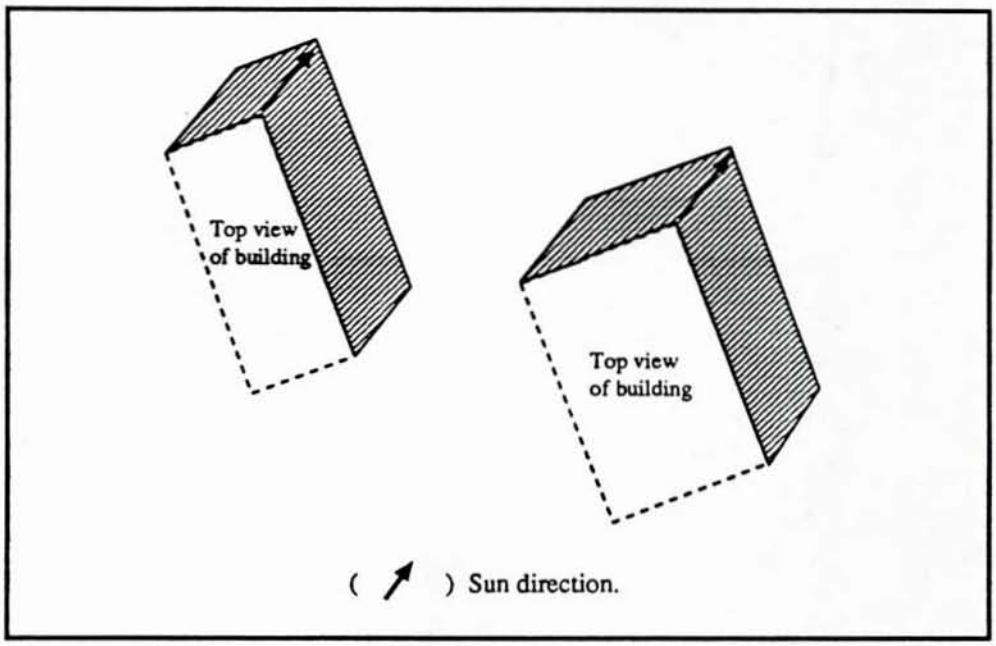


Figure 22: Estimating the sun direction using building corners and their shadows.

of which belongs to the building and which is the shadow. Since we need to make this decision to calculate the sun direction, we assume that the non-convex corner of the shadow is always the one which is shared by the building. This assumption is justified since we take the sun direction to be the one which is supported by the most number of corner pairs, and in a full aerial image, there are typically more convex shadow causing building corners (which correspond to non-convex corners on the border shadow boundary) than non-convex.

We applied our shadow boundary finding technique to the images in Figure 4 and obtained the results in Figures 23 through 27. We also obtained the sun direction for each of the images. Only the two images in figures 23 and 25 gave correct sun direction of 310 degrees and 166 degrees respectively. The sun direction is taken to be 0 degrees in the eastward direction, and to be increasing in the clock-wise direction. It was not possible to find the correct sun direction in the other images due to the lack of the type of corners which our sun direction estimation method relies on.

Figure 23: Shadow boundaries in the image of Figure 4.a.

Figure 24: Shadow boundaries in the image of Figure 4.b.

Figure 25: Shadow boundaries in the image of Figure 4.c.

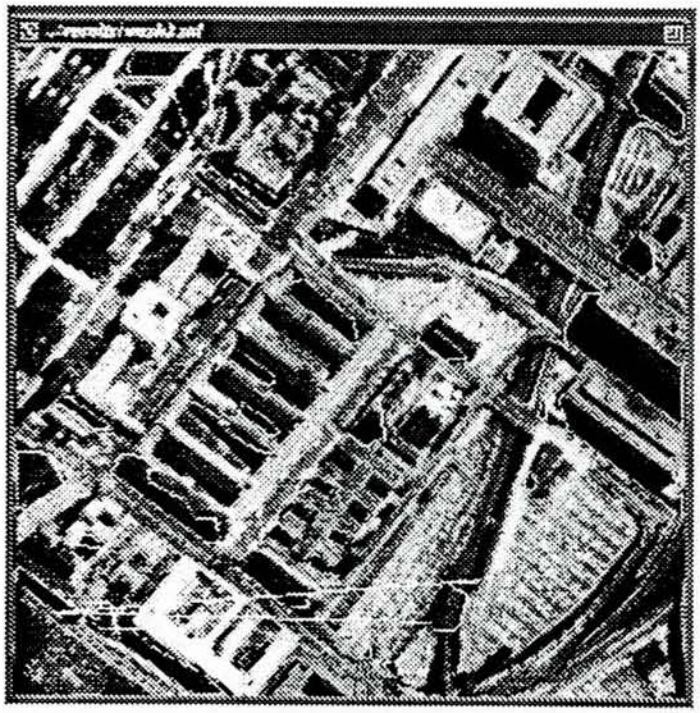


Figure 26: Shadow boundaries in the image of Figure 4.d.

Figure 27: Shadow boundaries in the image of Figure 4.e.

4.4 Determining Building Shape

By finding the shadow boundary for a building and estimating the sun direction, it is clear that we can determine which part of the shadow boundary is shared by its building. In our implementation this is performed by finding two bounding lines of each shadow region such that each of the two lines is in the direction of the sun. Each of these lines passes through a point on the shadow boundary. These points split the shadow boundary loop into two parts. The part of the loop which is closer to where the sun rays are coming from is the shadow building border.

To find the rest of the building boundary, we extrapolate based of the part of the boundary which has been found. We assume that the buildings are parallelogram shaped or are composed of parallelogram shaped blocks. This means that we can simply complete each of the parallelograms half of which is given in the each of the convex (relative to the building boundary) corners of the building-shadow border.

Figures 28 and 29 give the results of extracting the buildings in the images of

Figure 28: Building boundaries found for the image in Figure 23.

Figures 23 and 25 respectively. (We could not obtain meaningful building boundaries for the rest of the images since it was not possible to calculate the sun direction in these images.

4.5 Calculating Building Height

As explained in Chapter 2, we can use the shadow length to estimate the height of its corresponding building. In order to find the shadow length, we calculate the average distance between the shadow-building border part of the shadow boundary and the other part along the direction of the sun. In some situations, it may be the case that the shadows of more than one building are merged as a result of their proximity. This situation may be detected if the buildings have significantly different heights by detecting the significant difference in the shadow length when measured at different points along the boundary.

Figure 29: Building boundaries found for the image in Figure 25.

CHAPTER 5
CONCLUSION

5.1 Discussion of Results

It is clear from examining the results in the previous chapter that using only shadows to find the buildings in an aerial image often produces incomplete results. Only when the image is clear and the shadow cast is completely visible is it possible to find an accurate estimate for the part of the building shape reflected in the shadow or even to decide the existence of a building. Perhaps the most important step in the whole process is the shadow extraction step. This is because most of the processing performed beyond this initial step is completely dependent on how accurate the extracted shadows are. Any large distortion in the shape or number of the initially extracted shadow regions will directly affect the final performance of the algorithm. This condition of having to obtain very good shadows to be able to find good estimates of building properties may not be necessary if we can correct the errors in the shadow boundaries using reliable edges. Unfortunately, obtaining such reliable edges is also dependent on how clear the original image is and how sharp the edges in that image are.

Another problem which can be solved by obtaining reliable edges is determining the complete shape of buildings. As we indicated in previous chapters, a shadow region reflects only a portion of its building boundary. Ideally we would like to have enough edge information to reconstruct the missing parts of building boundaries. This is very difficult to do since the edges of the buildings which do not border

the shadow regions are typically very weak and therefore are difficult to detect by an edge detector. Extrapolating based on building-shadow borders is a reasonable option only if the buildings happen to be parallelogram shaped. However, if the buildings are irregular in shape then shadows do not provide any more information other than the border information.

5.2 Future Research

Most work which has been done in the area of aerial photograph interpretation so far remains highly restrictive, which makes it less useful in practical applications. Our algorithm utilizes a single aspect of the information contained in an aerial image to infer some information about the real world scene represented by this image. This algorithm is not meant to be an independent method of providing all information about all buildings in an image. However this algorithm along with others can be integrated into a full system of aerial photograph interpretation. A final complete interpretation system is not likely to be composed of separate modules but rather of closely connected modules each of which depends on information provided by the others to make its decisions.

Because of the complexity of any typical scene captured in an aerial image, it was important to deal with solving the problem of interpreting such images by dividing the problem into parts. From all the research work which has been done on various subparts of this problem a very sophisticated interpretation system can be implemented. This task of combining the various methods and sources of information into a full system will take much time and effort but it will provide a very important tool which will be of great use in several applications. It will also be beneficial if a system is implemented to take color images. This will solve certain problems such as distinguishing between vegetation and shadow areas which usually have similar intensities in gray-scale images.

5.3 Summary

I have provided a complete description of my thesis project which involved the investigation of the idea of using shadows to estimate building properties in aerial photographs. The first two chapters provided general introduction and full explanation of the background for this work. Chapter 2 also provided the theoretical bases for the connection between buildings and their cast shadows. Chapter 3 provided the idea of establishing "ground truth" and concentrated on extracting shadow regions. Chapter 4 described in detail an algorithm which used the extracted shadows to find buildings and estimate their height. Finally, this conclusion chapter provided some discussion and suggestions for future work.

LIST OF REFERENCES

[1] T. E. Avery. *Interpretation of Aerial Photographs.* Burgess Publishing Company, Minneapolis, MN, 1977.

[2] D. Ballard and C. Brown. *Computer Vision.* Printice Hall, Englewood Cliffs, New Jersey, 1989.

[3] P. Besl and R. Jain. "three-dimensional object recognition". *Computing Surveys*, 17(1):75–145, 1985.

[4] J. Beveridge, J. Griffith, R. Kohler, A. Hanson, and E. Riseman. "segmenting images using localized histograms and region merging". *International Journal of Computer Vision*, 2:311–347, 1989.

[5] J. Canny. "a computational approach to edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–697, 1986.

[6] C. Chow and T. Kaneko. "automatic boundary detection of the left ventricle from cineangiograms". *Computers and Biomedical Reasearch*, 5:388–410, 1972.

[7] Jr. David M. McKeown and Jerry L. Denlinger. "cooperative methods for road tracking in aerial imagery". In *IEEE Computer Vision and Pattern Recognition Conference*, pages 662–672, 1983.

[8] John A. Howard. *Aerial Photo-Ecology.* American Elsevier Publishing Company, Inc., New York, NY, 1970.

[9] A. Huertas and R. Nevatia. "detection of buildings in aerial images using shape and shadows". In *The 8th International Joint Conference on Artificial Intelligence*, pages 1099–1951, 1983.

[10] A. Huertas and R. Nevatia. "detecting buildings in aerial images". In *Computer Vision, Graphics, and Image Processing*, volume 41, pages 131–152, 1988.

[11] R. Bruce Irvin and Jr. David M. McKeown. "methods for exploiting the relationship between buildings and their shadows in aerial imagery". In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 19, pages 1564–1575, 1989.

[12] A. Jain. *Fundamentals of Digital Image Processing.* Printice Hall, Englewood Cliffs, New Jersey, 1989.

[13] Yuh-Tay Liow and Theo Pavlidis. "use of shadows for extracting buildings in aerial images". In *Computer Vision, Graphics, and Image Processing*, volume 49, pages 242–277, 1990.

[14] M. Nagao, T. Mstsuyama, and I. Ikeda. "region extraction and shape analysis in aerial photographs". In *Computer Graphics and Image Processing*, volume 10, pages 195–223, 1979.

[15] A. Shafer. *Shadows and Silhouettes in Computer Vision.* Kluwer Academic Publishers, Boston, 1985.

[16] R.K. Singh and R.S. Ramakrishna. "shadows and texture in computer vision". In *Pattern Recognition Letters*, volume 11, pages 133–141, 1990.

[17] D. Waltz. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, 1975.

[18] Michael T Checky William B. Thompson and William F Kaemmerer. "shadow stereo - locating object boundaries using shadows". In *Sixth National Conference on Artificial Intelligence*, pages 761–766, 1987.

Appendix B:

The Recovery of 3D Information using Stereo

Vineet Goel, Mubarak Shah
University of Central Florida

Contents

1 Introduction

The goal of the stereo vision is the recovery of three- dimensional information from images taken from different viewpoints. Two cameras located at two positions can be used to take two images, or one camera can be used to take two images from two different positions. As one takes images of an object from two different positions, the object is shifted in one image relative to that in other image. This shift is inversely proportional to the depth of the object, which is the distance of the object from camera. We call this shift disparity, and hence it is the measure of three dimensional depth.

The basic problem in stereo is that, one has to match a point from the left image to a point in the right image. One point in the left image can correspond to only one point in the right image. The problem of matching a point from left image to a point in the right image is called correspondence problem. All the stereo algorithms try to solve this problem.

If we take two images with the cameras which are aligned, the matching problem can be restricted to one dimensional search along any row, i.e. we do the matching of the points in the same row in both images, we don't match a point in one image to the point in other image which is not in the same row. This constraint is known as *epipolar constraint* .

Establishing correspondence problem is the main task in a stereo algorithm, this involve two questions: what to match? and how to match? Correspondence between two images can be established by matching specific features such as blobs or edges, or by matching small regions by direct correlation of image intensities without identifying features. After feature matching, the next step in stereo is to apply continuity constraint, which states that *disparity varies smoothly almost everywhere, and that only a small fraction of the area of an image is composed of bounderies that are discontinous in depth* . When disparities at some points are not known, a surface between known disparities can be interpolated using the smoothness criterion. Since these disparities are direct measure of the depth, we can interpolate a surface going through all points with given disparities.

This report summarizes our work during the first year of this project. The first six months of this project were devoted to the literature search on the stereo algorithms. A number of recent papers from the current literature were studied and their approaches analyzed. Three representative algorithms were selected for further study. During the next six months

the algorithms were implemented, and tested on a representative set of stereo pairs. This report describes our experiments, and the comparitive study of these algorithms. During the second year of this project we will focus on improving these algorithms by incorporating the smoothness and continuity criteria. We will also attempt to introduce some other 3D cues like *shading* in the stereo algorithms. It is expected that the performance of such method using multiple cues will be much better than the one using only single cue. We will also start an initial study related to interpolation of surface from the dispariy maps.

In the next section, we will discuss three stereo algorithms : Normalized Correlation Method, Difference Method and Prazdny's Method. Section 3 presents the results obtained for several images with these algorithms. We end this report with section 4 which conclude this report. We have also included the source codes of three algorithms, and Canny's edge detector written in **C** language in the Appendix A. This will make it easy to run these algorithms on any other system for further experimentation. In addition, the demos for running our programs are given in the Appendix B.

2 Algorithms

The three algorithms which are used here, basically try to match corresponding points in images and get the difference in their positions, which as explained in previous section is measure of the distance of a point in the object from camera. Cochran and medioni [2], have proposed an algorithm which declares a point as corresponding point to a ipoint at which highest Normalized correlation is obtained. The algorithm reported by Kayalap [3], takes sum of the absolute difference of the intensities of the neighbouring points of a point in the left image to the corresponding neighbouring points of a point in the right image. The point in the right image, which gives the minimum sum of difference, is the matching point. Prazdny's [1] algorithm chooses a matching point which gets the maximum support from its neighbours.

Each of the algorithm is discussed briefly in the following sections.

2.1 Normalized Correlation Method (Cochran and Medioni)

In this method, to match a point x in the left image to the point y in the right image, we first get the mask of given size around x. Then we compute the normalized correlation of that mask with the mask around point y in the right image. The point y at which we get the maximum normalized correlation value is the matching point. The difference in the position of x and y is the disparity value for x. If multiple points with maximum normalized correlation values are obtained, we select one disparity value for the point x, based on the smoothness criterion. The smoothness criterion assume that a point x has almost the same disparity value as of its neighbouring points. Finally, the edgels from the original intensity images can be extracted, and used for possible locations of the depth discontinuties.

In order to get disparity for a point (x, y), we compute

$$A(x_i, y) = \sqrt{\sum_{j=-m}^{m} \sum_{k=-m}^{m} R(x_i + k, y + j)^2}$$

$$c(x_i, y) = \frac{\sum_{j=-m}^{m} \sum_{k=-m}^{m} L(x + k, y + j) \times R(x_i + k, y + j)}{A(x_i, y)}$$

where

$$x - d \leq x_i \leq x + d,$$

d is the disparity, m is the mask size, $L(x, y)$ and $R(x, y)$ respectively are the gray values in left and right image at (x, y). The point (x_i, y) at which value of $c(x_i, y)$ is maximum is the matching point for (x, y), and $x_i - x$ is the disparity value for (x, y).

2.2 Sum of Absolute Difference Method (Kayalap)

This method is based on sum of absolute differences of intensity values around the small neighborhood of the two points in the left and right images. To get a point y in the right image corresponding to the point x in the left image, first the mask of given size around x is obtained. Then the sum of absolute differences (SAD) of this mask with the mask centered around point y in right image is computed. The point in the right image which has minimum SAD is the point y matched to the point x in the left image. The difference in the position of x and y is the disparity for x. The disparity for all the points in the left image are computed

in the similar fashion. In this case also, we may get multiple points with the same SAD for a point x. The smoothness criterion can be used to to resolve the disparity conflict.

The disparity for a point (x, y) is computed as follows:

$$D(x_i, y) = \sum_{j=-m}^{m} \sum_{k=-m}^{m} \mid L(x+k, y+j) - R(x_i+k, y+j) \mid$$

where

$$x - d \leq x_i \leq x + d,$$

d is the disparity and m is the mask size. The point (x_i, y) at which the value of $D(x_i, y)$ is minimum is the matching point and, $x_i - x$ is the disparity value for (x, y).

2.3 Prazdny's Method (Prazdny)

In this method, the disparity is computed at edge points only unlike the previous methods where the disparity was computed for each point. Therefore, the first step in this method is to apply the edge detector to left and right images. Prazdny's method is based on smoothness criterion. The point under consideration should have the same disparity value as of most of its neighbours. This implies that the neighbours should support a disparity value for a point under cosideration. Prazdny has given an expression which calculates support for a disparity value of a point based on above criterion. This expression is called similarity function. The sum of similarity function for each possible disparity is calculated. A final disparity value for which this summation is maximum is assigned to the point.

The similiarity function should meet three requirements :

1. The disparity similarity function should be inversely proportional to the difference of disparities of interacting points.

2. More distant points should exert less influence while nearby points should have more disambiguiting power.

3. The more distant the two interacting points are the less seriously should their disparity difference be considered. Because of inherent uncertainity : steeply-sloped surfaces will generate large disparity differences which should nevertheless contribute to disambiguation. For large seperations one should probably expect a flat support function.

The similarity function capturing all of these requirements is :

$$s(i,j) = \frac{e^{-(D_i - D_j)^2/(2c^2(i-j)^2)}}{c \mid i-j \mid \sqrt{2\pi}}$$

Here, $s(i,j)$ expresses the amount of support disparity D_i at point i, receives from disparity D_j at another point j, and $\mid (i-j) \mid$ is the distance between two points, c is scaling costant, usually taken between 0.55 and 0.85.

The first step in this method is to compute all the potential disparities for each point in the left image. We will explain this step by taking an example in one dimension. This example can easily be extended to two dimensions.

Let $\otimes$ denote exclusive nor operation, and let two one dimensional images of size 1×5 are

left(L) image

0	1	1	0	0

Right(R) image

0	0	1	1	1

then C^0 is defined as:

$$C^0(x,d) = L(x) \otimes R(x+d)$$

For this example the initial disparity map, C^0 is given by:

	1	**2**	**3**	**4**	**5**
2		1	1		
1	1	1	1		
0	1		1		
-1					
-2				1	

The first column (shown in bold) represents possible disparity values ranging from -2 to $+2$ in this case. While the first row (shown in bold) denotes the x location of a point.

The disparity values influence each other as follows. Suppose that a left image feature point at location i has a set of possible disparities D_i and we are interested in the amount of support a particular disparity d_i receives from the feature point j (with possible disparity set D_j), which is neighboring point of i. First, the disparity d_j from set D_j of point j is selected, and the support is computed using similiarity function. In the same way, the support for a disparity value d_i is computed using all disparity values d_j of all neighbors j of point i. Then all these supports are added to get the support for disparity d_i of point i. The disparity d_i for point i which receives the maximum support is the final disparity value assigned to the point i.

3 Results

The algorithms discussed in the previous sections were tested on a set of six stereo pairs: Renault, Sandwich, Pentagon, Sphere, Ruts, and Rocks. The Renault pair shows the auto part used in the Renault car, while the Sandwich pair shows a Sandwich resting on a flat surface, and the Pentagon pair is an aerial view of the Pentagon with some cars in the parking lot in the background. The Sphere pair is synthetically generated which shows a concrete shpere on the table. Finally, the Ruts and Rocks pairs are real images which show respectively the mound of rocks, and ruts. The images were acquired from Professor Ramesh Jain at University of Michigan, Ann Arbor. The original images were of the size 512×512, but were reduced to 128×128 for timing constraint. All the algorithms were run on Sun SparcStation-1.

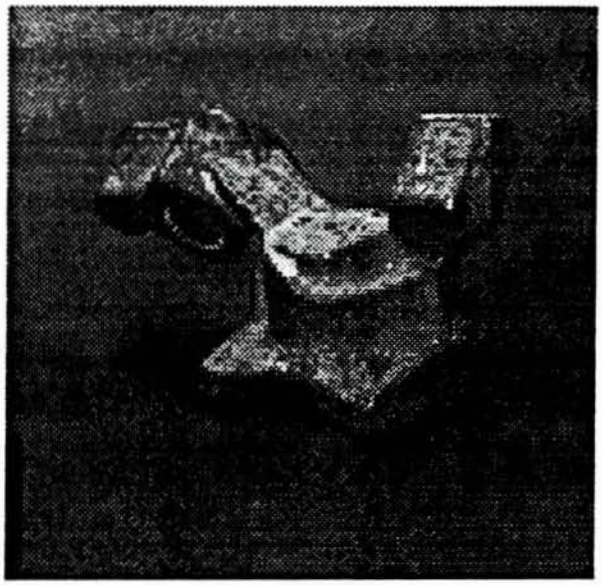
3.1 Normalized Correlation Method

The results are shown in Figures 2–7. The informations regarding the image size, disparity range, masksize and cpu time in seconds are given in the table shown in Figure 1.

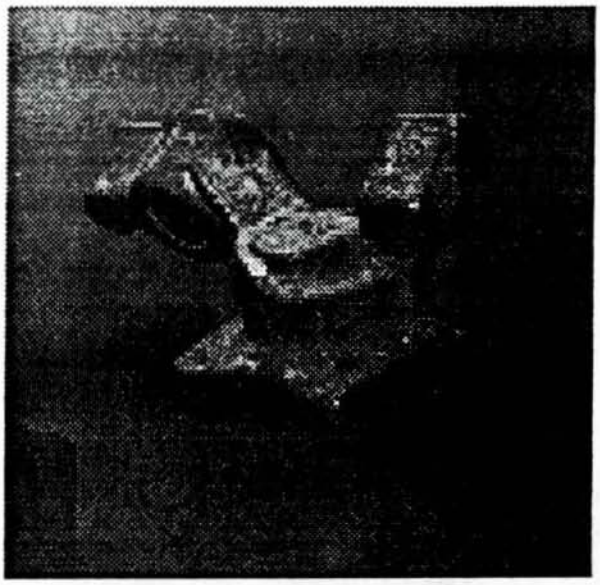
Disparity map for the Sandwich image shown in Figure 3(c), is almost continouous on the surface of sandwich. There is some discontinuity which can be avoided by using continuity criterion. In case of Pentagon image shown in Figure 4, it was observed that smaller mask size gave rise to the finer details of the Pentagon building. However, lot of discontinuities in the background were also observed. On the other hand, with large mask size the finer details were lost but background became smoother. Disparity map for Renault shown in Figure 2(c) does not have much discontinuity on the surface of Renault, and for larger mask the background is smoother also. The disparity maps for Sphere, Ruts and Rocks shown in Figure 5(c), 6(c) , 7(c) respectively, are not of good quality. It is difficult to perceive anything significant from disparity maps of Ruts and Rocks. The input images themselves are very difficult to analyze, these images have lot of symmetry. In this case, when selecting a particular disparity value the algorithm encourages multiple matches. Therefore, the results are not good.

Figure	*Image*	*Size*	*DisparityRange*	*Masksize*	*Time(sec)*
2	*Sandwich*	128 × 128	13	11	782
3	*Pentagon*	128 × 128	4	11	257
4	*Renault*	128 × 128	10	11	609
5	*Sphere*	128 × 128	17	11	2114
6	*Ruts*	128 × 128	17	11	1004
7	*Rocks*	128 × 128	38	11	2049

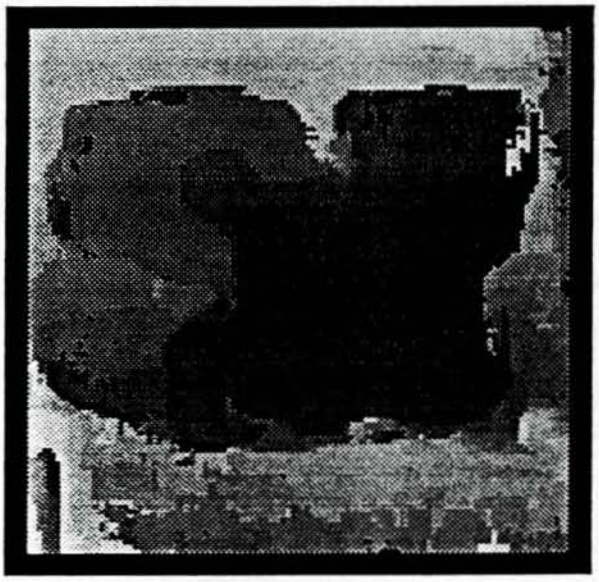
Figure 1: Summary of result for Normalized Correlation Method.



(a)

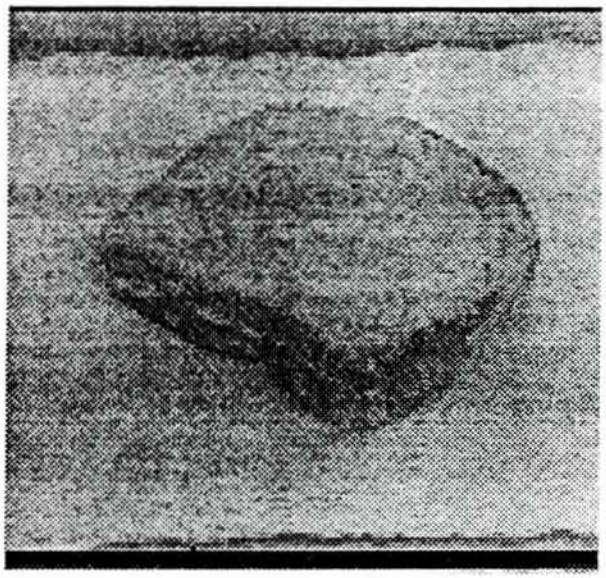


(b)

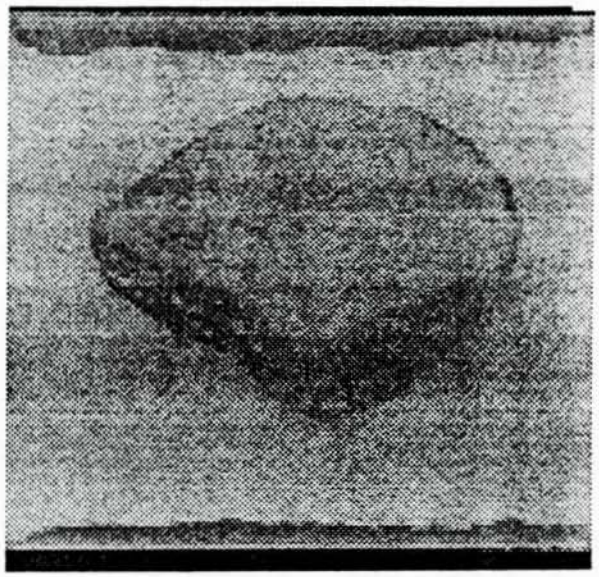


(c)

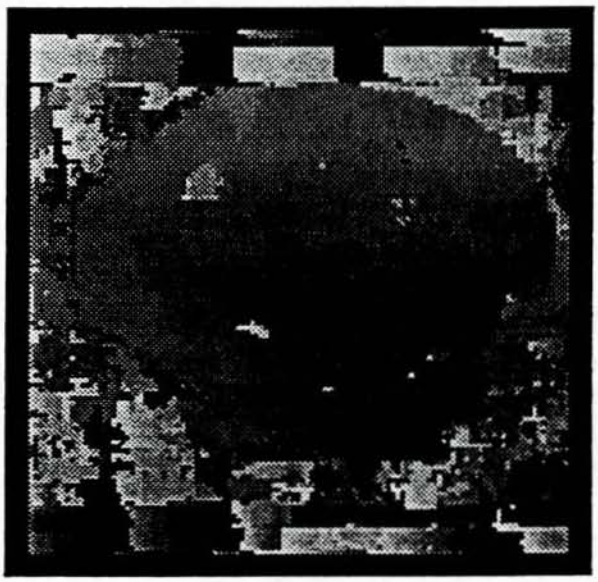
Figure 2: Result of Normalized Correlation Method for Renault Image. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.



(a)



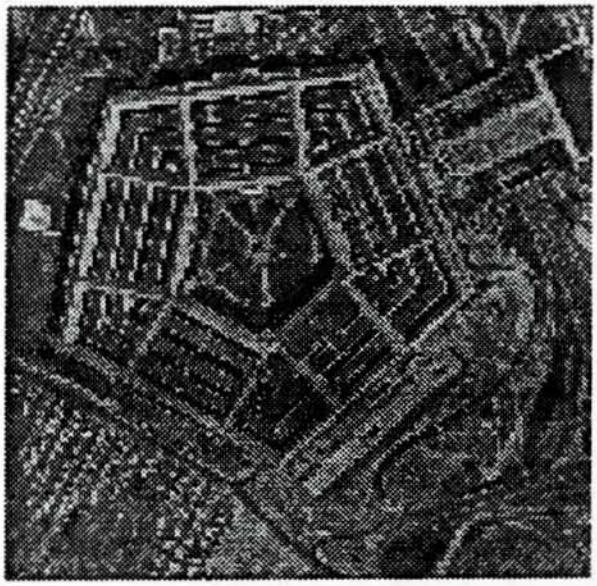
(b)



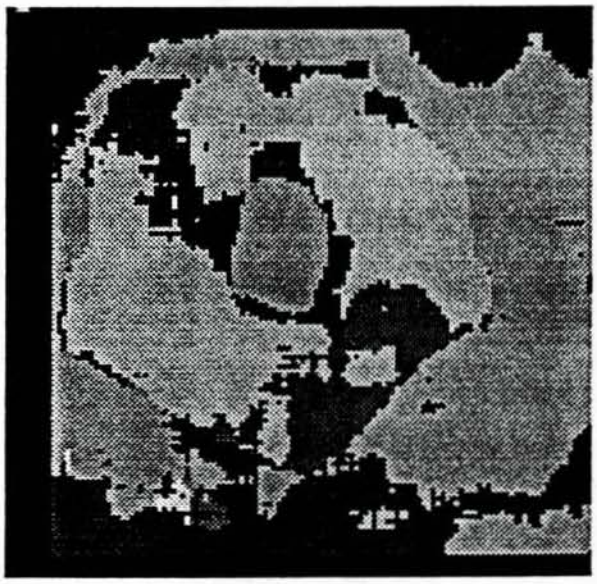
(c)

Figure 3: Result of Normalized Correlation Method for Sandwich Image. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.

(a)

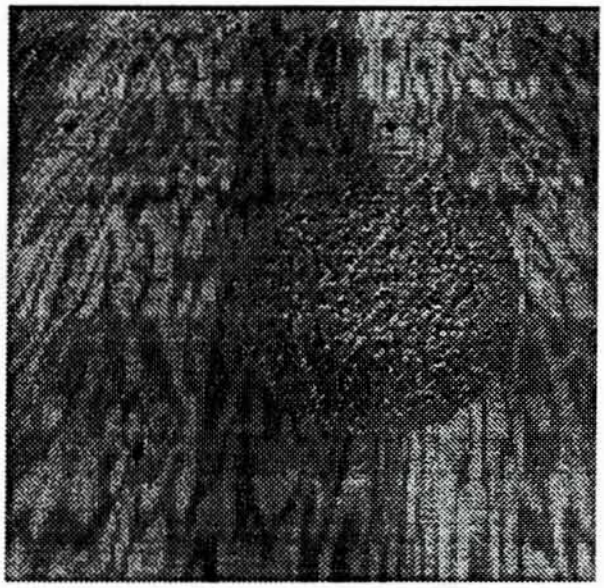


(b)

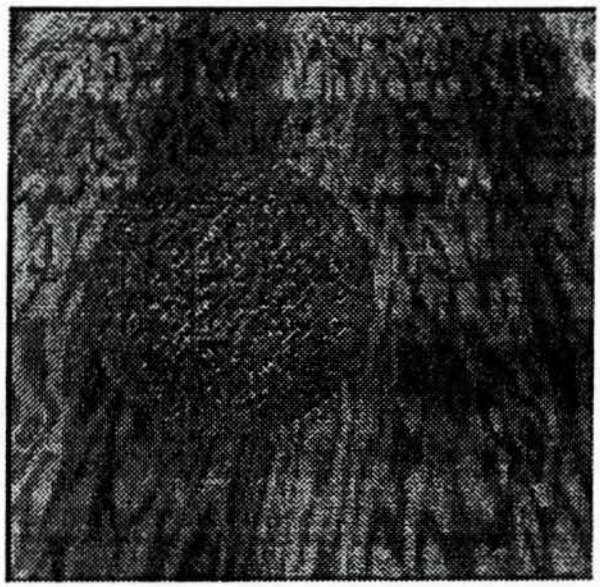


(c)

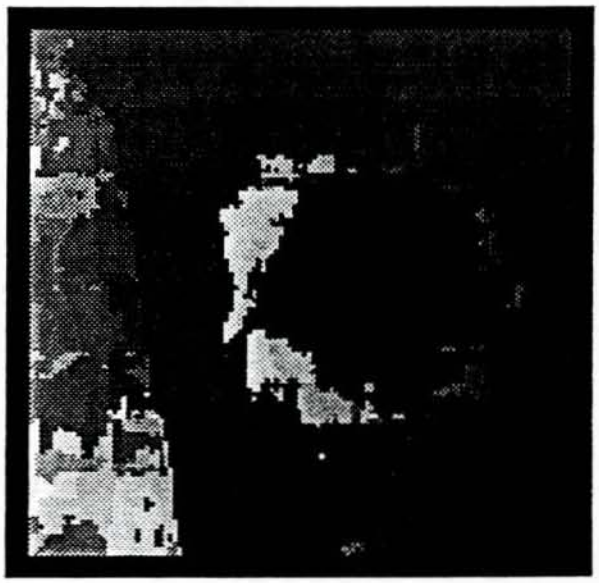
Figure 4: Result of Normalized Correlation Method for Penatgon Image. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.



(a)

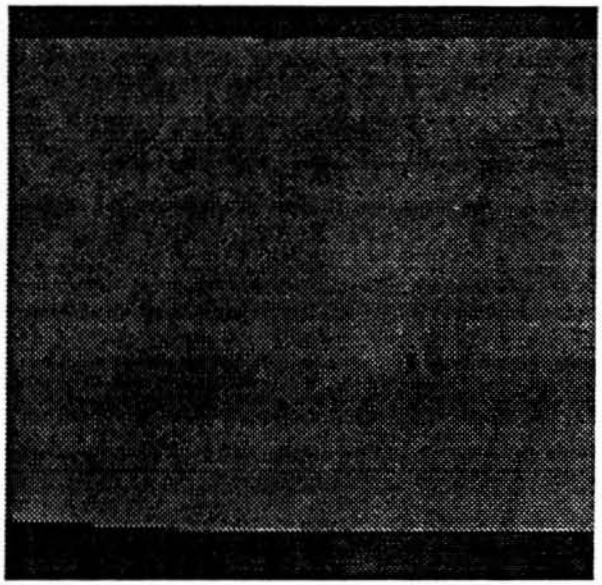


(b)

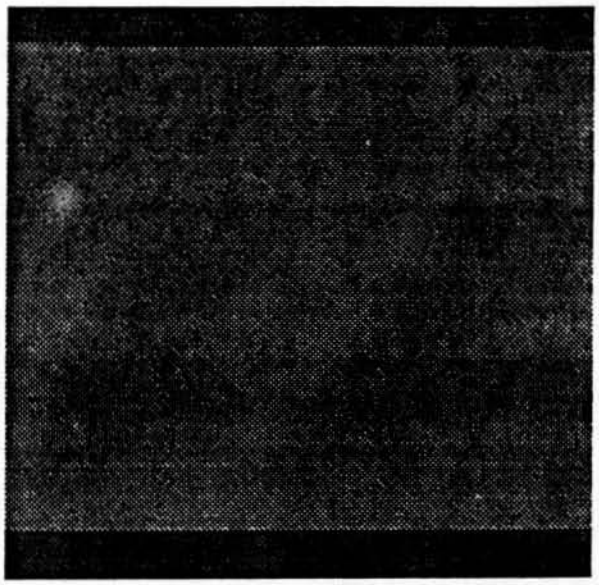


(c)

Figure 5: Result of Normalized Correlation Method for Sphere. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.



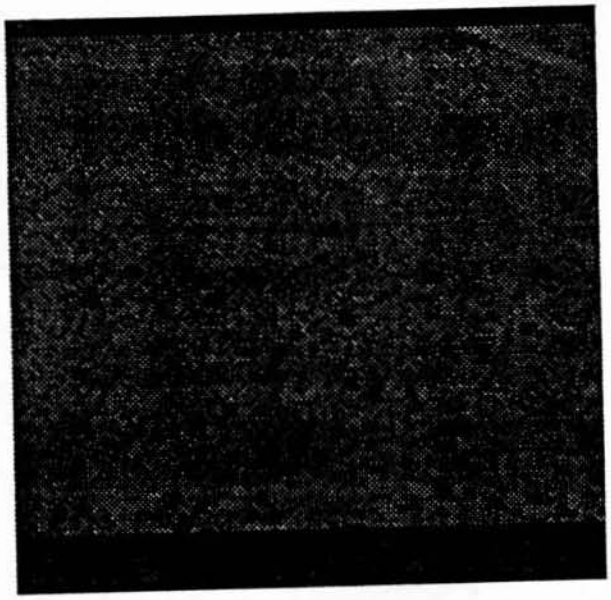
(a)



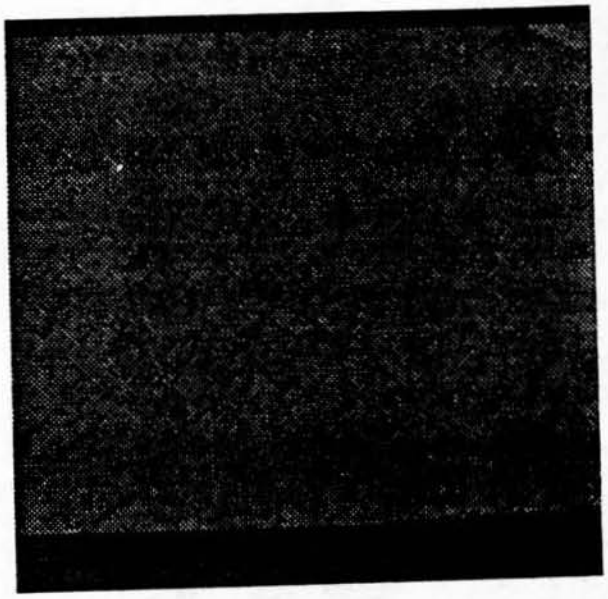
(b)

(c)

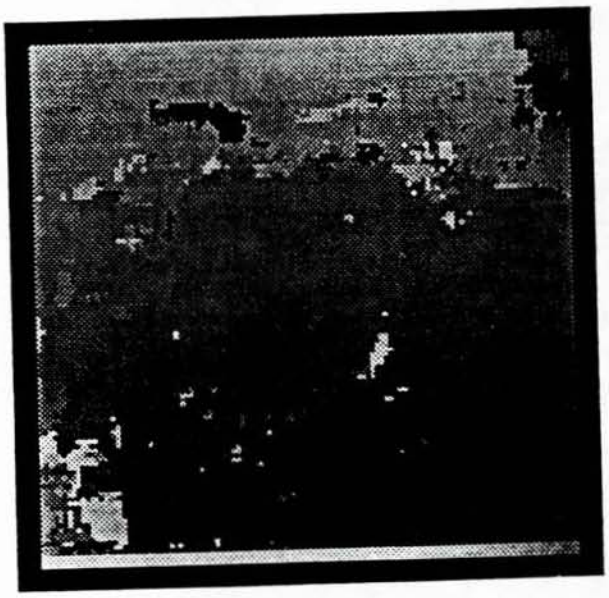
Figure 6: Result of Normalized Correlation Method for Ruts. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.



(a)



(b)



(c)

Figure 7: Result of Normalized Correlation Method for Rocks. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.

3.2 Difference Method

The results for this method are shown in Figures 9–14. The informations regarding the image size, disparity range, masksize and cpu time in seconds are given in the table shown in Figure 8.

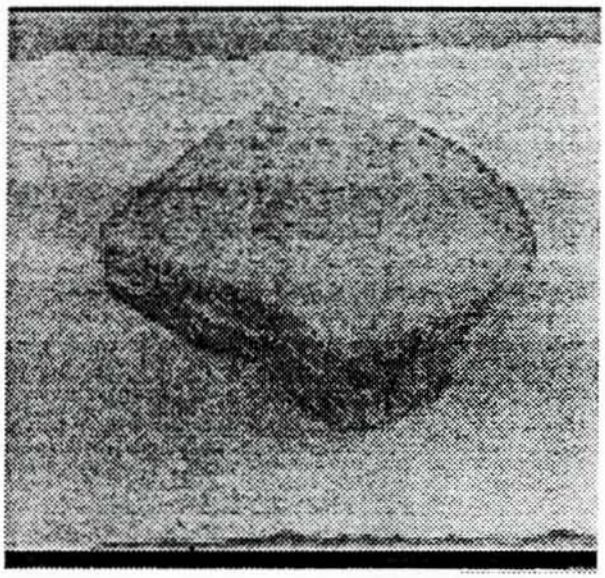
The disparity map for the Sandwich pair is shown in Figure 9(c). It is clear from the result that some of the details are missing, which can be filled in using the continuity criterion.

The disparity map for the Pentagon pair shown in Figure 10(c) does not contain finer details. The results for the Sphere pair shown in Figure 12(c) are very good with this method. The sphere is clearly visible.

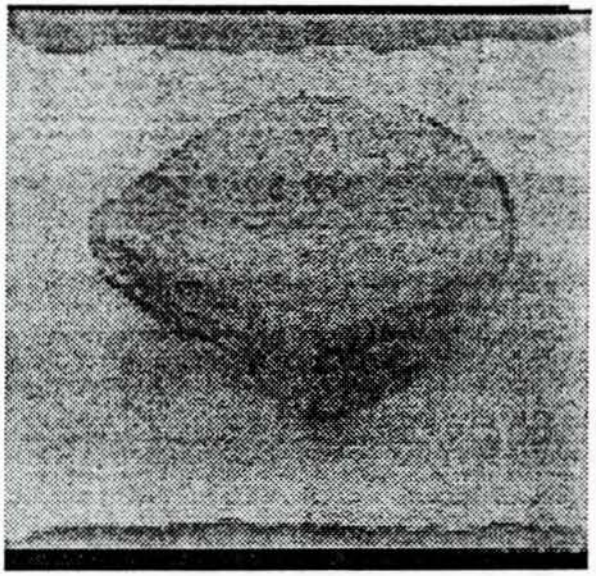
In summary, we get almost similiar results with this method as compared to the the correlation method discussed in the previous section. However, the difference method is the order of magnitude faster than the normalized correlation method. The normalized correlation method requires additional square root operation for each point, which slows it down.

Figure	*Image*	*Size*	*DisparityRange*	*Masksize*	*Time(sec)*
8	*Sandwich*	128 × 128	13	11	327
9	*Pentagon*	128 × 128	4	11	122
10	*Renault*	128 × 128	10	11	261
11	*Sphere*	128 × 128	17	11	391
12	*Ruts*	128 × 128	17	11	411
13	*Rocks*	128 × 128	38	11	800

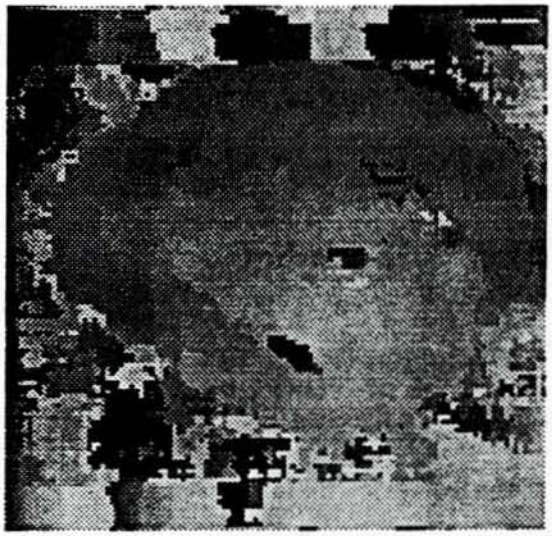
Figure 8: Summary of Results for Difference Method.



(a)



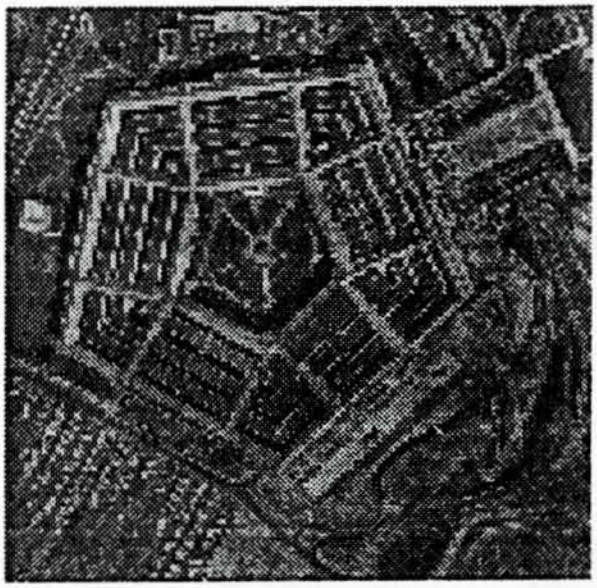
(b)



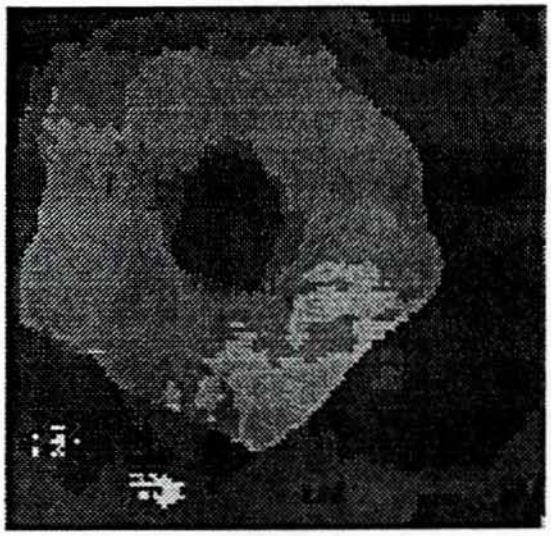
(c)

Figure 9: Result of Difference Method for Sandwich. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.

(a)

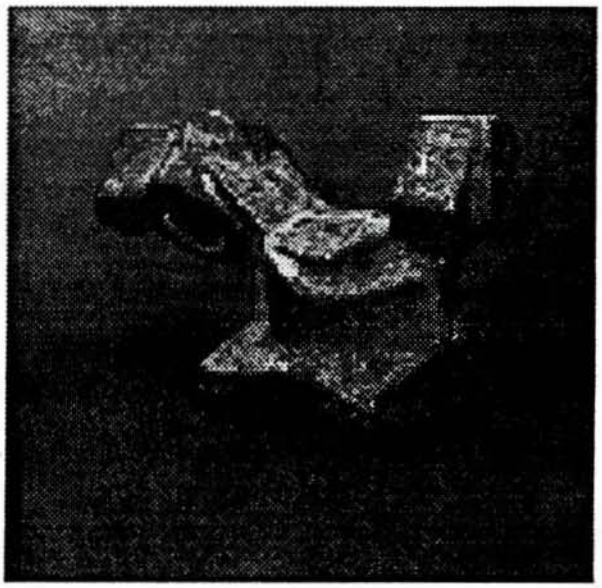


(b)

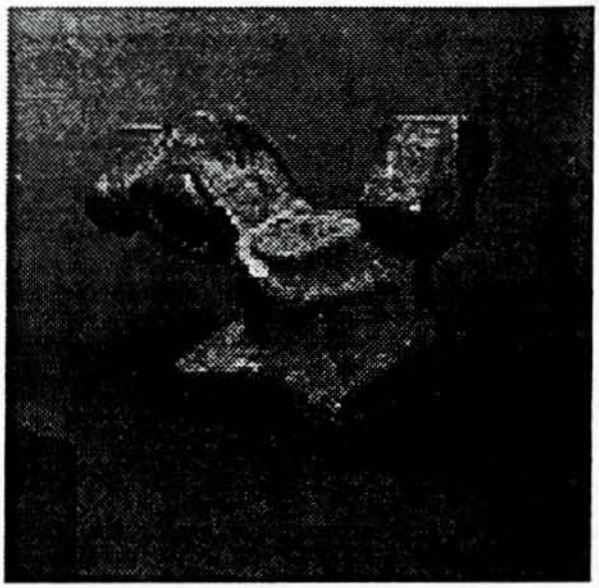


(c)

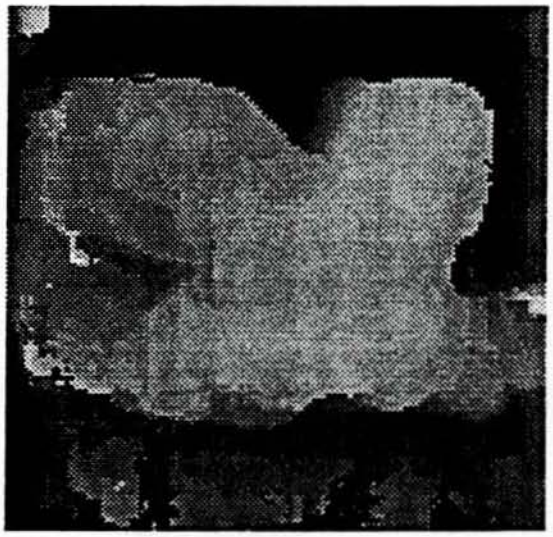
Figure 10: Result of Difference Method for Pentagon. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.



(a)

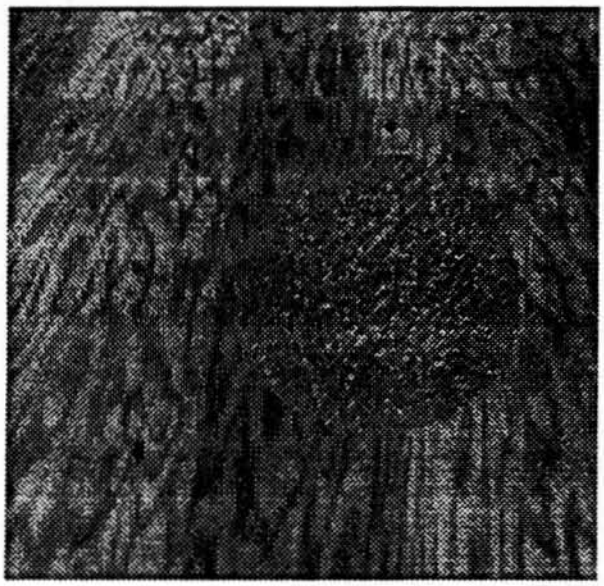


(b)

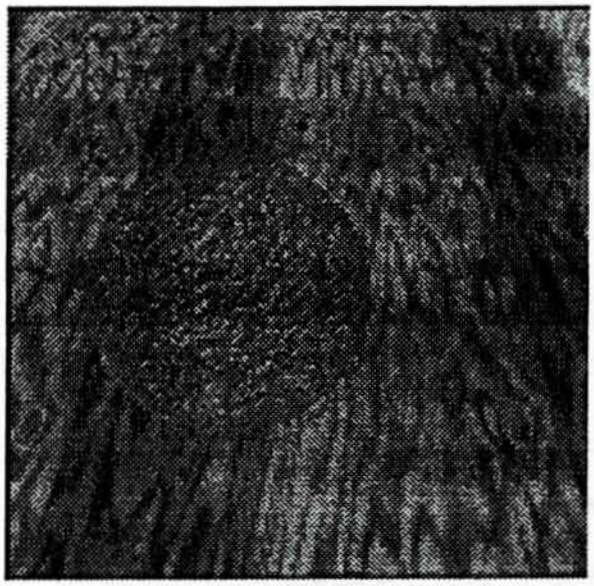


(c)

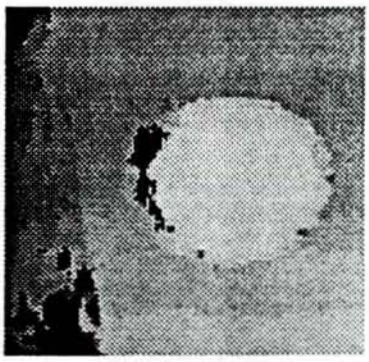
Figure 11: Result for Difference Method for Renault. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.



(a)

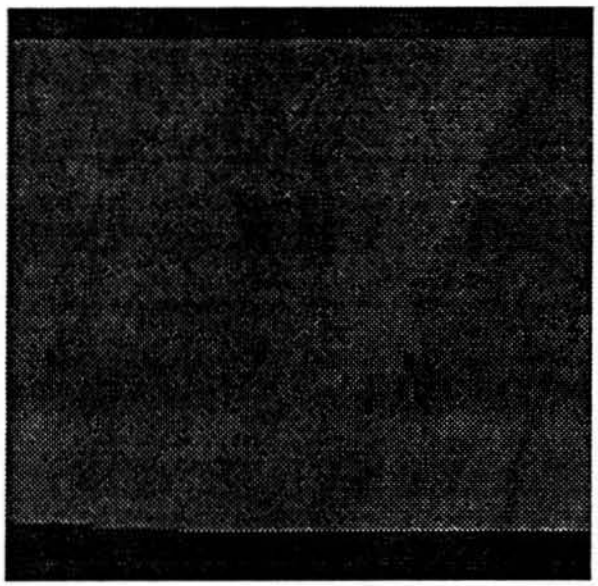


(b)

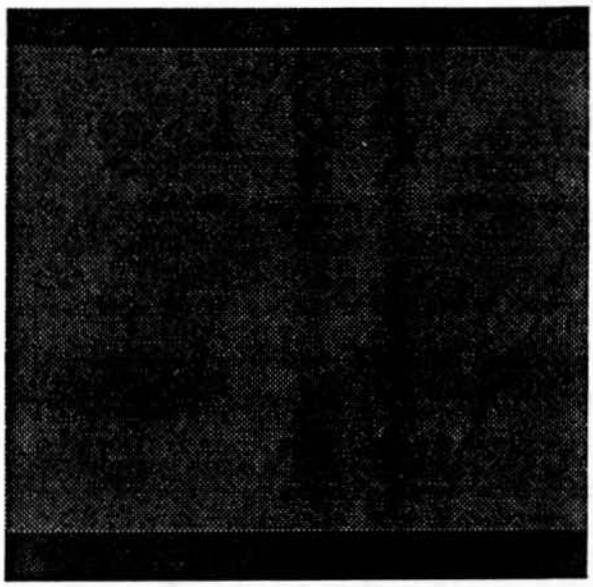


(c)

Figure 12: Result for Difference Method for Sphere. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.



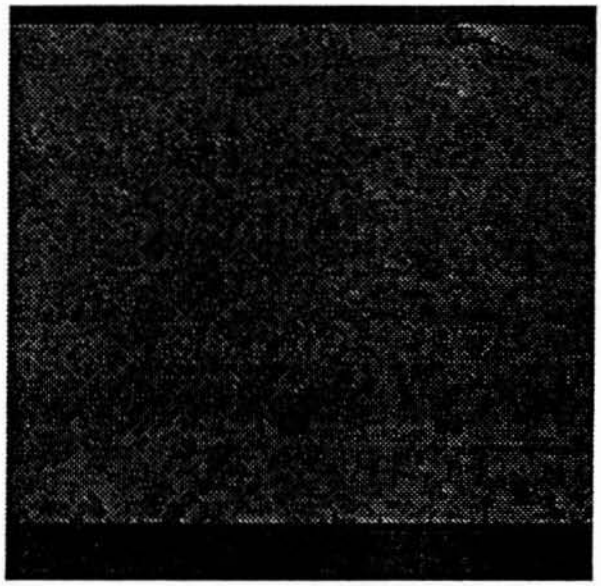
(a)



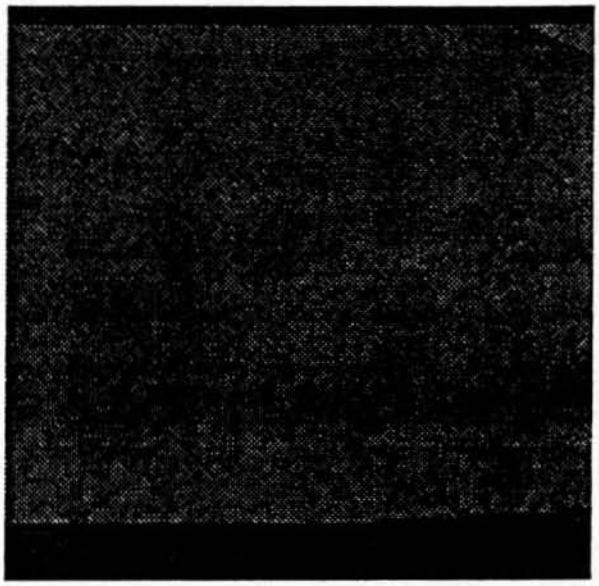
(b)

(c)

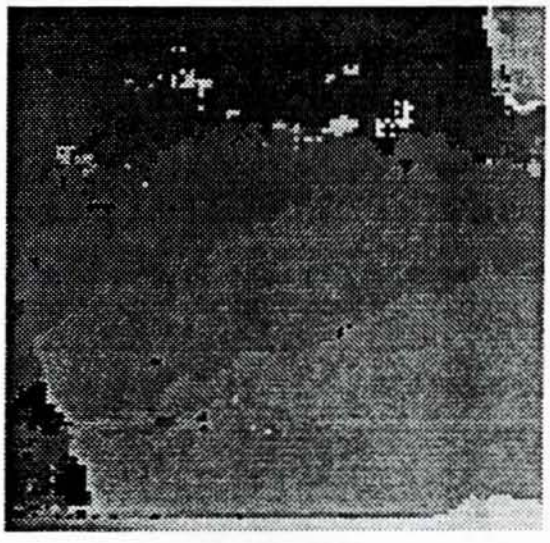
Figure 13: Result for Difference Method for Ruts. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.



(a)



(b)



(c)

Figure 14: Result for Difference Method for Rocks. (a) Left Image, (b) Right Image, (c) Disparity Map. Darker parts are closer to the viewer.

3.3 Prazdny Method

The results obtained with Prazdny's method are shown in Figure 16–21. The informations regarding the image size, disparity range, masksize and cpu time in seconds are given in the table shown in Figure 15. The value of constant c in the similiarity function is taken as 0.65. Canny's edge detector was applied to get the negative and positive edges for all the images.

The left and right images for the Sandwich pair are shown in Figures 16 (a)-(b), corresponding edge maps are shown in Figure 16(c)-(d), and the disparity map is shown in Figure 16(e).

The results for the Pentagon pair are shown in Figure 17. The disparity map obtained for this pair is quite good. The disparity map for Renualt pair is shown in Figure 18(e).

Next, the results for the Sphere pair are shown in Figure 19. From the edge maps of this pair, it was observed that disparity for this pair is very high. Therefore, a high disparity value range was given as input to the algorithm. The benefit of edge based method is that, we can easily observe from edge map what maximum disparity value is to be given as input. For this pair we get much better result as compared to methods discussed in previous subsections.

Quite good results were obtained with this algorithm for Ruts and Rocks stereo pairs shown in Figures 20 and 21(e) respectively, as compared to previous two methods.

Figure	*Image*	*Size*	*DisparityRange*	*Masksize*	*Time(sec)*
16	*Sandwich*	128 × 128	13	11	16
17	*Pentagon*	128 × 128	4	11	19
18	*Renault*	128 × 128	10	11	65
19	*Sphere*	128 × 128	13	11	29
20	*Ruts*	128 × 128	13	11	16
21	*Rocks*	128 × 128	13	11	18

Figure 15: Summary of results for Prazdny's Method.

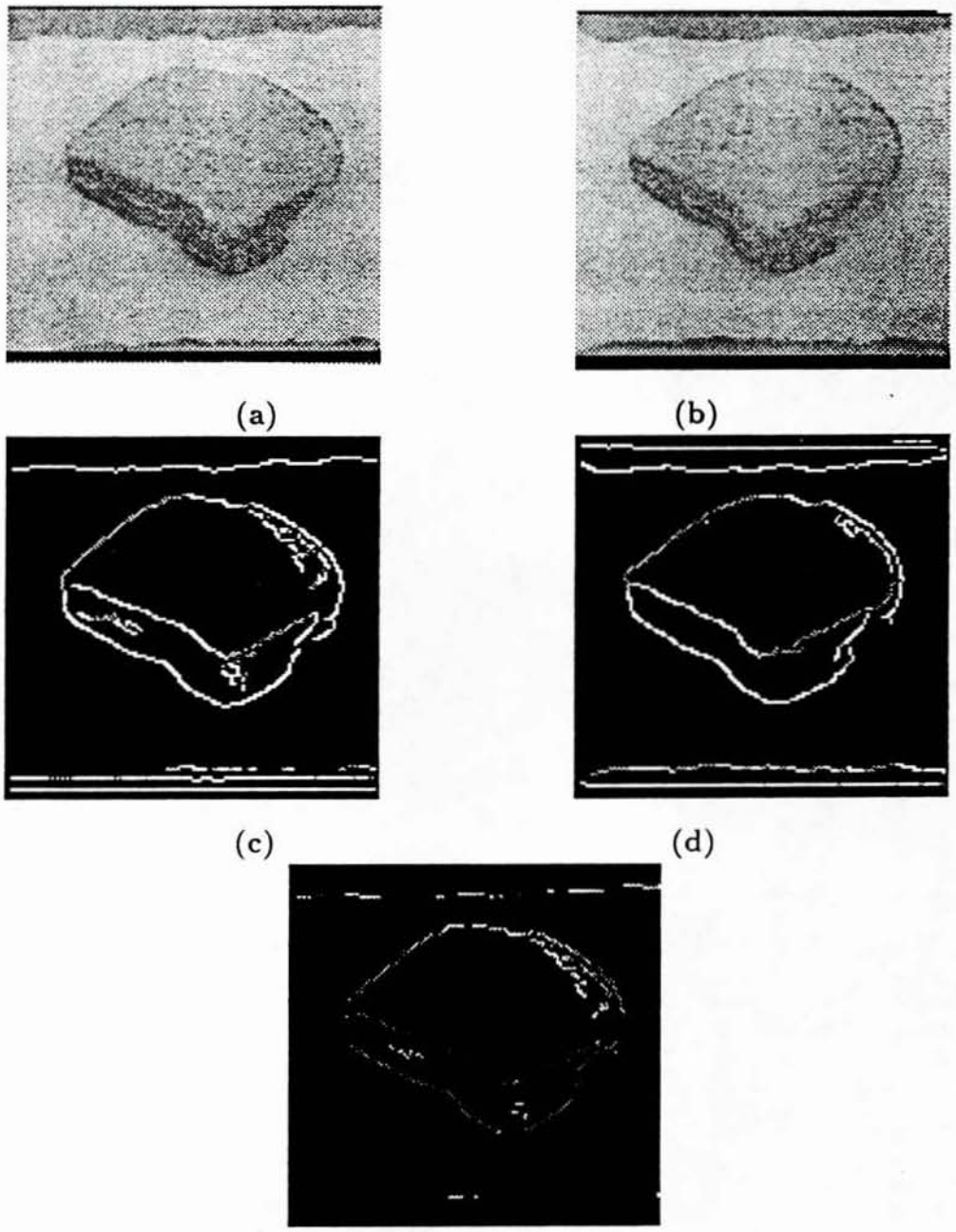


Figure 16: Result for Prazdny's Method for Sandwich pair. (a) Left Image, (b) Right Image, (c) Left Edge Map, (d) Right Edge Map. Positive edges are brighter than negative edges. (e) Disparity Map at edge points only. Brighter edges are far from the viewer.

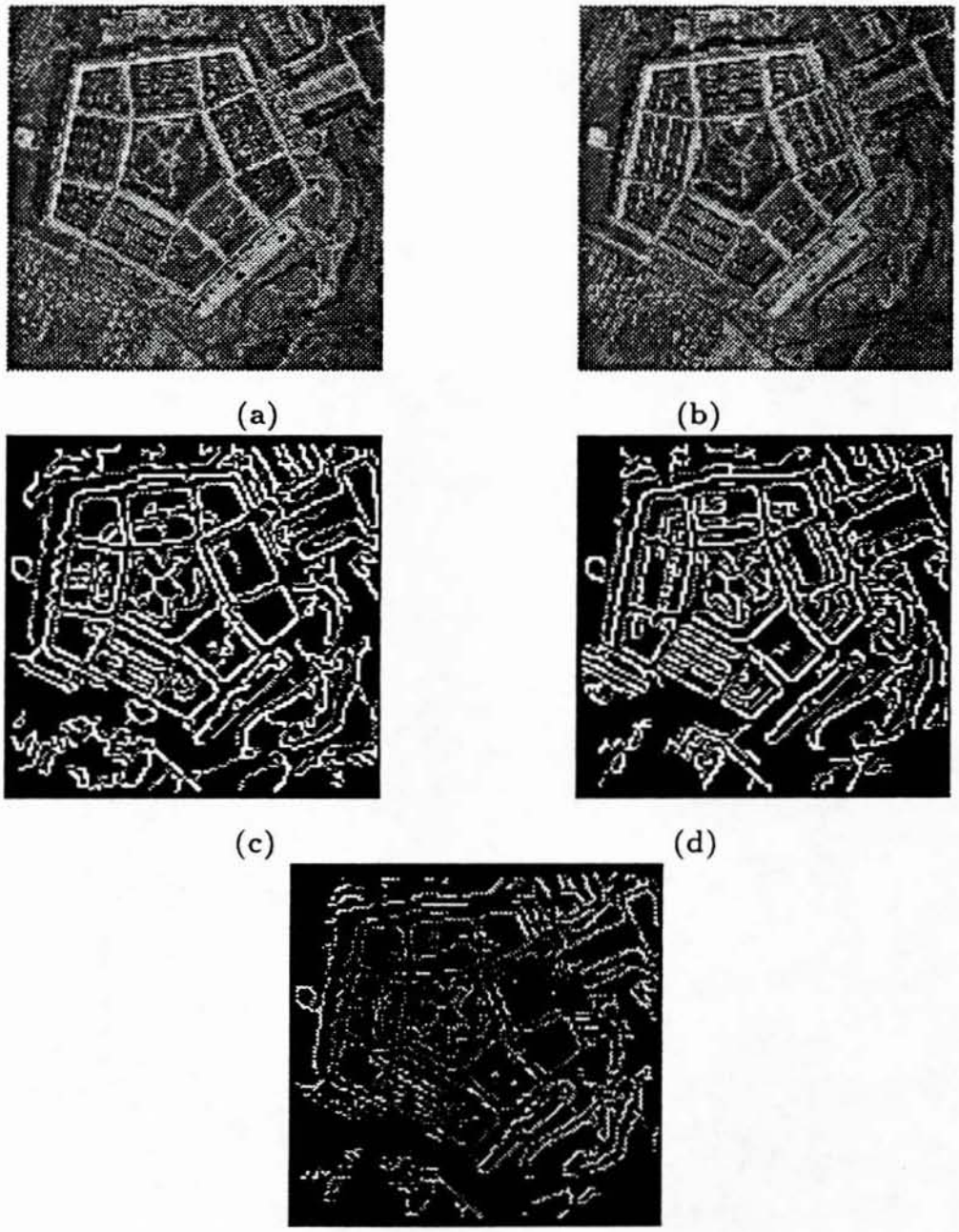


Figure 17: Result for Prazdny's Method for Penatgon pair. (a) Left Image, (b) Right Image, (c) Left Edge Map, (d) Right Edge Map. Positive edge are brighter than negative edges. (e) Disparity Map at edges only. Brighter edges are far from the viewer.

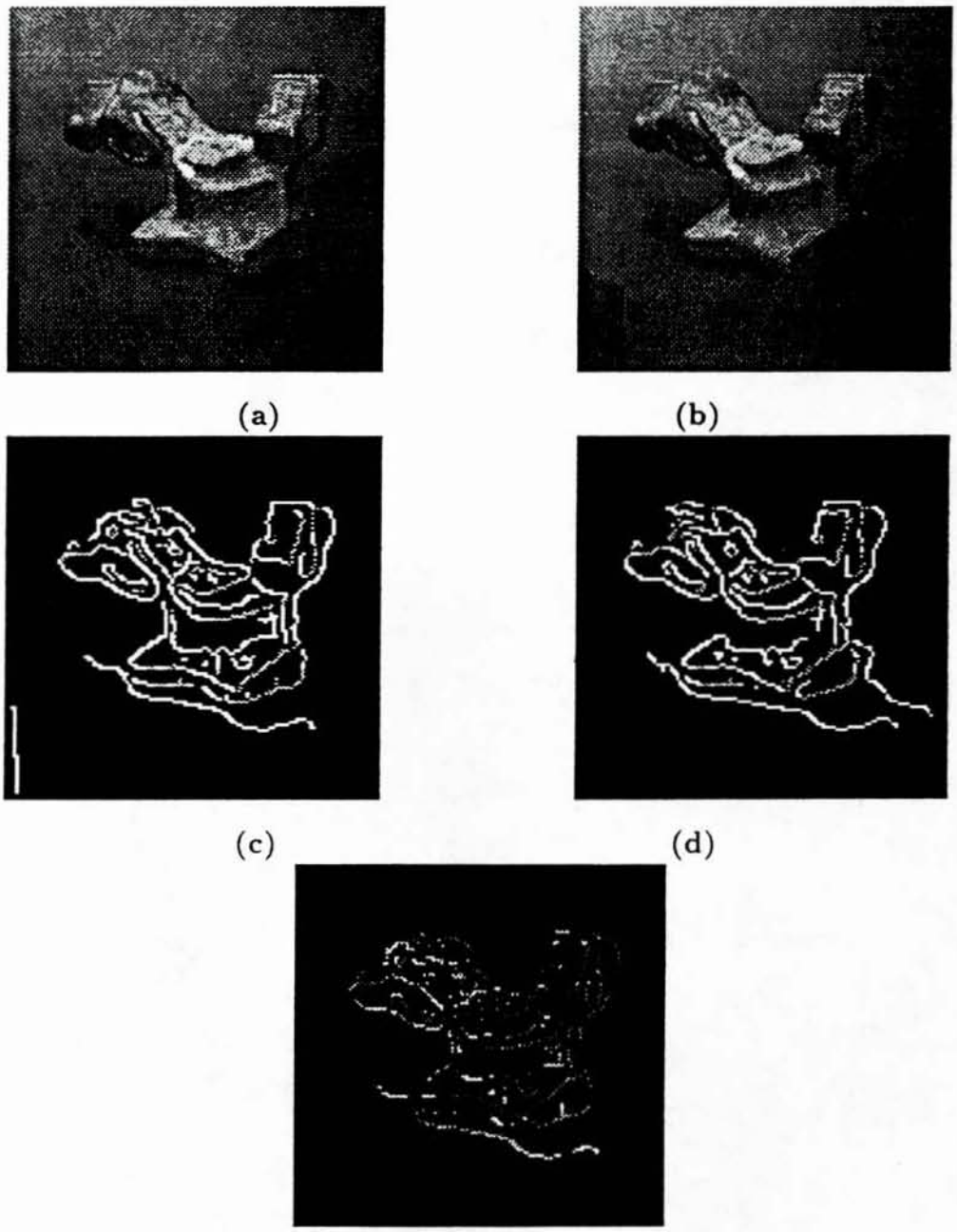


Figure 18: Result for Prazdny's Method for Renault pair. (a) Left Image, (b) Right Image, (c) Left Edge Map, (d) Right Edge Map. Positive edge are brighter than negative edges. (e) Disparity Map at edges only. Brighter edges are far from the viewer.

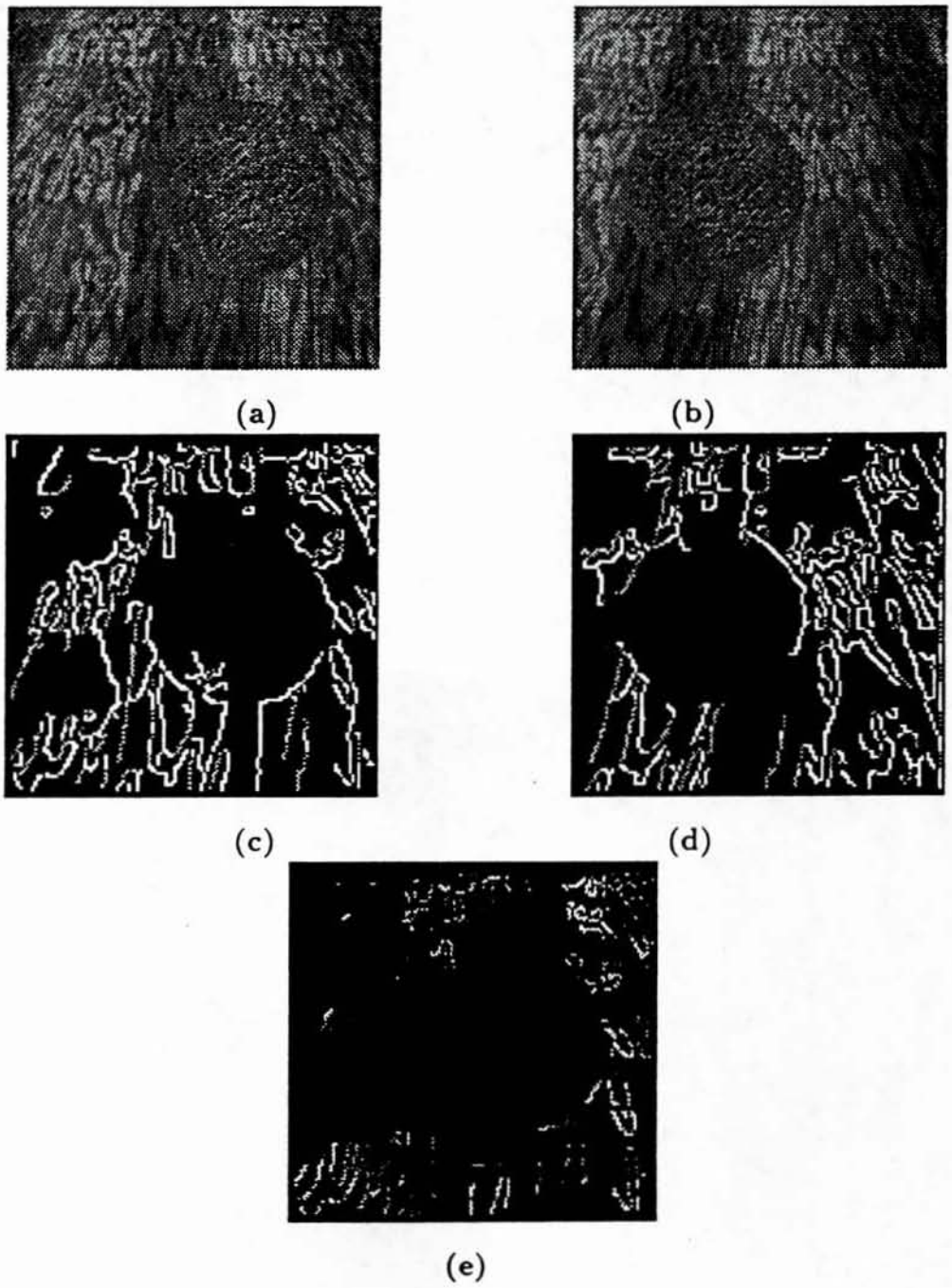


Figure 19: Result for Prazdny's Method for Sphere pair. (a) Left Image, (b) Right Image, (c) Left Edge Map, (d) Right Edge Map. Positive edge are brighter than negative edges. (e) Disparity Map at edges only. Brighter edges are far from the viewer.

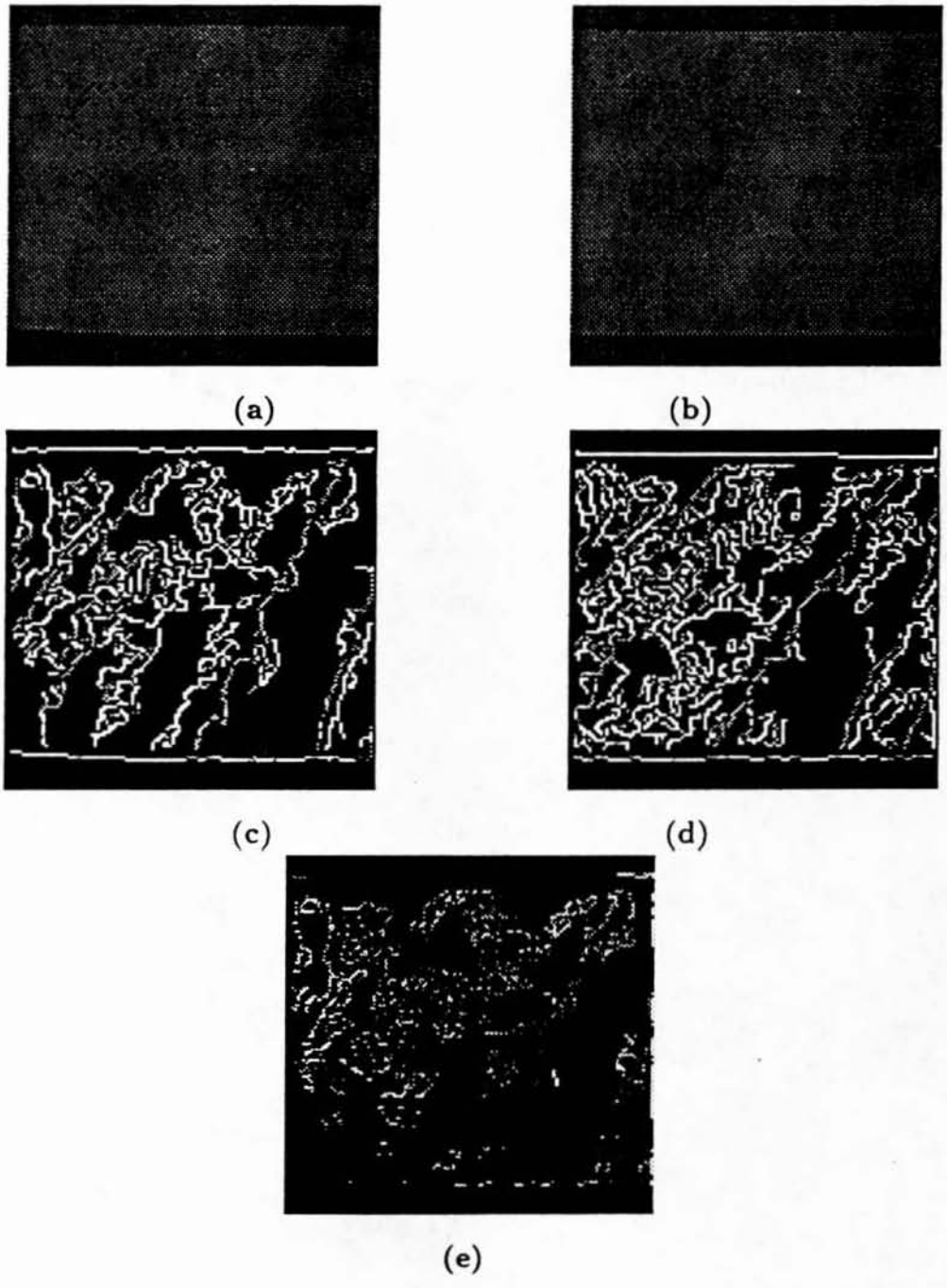


Figure 20: Result for Prazdny's Method for Ruts pair. (a) Left Image, (b) Right Image, (c) Left Edge Map, (d) Right Edge Map. Positive edge are brighter than negative edges. (e) Disparity Map at Edges only. Brighter edges are far from the viewer.

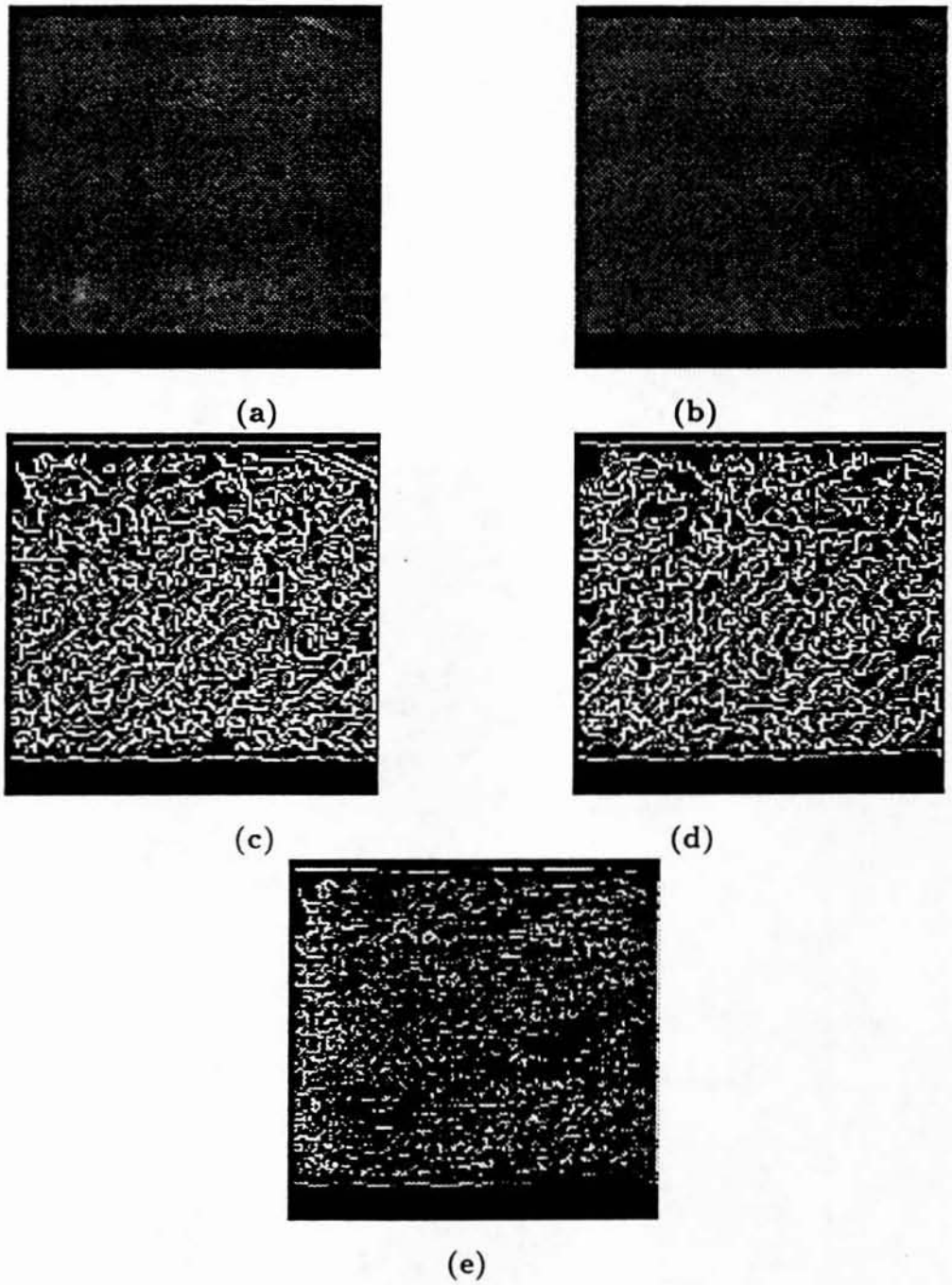


Figure 21: Result for Prazdny's Method for Rocks pair. (a) Left Image, (b) Right Image, (c) Left Edge Map, (d) Right Edge Map. Positive edge are brighter than negative edges. (e) Disparity Map at Edges only. Brighter edges are far from the viewer.

4 Summary and Conclusions

In this report we have described the implementation of three stereo algorithms. Two algorithms are correlation type, while the third one is edge based. The algorithms were also tested on a set of stereo pairs, and compared in terms of cpu time and the overall results. It was found that difference and normalized correlation methods give almost similar results, except that difference method is order of magnitude faster than the normalized correlation method.

In case of prazdny's method, which is edge based, the disparity values are computed only at edge points. Therefore, in order to obtain disparity values on other points the surface interpolation is necessary. The Difference and Normalized Correlation methods don't perform well for the textured images. However, Prazdny's method still give good result since it computes disparity at edge points only. One of the limitations of all these methods is that the maximum disparity range is to be known beforehand.

The execution time of the algorithms reported is reasonable for image size of 128×128. But, the algorithms will slow down significantly if the original size of images (512×512) is used. However, the execution time can be easily reduced to few seconds by using additional hardware boards e.g. Data Cube convolution boards for the Sun Workstation.

Our future work will focus on improving these algorithms by incorporating the smoothness and continuity criteria. We will also study some other 3D cues like *shading*, and attemp to combine these cues with stereo. It is expected that the performance of such method using multiple cues will be much better than the one using only single cue. We will also start an initial study related to segmentation of 3D surfaces followed by the interpolation step.

References

[1] K.Prazdny, *Detection of Binocular Disparity* , *Readings in Computer Vision*, 73-79, 1986.

[2] Steven D. Cochran and Gerard Medioni, *Accurate Surface Description from Binocular Stereo* , IEEE Workshop interpretation of 3-D Scenes, 16-23, 1989.

[3] Ali E. Kayalap, *High Speed Machine Perception Using Passive Sensing Technology*, SPIE Mobile Robots IV vol 1195. 33-43, 1989.