IMAGE UNDERSTANDING AND FEATURE EXTRACTION FOR APPLICATIONS IN INDUSTRY AND MAPPING

M.F.CALITZ

Presented to the University of Cape Town
in fulfilment of the requirements for
the Degree of Doctor of Philosophy

Cape Town
1995

DST 526 CALI
96/261

ACKNOWLEDGEMENTS

I am thankful to Professor Heinz Rüther of the Department of Surveying and Geodetic Engineering at the University of Cape Town, for providing me with the opportunity to undertake the research project and for acting as supervisor to this thesis. He also sacrificed his time to reading this thesis and providing invaluable advice.

The late Mr Nick Hanekom of Rembrandt inspired the research into industrial applications and deserves special recognition for his generous support of research. His sudden passing away before the completion of this project was strongly felt by all, for among other reasons, we had lost the enthusiastic optimism that he generated.

I am grateful to my colleagues, Sue Binedell, Dr Ryszard Florek, Kari Laatikainen, Val Atkinson, Sidney Smith, Greame van der Vlugt, Mark Matthews, Julian Smit and Nic van der Merwe, for their assistance and advice.

A special acknowledgement to Jacques Pienaar of Surveys and Land Information for initially drawing my attention to the advertisement for researchers in digital photogrammetry.

I would especially like to express gratitude and dedicate this work to my family, who have supported me through all my endeavours. My late father who always believed in the value of education beyond the materialism of this world. My mother whose positive encouragement was invaluable. My sister and her family for their support. A special dedication to my daughter.

January 1995 Michelangelo Franco Calitz

ABSTRACT

The aim of digital photogrammetry is the automated extraction and classification of the three dimensional information of a scene from a number of images. Existing photogrammetric systems are semi-automatic requiring manual editing and control, and have very limited domains of application so that image understanding capabilities are left to the user. Among the most important steps in a fully integrated system are the extraction of features suitable for matching, the establishment of the correspondence between matching points and object classification. The following study attempts to explore the applicability of pattern recognition concepts in conjunction with existing area-based methods, feature-based techniques and other approaches used in computer vision in order to increase the level of automation and as a general alternative and addition to existing methods.

As an illustration of the pattern recognition approach , examples of industrial applications are given. The underlying method is then extended to the identification of objects in aerial images of urban scenes and to the location of targets in close-range photogrammetric applications. Various moment-based techniques are considered as pattern classifiers including geometric invariant moments, Legendre moments, Zernike moments and pseudo-Zernike moments. Two-dimensional Fourier transforms are also considered as pattern classifiers. The suitability of these techniques is assessed. These are then applied as object locators and as feature extractors or interest operators. Additionally the use of fractal dimension to segment natural scenes for regional classification in order to limit the search space for particular objects is considered.

The pattern recognition techniques require considerable preprocessing of images. The various image processing techniques required are explained where needed.

Extracted feature points are matched using relaxation based

techniques in conjunction with area-based methods to obtain subpixel accuracy. A subpixel pattern recognition based method is also proposed and an investigation into improved area-based subpixel matching methods is undertaken. An algorithm for determining relative orientation parameters incorporating the epipolar line constraint is investigated and compared with a standard relative orientation algorithm.

In conclusion a basic system that can be automated based on some novel techniques in conjunction with existing methods is described and implemented in a mapping application. This system could be largely automated with suitably powerful computers.

CONTENTS

APPENDICES *Page*

1. INTRODUCTION AND SYNOPSIS OF THE INVESTIGATIONS

1.1 INTRODUCTION

Advances in digital imaging technology have recently enabled the development of semi-automatic photogrammetric systems. *Gruen, 1994,* gives an overview of the current state of the art. One desirable consequence of these developments is a completely "intelligent measurement robot" that upon some input instruction can derive measurements and interpret objects from two or more digital images.

Most semi-automatic photogrammetric systems are dedicated to specific applications such as the quality control of machined parts. Image understanding is the general term used for the activity that tries to relate classified objects to each other in a scene based on a priori knowledge of the context in which the image was acquired. Knowing where an image was acquired in itself provides a considerable amount of information about the likely interrelationships of the objects appearing in the scene.

An important component of any automated image understanding system is the classification of objects in the scene. This thesis investigates the use of statistical pattern recognition in conjunction with several other techniques for feature extraction and object classification. In addition image matching algorithms are investigated for establishing accurate correspondence between the extracted features. The various advantages and disadvantages of these methods are assessed. As an illustration of a system that may be automated, the digital elevation model of a preselected building in a stereopair of aerial images is determined.

A fully automated system for image analysis would consist of a number of the possible methods of feature extraction, classification and matching used cooperatively or selectively according to the domain of application. In addition to fulfil the

interpretive tasks of image understanding a knowledge database needs to be integrated into the system. Ultimately the system would act as an expert system which could be interrogated to obtain any desired information from presented images.

1.2 OUTLINE OF THESIS

In order to gain some perspective into the investigations undertaken the contents of the following chapters are briefly described.

Chapter 2 looks at the inherent problems encountered when attempting to analyse images. These problems being primarily due to the physics of the imaging environment and the nature of the imaging process. A theoretical discussion of the problem of extracting meaningful information from images is attempted to emphasize the limitations that exist. The problem of extracting three dimensional information from images and the necessary constraints that need to exist to make this possible are addressed.

Image matching is essential for obtaining three dimensional surface reconstructions. The various approaches to image matching are reviewed and the novel element of using statistical pattern recognition approaches in photogrammetry is introduced.

Chapter 3 has the overall goal of showing how a simple image understanding system can operate. Two pattern recognition quality control applications are described. The first application shows how a priori knowledge can be combined with extracted image primitives to obtain syntactic classifiers which are then interpreted or decoded according to a priori defined relationships. The actual problem investigated is the location and decoding of barcodes on boxes from images. A similar approach may be adopted to extract a relational descriptor of a building from an aerial image in terms of edge lengths and their mutual angles and to identify the particular building according to a

knowledge base of descriptors.

The second example is of model-based image classification. The problem is to locate and identify labels on cylinders. A knowledge base of label descriptors is built up and used to identify extracted descriptors in images. The statistical pattern recognition approach is adopted. This example illustrates in principle how descriptors can be extracted from a knowledge base and used for locating specific objects in a scene. A full three dimensional representation of an object, such as a building, may exist in a knowledge base. A descriptor can be extracted from a computer generated two dimensional view of the object and used for subsequent location and identification in a general image.

The specific applications that are described were solved using a newly developed combination of existing techniques and a priori information.

Having shown a simple application of an image understanding system **Chapter 4** considers using pattern vector descriptors as suitable features for the identification of specific objects or regions in a scene. This approach will enhance the ability of a general image understanding system to identify objects in 'noisy' images. In addition as part of a photogrammetric system extracted pattern vectors can be used in feature-based matching. The usual approach to feature-based matching concentrates on geometrical features consisting of points, lines, regions or relational descriptors consisting of a combination of these. By broadening the definition of feature one can integrate the relationships between the geometrical components of an object and represent these by components in a pattern vector. This latter approach appears to have been neglected in photogrammetric literature.

A novel adaptation of a pattern vector based object identification system is used to illustrate the approach in developing interest operators and as an object locator. The components of the pattern vectors are based on moments derived

from the image of an object in a scene and its edges. Geometric moment invariants are considered as suitable vector components. Other moment invariants are newly considered as descriptors and their suitability determined. Among the moments considered are Legendre moments, Zernike moments and Pseudo-Zernike moments which are orthogonal moments. Rotational and Complex moments are also investigated. The latter moments and geometric moments are not orthogonal.

A novel application of pattern vectors as interest operators is described. The method is illustrated by locating corners in images. Due to the sensitivity of this method to image noise an alternative new interest operator was developed based on the angles between neighbouring edgels or edge pixels. Extracting reliable interest points from images is essential to the automation of a surface reconstruction system. A relative orientation between two or more images can be established automatically once these interest points are matched. Surface reconstruction is then possible.

The above method of reconstructing surfaces does not result in any explicit identification of the objects in a scene. Integrating a knowledge base of classified pattern vectors or deriving pattern vectors from a object knowledge base or an image enables the pattern vector approach in conjunction with a priori constraints to identify objects in a scene.

Descriptors that can be derived from the spatial frequency information in an image are considered. A new two dimensional Fourier descriptor is described and its merits considered. As a further application of two-dimensional Fourier transforms fractal dimension segmentation of an image is investigated. This is particularly useful in an automated system for the identification of urban, forested and ground regions necessary in digital terrain model generation.

Having investigated how pattern vectors can be used to locate identifiable objects in images and relate these to existing

objects in a knowledge base, accurate image point matching is required to measure or reconstruct the objects in a scene. **Chapter 5** discusses accurate point matching techniques. Two approaches can be adopted:-

1. For a given point in one image an approximate match is obtained in another image and a subpixel matching algorithm applied to obtain accurate correspondence.

2. Points can be extracted from images to subpixel accuracy and an algorithm applied to establish correspondence between these extracted points.

Approach 1 is commonly adopted in most applications. Iteratively reweighted least squares(IRLS) matching is the generally accepted method for obtaining subpixel refinement. A novel application of a least absolute deviation(LAD) algorithm in image matching is investigated and compared with other related methods.

A novel nearest neighbour steepest descent search algorithm is proposed for using pattern vectors in obtaining matching to subpixel accuracy. Various versions of this method are compared with the IRLS and LAD methods.

Resulting from the pattern vector approach a new distance transform is derived. Applications of this transform are investigated.

Approach 2 mentioned above is also investigated by the development of a novel application of a connected components method to obtain subpixel locations of circular targets in images. A novel application of dynamic programming is then used to establish correspondence between the extracted points. Applications of this approach are discussed.

Chapter 6 considers continuous relaxation matching of points

extracted using an interest operator. Methods for obtaining the relative orientation between two images from the matched points is described. A novel method is suggested of integrating the epipolar line constraint in the relative orientation algorithm in an attempt to ensure compatibility between the matching and relative orientation. The conventional technique is described for obtaining the three dimensional reconstruction from the matched points once the relative orientation is known. In principle the contents of this chapter could be combined in a general automatic system for generation of digital elevation models of aerial images and close range images. No interpretive ability is included in the system in this form. Integrating a knowledge base and pattern recognition techniques can lead to object mensuration and identification.

Chapter 7 describes the application of the methods in **chapter 6** combined with the localization of a specific building in an aerial scene using the pattern vector approach. Once a digital elevation model of the region including the building is obtained, analysis of the three dimensional surface can determine the position of the building and a resulting digital terrain model obtained by removing the building from the scene. A strategy used to achieve this goal can be largely automated. The method entails a large amount of numeric processing requiring computational power in the region of several megaflops (million floating point operations per second). The relaxation matching algorithm is also suitable for parallel processing as is the pattern vector search technique. Array processor boards can greatly enhance the efficiency of these methods.

Chapter 8 outlines the main conclusions and results that have emerged from this investigation and makes recommendations as to possible areas of future research and development.

Appendices are attached that contain some of the core and generally useful subroutines used in the various computational investigations described.

Throughout this thesis the notation invoked in figures consisting of diagrams is specific to the context in which the diagrams are explained.

2. DIGITAL IMAGE MATCHING

2.1. INTRODUCTION

Matching corresponding points in a set of images of the same scene or object is central to obtaining a three dimensional reconstruction of the scene or object. Analytical photogrammetry requires a human operator to perform the matching on photographs. Available video and computer technology has made it possible to perform the matching on digitized images, currently in semi-automatic systems. In order to gain a perspective of the importance of digital image matching to modern developments in research and industry, a brief review of areas requiring matching follows.

Currently substantial research is being undertaken in the field of artificial intelligence with the objective of using computers to do interpretive tasks normally requiring a human operator. A wide variety of activities are being investigated, and with the integration of video technology and computer technology through real-time digitizing devices such as CCD(Charge Coupled Device) cameras, machine vision has become a realisable goal.

Most of the tasks that use machine vision are problem specific and only work in controlled environments or on problems with *a priori* constraints examples of which can be found in *[Gustafson, 1988] [El-Hakim, 1986] [van der Vlugt and Rüther, 1992] [Gruen and Baltsavias, 1985] [Streilen, 1992] [Loser and Luhmann, 1992] [Pattern Analysis and Machine Intelligence (PAMI) Vol. 10, No.1, January 1988]*. Automatic object reconstruction from images is one of the main areas of research in computer vision with current application in areas such as robotics, topographic map generation from aerial or satellite images, quality control of machined parts, medical diagnosis using Nuclear Magnetic Resonance(NMR) images. In order to obtain the three dimensional structure of an object with acceptable precision two or more images of an object are required. The images must be taken from different positions. There are methods

of obtaining the three dimensional shape of an object from a single image such as shape-from-shading *[Horn, 1975] [Horn and Brooks, 1986] [Szeliski, 1991]*, but these require some *a priori* information and are generally not sufficiently robust or accurate. Model based techniques can also be used on single images *[Gottschalk and Mudge, 1988] [Wolfe, Weber-Sklair, Mathis and Magee, 1988] [Pattern Analysis and Machine Intelligence (PAMI) Vol. 14, No.2, February 1992]*.These require well defined representations of the objects usually in terms of local features , at various possible viewing angles, to be maintained on a database. Single image techniques can be used to identify three dimensional objects in scenes.

Digital photogrammetry provides techniques for accurately measuring the dimensions of three dimensional objects or scenes from multiple images. *Gruen, 1994* , gives a review of current developments in digital photogrammetry. Fundamental to these methods is obtaining precise correspondence of matching points between images. A process usually referred to as *registering* the images.

2.2 PHOTOGRAMMETRY AND IMAGE MATCHING

The underlying concept applied in photogrammetry, which could be defined as the subject area concerned with measuring objects or scenes using photographs or images, is the same as that used to measure the distance of nearby stars in astronomy, viz. parallax. Figure 2.2.1 shows how the apparent position of a nearby star will shift relative to the distant stars when viewed from opposite points on the Earth's orbit. This shift is known as parallax. As the distance to the Sun is known, measuring the angular shift Θ enables one to determine the approximate distance to the nearby star as b/Θ (Θ is in radians and very small usually seconds of the arc). In order to determine the parallax two different views of the star is required. To estimate the distance to the star the relative positions of the viewpoints has to be known. Analogously two images of the same relatively near,

extended object or scene taken from two different known positions will enable the distance to various points on the object to be determined. Consequently the three dimensional structure of the visible parts of a scene or object can be calculated. The main difference between the two scenarios is that in the case of astronomy there is less of a problem locating the same star, which is close to being an ideal point, in each image. In the terrestrial case identifying the same object point in each image is difficult and is the subject of image matching. In images of extended objects regions or lines can also be used in matching to infer the location of points.

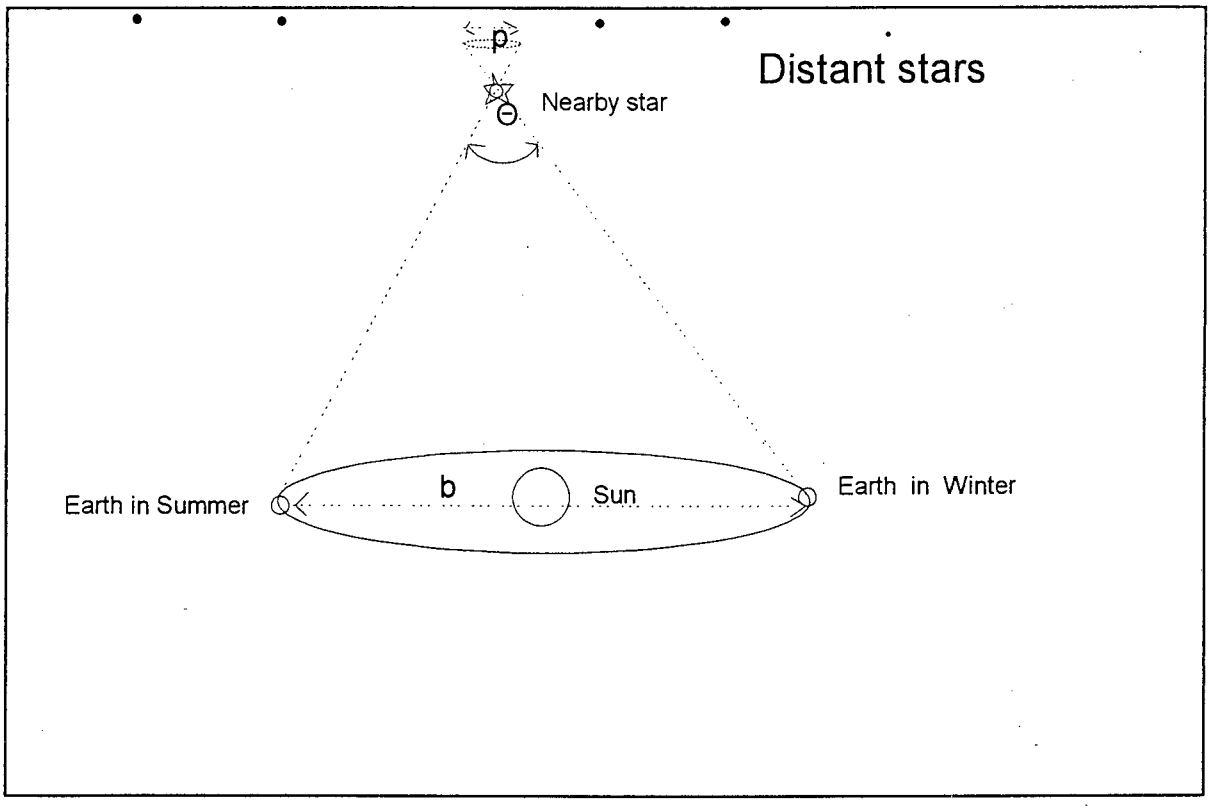


Figure 2.2.1 *Parallax in astronomy. p=parallax , b=baseline.*

2.3. STEREOPSIS AND MATCHING

In computer vision, the ability to determine depth of a point from two perspective image projections of the scene containing the object is known as stereopsis *[Haralick and Shapiro, 1993]*. As with the astronomical analogy the disparity or parallax between corresponding points in each image enables the depth of the scene to be inferred from the three-dimensional geometrical

relationships between the cameras and the objects. Figure 2.3.1 shows the simplified relationship between two one dimensional images and an object point in two dimensions enabling stereopsis. Object point P is captured on the images at points P' and P". The centres of projection are at O' and O" a distance b apart in the x direction of a cartesian coordinate system XYZ. The focal length of the camera is f. The parallax or disparity between the two image points is represented by d. (Image positives are assumed in the diagram.)

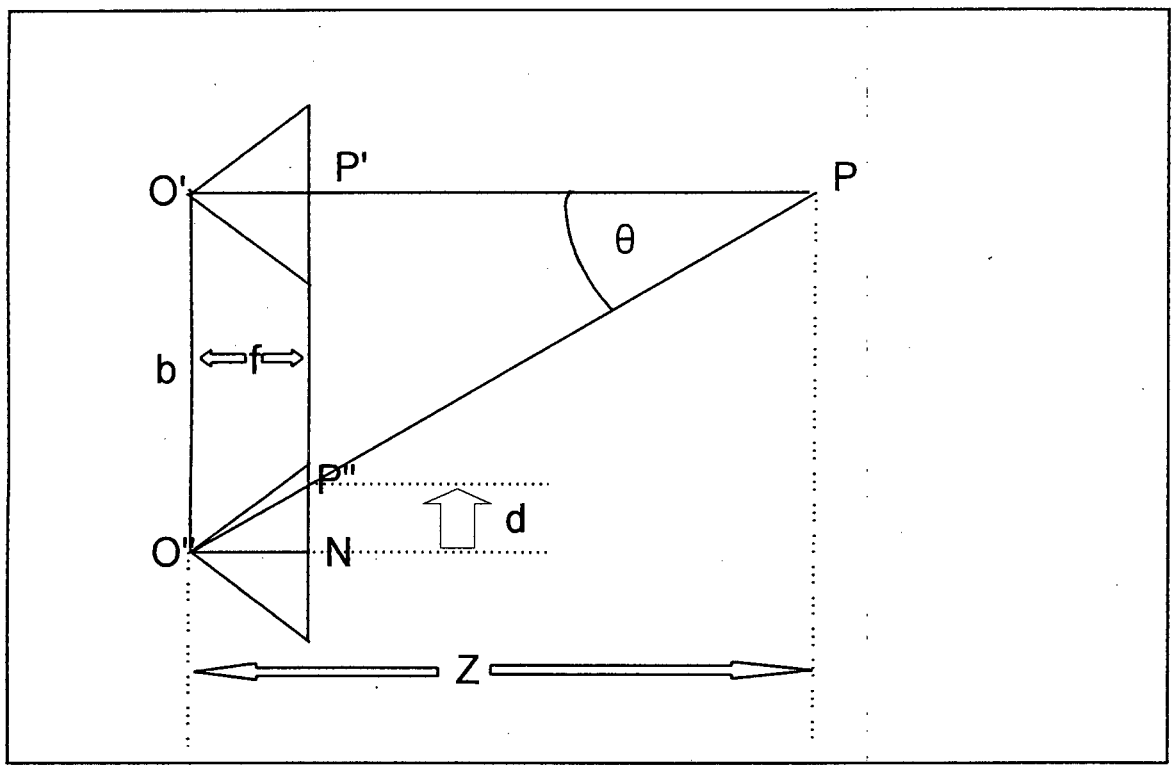


Figure 2.3.1 *Obtaining stereopsis in two dimensions.*

From the similar triangles ΔPP'P" and ΔO"P"N

$$\frac{d}{f} = \frac{b-d}{Z-f}$$

$$\Rightarrow \quad Z = \frac{bf}{d} \qquad \textbf{(2.3.1)}$$

Equation (2.3.1) shows how the depth coordinate z can be estimated from the disparity in two dimensions. The collinearity

equation given by equation (2.4.1) below expresses the geometrical relationship between the object and image in three dimensions, from which an expression relating disparity to the three dimensional object coordinates can be derived. These equations imply that if all the disparities of points between two images are available then, with the a priori knowledge of the camera positions and focal length, the three dimensional structure of the objects in the images can be reconstructed. This ideal is unattainable in practice as:

1. Only a relatively sparsely distributed number of points in the image are accurately recognizable and locatable.

2. Not all the points appear in both images.

3. Accurately determining the disparities requires matching the recognizable points to a high degree of accuracy which is not a completely solved problem

4. Errors introduced by the imaging systems in the form of noise and geometrical distortions limit the maximum attainable accuracy.

5. Interpolation of the structure or disparities between the matched points is an ill-posed inverse problem with no unique solution.

Although these seemingly insurmountable problems do exist in general, by significantly constraining the object reconstruction problem using a priori knowledge many very useful areas of application exist as mentioned in section 2.1 above.

2.4. IMAGE FORMATION

As indicated in section 2.3 the problem of digital image matching is further complicated by the physical process involved in image formation. The information or raw data in an image can

be regarded as a two dimensional grey-scale distribution function. The positions of the pixels and their intensity values are the data that have to be interpreted.

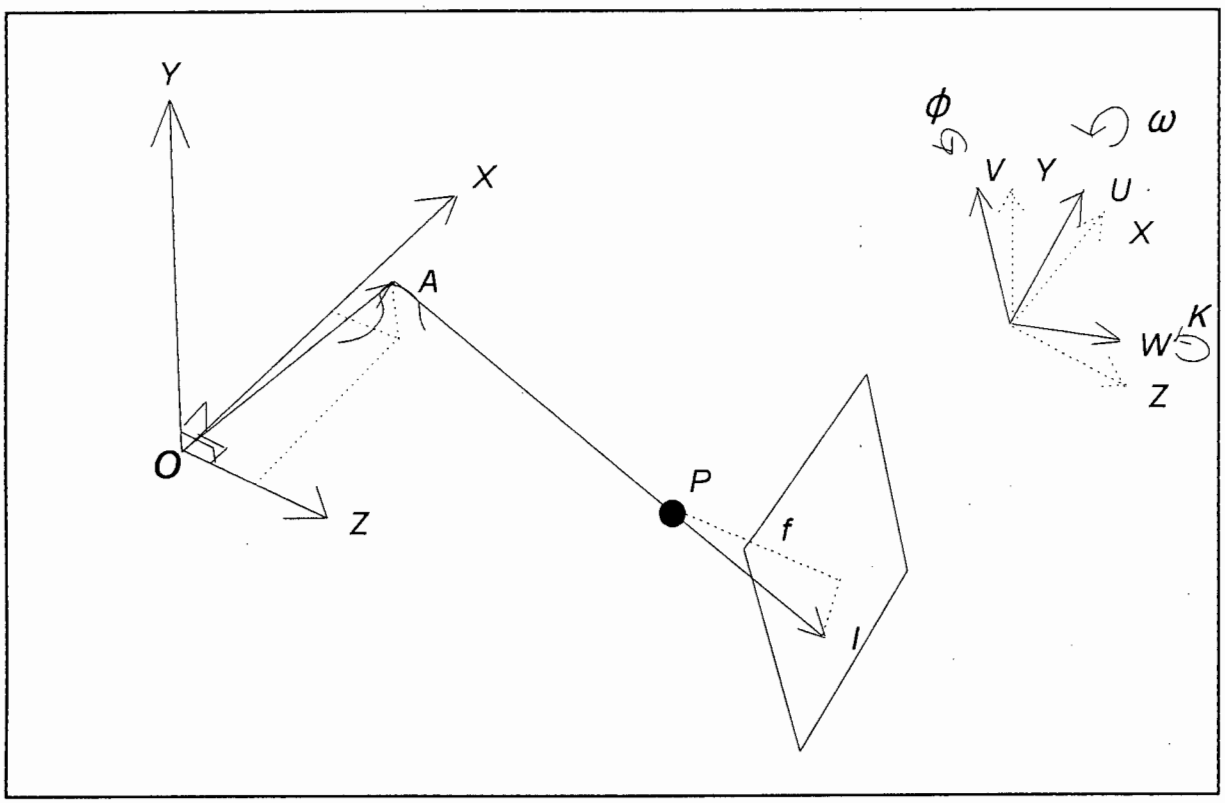


Figure 2.4.1. *Geometry of image formation of point A.*

Figure 2.4.1. illustrates the geometrical relationship between object and image space that exists during the imaging process of a pinhole camera. This model is adequate for most applications where the lens diameter is considerably less than the distance to the nearest object point. In very close range applications the paraxial optical assumption or first-order optics will break down *[Hecht and Zajac, 1974]*. Higher order geometrical aberrations will then need to be considered in the imaging process. Models allowing for these aberrations have been devised for close-range applications *[Brown, 1971 and 1976] [Beyer, 1992(a)]*.

The collinearity constraint equation expressing the relationship between image and object space is given by

$$\underline{U_I} = \underline{U_P} + R\,(\,\underline{X_A} - \underline{X_P}\,) \qquad \textbf{(2.4.1)}$$

where

$\underline{U_I} = (\,u\,,\,v\,,\,w\,)$ are the coordinates of the image point in the image space defined by the U, V, W axes.

$\underline{U_P} = (u_p, v_p, f)$ are coordinates of the projection centre of the camera on the image.

$\underline{X_A} = (\,x_A, y_A, z_A)$ are the coordinates of the object in the object space coordinate system X, Y, Z.

$\underline{X_P} = (x_P, y_P, z_P)$ are the coordinates of the projection centre of the camera in object space.

R is the rotation matrix with elements that are determined by the rotation angles ω , ϕ , κ . The elements are given in chapter 6.

The collinearity equation gives the coordinates of the image in the ideal situation of a pinhole camera. With real image acquisition systems geometrical distortions are introduced due to the optical system, scale differences in the x and y directions, the nature of the sensor and the digitization process. These distortions are modelled by adjustments to the image coordinates. Basic models integrate the effects of these distortions into radial symmetric lens distortion parameters and decentering parameters. More complex models have been developed

and are fully described in *Brown, 1971 and 1976, Beyer, 1992(a) and 1992(b)*.

Although the geometric position of a point in an image is well described by the above, the intensity value at that point is more difficult to model. This is mainly due to factors that are functions of the environment and virtually uncontrollable excepting in a laboratory setting where some control can be attained.

The intensity distribution function of an image is due to the radiant energy captured by a lens and focused onto a photochemical or photoelectric sensor. The radiant energy has a spectral distribution which is affected by the initial source of light; the light scattering characteristics of the object; the absorption and transmission characteristics of the medium between the light source and object and between the object and lens; the aberrations and transmission properties of the lens system. The image is further modified by the spectral characteristics of the sensors and the electronic characteristics of the digitizing process. The discretization of the image via a scanner or CCD device leads to further information loss due to the integration of the intensity distribution over the area of a pixel unit.

The electronic and spectral characteristics of the image acquisition system can be determined empirically *[Beyer, 1992(b)]*. Modelling the radiant energy that actually reaches the lens is more difficult and requires *a priori* assumptions about the nature of the object surface. The reflecting properties of the objects in a scene will determine the radiant energy emitted in the direction of the camera. These properties have been modelled by means of a reflectance function. An explanation of the three most accepted models viz. the Lambertian model; the Torrance-Sparrow model and the Beckman-Spizzichino model is given in *Nayar S.K., Ikeuchi K, Kanade T, 1991*.

As a consequence of the above it can be deduced that image formation is an extremely complex process. An image can be regarded as a sensorially filtered representation of reflected light of a scene obtained from a specific viewpoint. In the image formation process an enormous amount of information about the scene is lost due to the filtering and noise characteristics of sensors and most significantly due to the projection of a three dimensional space onto two dimensions at a reduced size. When viewing the image human intelligence can regenerate much of the lost information and can recognize and interpret objects and their relationships. This is primarily as a result of learned experience. The exact mechanism by which the human mind can perform these tasks effortlessly is the subject of on going research. The methods used in digital pattern recognition and image matching are aimed at emulating this ability of the human intelligence and therefore falls within the ambit of artificial intelligence.

2.5. GENERAL OBJECT RECONSTRUCTION PROBLEM

The final objective of digital photogrammetry is the complete three dimensional description of the objects surface in a scene. Some of the problems encountered in reaching this goal have been mentioned above. In order to gain some insight into the problem an abstract formulation of the process is considered following the approach of *Zheng, 1993*.

Let X be the set of object model parameters and let Y be the set of observed image parameters. The imaging process can be regarded as a mapping function f : X -> Y . The forward problem is the prediction of the Y parameters from the X parameters under a mapping f. The inverse problem is the recovering of the parameters in X from the observed parameters Y. The inverse problem is the problem that needs to be solved in object reconstruction from images and its solvability is dependent on the nature of f.

(a) f is called surjective if for every Y there exists at least one X. This implies that there are one or more X that can result in Y when the mapping f is applied, and ensures the existence of at least one element $X = f^{-1}(Y)$, where f^{-1} is the inverse mapping function.

(b) f is injective if for every Y the inverse mapping $f^{-1}(Y)$ is empty or a singleton. This implies a one-to-one mapping of X onto Y, but does not ensure the existence of an X for every Y. If the inverse image exists it is unique.

(c) f is bijective if it is both surjective and injective. This implies that the inverse mapping $X = f^{-1}(Y)$ exists and is unique.

If © is true then a unique solution is determinable. If (b) is true the system is overdetermined. An example of this situation would be when a parametric model of f is applied to X to give an image Y which is close to but not equal to the actual observed image Y. If (a) is true the system is underdetermined as there are possibly more than one X that can give rise to a particular image Y. An example would be images of reduced scale special effects models used in cinematography to recreate images of real full scale objects.

Generally only a subset of the parameters X are desired viz. object coordinates. The other parameters are largely indeterminate such as luminance, reflectance , atmospheric conditions affecting the light transmission medium, and the factors outlined in 2.4 that contribute to image formation. In general therefore there are many possible combinations of the parameters in X that could give an observed image Y. The general inverse problem of object reconstruction is governed by a surjective mapping, and is consequently an ill-posed inverse problem. Ill-posed in the sense that it is an underdetermined problem.

By constraining the imaging environment many of the

indeterminate parameters in X can be held constant and the problem can be converted into an injective problem which is well-posed. An example is the application of the geometrical constraints of the imaging environment e.g. the collinearity condition and the coplanarity of the image plane to images of the same scene separated in time and distance by a small interval, enabling the recovery of some of the three dimensional information of a scene. Consequently existing digital photogrammetric systems are applied in special applications with constrained environments. The environments of close-range photogrammetric systems are more easily constrained by specially designed backgrounds, lighting , pattern projections and object targeting.

Assuming that the imaging environment is suitably constrained the main steps in object reconstruction that were proposed by *Marr and Poggio, 1979*, based on their computational model of human stereo vision are shown in figure 2.5.1. *[Grimson, 1981]*

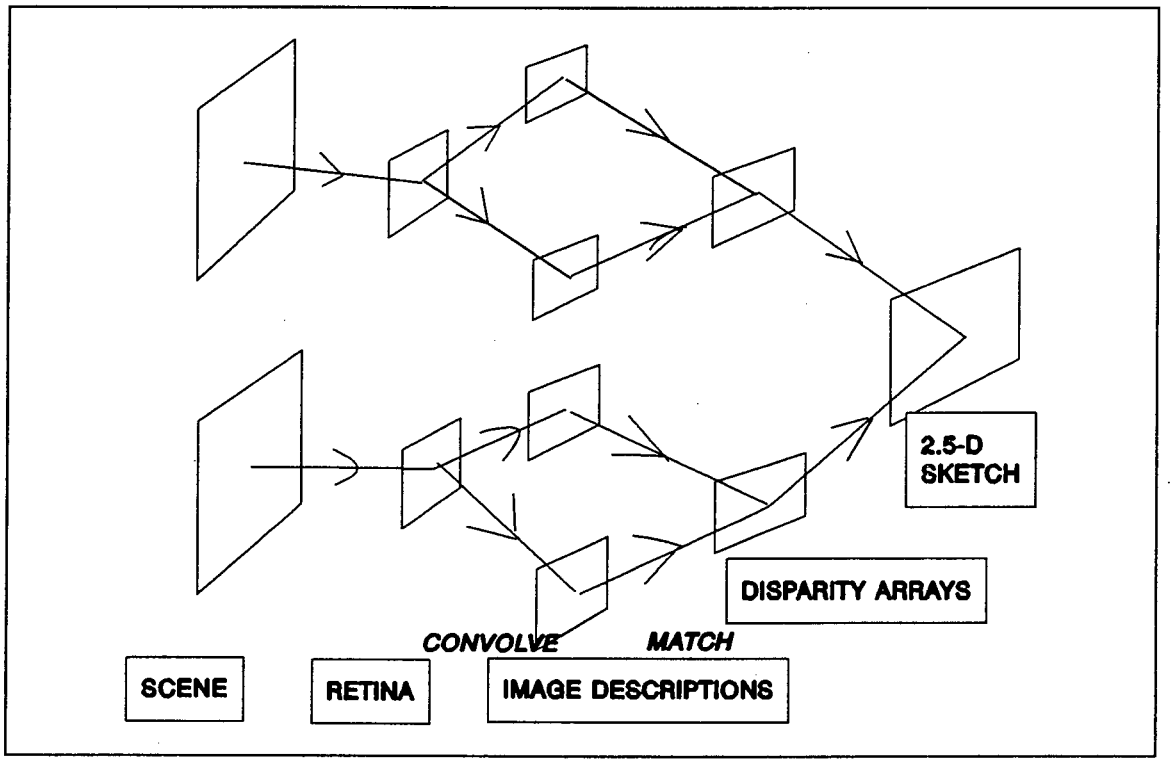


Figure 2.5.1. *Steps towards object reconstruction based on the computational model of human stereo vision proposed by Marr and Poggio.*

The scene viewed is projected onto the retinas of both eyes. The retina in conjunction with the brain effectively convolves the image into zero-crossing image descriptions of the scene at various resolution channels, two of which are represented in the diagram. The theory contends that most of the information of a scene is gained from the points with sharp gradients. *Marr and Poggio, 1979*, postulated the Laplacian of the Gaussian gradient operator to simulate the process of visual convolution of the human brain given by

$$\nabla^2 G(r,\theta) = \left(\frac{r^2-2\sigma^2}{\sigma^4}\right) e^{-r^2/2\sigma^2} \qquad \textbf{(2.5.1)}$$

where

$G(r,\theta)$ is the rotationally symmetric Gaussian function with r the radial coordinate and θ the angular coordinate.

σ is the parameter determining the width of the Gaussian function and acts as a resolution filter. The larger σ the lower the resolution of the resultant convolved image.

The convolved images are termed the primal sketches of the scene. Correspondence between matching points is then established by the brain. The exact basis of this is unknown. Disparity arrays or maps are produced to form a raw 2.5-D sketch of the scene. Some interpolation technique is then employed by the human intelligence to produce a complete 2.5-D sketch of the scene. The terminology 2.5-D is used because an accurate full 3-D reconstruction of the scene is impossible even for the human mind as there is inevitably loss of information

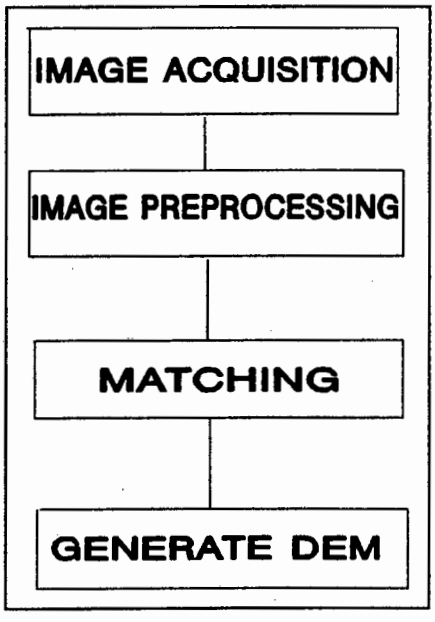


Figure 2.5.2. *Basic modules required for computational object reconstruction.*

in the visual process.

A generalization of this model to any visual system results in analogous processes for object reconstruction as shown in figure 2.5.2. Acquisition of a stereo pair of images is assumed.
The digital image is acquired directly using CCD (Charge-Coupled Device) cameras and digital frame grabbers or indirectly by digital scanning of conventional photographs.

Next the images are processed and matching primitives are extracted from the image. The methods used here are system dependent.

In the following step the matching primitives are matched using correspondence or matching algorithms.

After matching the three dimensional coordinates of the matched points are determined with a priori knowledge of the position and orientation of the cameras. A complete DEM (Digital Elevation Model) can then be determined by suitable interpolation between the calculated coordinates of the matched points. The interpolation problem is ill-posed as there are many possible surfaces that can pass through a finite number of points in three dimensions. The surface interpolation problem is discussed in detail by *Grimson, 1982* and *Terzopoulos, 1988*.

2.6. MATCHING

As indicated by the above discussion the accurate matching of corresponding points is an essential requirement for object reconstruction. Various different approaches are used for matching. The method chosen is problem-specific.

Digital image matching can be divided into two major operations.

1. Extraction of matching primitives in each image.

2. Establishing correspondence of the matching primitives between each image.

The two stages are interdependent with specific matching algorithms being more suitable for particular matching primitives. In practical systems different matching techniques can be used cooperatively to form an overall matching strategy.

In its most general form matching has to establish a one-to-one mapping from an item in one list with an item in another list. Each list consists of matching primitives associated with a particular point location in an image.

The most primitive matching algorithm would be to exhaustively inspect every possible configuration of the items in both lists. If there are N primitives in each list then there are N! possible configurations. Figure 2.6.1 shows the possible matched configurations for N=3. The best match is then chosen based on some decision function. The decision function will depend on the matching primitives but generally needs to assess the similarity between matching primitives and the consistency of the matched configuration. Similarity is taken into account giving more weight to matching primitives that are very similar than to those that are not. Consistency is accounted for by considering a relationship between a neighbourhood of the points associated with the matching primitives, such as the variance of the matched point disparities of these points. The smaller this variance the more weight can be attached to the configuration. The details of the weighting and measures used in the decision function will be problem specific.

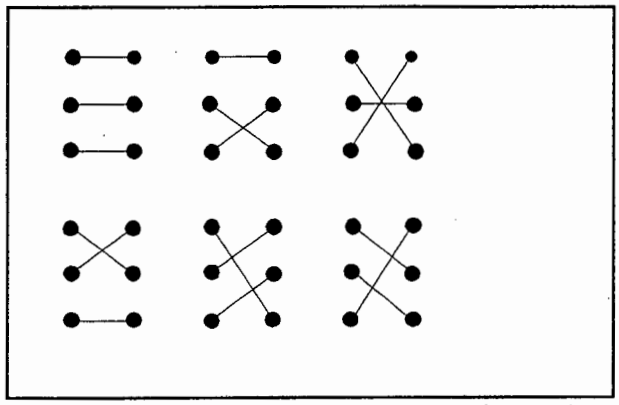


Figure 2.6.1. *Possible matching configurations between two lists with 3 elements each.*

The primitive matching algorithm described is generally impractical as the number of possible matching configurations becomes exponentially large as N increases. For N=100 the number of possible matched configurations is of the order of 10^{158}.

More formally matching can be regarded as a consistent-labelling problem (CLP) *[Haralick and Kartus, 1978, Haralick and Shapiro,1979 and 1980]*. A consistent-labelling problem is defined as

$$CLP = (U , L , T , R) \qquad \textbf{(2.6.1)}$$

where

$U = \{ u_1 , u_2 , u_3 \ldots u_M \}$ a set of M units.

$L = \{ l_1 , l_2 , l_3 \ldots l_M \}$ a set of M possible labels.

U can be regarded as the set of matching primitives from one image and L those from another image.

T is a unit constraint relation such that if an N-tuple $\{ u_1, u_2 \ldots u_N \}$ belongs to T, then the units $u_1 , u_2 , \ldots u_N$ mutually constrain one another. An example would be the set of points belonging to an extracted edge in an image.

R is the unit-label constraint relation and has elements equal to the allowable labelings of ordered size-N subsets of U. For example if the ordered size-N subset $\{ (u_1,l_1) , (u_2,l_2) , \ldots (u_N,l_N) \}$ belongs to R then the units $u_1, u_2, \ldots u_N$ can be assigned the corresponding labels $l_1, l_2, \ldots l_N$. An example would be the relationship between two matching edges, where u_i would come from one edge and l_i from the corresponding edge.

In terms of this definition matching can be defined as a CLP where the set of all consistent labelings f: U -> L satisfying the constraints specified by T and R has to be found.

Haralick and Shapiro, 1979 and 1980, have investigated various tree search techniques of solving this abstract formulation of the matching problem.

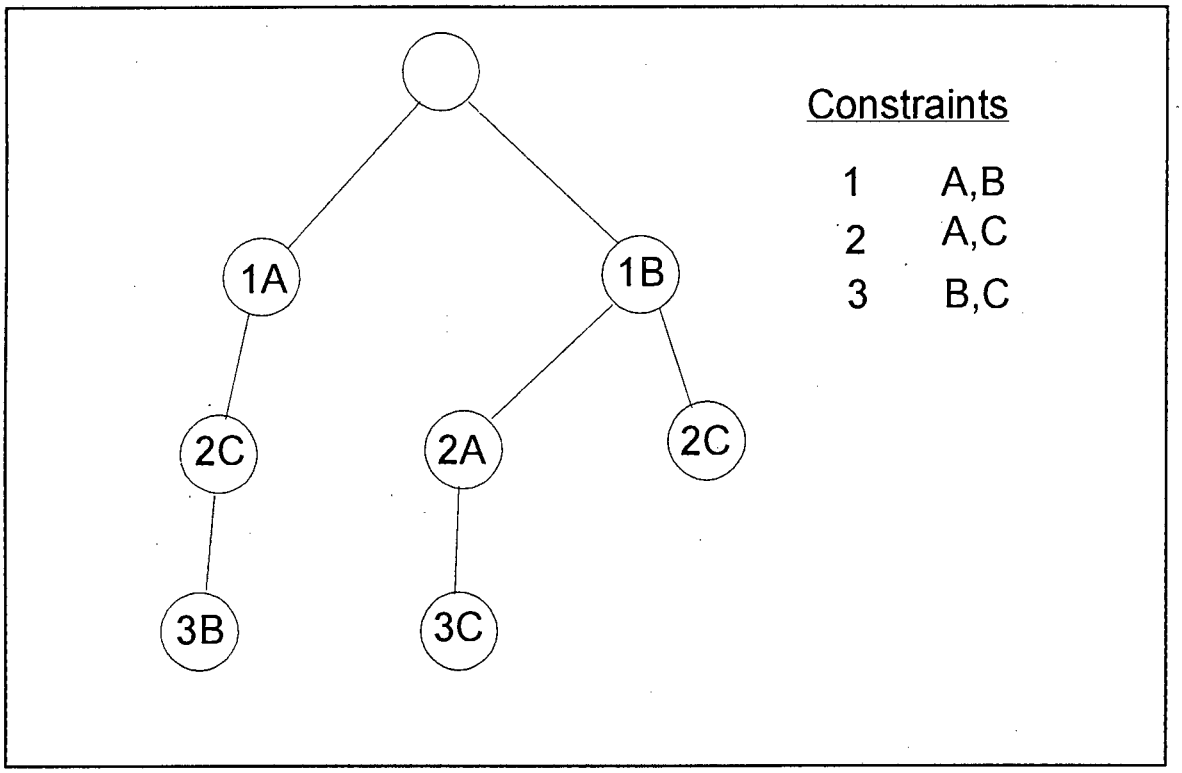


Figure 2.6.2. *Tree search for labelling of 3 units subject to the given constraints.*

Figure 2.6.2 shows a simple tree search where U={1 ,2 ,3} , L={A ,B ,C} and T and R are the constraints listed. Two possible matches are found { 1A, 2C, 3B} and { 1B, 2A, 3C}. The constraints have limited the 3^3 possible branches to 7. This illustrates the importance of constraints in making matching problems tractable. In digital image matching applications the use of geometric constraints, in addition to similarity and consistency constraints, is made to obtain solutions relatively rapidly.

Having established the general nature of the image matching problem a few of the more commonly used methods are briefly described. There are no definite boundaries when classifying matching algorithms as they are interrelated and used cooperatively in most practical digital matching applications. Table 2.6.1 lists the methods that are discussed. Epipolar

matching is included although this is really a matching strategy that can be used in conjunction with any of the matching techniques listed. There are a number of review articles which cover most of the commonly used matching techniques, some of which are *Dhond and Aggarwal, 1989* , *Lemmens, 1988, Pilgrim, 1992*, and a review of their implementation in practical systems is given in *Gruen, 1994*.

MATCHING ALGORITHM	MATCHING PRIMITIVES
Correlation	Regions of pixel intensity. Numerical regional features.
Relaxation	Extracted point features or descriptors.
Heirarchical or Multiresolution	Pixel intensity or extracted features.
Relational	Extracted feature descriptors.
Dynamic Programing	Extracted point features or descriptors.
Model Based	Extracted features from generated viewclasses.
Integrated Matching	Regions of pixel intensity.
Epipolar Matching (A matching strategy)	All types.

Table 2.6.1. *List of some commonly used matching algorithms.*

A. CORRELATION MATCHING (AREA-BASED MATCHING)

The matching primitive is the grey-scale intensity values of an image window. For any particular window in the left image, of a stereopair, the right image is searched using a equally sized

window ,and the similarity between the windows is measured using either a cross-correlation function or an intensity difference function. A window is accepted as a matching window when either the correlation function is greater than a predetermined threshold value or the distance function is less than a similarly determined threshold value. Geometric and consistency constraints are applied to disambiguate multiple matching points. *[Helava, 1978]*

Once a matching window is found a parametric affine mapping is assumed between the windows. Using the method of least-squares or any estimator technique a subpixel match can be obtained while solving for the affine parameters. *[Förstner, 1982] [Ackermann, 1984] [Gruen, 1985]*

B. RELAXATION MATCHING

Relaxation matching is an extension of the tree-searching algorithms mentioned above. No parametric mapping function is assumed. Matching primitives are extracted from images using interest operators , edge detectors or feature extractors. The matching primitives may be points *[Förstner and Gülch, 1987] [Barnard and Thompson, 1980]* or more complex symbolic structures *[Faugeras and Price, 1981] [Bhanu and Faugeras, 1984].*

Discrete relaxation is an extension of the forward checking tree search algorithm. The algorithm checks for the compatibility of future label assignments ,according to the constraints of the problem and the instantiated matches. The process is iterative because for every newly instantiated match future possible matches have to be re-evaluated and discarded if incompatible. The process terminates when there are no further changes to the established matches. A clear illustration of the process is given by *Waltz, 1975,* when applied to labelling of lines connecting vertices on polyhedral objects. A similar application is given by *Rosenfeld, Hummel and Zucker, 1976. Medioni and Nevatia, 1984,* apply the method to matching constructed graph descriptors in

aerial images.

Continuous relaxation or stochastic labelling assigns initial probabilities to possible label assignments based on some measure of similarity. These probabilities are iteratively updated by a local compatibility checking procedure . A label assignment is accepted as a match when the probability exceeds a threshold value. *[Barnard and Thompson, 1980] [Bhanu and Faugeras, 1984].*

Relaxation matching is very flexible in that a variety of similarity measures and consistency constraints can be incorporated into the problem without significantly changing the general structure of the algorithm.

C.HIERARCHICAL MATCHING

Matching can be guided by applying a matching algorithm on low resolution versions of the stereopair images. The results of this matching are then used to locate approximate initial matching points in the next higher resolution images and the matching algorithm is again applied. This process continues through to the maximum resolution images *[Ackermann, 1984]. Baltsavias, 1991,* gives a detailed review of this method.

The multiresolution images are generated differently according to the matching primitives used. For features such as edges different resolution edges can be obtained applying the Laplacian of Gaussian edge detector and altering the value of σ in equation (2.5.1). For grey-scale correlation techniques coarser resolution images are created by averaging pixel values over the integrated neighbourhood. Wavelet decomposition can also be used to generate multiresolution images. *[Zhou and Dorrer, 1994] [Mallat, 1989]*

D. RELATIONAL MATCHING

The matching primitives used are usually structural

descriptors of features such as edges and their neighbourhood relationships. The matching primitives can be represented in a graph-based terminology*[Haralick and Shapiro, 1993] [Hellwich and Faig, 1994]* and the matching is performed using a tree search technique subject to predefined constraints, measures of similarity and measures of consistency.

Relational matching is strongly dependent on the extraction of nearly noise free features and is difficult to implement on real images without additional constraints or a priori heuristics.

E. DYNAMIC PROGRAMMING MATCHING

The matching primitives used are usually edges*[Benard, 1984] [Ohta and Kanade, 1985]*. The images are photogrammetrically normalized (which is explained below) to reduce the image search space to one dimension. For each edgel in the left image the corresponding line in the right image is scanned for possible matching edges subject to disparity constraints. A cost is evaluated for each possible match based on some similarity and consistency criterion. A tree-like graph is generated with branches linking possible matching pairs. The path linking the branches with total minimum cost gives the optimal match.

Dynamic programming is flexible with regard to the definition of the cost function and the method can be advantageously employed in relational matching.

F. MODEL-BASED MATCHING

Locating an object in an image which is represented by a full digital three dimensional model description is primarily used in pattern recognition applications. Two dimensional viewclasses of the model can be generated and a matching algorithm used to find the matching structures in the image. Relational tree search algorithms are often used *[Haralick and Shapiro, 1993]*. One could

extend this method to the localization of matching objects in a stereopair.

The computational demands of this method have led researchers to recast the algorithm as a neural network for parallel computation *[Nasrabadi, Li and Choo, 1990]*.

An alternative object matching algorithm based on a concept similar to the Hough transform is affine-invariant matching or geometric hashing*[Lamdan, Schwartz and Wolfson, 1988]*, but this method is extremely sensitive to image noise.*[Grimson and Huttenlocher, 1990]* A similar method based on clustering in a parametrized image mapping parameter space was applied to stereopair satellite images.*[Stockman, Kopstein and Benett, 1982]*

An industrial application of the model based viewclass approach to pattern recognition is described in chapter 3.

G. INTEGRATED MATCHING

A method which uses least squares estimation in object space to integrate image matching and object surface reconstruction has been under investigation by a number of researchers .*[Ebner and Heipke, 1988] [Helava, 1988] [Wrobel, 1987]* The object surface is divided into a grid composed of facets called groundels. A reflectance model and the collinearity equations are used to model groundel intensities from the pixel data in the images. A least squares estimator is then used to solve for the unknowns in the model.

This method requires a large amount of computation and is usually applied in an hierarchical scheme. Considerable improvement in the modelling developed to date is required to allow for the large radiometric and geometric variations in real images.

H. COMBINED METHODS

Correlation matching and integrated matching can be regarded as area-based matching techniques as they are based on the regional pixel intensity values. Excepting for hierarchical matching the rest of the methods depend on the extraction of features by the use of edge-detectors, interest operators and other image processing techniques. Hierarchical matching is a matching strategy that can be used with both area-based and feature-based methods. In most digital matching applications different combinations of the above methods are used cooperatively.

In most systems feature-based methods are usually used to obtain the initial matching points and a least-squares area-based technique is used to refine the match to subpixel accuracy. Geometrical constraints are used to limit the search space. *Gruen , 1994*, lists many applications used for close-range photogrammetry. *Li and Schenk, 1990*, describe a system for aerial image matching. *Greenfeld, 1991*, proposes an automatic system using the different matching techniques where appropriate. *Van der Vlugt and Rüther, 1994*, describe a close range application using combined methods.

I. EPIPOLAR LINE MATCHING

Figure 2.6.3 shows the geometry of a stereopair of images I' and I". The perspective centres of the cameras are located at O' and O''. H' and H" are the principal points in the images and act as the origins of the respective image coordinate systems (u',v') and (u",v"). The object point P projects onto the images I' and I" at P' and P" respectively. The points F' and F" are the points where the baseline b intersects the image planes I' and I" respectively , and are known as the epipoles. The triangles ΔPO'O", ΔP'O'F' and ΔP"O"F" lie in the same plane. The intersection of this plane on the images I', I" define the epipolar lines in the images. This implies that for a point P'

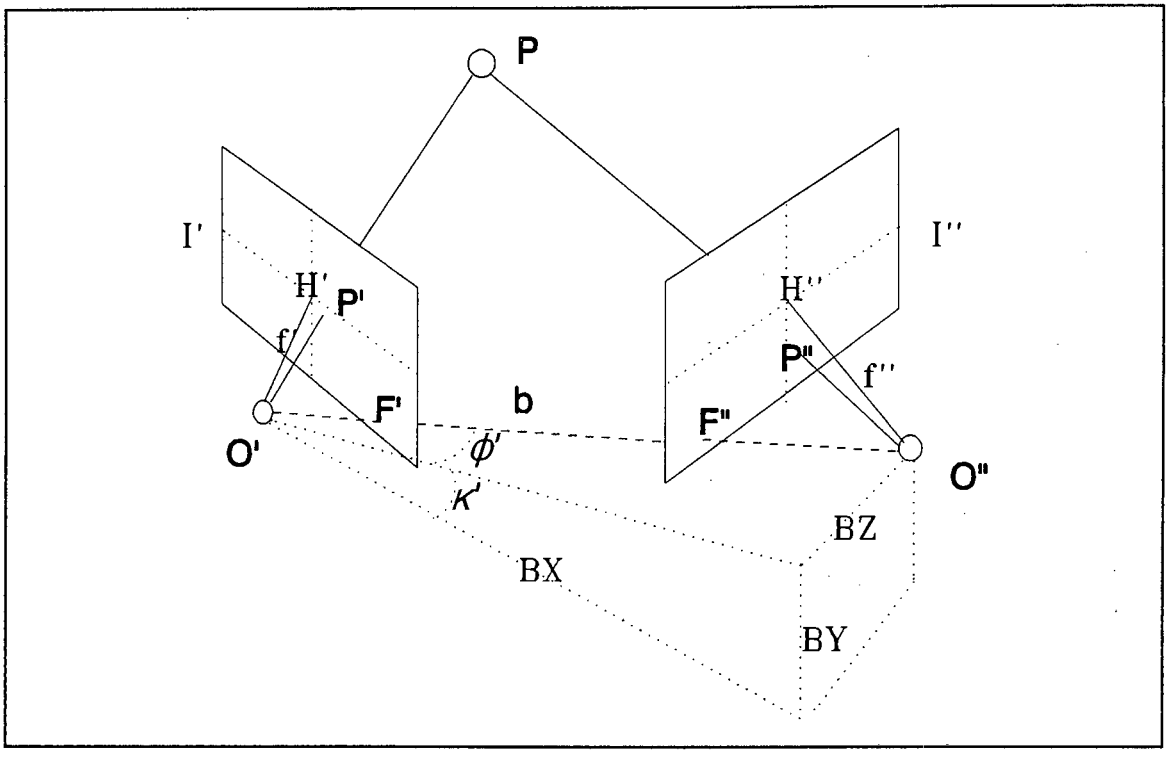


Figure 2.6.3. *Epipolar geometry of a stereopair of images I' and I".*

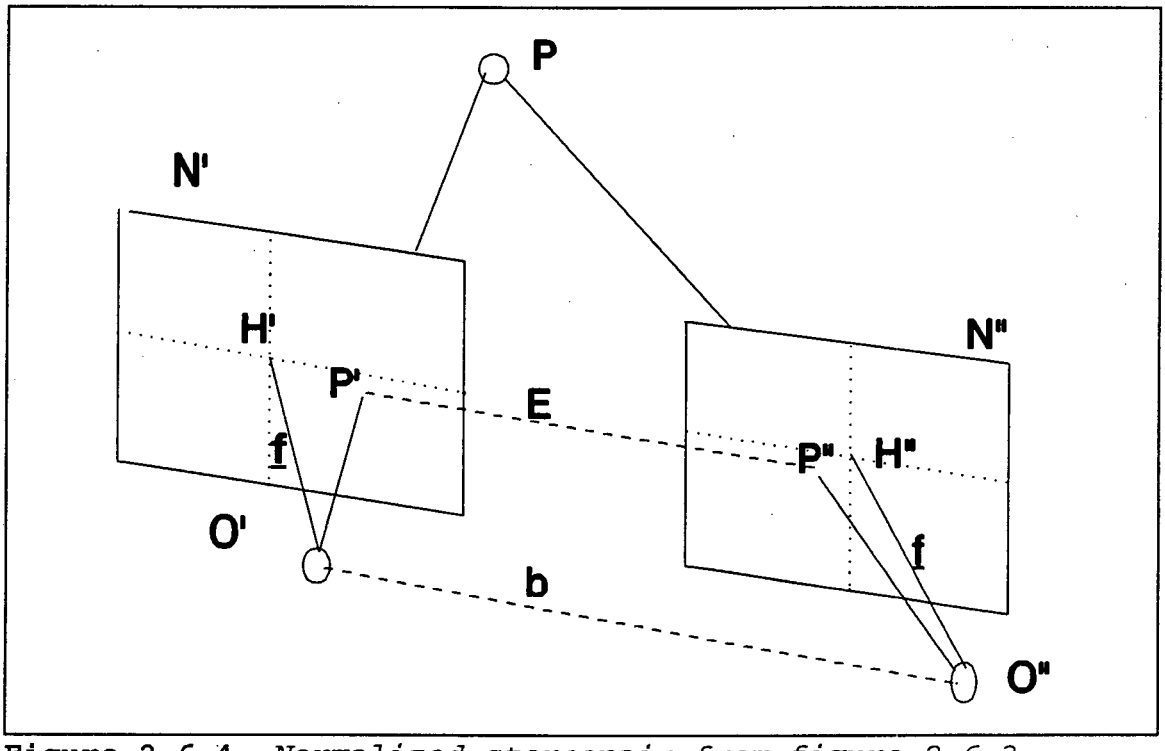


Figure 2.6.4. *Normalized stereopair from figure 2.6.3.*

in the left image the corresponding point P" must lie somewhere along the epipolar line P"F" in image I". In order to determine the epipolar lines in image I" corresponding to those in I' the relative orientation between the two cameras needs to be known.

The relative orientation is defined by choosing the perspective centre O' of the left camera as the origin. The coordinates of the perspective centre O" of the right camera is then (BX,BY,BZ). The orientation of O"H" with respect to O'H' is given by angles (ω,ϕ,κ). By fixing one of these six parameters arbitrarily ,the other five parameters values defining the relative orientation of the right image with respect to the left image can be determined. As shown in chapter 6 these values can be determined from matched points in the image by using the coplanarity condition. The equations for calculating the epipolar lines from the relative orientation parameters are given in chapter 6 and can be found in *Wong and Ho, 1986*.

If the images can be transformed into planes that are parallel to the baseline b , the epipoles will be at infinity and the epipolar lines will be parallel to b. By transforming the images into the same plane parallel to b the corresponding epipolar lines will be horizontal. Once this transformation is complete the search space for corresponding points will be one-dimensional as the points will lie on the same horizontal line at the same distance from the perspective centre in each image. The transformation is know as normalizing the images. The normalized images are represented as N' and N" in figure 2.6.4.

Schenk, 1990, derives formulae required to obtain the normalized images and are shown below.

$$\phi' = \arctan\frac{BZ}{BX}$$

$$\kappa' = \arctan\frac{BY}{(BX^2 + BY^2)^{1/2}} \qquad \textbf{(2.6.2)}$$

$$R_N = R_{\kappa'}R_{\phi'}$$

where

$$R\phi' = \begin{vmatrix} \cos\phi' & 0 & \sin\phi' \\ 0 & 1 & 0 \\ -\sin\phi' & 0 & \cos\phi' \end{vmatrix} \qquad R\kappa' = \begin{vmatrix} \cos\kappa' & -\sin\kappa' & 0 \\ \sin\kappa' & \cos\kappa' & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$R_T = R_N R^T \qquad (2.6.3)$$

where

R_T is the required transformation.

R^T is the transpose of the rotation matrix of the exterior orientation. The elements of the rotation matrix are shown in chapter 6.

The transformation R_T is applied to both images to give the normalized images.

$\underline{f}$ is the principal distance assumed for the normalized image and can be taken as equal to the camera principal distance.

The transformation therefore entails transforming the images into a vertical aerial position by applying their respective exterior orientation rotation matrices R^T and then transforming into the normalized position using R_N.

If only the relative orientation parameters are available then the right image can be rotated into a position parallel to the left image before the normalization transformation is applied to both images.

Once the image orientations are known then most image matching strategies should incorporate the epipolar line constraint to dramatically limit the search space. The epipolar lines can be determined generally without normalizing the images and the

matching algorithm need only search along this calculated line. Alternatively to accelerate the matching procedure the images can be normalized as described above allowing the matching algorithm to search only along the epipolar lines. The resampling required to generate the normalized images can be time consuming, but once these images are available the matching procedure is considerably faster. Naturally the speed is dependent on the matching algorithm chosen and the specific application.

2.7 PATTERN RECOGNITION BASED STEREO MATCHING

In many pattern recognition applications a pattern is described in terms of a feature vector with numerical component values. In computer vision this method is usually referred to as statistical pattern recognition. Statistical pattern recognition does not appear to be used much in photogrammetric applications, although it seems to be potentially useful, especially for the location of objects in scenes. As mentioned before the primary goal of this thesis is to show how the pattern recognition approach can be used in photogrammetry.

Most of the novel approaches in image matching concentrate on relational object descriptors with emphasis on the geometrical relationships. There is considerable justification for this because geometrical primitives such as points ,lines ,corners and polygons are essential for accurate point matching. The major stumbling block that these systems face when dealing with real images is the quantization of the image into discrete pixels and the quantization of the radiant information into a limited number of grey-scale units. In addition considerable noise is introduced into the image by environmental and digitization factors. Extracting reliable geometric primitives from an image under these conditions poses difficult problems. Many feature extraction techniques have been developed to overcome these problems and many successful applications have been developed.

As an alternative approach and because of the inherently

stochastic nature of the digital image acquisition process, statistical pattern recognition can be applied. In its simplest form an image region is selected in the left image of a stereopair surrounding the object of interest. A training algorithm which extracts a feature vector description of the region is then applied. Once the feature vector is obtained the right image is searched with a similar regional window and feature vectors are extracted sequentially. A nearest neighbour rule is used to locate similar objects. Application of this rule may be subject to a threshold value so that all regions that are closer than the threshold to the model vector are accepted as possible matches. Other constraints such as acceptable disparity ranges can be used to disambiguate matches. The closest of the remaining matches is then chosen as the correct location. If the relative orientation of the image is known then the epipolar line constraint can be used to limit the search to the epipolar line. Details of this method are given in chapter 4.

In model based photogrammetric systems a viewing aspect of the three-dimensional object can be generated and a feature vector determined from this view. This vector with a suitably determined threshold and possible a priori constraints, can then be used to search both the left and right images to locate the desired object. A model based pattern recognition application is described in chapter 3.

By using statistical pattern recognition in conjunction with relaxation techniques and area-based matching a 3-D reconstruction of an object can be generated as will be described in chapter 7. Although the reconstruction is termed 2.5-D as described above, 3-D will be used from now for convenience and because it has a clearer intuitive meaning.

2.8 CONCLUSION

Digital image matching was analyzed in general to gain some insight into the nature of the problem. Various digital image

matching techniques were briefly reviewed in order to demonstrate the scope of the techniques available. The applicability of statistical pattern recognition in digital photogrammetry was discussed and as will be shown can be used advantageously in photogrammetric applications.

3. PATTERN RECOGNITION

As an introduction to using pattern recognition principles in photogrammetry two industrial applications of pattern recognition were investigated. The essential basis of pattern recognition systems is outlined and illustrated with quality control applications.

3.1. PATTERN RECOGNITION PRINCIPLES

Digital pattern recognition is essentially a data-processing problem which can be divided into three basic phases *[Sing-Tze Bow, 1984]* as shown in figure 3.1.1.

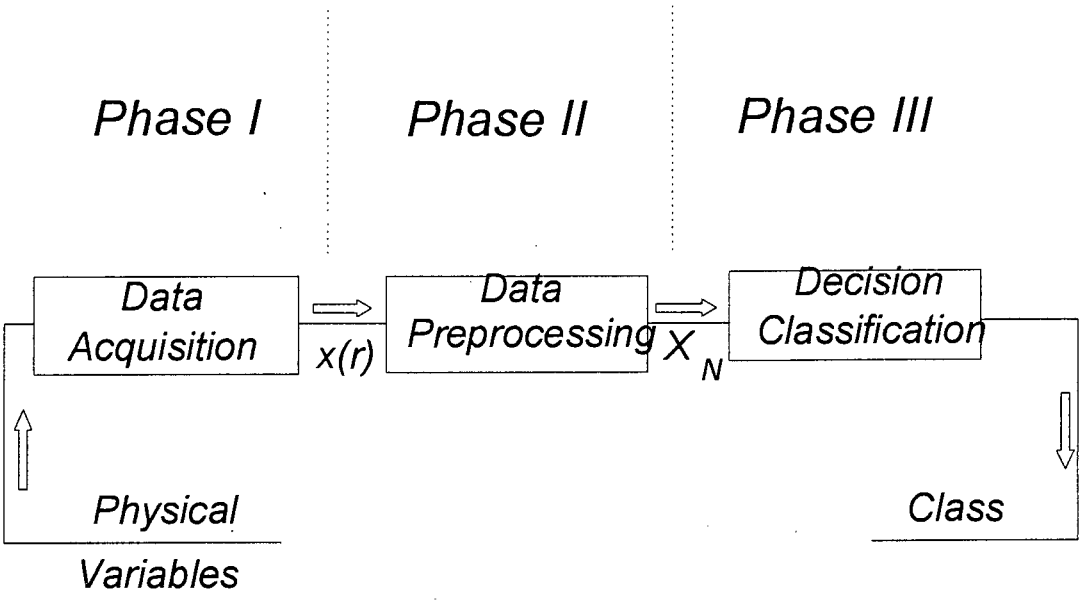


Figure 3.1.1. *Main phases in a general pattern recognition problem.*

The data in digital pattern recognition are digitized analogue signals. Examples of this are the light intensity variations on the image plane of a camera in image acquisition systems. The digitization of the physical variables by means of sensors in the environment forms the initial phase of data acquisition. The

output of this phase is the digital representation of the signals acquired by the sensors, x(r). In most modern close-range photogrammetric applications CCD (charge-coupled device) cameras are used as the sensors. Aerial photogrammetry uses high quality analogue photographic equipment to capture images. These photographs are usually digitized using flat-bed scanners.

The methods used to process the digital data are primarily application dependent. Features are extracted from the digital data in phase II. Various digital image-processing techniques are employed in this phase ,and the output is a vector or descriptor consisting of a number of features.

Phase III performs the function of classifying the extracted feature vectors or descriptors according to decision rules. Most pattern recognition applications use trainable pattern classifiers *[Tou and Gonzalez, 1981]*.In these cases model class vectors are predetermined by a training algorithm, also known as a teach-in procedure. Decision rules are then formulated using these model vectors. There are many possible decision functions some of which are explained in *Tou and Gonzalez, 1981*. Non-trainable classification rules can also be generated using various clustering algorithms.

The specific methods selected in all the phases mentioned above are a function of the application. Phases I and II can be regarded as a filtering and transformation procedure, where only a narrow band of information is extracted from the environment, and transformed into data elements that can be interpreted by classification rules.

There are two main interrelated approaches to pattern recognition problems, the **decision theoretic approach** and the **syntactic approach**. Both require the phases outlined above but the details of implementation are different. The decision theoretic approach relies on the essentially statistical nature of image data. The digital data is transformed into feature

vectors with N stochastic numerical components. These are processed according to decision functions in an N-dimensional vector space. The syntactic approach relies on extracted structural primitives from the image data to form an alphabet. These are combined to form sentences or strings to describe objects. A combination of sentences or strings forms a language which is governed by rules known as a grammar. 'Parsing' is the term used to denote the procedure that establishes whether a sentence belongs to a particular language. The statistical nature of image data can be incorporated with stochastic grammars. More detail on the syntactic approach can be found in *Tou and Gonzalez, 1981*.

The above two approaches to pattern recognition provide a neat formalism to the subject. In most practical applications the formalism is adapted to the heuristics of the problem and aspects of both formalisms may be combined. In the applications to be discussed aspects of both formalisms are used.

3.2. BINARY IMAGE PATTERN RECOGNITION

In order to illustrate the utility of digital pattern recognition in locating and interpreting patterns in a scene, two quality control applications in industry are described. Both methods can be regarded as applications of model-based object recognition. The object to be found is known beforehand with a predefined or predetermined description.

The first application entails the localization and decoding of barcodes on packaging boxes. The barcode is predefined and the environmental constraints reduce the problem to identification on a two-dimensional surface as would occur in optical character recognition. A similar approach could be adopted to locate a predefined structure consisting of high contrast substructures in an aerial image.

The second application involves the identification of labels on a cylindrical surface. Predetermined descriptors are derived

interleaved refers to the fact that both the light and dark bars are used to represent numbers. The dark bars take precedence reading from left to right. The code convention is shown in table 3.2.1.1.

Character	Code
0	00110
1	10001
2	01001
3	11000
4	00101
5	10100
6	01100
7	00011
8	10010
9	01010
START	110
STOP	101

Table 3.2.1.1. *2 of 5 Interleaved barcode.(1=wide bar, 0=narrow bar)*

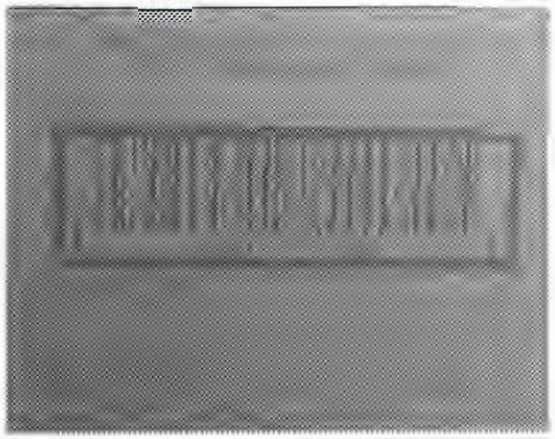


Figure 3.2.1.2. *Typical 2 of 5 interleaved barcode on a box.*

Figure 3.2.1.3 shows the flow diagram of the controlling software of the barcode identification system. The module POLL keeps reading the contents of the port connected to the photocell detector until a change of state is detected. The state changes when a box breaks the circuit by passing in front of the light source. POLL then directs CAPTURE to grab an image of the box face. The module BARCODE then analyses the captured image to locate and decode the barcode. The results of the decoding are then sent to the module OUTPUT which controls the appropriate response. OUTPUT is tailored to the requirements of the system.

The analysis of the images was done using very rudimentary image processing routines in order to minimize the computation time required. Each CAPTURE-BARCODE cycle was needed to be completed in less than a second.

from models of the labels on a cylinder. As different parts of the label can be visible at any time , descriptors from different views have to be predetermined. This is an application of model-based pattern recognition where different viewclasses are derived from a known model. Viewclasses are discussed further later in this chapter. Considerable research is being undertaken in the area of model-based techniques especially in remote sensing ,but can be applied in photogrammetry also. *[McKeown, 1988] [Muller, 1988]*

3.2.1 BARCODE IDENTIFICATION

In industrial applications images are often transformed into binary images by thresholding to extract specific information. As will be shown later this can also be applied to images in photogrammetric applications. To illustrate the principle an application where a barcode is located and decoded on packaging boxes will be described.

Figure 3.2.1.1 shows the basic structure of the system. The box moves along the conveyor belt. When it passes the photocell switch it triggers the computer which instructs the framegrabber to freeze and grab an image. The software then processes the image ,extracting and decoding the barcode.

The packaging boxes had a 2 of 5 interleaved barcode as shown in figure 3.2.1.2. A concise explanation of some commonly used barcodes can be found in *Burkinshaw, 1992.*

The 2 of 5 interleaved code represents each number from 0 to 9 by 5 vertical bars 2 of which are wide bars. The term

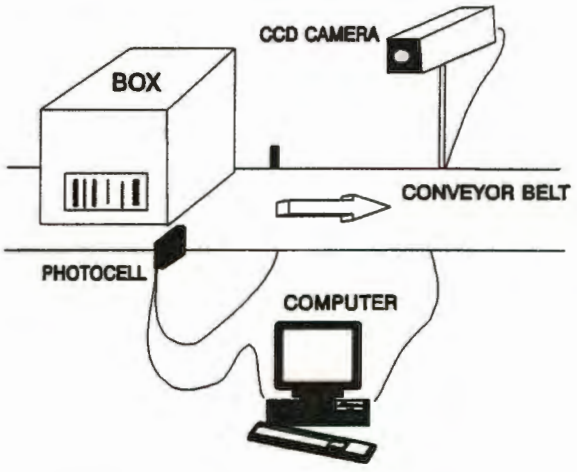


Figure 3.2.1.1. *Basic structure of barcode reading system.*

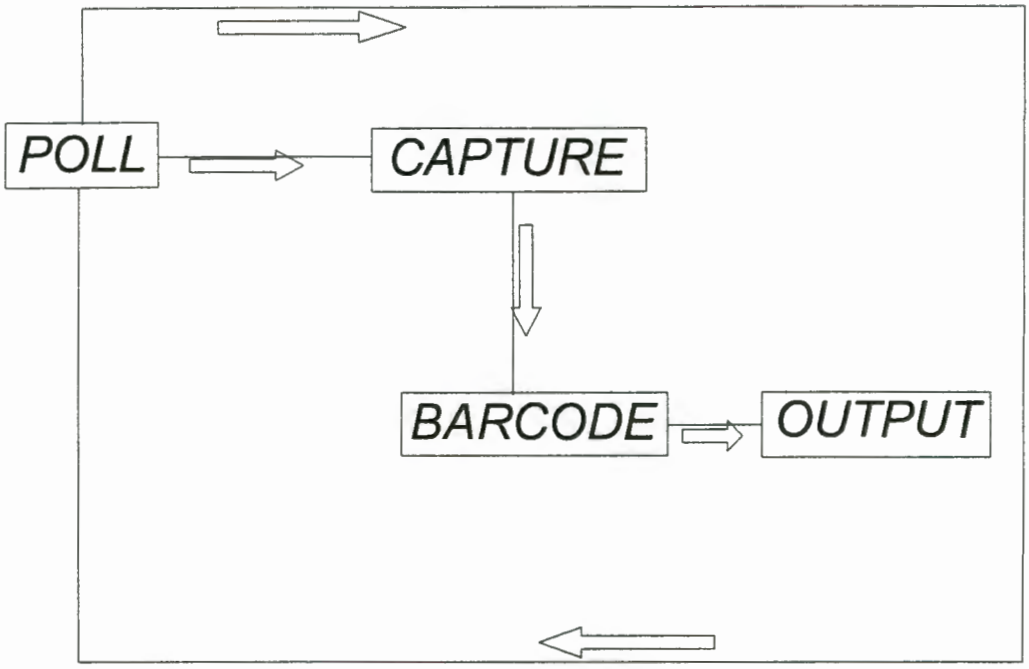


Figure 3.2.1.3. *Flow diagram of the barcode software modules.*

The BARCODE module is split into two main subroutines, one to locate the barcodes and the other to decode the barcode. To locate the barcodes the image is scanned and thresholded line by line in both the horizontal and vertical direction, as the barcodes are either horizontally or vertically orientated on the box.

The threshold value is determined by first finding the maximum and minimum pixel value in a line. A contrast transfer function is calculated as

$$CTF = \frac{(pmax - pmin)}{(pmax + pmin)} \qquad \textbf{(3.2.1.1)}$$

where

CTF is the Contrast Transfer Function

pmax is the maximum pixel value.

pmin is the minimum pixel value.

If the CTF is less than 0.3, which was established heuristically, the line has too low a contrast for effective thresholding. If the CTF > 0.3 then the threshold value is determined as

$$THRESH = \frac{(pmax + pmin)}{2} \qquad \textbf{(3.2.1.2)}$$

where

THRESH is the threshold value for the line.

Pixel values greater than THRESH are equated to the maximum pixel value and pixel values less than THRESH are made zero. By starting at one end of the line and counting the number of adjacent pixel value transitions from low to high and high to low across the entire binarized line, the number of bars in the line can be determined. If the number of bars counted falls within the range 35 to 46 allowing for poor contrast and noise, then the line is assumed to cross a barcode. The start and end lines of the barcode is then determined and the barcode is located.

All numerical parameters are determined from the heuristics of the particular application.

Having determined the location and orientation (horizontal or vertical) of the barcode , a number of lines crossing the barcode at different positions are decoded. More than one is required to allow for local image noise and localized printing defects in the barcode pattern.

By the determining the pixel widths of the bars in a line and using a priori knowledge the barcode can be decoded. The a priori information is as follows:-

(a) The ratio of the number of narrow to wide bars is 3:2.

(b) The average wide bar is about 2.5 times the width of a narrow bar.

The width of a bar in a binarized line is determined by counting the number of adjacent pixels with the same value. The average bar width over the whole line is then determined. Using the a priori information above expressions for the average values of the narrow and wide bar widths and their variances are determined as follows:-

$$\overline{nb} = \frac{5\overline{ab}}{8}$$

$$\overline{wb} = \frac{25\overline{ab}}{16} \qquad \textbf{(3.2.1.3)}$$

$$\sigma_{nb} = \frac{10\sigma_{ab}}{31}$$

where

$\overline{ab}$ is the average bar width over the line.

$\overline{nb}$ is the average narrow bar width.

$\overline{wb}$ is the average wide bar width.

$\sigma_{ab} = \frac{\Sigma ab_i}{n} - \overline{ab}$ is the variance of the average bar width with n bars.

σ_{nb} is the variance of the narrow bar width.

$\overline{ab} = \frac{\Sigma ab_i}{n}$ is the average of n bar widths.

The variance of a wide bar is heuristically taken as $\sigma_{wb} = 4\sigma_{nb}$

A significant amount of the variance is accounted for by the quantization error introduced by the digitization of the image. The effect of this source of variance will be more significant for the narrow bands than the wide bands.

The pixels within a bar are counted up by stepping along the line, and using the above parameters the bar can be identified as narrow or wide. After counting over the start code bars the remaining identified bars are formed into groups of 5. Depending on the ordering of the bars within each group a number is associated with each group according to the predefined code laid out in table 3.2.1.1. The barcode is then determined.

If one or more of the lines decoded compares exactly with any of the stored recognizable barcode numbers the box type is determined. This is then output to the OUTPUT module where an appropriate user defined operation is executed.

Four barcode types were used to test the initial system. The foreground and background colours have a significant effect on the observed contrast. The results of the tests are given in table 3.2.1.2. FG and BG stands for foreground and background respectively. CTF is the previously defined , equation (3.2.1.1),contrast transfer function expressed as a percentage. The RELIABILITY is given as a percentage and is the number of correctly identified barcodes divided by the total number of barcode readings taken. The number of readings for determining RELIABILITY was 40. 50 lines are decoded at each reading. The readings were taken with optimal environmental settings such as light level, focusing and resolution.

As shown in table 3.2.1.2 one barcode has a very low RELIABILITY value. In this particular case the global contrast over a line is good but the local contrast between narrow bars is poor. The pixel intensity profile along one of these lines is shown in figure 3.2.1.4 below. The narrow bar peaks are considerably lower than the wide bar peaks, resulting in the difficulty of obtaining a reliable threshold value. In the case of the other codes the bar peak heights are similar. On careful investigation of the printing ink used for the barcodes in the bad case, it was found that the background colour was showing through the foreground colour. This was not apparent in the case of the other barcodes. Using better printing ink would circumvent this problem.

BARCODE	COLOUR		CTF	RELIABILITY
	FG	BG		
16001237063333	grey	white	40.6	67.7
16001237072335	blue	brown	98.8	98.6
16001237012324	green	brown	49.8	2.5
16001237006323	Dark brown	brown	58.6	74.1

Table 3.2.1.2. *Barcode reading results. Terms are defined in the text.*

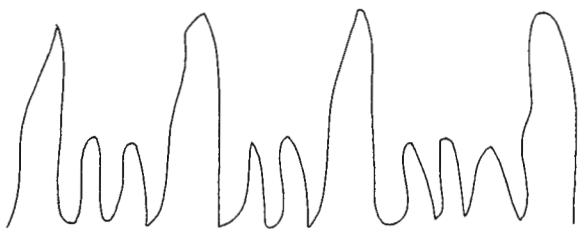


Figure 3.2.1.4. *Pixel intensity profile of line along barcode.*

The above technique of decoding barcodes can be used in a general sense to identify patterned boxes. By converting a pattern on a box into a binary image unique descriptors can be derived and

used in conjunction with some a priori information for identification. Further research in this direction is required. Some of the recent developments in information encoding and decoding using barcodes is given in *Pavlidis, Swartz and Wang, 1990 and 1992.*

The method used in the above application is mostly related to the syntactic approach to pattern recognition. The pattern primitives are the bars and their ordering determines the pattern class. The string of pattern primitives forms a sentence. The sentence is parsed according to the grammar defined by the barcode to obtain the numbers that the barcode represents.

In general terms this example is a simple application of model-based pattern recognition where the object to be found is predefined and two dimensional.

3.2.2 LABEL IDENTIFICATION

As an example of a more complex pattern recognition application a system for identifying cigarette types as they are packed into their cigarette boxes is described. In this application ,model label descriptors are built up and stored using a teach-in phase. Identification and classification is performed by comparing extracted descriptors with the model descriptors by template matching.

Figure 3.2.2.1 shows the basic layout of the cigarette label identification system. The packing machine puts an empty cigarette box in position and places a required number of cigarettes into the box. The cigarette labels will be randomly distributed around the axis of rotation of the

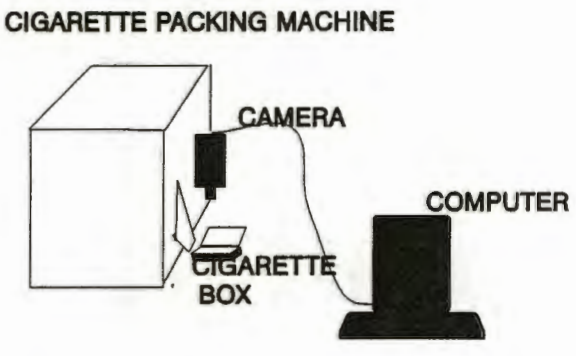


Figure 3.2.2.1. *Basic layout of cigarette label identification system.*

cigarette cylinder. The image of the top layer of the filled box is captured and analyzed to identify the exposed labels on the individual cigarettes. The objective of the system is to ensure that the correct cigarette type is packed into its corresponding box. When an incorrect cigarette type is identified the packing machine is halted so that the batch of cigarettes placed in the machine can be checked and replaced. As large batches of cigarettes are placed in these machines mixing of brands often occurs thus justifying the necessity of a monitoring system.

Quite clearly not all cigarette labels will be exposed at any one time and the system depends on the random statistical behaviour of the positions of the cigarettes when packed. In this situation a number of mixed batches will get through ,but as each batch consists of a very large number of cigarettes the probability of a wrong brand revealing itself will be high. Modelling of this statistical behaviour can be regarded as a separate problem in this particular description, as the main emphasis is to explain the recognition process.

Figure 3.2.2.2 shows a captured image of the top layer of a cigarette box. Not all the cigarettes are viewed as the label identification required a fairly high resolution. This will reduce the overall sample size in the statistical modelling of the process referred to above. The sample size is therefore a function of the camera resolution.

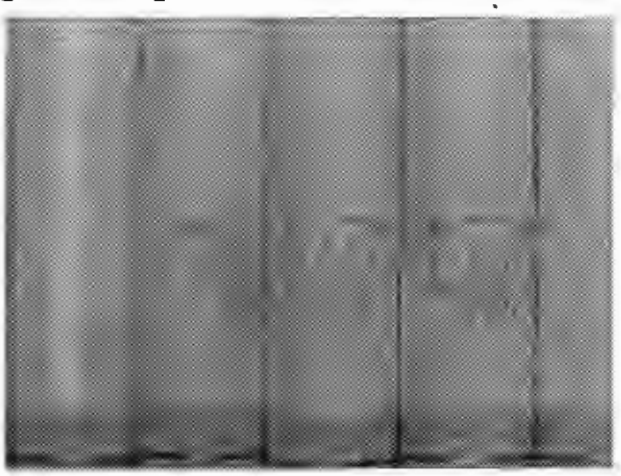


Figure 3.2.2.2. *Cigarette labels captured from the top layer of a cigarette box.*

As shown different views of the label will be visible according to the angle of rotation of the cigarette cylinder. Descriptors for all possible views of the labels are required for template

matching. In designing a suitable descriptor for the label the constraints of very limited computation time had to be considered as the packing machines operate at high speed. Therefore a fairly simple label descriptor is required.

Before extracting the label descriptors from the images the labels have to be located. Localization of each cigarette is possible due to the large gradients in the grey-scale image of the cigarettes that exist as a result of the small space between each cigarette in a box. The gradients of a number of lines across the image are analyzed and the cigarette boundaries are taken where the gradient is larger than a heuristically determined threshold. As the labels are placed at a similar position on most brands of cigarette, the camera is situated so that it will capture the image of the region containing the labels. Heuristically determined vertical limits are placed on the area that is image processed. Figure 3.2.2.4 shows the areas containing the labels that have been processed.

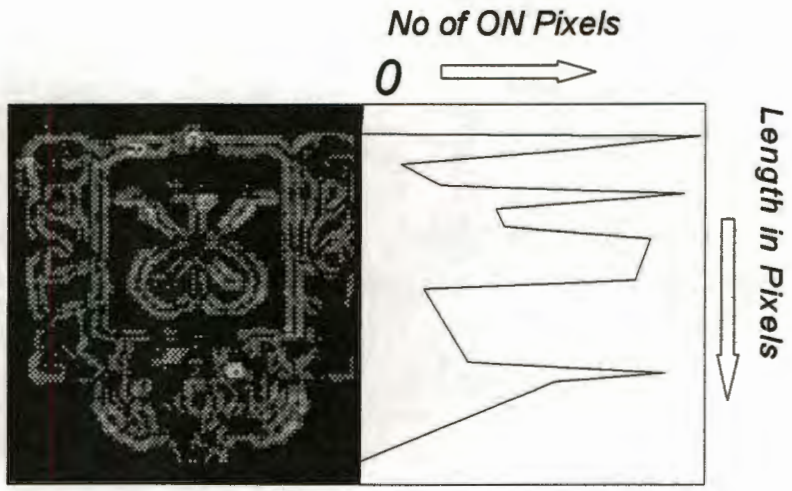


Figure 3.2.2.3. *One dimensional label signature derived from edge pattern of label.*

A one dimensional signature pattern is derived from the two dimensional label as shown in figure 3.2.2.3. A simplified edge detector is used to determine the gradients that depict the label and a threshold is used to eliminate all noisy edgels (edge pixels). The threshold is determined by analysing the histogram of the edgel distribution. The edge detector is based on calculating the gradient as the difference in the grey-scale values between a particular pixel and its North, South, East and West neighbours, and taking the maximum difference as the gradient. The edge image thus obtained is converted into a binary image with minimal noise by thresholding as shown in figures 3.2.2.3 and 3.2.2.4. This binary image is converted into a one dimensional signature which is a function of the number of ON pixels across the label against the length of the label as shown in figure 3.2.2.3. This can be regarded as a one dimensional projection of the two dimensional label and its use in pattern recognition is mainly motivated by its ease of computation making real-time applications possible. Some applications of the signature projection technique are described by *Sanz and Dinstein, 1987, Pavlidis, 1968, Rosenfeld, 1974.*

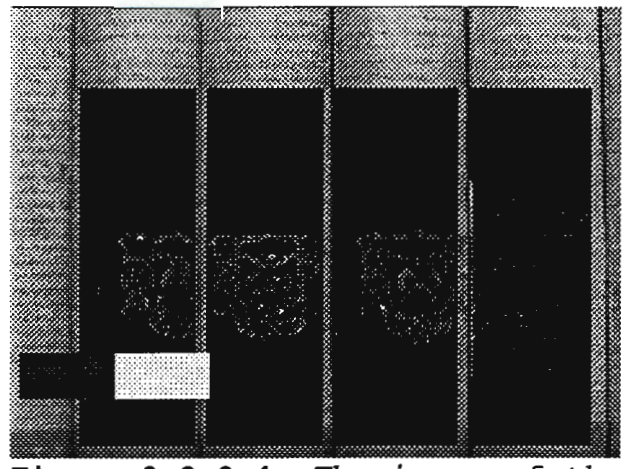


Figure 3.2.2.4. *The image of the cigarettes after processing and identification. As indicated by the shaded blocks the first two labels are correctly identified.*

Figure 3.2.2.4 shows the results of an identification run where the first two labels are correctly identified. The horizontal bar at the left hand bottom corner of the figure indicates the result of the identification run. The two dark grey squares on the left of the bar show that the two cigarettes shown on the left were correctly identified. The two white squares on the right of the bar show that the two cigarettes shown on the right were not

identified. The third one was not identified, due to the inclusion of some of the edgels resulting from the cigarette boundaries and not solely those from the label. The fourth cigarette shown on the right of the image, was correctly not identified due to the absence of a label.

A descriptor was derived from the signature. A label is assumed to commence at the point where the signature value is greater than some experimentally determined value. The label ends when the signature value falls below a similar value. The descriptor is defined for the region of the signature occupied by the label. The descriptor consists of the following components:-

1. The pixel length of the label.

2. The first four moments of the signature given by

$$M_n = \sum_{i=1}^{L} i^n f_i \qquad \textbf{(3.2.2.1)}$$

where

M_n is the nth moment.

i is the pixel position along the length of the signature.

L is the length of the label in pixel units.

f_i is the proportion of the total signature pixels at point i. This is equal to the number of on pixels at point i divided by the total number of on pixels in the signature.

3. The first four moments of each half of the signature.

4. The mean number of on pixels in each of m equal subregions over the signature. The mean number is given by

$$\bar{p} = \frac{\sum_{i=st}^{end} p_i}{end-st+1} \qquad (3.2.2.2)$$

where

$\bar{p}$ is the mean number of on pixels in each subregion.

st is the starting point of the subregion.

end is the endpoint of the subregion.

p_i is the number of on pixels at i.

5. The variance in the mean number of pixels over the same m subregions given by

$$\sigma^2 = \frac{\sum_{i=st}^{end} p_i^2}{end-st+1} - \bar{p}^2 \qquad (3.2.2.3)$$

where
σ^2 is the variance.

6. The centre of mass in each of the m subregions given by

$$COM = \frac{\sum_{i=st}^{end} ip_i}{\sum_{i=st}^{end} p_i} \qquad (3.2.2.4)$$

where
COM is the centre of mass.
m = 8 seemed to give good results at the resolution adopted with the length of the signature L in the range [100,150] pixels.

Descriptors of prototype label patterns are created by means of a teach-in procedure. A prototype pattern has to be generated for each possible view of the label. Prototype patterns are limited to views with the label wholly or partially visible, at every 15 degrees of rotation of the cigarette cylinder. The prototype patterns do not change significantly for angles of rotation less than 15 degrees. These prototypes are compared with the extracted descriptors from an image using an hierarchical decision rule.

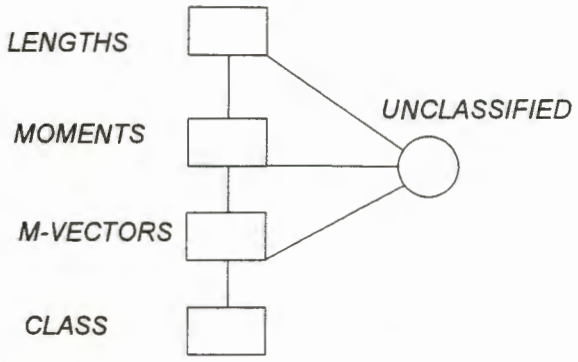


Figure 3.2.2.5. *Hierarchical decision rule for the label classification.*

Figure 3.2.2.5 shows the structure of the decision rule. The circular endpoint indicates that the extracted descriptor is of an unknown class. Firstly the lengths of the extracted labels are compared with the stored prototype lengths. If the length of the extracted labels falls within a calculated acceptance range, the classification procedure continues to the next stage or else the process is terminated as unclassified. The process continues in a similar vein with the comparison of moments followed by the comparison of the subregion vectors. At each stage the label is either rejected as unclassified or else passes through all the comparison stages and finally classified as one of the prototype classes.

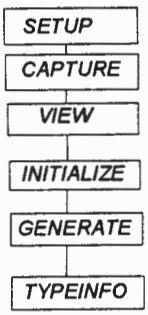


Figure 3.2.2.6. *Modular flow of the teach-in system to capture label descriptors.*

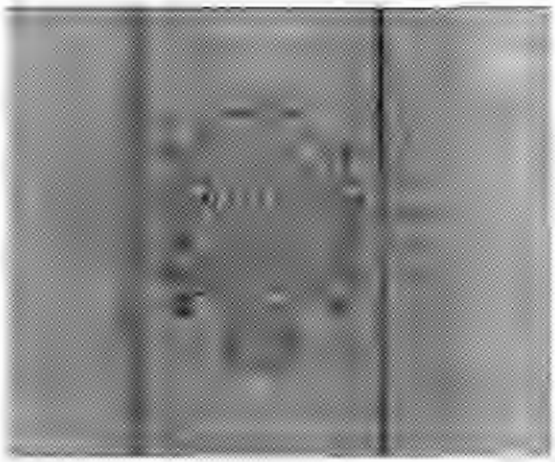


Figure 3.2.2.7. *Label in position for image capturing.*

Figure 3.2.2.8. *Flattened out label image.*

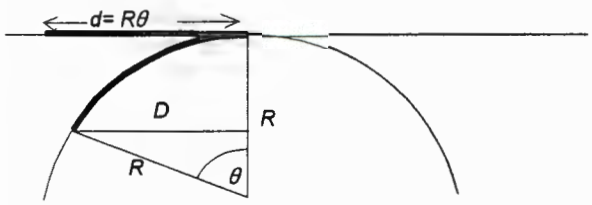


Figure 3.2.2.9. *Cross-section of a cylinder showing the dimension parameters and angles used to flatten out label.*

Figure 3.2.2.6 shows the flow of the modules used in capturing prototype label descriptors. SETUP prompts for width and height of image and the pixel diameter of the cigarette for the image shown in figure 3.2.2.7. CAPTURE captures the image and converts the cylindrical label pattern into a flattened pattern as shown in figure 3.2.2.8. The label is captured from an image taken at close proximity to the cigarette in order to obtain a large image of the label. Figure 3.2.2.9 shows the dimensional parameters and

angles used for the flattening process. In essence the label is unrolled from the cylinder in this process. The radius of the cylinder in pixels can be determined from the image diameter of the cigarette. For any particular point on the label the angle θ can be determined as

$$\theta = \arcsin\frac{D}{R} \qquad \textbf{(3.2.2.5)}$$

where

R is the pixel radius of the cylinder.

D is the orthogonal pixel distance from the axis of the cylinder.

The flattened label is the rolled out circumference of the cylinder containing the label. The flattened out distance from the axis to place label point is given by

$$d = R\theta \qquad \textbf{(3.2.2.6)}$$

where

θ is in radians.

d is the orthogonal pixel distance of the flattened label from the cylinder axis.

Applying the above transformation results in increasingly large gaps in the image as the cylinder boundary is approached. Resampling by direct linear interpolation is applied to fill in the gaps. Some reflectance model needs to be assumed to calculate the correct pixel intensity on the flattened surface. The Lambertian model was initially assumed where the radiance is directly proportional to $\cos\theta$, but by experimentation a more suitable radiance dependence was obtained with proportionality to $\cos(\theta/2)$. Figure 3.2.2.10 gives the resulting image of the rolled out label over the whole circumference of the cylinder.

Once an image of the flattened label is available, the label can be placed on a cylinder to simulate the differently exposed views of the label. VIEW places the label on a cylinder of any pixel diameter smaller than the pixel diameter of the original cigarette from which the label was extracted. A smaller pixel diameter is used as a label cannot be recreated at a higher resolution than that at which it was acquired. In addition the actual cigarette identification is performed at a lower resolution so that as many cigarettes as possible from the top layer of the packing box can be checked. The minimum pixel diameter is limited by the loss of meaningful detail making identification impossible. Edge detection and thresholding is carried out so that a suitable threshold value is determined interactively.

In order to place the label on a cylinder a pre-selected diameter is required and set at the pixel diameter observed in the images acquired when the final system is in operation. The pixel height of the label is scaled down to maintain the aspect ratio of the label. A two dimensional angular dependent image function is generated from 0 to 359 degrees and from 0 to the height of the flat label. Effectively the resolution of the flattened out label shown in figure 3.2.2.10 is reduced to the scale determined by the selected pixel diameter mentioned above.

All pixels that fall within a unit degree and unit height are averaged and that value is used to determine the intensity in the image function. The average intensity at the border of the flattened label image are used for all non-label pixel values. The resultant image function is shown in figure 3.2.2.10.

The two dimensional image function can then be used to generate a label on a cylinder starting at any particular angle. The pixel values are determined by calculating the angular interval between each pixel and averaging all contributing pixels. An example of a label placed on a cylinder at a different angle is shown in figure 3.2.2.11.

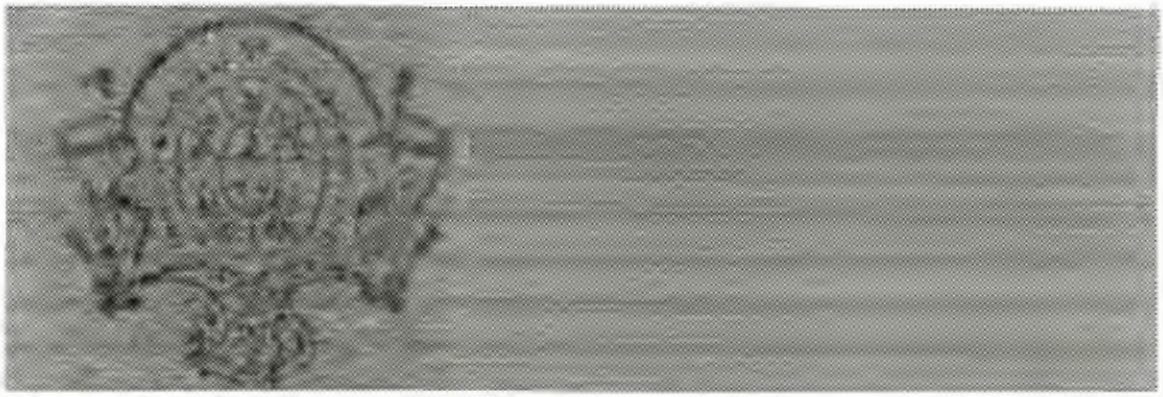


Figure 3.2.2.10. *Angular dependent image function for degree intervals from 0 to 360.*

Figure 3.2.2.11. *Label placed on cylinder to obtain different view.*

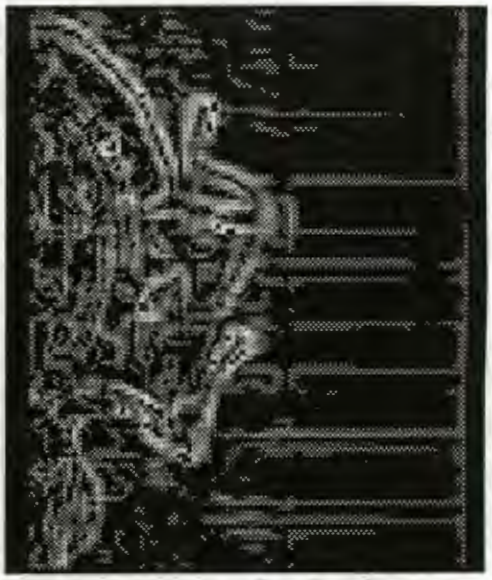


Figure 3.2.2.12. *Thresholded edges of figure 3.2.2.11.*

INITIALIZE is the module that requests parameters relevant to the prototype sample generation. An example is the prototype for a view with the cylinder at 15 degrees which is taken as the mean of the descriptors from 10 to 20 degrees at degree intervals. The angle and the ranges can be varied. Justification for this lies in the fact that the one dimensional signature does not change significantly over small angle intervals. The variances of the sample are also stored to determine limits of acceptance for the decision rule. The number of prototypes and the step in angular rotation of the cylinder at which each prototype is to be

determined is required by the program.

Using the parameters entered from INITIALIZE , GENERATE calculates all the required prototype descriptors and their variances.

TYPEINFO requires the user to update the on-line database of label types.

Once the teach-in phase has been completed the system can be used to check the labels of cigarettes in the system depicted in Figure 3.2.2.2. A typical individual run gives results as shown in figure 3.2.2.4, where two out of three labels were correctly identified. Errors of omission , where non-identification occurs when it should not are quite common. Errors of commission where misidentification takes place are minimal.

The system is very sensitive to lighting conditions. A powerful halogen lamp was used to illuminate the cigarette boxes. The environment needs to be optimized to improve the overall performance of the identification phase. An experiment in a sub-optimal environment gave 60% of possible labels as correctly identified.

3.3. APPLICATION TO DIGITAL PHOTOGRAMMETRY

The above cigarette label identification and barcode reader applications are simplified forms of pattern matching from different view classes and are model-based matching techniques. The concept of view classes was introduced by *Koenderink and Van Doorn, 1979*, in which a three dimensional object is represented by an aspect graph consisting of classes of two-dimensional views. Only a restricted number of simple polyhedra have been represented in this form. *[Plantinga and Dyer, 1986] [Gigus and Malik, 1988] [Gigus, Canny and Seidel, 1988] Ikeuchi, 1987 and Ikeuchi and Hong, 1989* have attempted to use view classes in matching to establish correspondence in a stereopair of images.

Although a pattern recognition approach as applied above can be used to locate corresponding pattern regions, obtaining point matching will depend on having a detailed geometric feature description of each view class. Consequently this method also faces similar problems to geometrically based feature matching techniques when used with real images. Among the problems are broken lines, occlusions, hidden lines and shadows. This method would be applicable in a highly constrained environment analogous to the label identification application.

Locating and extracting objects from an image appears to be the most promising application area of pattern recognition in digital photogrammetry. This aspect is investigated more fully in chapter 4 with extensions to possible point matching applications through interest operators.

4.FEATURE-BASED MATCHING USING PATTERN RECOGNITION

Feature-based matching is one of the standard techniques for image matching *[Haralick and Shapiro, 1993]*. In most applications it uses extracted point features from two or more digitized images and tries to establish an accurate correspondence between matching points. Most point features are extracted by using interest operators and edge detectors. These operators are based on the grey-scale nature of the image in small windows, typically 3 X 3 to 10 X 10 pixels (picture elements) in dimension. The following investigation attempts to show that statistical pattern recognition techniques ,using two dimensional moments as features, can be used to locate specific objects in an image, and to design specific interest operators which can be made to extract feature points to sub-pixel accuracy.

4.1 FEATURE - BASED MATCHING PRINCIPLE [*Haralick and Shapiro ,1993*]

The steps required in feature-based matching are:-

1. Features are extracted from each image using an interest operator or an edge detector.

2. The correspondence between the selected feature points is determined based on some criteria of similarity and consistency *[Van der Merwe and Rüther, 1994]*. The similarity criterion is based on attributes of the feature points and consequently on the similarity of the grey-scale neighbourhood of the points. The consistency criterion is based on the piece-wise continuous nature of the mapping function between the images. The mapping function determines how the position of a point in one image is transformed in the other image.

3. Once the mapping function has been determined at the corresponding extracted points, the disparity or parallax

map between the two images is determined by interpolation. The digital elevation model of the objects in the scene can then be obtained assuming that the absolute orientation of the images has been determined. Alternatively the digital elevation model can be found if the relative orientation of the images is determined and the object space coordinates of identifiable matching points in the images are known. In an automated system the relative orientation can be determined based on the results of step 2. above as will be discussed in chapter 6.

Statistical pattern recognition can be used for all three steps, although the computation time required will be longer than for existing methods, the accuracy could improve.

4.2 INTEREST OPERATORS

An image can be regarded as a mixture of different pattern classes. Examples of these classes are corners, edges , textured regions or smooth areas. Ideally an interest operator should be designed to

1. Extract a particular class while ignoring other classes.

2. Extract patterns localized to particular points in the image to sub-pixel accuracy.

Due to the effects of variations in perspective and illumination between two images this ideal is rarely completely achievable. For similar reasons deriving techniques for assessing the degree of accuracy in feature extraction from image data are difficult to formulate.

The information in a digitized image is contained in the spatial distribution of grey-scale intensity pixels in two

dimensions. Let this grey-scale intensity function be represented by $I(x,y)$. Any particular pattern class will be a subset of this intensity function $p(x,y) \subset I(x,y)$. The underlying principle behind the pattern recognition approach is that similar objects in three dimensional space will transform into similar patterns in two dimensional image space by virtue of the perspective transformation. In reality this is usually true only in a limited subset of all the possible camera-to-object-world configurations. An example of a configuration where the transformation of patterns from object space to image space could be consistent would be in two or more aerial photographs of the same scene taken at high altitude. In this case the scene is reduced to an almost two dimensional space with relatively small variations in disparity or parallax between the images. A configuration where objects will transform differently would occur in a stereopair of images in close range imaging where a change in the position of the camera can alter the view of any particular object dramatically. The pattern recognition approach could only be adopted effectively in the case of close range imaging systems where the camera configurations ensure consistent image transformations. As a result of uncontrollable factors in the environment the radiometric properties of images may vary significantly and alter the patterns in images. In most applications an attempt is made ensure fairly homogeneous lighting conditions.

The pattern classes that are identified with some of the more popular interest operators are

1.Straight line segments (edges) extracted with edge detectors. *[Haralick and Shapiro, 1993] [Canny, 1986]*

2.Junction points where straight line segments meet (corners) as with the Förstner operator. *[Förstner, 1987]*

3.Regions with large variations in their grey-scale as with the Moravec operator. *[Moravec, 1980]*

Similarly pattern classes can be identified using statistical pattern recognition principles. The latter approach usually entails transforming the raw pixel data into abstract numerical parameters as opposed to investigating the extracted geometrical features in the image.

4.3 BASIS FOR PATTERN RECOGNITION METHOD

A pattern class $p(x,y)$ can be represented by an N dimensional feature vector $\underline{v}_p$. Each component is a different pattern characteristic. The selection of these pattern characteristics are usually based on a heuristic approach towards the specific problem that needs to be solved *[Nandhakumar and Aggarwal, 1985]*.

For any particular pattern the feature vector descriptor has to be learnt by a training procedure. There is a number of different approaches to the training procedure but the basis of these methods assume that the feature vector is the mean vector drawn from a sample of a multivariate normally distributed population of feature vectors.

The multivariate normal distribution for a feature vector can be stated as follows *[Fukunaga, 1990]*.

$$N_{\underline{v}_p}(\underline{m}_p,\ \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2} d^2(\underline{v}_p)\right] \qquad \textbf{(4.3.1)}$$

where

$N_{\underline{v}_p}(\underline{m}_p, \Sigma)$ is the probability density of the feature

vector $\underline{v}_p$ in a normally distributed population with mean vector $\underline{m}_p$ and covariance Σ.

$\underline{m}_p$ is the mean feature vector.

Σ is the covariance matrix.

$|\Sigma|$ is the determinant of the covariance matrix.

$d(\underline{v}_p)$ is a distance metric defined as follows

$d(\underline{v}_p)^2 = (\underline{v}_p - \underline{m}_p)^T \Sigma^{-1} (\underline{v}_p - \underline{m}_p)$ usually referred to as the

Mahalanobis distance.

Σ^{-1} is the inverse of the covariance matrix.

T represents transposition.
n is the number of features in each vector or the dimension of the feature space.

The probability that $\underline{v}_p$ belongs to the class associated with vector $\underline{m}_p$ is given by

$$P(\underline{m}_p) = 2\int_{\underline{v}_p}^{\infty} N(\underline{m}_p, \Sigma)\, d\underline{x} \qquad \textbf{(4.3.2)}$$

A suitable selection of features and a suitably representative sample of the feature space need to be extracted from the image data. Suitable in this context is dependent on the nature of the pattern and the problem to be solved and is heuristically

determined in most applications. The optimality of the selected features towards solving a specific problem is largely indeterminate and the ultimate suitability of the set selected is judged on whether they can be used to identify the required patterns reliably.

The feature space extracted from the image data to act as a training sample is also problematic. A number of different samples can be extracted from an image or samples can be synthetically produced with additive noise. This will be discussed further when actual examples are considered. In general the sample set will not be drawn from a multi-variate normal distribution and non-parametric techniques will be needed ,based on some suitable distance metric, to establish decision boundaries in the classification scheme. Possible decision functions are considered later with the examples.

Once a suitable pattern vector and covariance matrix have been determined they can be used to find similar patterns in images. As an illustration consider the feature vector as belonging to only two possible classes. Either the feature vector belongs to the pattern class required or it does not. A suitable method of making this decision is based on the distance,as defined above,between the extracted vector and the mean pattern vector. The smaller this value the higher the probability that the vector does belong to the pattern class. A threshold value for this distance needs to be determined so that any extracted vector that is further than the threshold from the mean vector is excluded from the pattern class. This threshold can be determined in terms of the variance of the training sample or determined heuristically.

4.4 FEATURE SELECTION

The selection of features used is somewhat arbitrary and is usually dictated by properties of the pattern and the nature of the available image data. Once suitable features have been chosen

as components of a feature vector the statistical pattern recognition approach outlined above can be applied.

Figure 4.4.1 shows the main steps required in the application of the method.

1. FEATURE SELECTION.
The components of a pattern vector are chosen.

2. SAMPLE FEATURE VECTOR GENERATION.
A sample of feature vectors are generated from the image data containing the object pattern that is to be identified. The sample is then used to determine the mean pattern vector and the covariance matrix.

3. PATTERN IDENTIFICATION
The pattern vector determined in 2 is then used to identify similar patterns in images taken from a similar perspective.

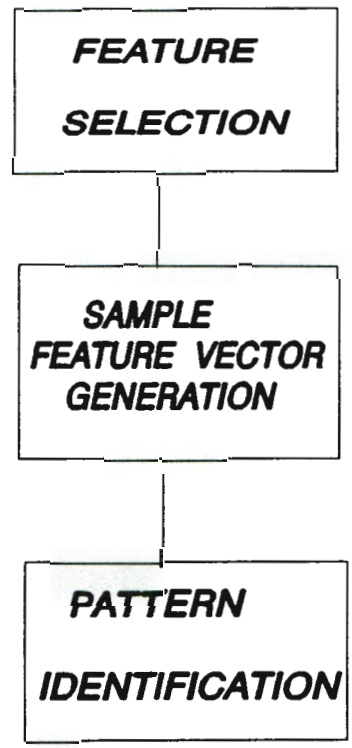


Figure 4.4.1. *Main steps in a statistical pattern recognition.*

The above method can be applied generally to identify object patterns in images. As an illustration the design of a corner interest operator will be discussed in detail and applied to both synthetic and real images. This interest operator can then be used to extract corner points in images. Different feature vectors will be considered for this particular application.

4.4.1.GEOMETRIC MOMENT INVARIANTS

For a two dimensional grey-scale image function f(x,y) the geometric moments are defined as *[Prokop and Reeves, 1992]*

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\, dxdy \qquad \textbf{(4.4.1.1)}$$

where

m_{pq} is the two-dimensional Cartesian moment of order p+q.

In a discretized form the moment for an M x N pixel image is

$$m_{pq} = \sum_{y=0}^{M-1}\sum_{x=0}^{N-1} x^p y^q f(x,y) \qquad \textbf{(4.4.1.2)}$$

The central moments are defined as

$$\mu_{pq} = \sum_{y=0}^{M-1}\sum_{x=0}^{N-1} (x-m_{10}/m_{00})^p (y-m_{01}/m_{00})^q f(x,y) \qquad \textbf{(4.4.1.3)}$$

where

m_{10}/m_{00} is the x coordinate of the centre of mass.
m_{01}/m_{00} is the y coordinate of the centre of mass.

The central moments are invariant under coordinate translations as the centre of mass or centroid of the pattern acts as the origin. If the pattern is translated in the image the centroid will be translated but the central moments will remain invariant.

Hu, 1962, has developed two dimensional moment invariants that have been successfully used in pattern recognition

applications *[Prokop and Reeves, 1992]*[*Dudani, Breeding and McGhee, 1977]*.Most of the applications have been on binary patterns and not on grey-scale patterns. In order to derive suitably robust binary patterns from a general grey-scale image a method for thresholding the image is required. The thresholding technique used will be discussed later in the text.

The Hu moment invariants are invariant with respect to translational, rotational and scale transformations in the image.

a) SCALE INVARIANCE

In order to ensure scale invariance the central moments have to be normalized. With reference to *Hu, 1962*, and *Teague, 1980*, let two image functions of a pattern at different scales be represented by f'(x,y) and f(x',y') where $x'=x/\lambda, y'=y/\lambda$ and λ is a scale factor then by the definition of central moments

$$\mu'_{pq} = \iint x^p y^q f(x/\lambda, y/\lambda)\, dxdy = \lambda^{2+p+q}\mu_{pq} \qquad \textbf{(4.4.1.4)}$$

$\lambda = \mu_{00} = m_{00}$ is the moment of order zero and can be regarded as representing the size of the pattern. The following normalized moments will therefore be invariant to scale changes.

$$\overline{\mu}_{pq} = \mu_{pq}/\mu_{00}^{(p+q+2)/2} \qquad \textbf{(4.4.1.5)}$$

This normalization method has the disadvantage of increasing the dynamic range of the central moment magnitudes.An alternative normalization method used by *Dudani,Breeding and McGhee, 1977*, uses the radius of gyration as the normalizing factor. The radius of gyration is given by

$$r = (\mu_{20} + \mu_{02})^{\frac{1}{2}} \qquad \textbf{(4.4.1.6)}$$

the normalized moments are then determined as

$$\bar{\mu}_{pq} = \mu_{pq} / r^{(p+q+2)/4} \qquad \textbf{(4.4.1.7)}$$

The dynamic range of the moment values is reduced in this case.

b) ROTATION INVARIANCE

By investigating the behaviour of central moments under an orthogonal rotation transformation *Hu, 1962*, developed functions consisting of central moments that are invariant under rotation.

The rotation transformation is

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta \qquad \textbf{(4.4.1.8)}$$

where

(x , y) are the initial coordinates to which the transformation is applied.
(x', y') are the rotated coordinates.
Θ is the angle of rotation.

Hu showed that the following functions of the central moments are invariant under rotation:-

$$\mu_{20} + \mu_{02} \qquad \textbf{(4.4.1.9)}$$

$$(\mu_{20} - \mu_{02})^2 + 4(\mu_{11})^2 \qquad \textbf{(4.4.1.10)}$$

$$(\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{02})^2 \qquad \textbf{(4.4.1.11)}$$

$$(\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \qquad \textbf{(4.4.1.12)}$$

$$(\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{02})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \qquad \textbf{(4.4.1.13)}$$

$$(\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \qquad \textbf{(4.4.1.14)}$$

$$|(3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{03} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]| \qquad \textbf{(4.4.1.15)}$$

Employing the normalized central moments in the above relations will result in expressions that will be invariant to translation, size and rotation. The geometrical invariants given by equations (4.4.1.9) to (4.4.1.15) normalized using equation (4.4.1.5) have been used in this investigation.

4.4.1.1 BASIC METHOD FOR PATTERN LOCALIZATION

The main steps required to generate model pattern vectors and to use these in locating similar patterns in images are shown in figure (4.4.1.1.1).

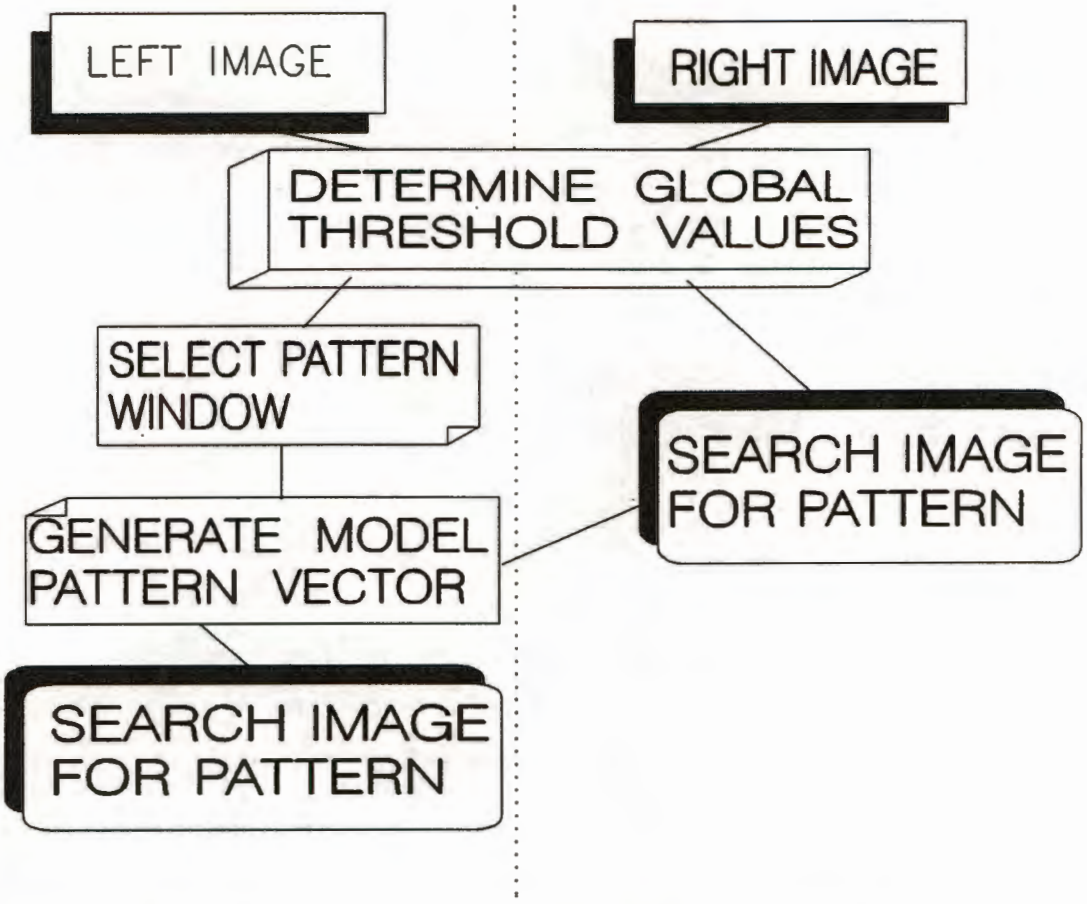


Figure 4.4.1.1.1. *The main steps for locating patterns in images.*

1. The global grey-scale threshold value of the left image is determined. Similarly the grey-scale threshold value of the right image is determined. An edge detector is applied to both images and their respective grey-scale threshold values determined.

2. A window of a selected size is placed around the pattern or object of interest in the left image.

3. Using the global threshold value appropriate to the left image the window is converted into a binary image. The first seven components of the pattern vector are then determined, consisting of the geometric moment invariants given by equations (4.4.1.9) to (4.4.1.15). The original window is restored and the edge

detector applied over the window. The window of edges is again binarized by applying the appropriate global threshold for the left image determined in step 1. The second seven components of the pattern vector are then determined , consisting once again of the moment invariants. Every 14 component pattern vector is determined in this manner.

In order to determine the covariance matrix and a mean vector representation of the pattern, a sample of pattern vectors have to be extracted. This sample is generated by extracting pattern vectors for successive windows of the same size from the neighbourhood of the pattern of interest. Details of this will be given later. A covariance matrix is determined from the sample of pattern vectors.

4. The left and right images can now be searched by applying the pattern vector extraction procedure in step 3 over a window moving across the images. The distance between the model pattern vector and the extracted vector is used as a measure of similarity. The smaller this value the greater the similarity. A distance threshold value is determined based on the covariance matrix. If the distance is less than this threshold then the point centred on the window is taken to represent the centre of the region with a similar pattern structure.

Details of the above steps are presented below with an initial investigation of the performance of a corner pattern locator or interest operator.

4.4.1.2 PREPROCESSING OF IMAGES

As mentioned above the elements of the feature vectors suggested here consist of the seven moment invariants, given by equations (4.4.1.9) to (4.4.1.15), of the thresholded image pattern , and the seven invariant moments of the thresholded edge pattern. The feature vector therefore has fourteen components. Before these components are calculated the selected image window

has to be converted into a binary image using a suitable threshold value. The threshold value used is the global value determined by applying the method suggested by *Otsu, 1979*, to the whole image.

The thresholding method is based on minimizing the within-group variance of the grey-scale frequency histogram P(i) of the image, where i is the grey-scale value. The within group variance σ_W is determined as:-

$$\sigma_W^2(t) = q_1(t)\sigma_1(t)^2 + q_2(t)\sigma_2(t)^2 \qquad \textbf{(4.4.1.2.1)}$$

where

t is the threshold value.

$$q_1(t) = \sum_{i=0}^{t} P(i)$$

$$q_2(t) = \sum_{i=t+1}^{l} P(i)$$

$$\mu_1(t) = [\sum_{i=0}^{t} iP(i)]/q_1(t)$$

$$\mu_2(t) = [\sum_{i=t+1}^{l} iP(i)]/q_2(t)$$

$$\sigma_1(t)^2 = [\sum_{i=0}^{t} [i - \mu_1(t)]^2 P(i)]/q_1(t)$$

$$\sigma_2(t)^2 = [\sum_{i=t+1}^{l} [i - \mu_2(t)]^2 P(i)]/q_2(t)$$

P(i) is the grey-scale histogram function where i is the grey-scale value.

l is the maximum grey-scale value. (l=255 in this investigation)

The threshold value t is taken as the value of i that results in the minimum value of σ_W .It is found by simply calculating

σ_W for all values of the grey-scale and taking the minimum.

Global threshold values are determined for a specified image using the above method.

A selected grey-scale image window is thresholded using the global threshold values determined and the moments of this silhouette image are extracted to give the first seven components of the feature vector.

An edge detector, generally known as the Sobel operator, was used to convert the grey-scale image into an intensity map of the grey-scale gradients in the image. The Sobel operator works by convolving a 3 x 3 pixel mask with the digital image function. The masks for the x and y gradients are shown in figure 4.4.1.2.1.

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

Figure 4.4.1.2.1. *Gradient Masks for Sobel operator (a) y gradient (b) x gradient.*

The convolution can be represented as follows in discrete form

$$G(x_0,y_0) = \sum_{j=-1}^{1}\sum_{i=-1}^{1} M(i,j)f(x_0+i,y_0+j) \qquad \textbf{(4.4.1.2.2)}$$

where

$G(x_0,y_0)$ is the gradient in either the x or y direction at point (x_0,y_0).

f(x,y) is the grey-scale image function.

M(i,j) is the value of the Sobel mask at point (i,j) in the mask assuming the centre of the mask as the origin.

The Otsu thresholding technique is then applied to the gradient intensity image to determine a global threshold value for the gradient image.

By thresholding the gradient image of the selected window the second seven invariant moment functions are extracted to complete the feature vector.

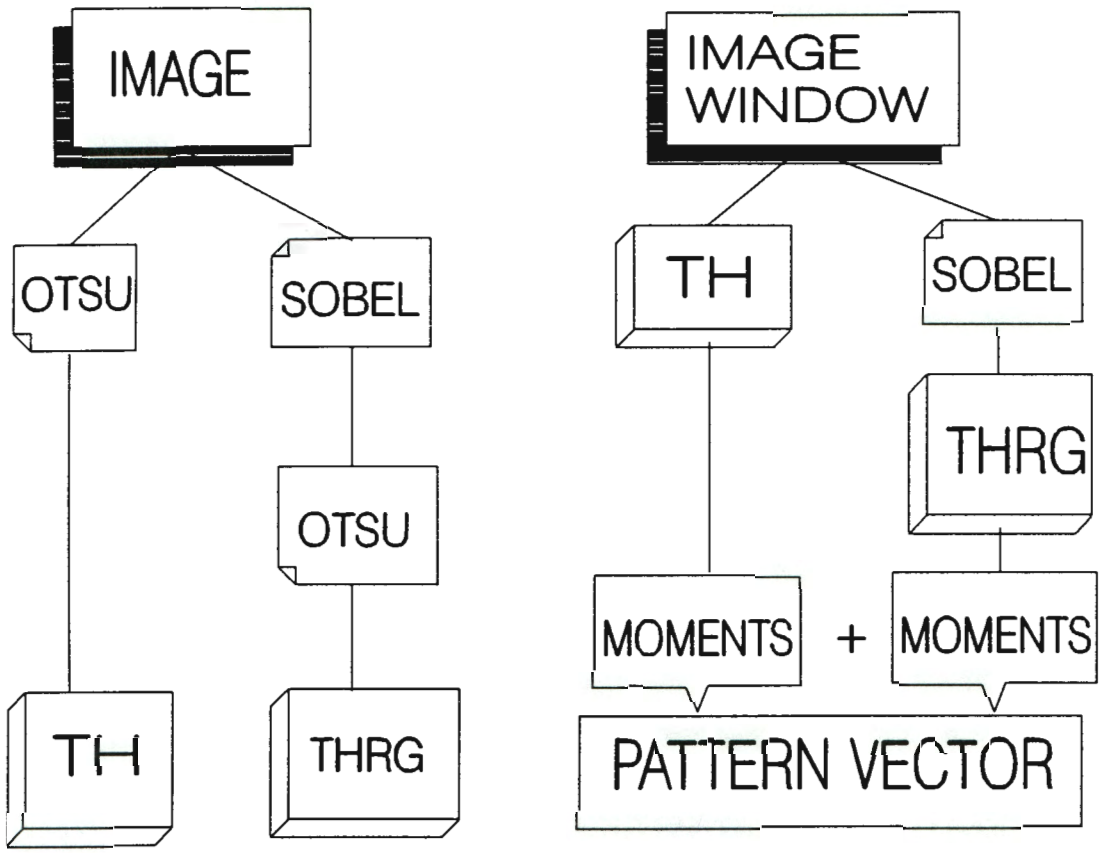


Figure 4.4.1.2.2. *Image processing required to extract pattern vector. Global threshold values determined in the left diagram and applied in the right diagram.*

4.4.1.3 APPLICATION OF A MOMENT INVARIANT BASED INTEREST OPERATOR

The synthetic image shown in figure 4.4.1.3.1 was used to test the interest operator, based on the pattern vector approach described above, as a corner detector. The image is a 50 x 50 pixel image with a 20 x 20 pixel square.

The upper left corner of the square in the image was used as the model pattern. The corner point is positioned at (4,4) in a 9 X 9 window with the extreme top left hand point of the window acting as the origin of a cartesian coordinate system with x increasing to the right and y increasing downwards.

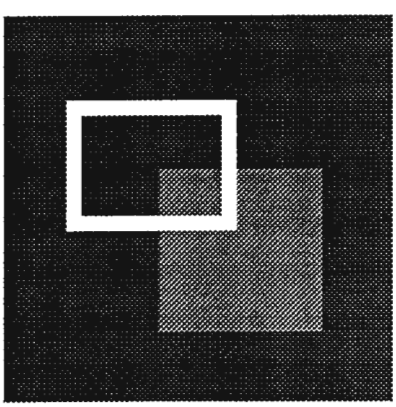


Figure 4.4.1.3.1. *Test image.*

The 14 component moment invariant vector was reduced to a 9 component vector. The 5

moment invariants of the segmented window consisting of third order central moments are excluded due to the four-fold symmetry. The third order central moments would be zero in this case. The situation is depicted in figure 4.4.1.3.3. The centroid of the corner in the window falls right in the centre of the square leading to the 4-fold symmetry.

The edge image of the corner has the centroid as shown in figure 4.4.1.3.4. In this case the third order moments will be non-zero.

In order to extract a representative vector for a corner an n X n window is placed around the corner with the corner at the centre of the window. The global threshold value of the whole image is used to segment the window into a binary image.

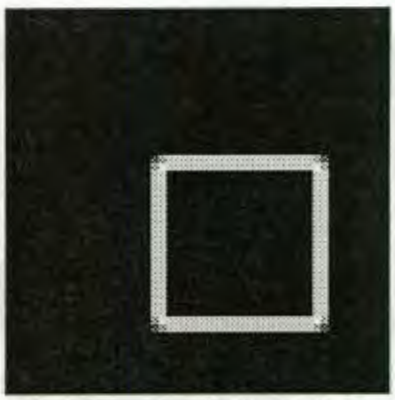


Figure 4.4.1.3.2. *Edge image of figure 4.4.1.3.1.*

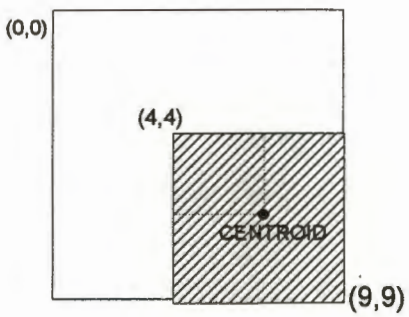


Figure 4.4.1.3.3. *9 X 9 window surrounding a corner point of image in Figure 4.4.1.3.1.*

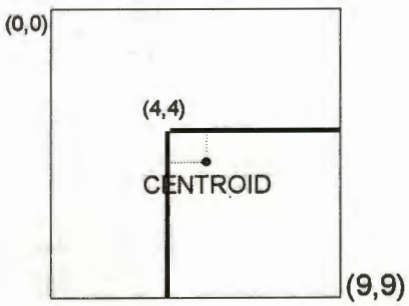


Figure 4.4.1.3.4. *9 X 9 window surrounding a corner point of image in Figure 4.4.1.3.2.*

The first seven invariants of the vector are calculated from the moments of this binary window. Similarly the global threshold of the whole edge image is used in order to extract the second seven invariants of the feature vector. As mentioned above only the first two of the first seven invariants extracted are used due to the symmetry of the corner.

The above process of placing a window around a specific point in the image and performing the extraction operations in this window ,as described above, will be referred to as the feature extraction operator.

The method can be applied to locate similar patterns in any number of images other than the image from which the pattern was extracted. As there is a likely possibility that a scale difference exists between the original image , from which the pattern was extracted, and any other image, the search window size has to be adjusted to allow for the scale difference. In pattern recognition applications of moment invariants the background to a located object is usually uniform. Examples are printed characters on paper, ships at sea and aircraft in the sky or on a runway. In these cases the background will not contribute significantly to the moments calculated especially in a thresholded image and true scale invariant moments can be determined. In the case of photogrammetric images the backgrounds can be rich in a variety of features and these will contribute to the calculated moments if same size windows are used to search images which have significantly different scales. The adjustment required would be the ratio of the scales. If w_l is the side length of the window used for extracting the model pattern in the left image which has a scale S_l , then w_r the side length of the window used in the right image will be given by

$$w_r = \frac{S_r}{S_l} w_l \qquad \textbf{(4.4.1.3.1)}$$

In the following text it is assumed that this window size adjustment is made prior to searching the right image if the scale difference between the two images is significant.

Once the representative pattern feature vector is extracted a nearest neighbour approach based on a distance metric is used to find similar patterns in an image. There are different approaches to determining a suitable distance metric depending on how the

sample data is processed and also on the sample selection procedure.Sample selection will be discussed later in the text.

If the *Mahalanobis distance* defined in equation (4.3.1) is used as a distance metric ,the inverse of the covariance matrix of the sample needs to be determined. The covariance matrix is calculated as

$$\Sigma_{ij} = \sum_{l=1}^{n} (\underline{v}_l(i) - \underline{m}(i))(\underline{v}_l(j) - \underline{m}(j))/n \qquad \textbf{(4.4.1.3.2)}$$

where

Σ_{ij} is the element of the covariance matrix in the ith row and jth column

$\underline{v}_l(i)$, $\underline{v}_l(j)$ are ith and jth components of the lth vector sample respectively.

$m(i) = \sum_{l=1}^{n} \underline{v}_l(i)/n$ is the sample mean value of the ith component.

n is the number of vectors in a sample.

The *Mahalanobis distance* defined as follows

$$d(\underline{v}_p)^2 = (\underline{v}_p - \underline{m}_p)^T \Sigma^{-1} (\underline{v}_p - \underline{m}_p)$$

requires the calculation of the inverse of the covariance matrix. Rather than calculate the inverse directly the covariance matrix is diagonalized so that

$$\Sigma\Phi = \Lambda\Phi \qquad \textbf{(4.4.1.3.3)}$$

where

Λ is the principal diagonal eigenvalue λ_i matrix of the covariance matrix.

Φ is the matrix whose columns are the eigenvectors of the covariance matrix.

Diagonalization facilitates the calculation of the *Mahalanobis distance* as will been shown below.

The eigenvectors form the basis of an orthogonal vector space and satisfy the orthonormality condition

$$\Phi^T\Phi = I \qquad \textbf{(4.4.1.3.4)}$$

In order to calculate the distance in the orthogonal space in terms of components which are not correlated the feature vectors are transformed into the orthogonal space by

$$\underline{Y} = \Phi^T \underline{v} \qquad \textbf{(4.4.1.3.5)}$$

where $\underline{Y}$ is the transformed vector.

In this space the covariance matrix becomes

$$\Phi^T\Sigma\Phi = \Lambda \qquad \textbf{(4.4.1.3.6)}$$

which is the diagonal eigenvalue matrix.

The inverse of the covariance matrix is then just Λ^{-1} in the orthogonal space.

The eigenvalues and eigenvectors of the covariance matrix are obtained using the method developed by C.G.J. Jacobi and can be found in *[FORTRAN Scientific Subroutine Library, Peerless Engineering Services, John Wiley & Sons 1984].*

The eigenvalues of the covariance matrix are the sample variances of the components defined by the eigenvectors. The error ellipse defined by its eigenvalues in a two component space is depicted in figure 6.

The error ellipse in the eigenvector space can be transformed into a circle of radius 1 by the *whitening transformation*. This is achieved by adding the further transformation $\Lambda^{-1/2}$ so that

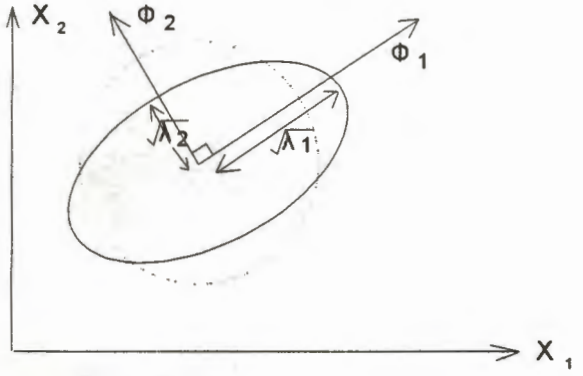


Figure 4.4.1.3.5.
——— Original Distribution .
- - - - Whitened Distribution.

$$\underline{Y} = \Lambda^{-1/2}\Phi^T \underline{V_p} \qquad \textbf{(4.4.1.3.6)}$$

The covariance matrix in the transformed space then becomes

$$\Sigma_Y = \Lambda^{-1/2}\Phi^T\Sigma\Phi\Lambda^{-1/2} = \Lambda^{-1/2}\Lambda\Lambda^{-1/2} = I \qquad \textbf{(4.4.1.3.7)}$$

Euclidean distances in the transformed space will then transform into the *Mahalanobis distance* in the original feature space as follows:-

$$\|\underline{Y}\| = \underline{Y}^T\underline{Y} = \underline{V_p}^T\Phi\Lambda^{-1}\Phi^T\underline{V_p} = \underline{V_p}^T\Sigma^{-1}\underline{V_p} \qquad \textbf{(4.4.1.3.8)}$$

More detail on the derivation of the above relations can be found in *[Fukunaga, 1990].*

The feature extraction operator is applied over the whole image sequentially from top to bottom moving from left to right. As

part of the extraction process the feature vector is transformed by the *whitening transformation* as described above and the *Euclidean distance* between the transformed sample vector and the transformed extracted vector is calculated.

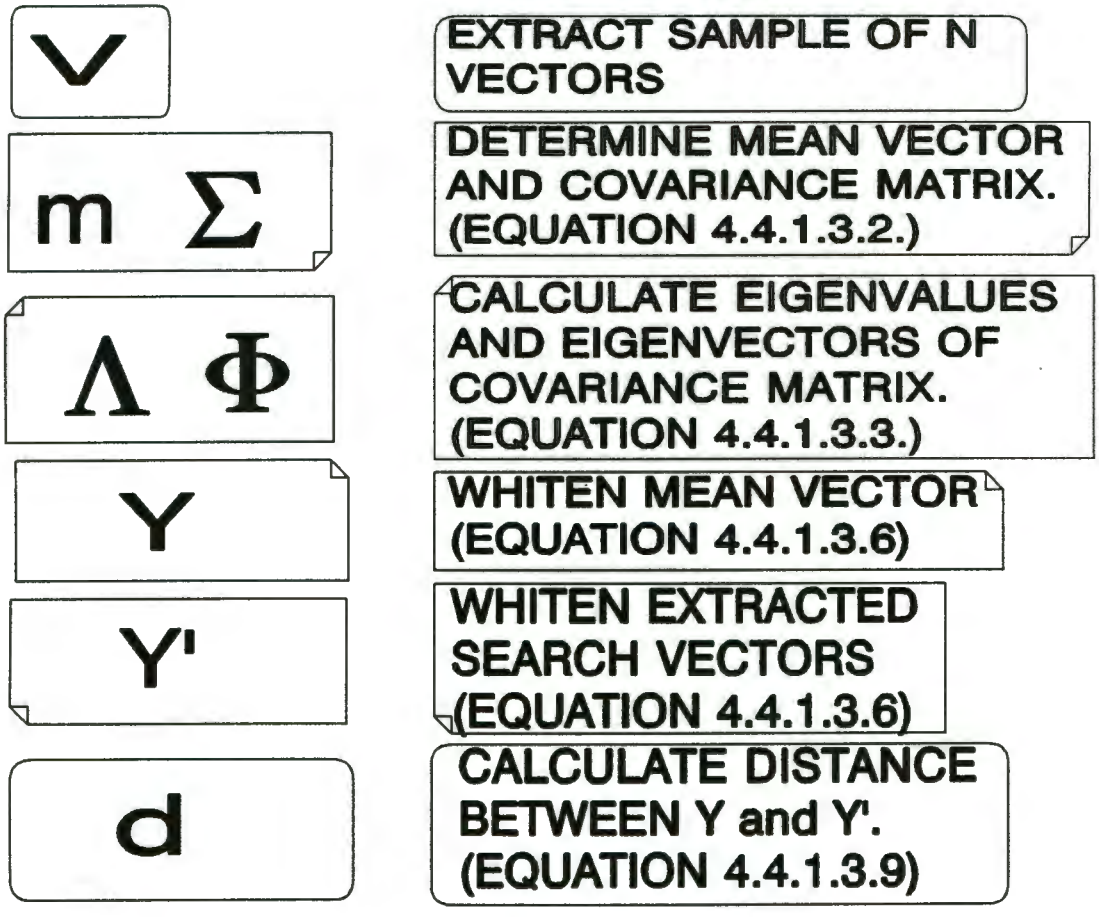


Figure 4.4.1.3.6. *Steps required in arriving at distance between mean sample vector and any extracted vector.*

The main steps required in finding the distance between a model or mean vector and any subsequently extracted pattern vector are shown in figure 4.4.1.3.6. As the model or mean vector is determined from a training sample extracted from the image, the eigenvalues and eigenvectors are determined only once at the start of the procedure. The model vector is whitened and used in this form for comparison with subsequently extracted vectors when images are searched. The comparison is based on the distance given in component form by

$$d^2 = \sum_{i=1}^{N} \frac{(Y(i)-Y'(i))^2}{\lambda_i} \qquad \textbf{(4.4.1.3.9)}$$

Having obtained the distances between the desired pattern vector and the pattern vectors from the image , a method for classifying the pattern vectors on the basis of these distances is required. A distance threshold is set so that any vectors with distances less than this threshold will be classified as belonging to the required pattern class. Patterns more distant than this threshold will be excluded.

The threshold distance is calculated from the initial calibration or training sample selected as follows

$$Threshold = \overline{dist} \pm \alpha\sigma_d \qquad \textbf{(4.4.1.3.10)}$$

where

$\overline{dist}$ is the mean distance of the sample vectors from the sample mean vector.

σ_d is the standard deviation of the mean sample distance.

α is the multiple which is determined heuristically.

As mentioned before, selecting a suitable sample to determine the reference or mean pattern feature vector is a function of the desired objective of the pattern classifier. If it is only necessary to find interest points similar to the required pattern then a fairly crude sample selection can be used. For locating patterns to subpixel precision requires a more sophisticated sampling procedure.

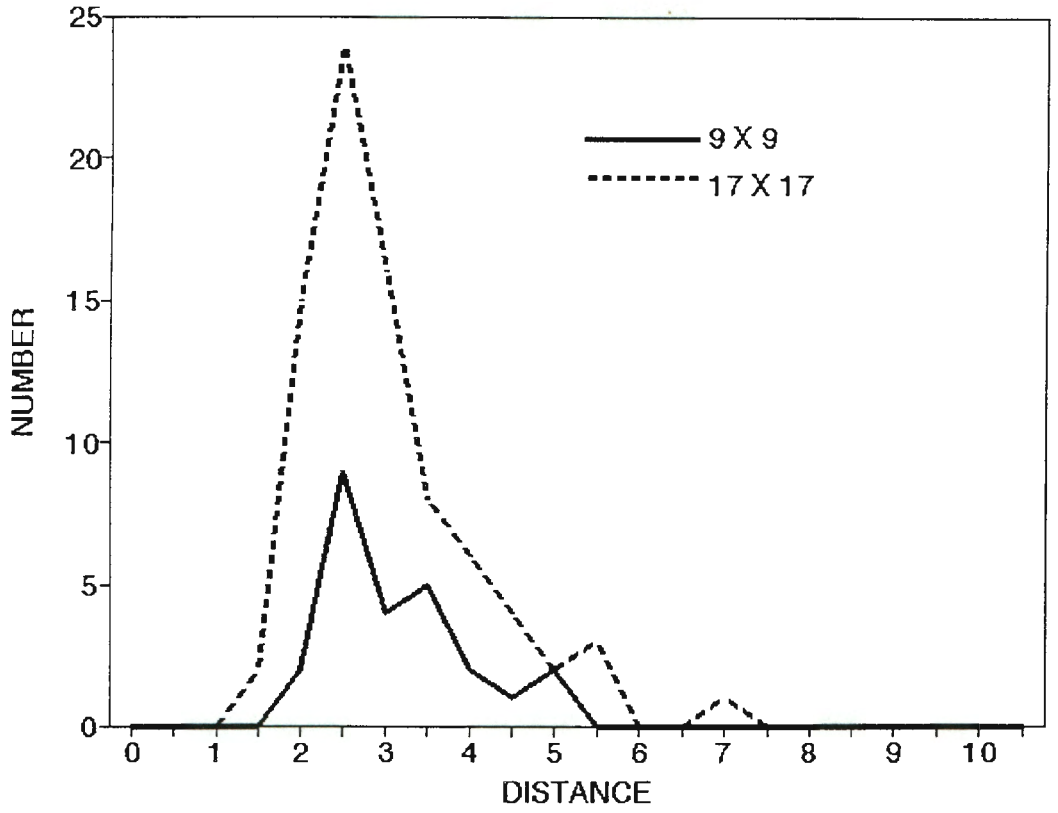
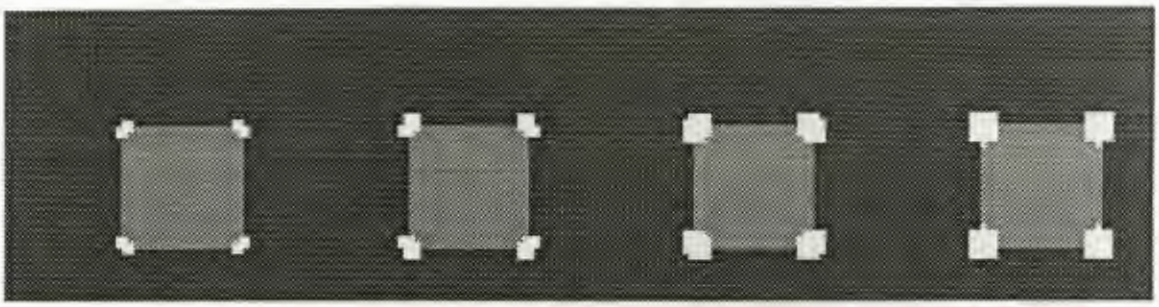


Figure 4.4.1.3.7. *Distribution of distance for different size window samples.*

A crude sample can be obtained by extracting vectors in the image neighbourhood of the required image pattern. If the window size is w X w then a sample can be obtained by extracting vectors by moving w/4 above and below and w/4 left and right of the centre of the desired pattern giving (integer arithmetic) $(2*w/4+1)^2$ vectors, e.g. if w=9 then there will be 25 vectors in the sample. (-w/4, w/4) was chosen as the neighbourhood sample selection range as the patterns should not change significantly in each window. Figure 4.4.1.3.7 shows the distribution of the distance of the sample vectors from the sample mean vector for samples taken around the top left corner point of the test image in figure 4.4.1.3.1. There are 25 vectors in the 9 X 9 window sample. There are 81 vectors in the 17 X 17 window sample.

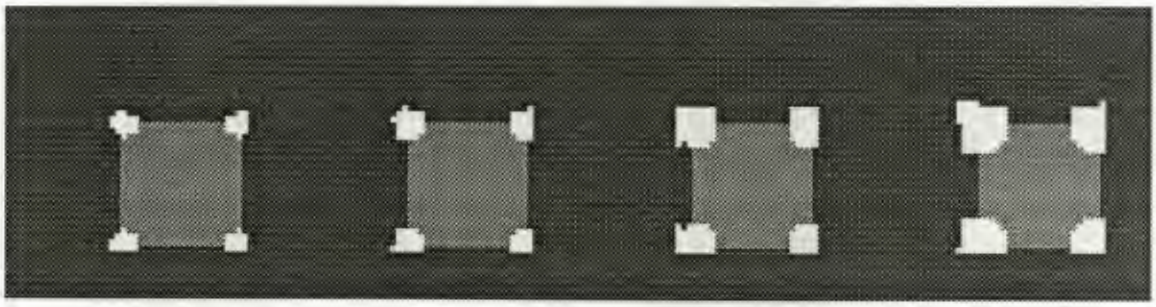
The standard deviation and the mean distance of the sample

distance distribution is used in determining the selection threshold in equation (4.4.1.3.10) for classification purposes.



α=0.1 α=1.0 α=3.0 α=10.0

Figure 4.4.1.3.8. *Nearest neighbour points for different values of α in equation (4.4.1.3.10). 9 X 9 window sizes were used.*



α=0.1 α=1.0 α=3.0 α=10.0

Figure 4.4.1.3.9. *Nearest neighbour points for different values of α in equation (4.4.1.3.10). 15 X 15 window sizes were used.*

Figures 4.4.1.3.8 and 4.4.1.3.9 show the results of using various values of α and different window sizes. As the threshold is increased more points are located as would be expected. As the window size is increased more points are located as the pattern range is increased i.e. the relative change in the pattern vector is less per pixel moved for the 15 x 15 window than for the 9 x 9 window. Three of the point clumps located using 15 x 15 windows are truncated due to the window reaching the boundary of the image.

In order to establish the point centred on the window with the closest pattern vector to the sample mean vector or model pattern vector, the nearest neighbour point in any particular clump of points needs to be found. A filtering process is applied. {EP} is the set of extracted points after application of the interest operator over the whole image.{NN} is the set of nearest neighbour points.

$$[\ \forall\ \underline{p}, \underline{q} \in \{EP\}\ ]\ \wedge\ [\ \|\underline{p}-\underline{q}\| < N\ ]$$

$$\underline{p} \in \{NN\}\ if\ d_p < d_q$$

where

$\underline{p}$ and $\underline{q}$ are vectors describing the position of point in the image.

$d_{\underline{p}}$ is the distance in feature space between the pattern vector associated with point $\underline{p}$ and the mean sample vector.

$\|...\|$ indicates the Euclidean distance in image space.

N is a suitably chosen neighbourhood range.

The isolation of the nearest neighbour pixel points are shown in figure 4.4.1.3.10, using a neighbourhood range of 5 pixels. The corner points are well identified in this simple test case.

An example of applying this technique to isolating points in real images is shown in figure 4.4.1.3.11. The same building is taken from a stereopair and the interest operator with w=9 and α=1 is applied. Although the corresponding point on which the classifier was trained i.e. the left bottom corner of the left building is identified in both images, some corner points have been missed.

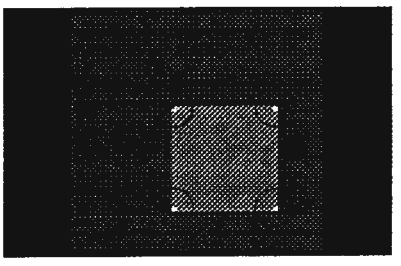
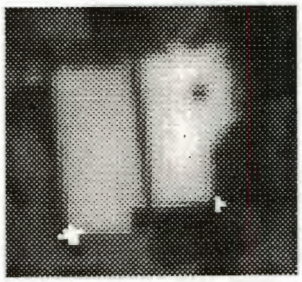


Figure 4.4.1.3.10. *Isolated nearest neighbour points.(9 X 9 window, α=1.0).*



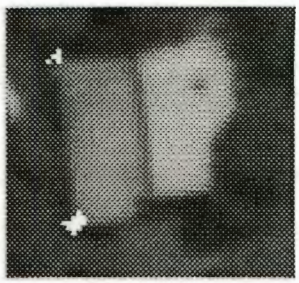


Figure 4.4.1.3.11. *Interest clumps identified using interest operator. (window 9X9 ,α=1.0)*

In order to improve the ability of the operator to locate more corners different sampling procedures can be used. One method that was tried in this investigation entails transforming the window surrounding the interest point by scaling and rotation transformations. During this process the centre of the window remains fixed. A window size with lengths of twice the interest operator window length w is transformed and then the feature extractor operator is applied to each transformed window in order to obtain the sample vectors. The transformation function is given by equation (4.4.1.3.11).

As the transformation leads to some pixels in the transformed image having no intensity a resampling procedure to fill in these values is required. A bilinear interpolation method was used by applying the inverse transformation given by equation (4.4.1.3.12). This equation transforms the point back into the original image space to a non integral pixel coordinate point. The pixel intensity values are calculated by interpolation between the four nearest integral pixel positions by equation (4.4.1.3.13).

$$(x' - x_0') = \alpha((x - x_0)\cos\theta - (y - y_0)\sin\theta)$$

(4.4.1.3.11)

$$(y' - y_0') = \alpha((x - x_0)\sin\theta + (y - y_0)\cos\theta)$$

$$(x - x_0) = \frac{1}{\alpha}((x' - x_0')\cos\theta + (y' - y_0')\sin\theta)$$

$$(y - y_0) = \frac{1}{\alpha}(-(x' - x_0')\sin\theta + (y' - y_0')\cos\theta)$$

(4.4.1.3.12)

where
(x,y) are the pixel coordinates of a point in the original image.
(x',y') are the pixel coordinates of a point in the transformed image.
(x_0,y_0) are the pixel coordinates of the point in the centre of the window in the original image.
(x_0',y_0') are the pixel coordinates of the point in the centre of the window in the transformed image.
α is the scaling factor in the range [0.8,1.2] so as not to alter the image too dramatically.
Θ is the angle through which the original image window is rotated.

$$g(x,y) = (1 - y + j)[(1 - x + i)g(i,j) + (x - i)g(i+1,j)]$$
$$+ (y - j)[(1 - x + i)g(i,j+1) + (x - i)g(i+1,j+1)]$$

(4.4.1.3.13)

where
g(x,y) is the grey scale value at a non integral pixel position (x,y) in the original image.
g(i,j) is the grey scale value at the integral pixel position (i,j).
i = truncated value of x.
j = truncated value of y.

Similarly sample vectors can be generated by assuming an affine transformation between the two images. The affine transformation is defined by the following relationships

$$(x' - x_0') = a_1(x - x_0) + a_2(y - y_0)$$

$$(y' - y_0') = b_1(x - x_0) + b_2(y - y_0) \qquad \textbf{(4.4.1.3.14)}$$

where the coefficients a_1 , a_2 , b_1 , b_2 integrate the scaling , rotation and shearing aspects of the transformation.

As with the previous method the transformation leads to pixel positions with no intensity in the transformed image. A bilinear resampling procedure is used to calculate these missing intensities by transforming back into the original image using the following inverse transformation:-

$$(x - x_0) = (b_2(x' - x_0') - a_2(y' - y_0')) / (a_1b_2 - a_2b_1)$$

$$(y - y_0) = (a_1(y' - y_0') - b_1(x' - x_0')) / (a_1b_2 - a_2b_1) \qquad \textbf{(4.4.1.3.15)}$$

Experimentation with these sampling methods shows that there is no real advantage gained by the additional computation overhead incurred when using the transformation sample generation methods. For all subsequent experiments the neighbourhood sampling method is used. In addition the slight perspective variations in a real image and the presence of different shadow areas can change the moment structure of a window significantly and render the excess computation superfluous.

4.4.1.4 OBJECT LOCALIZATION

A similar approach to that described above for interest operators can be used to locate the conjugate image position of an object, selected in one image, in another image. The window chosen needs to include a complete object such as a building. A sampling procedure is then used to build up the pattern vector.

the selected object. A nearest neighbour rule for the distance in pattern space is used as a decision function. Additional information, such as the expected disparity between the two images, can be used to disambiguate multiple locations. An approximate value for the expected disparity can be obtained from direct inspection of the images or by interpolation in a disparity map derived from a continuous relaxation matching of corresponding interest points between two images.

4.4.1.5 QUANTIZATION ERROR IN MOMENT CALCULATIONS

Due to the discrete nature of the digital images discrete summations are usually used to evaluate the moment integrals. *Teh and Chin, 1986*, investigated the errors introduced by discrete calculations and compared these with numerical integration techniques such as Simpson's rule. They found that numerical techniques reduced the quantization errors significantly.

The geometric moment invariant operator described in section 4.4.1.3 above was calculated using the simple discrete summation method and the numerical method. A comparison was made between the two methods against varying window sizes centred at the top left hand corner of the square in figure 4.4.1.3.1. Figure 4.4.1.5.1 shows the results for three different window sizes. Only the maximum % difference is plotted. As the window size increases the % difference becomes smaller. This is expected as the weight of the information content carried by each pixel is reduced as the number of pixels increase.

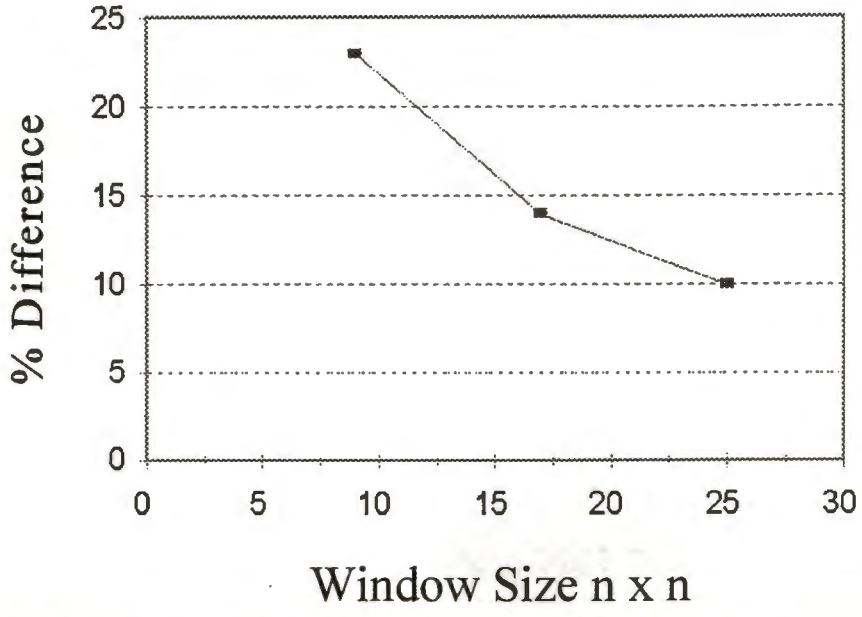


Figure 4.4.1.5.1. *% Maximum difference between discrete and numerical moment invariant calculations against window size.*

One clear advantage of using the numerical integration technique is that it appears to smooth out high frequency noise components in the pixel data. The discrete technique is computationally faster but appears to be less robust especially for small windows (n X n with n < 20).

4.4.1.6 NOISE SENSITIVITY OF GEOMETRIC MOMENT INVARIANT OPERATOR

As is evident in figure 4.4.1.3.11 the quality of all real images are degraded by noise. The noise content of the image will hamper the ability of any pattern recognition operator.

In order to gain some insight into the sensitivity of the geometric moment invariant technique to noise, white noise was added to the ideal image in figure 4.4.1.3.1 and the performance of the operator assessed for different levels of noise. White noise has a Gaussian

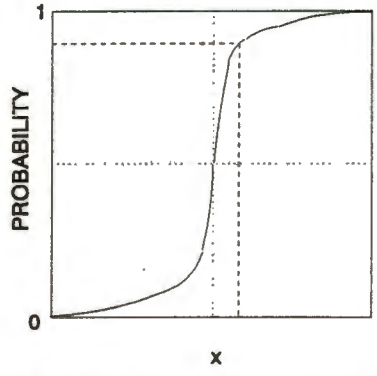


Figure 4.4.1.6.1. *Cumulative normal distribution used to relate probability to noise value x.*

amplitude distribution at any particular spatial frequency and a uniform spatial frequency distribution.

The probability density function for white zero-mean noise has the form of a Gaussian density function $\exp(-\frac{x^2}{2\sigma^2})$, where x is the noise signal level and sigma the standard deviation of the distribution. White noise has the same distribution for all spatial frequencies.

To simulate the noise in a synthetic image a Monte-Carlo technique based on pseudo-random numbers was used. A normalized cumulative distribution function was calculated with the zero mean probability density taken as

$$p(x) = A \exp(-\frac{x^2}{\sigma^2}) \qquad \textbf{(4.4.1.6.1)}$$

where

p(x) is the probability density at the grey-scale value x.

σ is the standard deviation of the distribution.

A is a normalization constant so that the cumulative distribution will sum to 1 at the upper limit of the range in x chosen. A look up table (LUT) was generated for the relationship between the x values and the cumulative probability. The LUT expresses the relationship shown in figure 4.4.1.6.1 at discrete values of x. By scanning the image at all possible spatial frequencies each pixel value in the image is randomly adjusted. The adjustment is determined using a pseudo-random number generator to select any probability value between 0 and 1 and the corresponding x value found from the LUT.

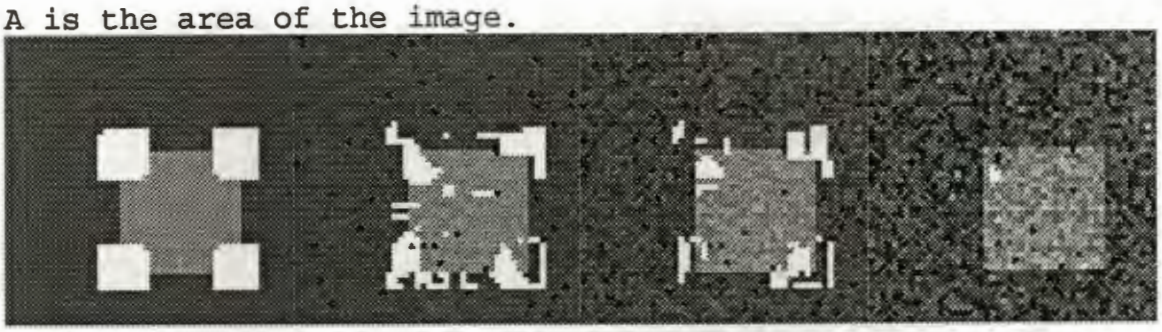
Noise was added to the image in figure 4.4.1.3.1 with predetermined signal-to-noise ratios and the moment invariant interest operator applied to these images. The results of this experiment are shown in figure 4.4.1.6.2. In this context the signal-to-noise ratio is defined as *[Teh and Chin, 1988]*

$$SNR = \frac{\frac{1}{A}\iint [f(x,y)]^2 dxdy}{\sigma^2} \qquad \textbf{(4.4.1.6.2)}$$

where
f(x,y) is the two dimensional grey-scale description of the image under consideration.

σ is the standard deviation of the noise distribution.

A is the area of the image.



SNR=∞ SNR=200 SNR=30 SNR=7

Figure 4.4.1.6.2. *Moment based interest operator performance for different Signal-to-Noise ratios (9 x 9 window, α=0.2).*

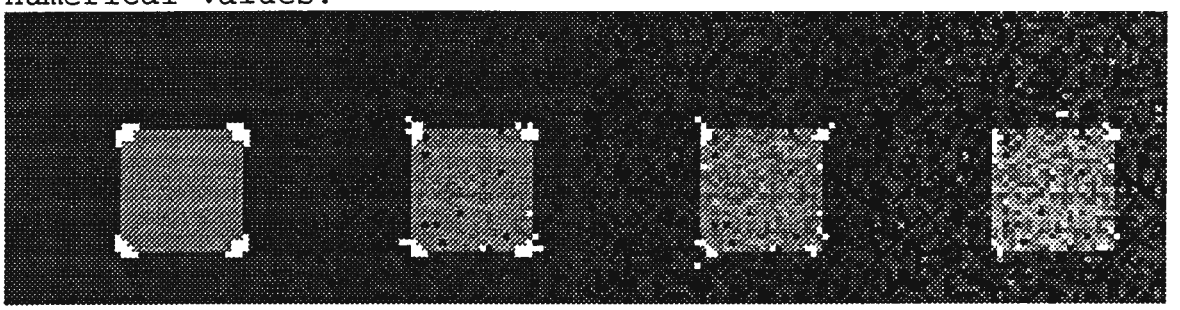
The results in figure 4.4.1.6.2 show that the geometric invariants are fairly robust in locating correct pattern regions for SNR >= 30. Most real images that are useable in photogrammetric applications have SNRs greater than 30.

4.4.1.7 APPLYING GEOMETRIC MOMENTS DIRECTLY TO GREY-SCALE IMAGES

The method described above of applying geometric invariants to the thresholded windows can be applied directly to the grey-scale images. The thresholding method above effectively converts the grey-scale image into a binary image. As the radiometric variations in some images can be quite diverse this will lead to loss of valuable information contained in the grey-scale values.

By omitting the preprocessing of the windows and calculating the moments directly on the grey-scale window a pattern vector of the seven Hu invariants is obtained. Intuitively it would be expected that the silhouette and edge information are integrated

into this descriptor as they are extracted from the same data. The grey-scale values are normalized by dividing by the maximum pixel intensity viz. 255 in this case. This ensures well-behaved numerical values.



SNR=∞ SNR=200 SNR=30 SNR=7

Figure 4.4.1.7.1. *Moment based interest operator performance for different Signal-to-Noise ratios (9 x 9 window, α=0.2).*

The results of applying this operator are shown in figure 4.4.1.7.1. The corners are well located for SNR >=30. This method can be used in place of the binary technique as there is less computation required and is consequently about twice as fast.

4.4.2.ORTHOGONAL MOMENT INVARIANTS.

Numerical features other than geometric invariants can be extracted from images and included as components of a feature vector. Some of these may be more efficient as descriptors or can be used as additional components in a feature vector. Analogous to geometric moments other moments have been defined and used in pattern recognition. The suitability of these moments to object location in images and hence for image matching are investigated.

Teague, 1980, asserts that a two-dimensional grey-scale image function f(x,y) can be expressed in terms of an infinite series involving monomial expressions of the form $x^n y^m$.

$$(x,y) = f_{00} + f_{10}x + f_{01}y + f_{20}x^2 + f_{11}xy + \ldots \qquad \textbf{(4.4.2.1)}$$

With the dimensions of the rectangular image window normalized to the range [-1,1] the moments can be expressed as

$$\int_{-1}^{1} dx \int_{-1}^{1} dy f(x,y)\, x^j y^k = M_{jk} \qquad \textbf{(4.4.2.2)}$$

In order to get a representation of the image by means of the series expansion in (4.4.2.1), the series is truncated at a chosen term and the coefficients are found in terms of the moments of the image by solving a set of coupled simultaneous equations. The equations are generated by calculating the moments of equation (4.4.2.2) by multiplying equation (4.4.2.1) by the appropriate monomial and integrating both sides as shown in

$$M_{jk} = \int_{-1}^{1}\int_{-1}^{1} f(x,y)\, x^j y^k dx dy = \int_{-1}^{1}\int_{-1}^{1} [f_{00} + f_{10}x \ldots]\, x^j y^k dx dy \qquad \textbf{(4.4.2.3)}$$

equation (4.4.2.3).

Each time more terms are added to the series a full set of coupled equations has to be solved. The number of equations increases substantially as more terms are added. Exact representations of the image are obtainable by means of the expansion in equation (4.4.2.1).

In order to overcome the disadvantages of using geometric moments directly to represent images the series expansion (4.4.2.1) can be expressed in terms of an orthonormal set of polynomials. Polynomials that have been successfully applied for this purpose are the Legendre polynomials and the Zernike polynomials.

4.4.2.1 LEGENDRE MOMENTS

Equation (4.4.2.1) can be rewritten as

$$f(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \lambda_{mn} P_m(x)\, P_n(y) \qquad \textbf{(4.4.2.1.1)}$$

$$\lambda_{mn} = \frac{(2m+1)(2n+1)}{4} \int_{-1}^{1}\int_{-1}^{1} dx dy f(x,y) P_m(x) P_n(y) \qquad \textbf{(4.4.2.1.2)}$$

$$P_n(x) = \frac{1}{2^n} \sum_{m=0}^{n/2} (-1)^m \frac{(2n-2m)!}{m!\,(n-m)!\,(n-2m)!} x^{n-2m} \qquad \textbf{(4.4.2.1.3)}$$

are the Legendre polynomials which satisfy the orthonormality condition

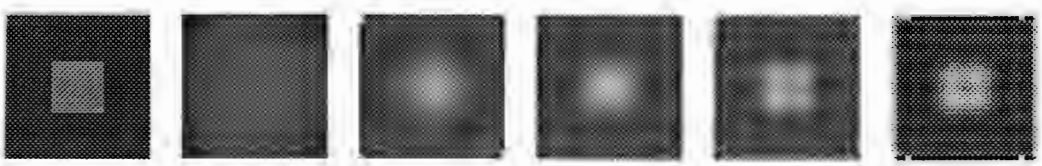
$$\int_{-1}^{1} dx P_m(x) P_{m'}(x) = \frac{2}{2m+1} \delta_{mm'} \qquad \textbf{(4.4.2.1.4)}$$

In practice equation (4.4.2.1.1) is truncated and the representation function is

$$f(x,y) \cong \sum_{N=0}^{N_{max}} \sum_{n=0}^{N} \lambda_{N-n,n} P_{N-n}(x) P_n(y) \qquad \textbf{(4.4.2.1.5)}$$

where N_{max} is the order at which the series is truncated.

When N_{max} is increased the previously calculated coefficients λ_{mn} need not be recalculated due to the orthogonality of the Legendre polynomials.



N_{max} = 4 8 12 16 20

Figure 4.4.2.1.1. *Reconstructed image on the left using equation (4.4.2.1.5) for different values of* N_{max}.

For any selected pattern in an image the legendre moments λ_{mn}

For any selected pattern in an image the legendre moments λ_{mn} can be determined. Simpsons rule is used to determine the double integral of equation (4.4.2.1.2). Figure 4.4.2.1.1 illustrates how equation (4.4.2.1.5) reconstructs an image for various values of N_{max}. As the number of moments included increases the better the reproduction of the original image. Legendre moments can therefore be used as descriptors of patterns in a scene and applied as pattern recognition attributes in order to find similar patterns in other images. The moments can be used as the elements in a feature vector and applied in a similar manner to the geometric moment invariants described before.

Unfortunately Legendre moments are not invariant to scaling and rotation in general. Only images with similar scale and perspective view can be compared. Figure 4.4.2.1.2 shows the Legendre moments up to order 8 of selected image windows. The values marked by ■ and + are between image windows of different size surrounding the corner in figure 4.4.1.3.1. These are all fairly close to each other as would be the case in an exact match. The values earmarked by * show a clear difference between the image area used for ■ and a selected area completely surrounding a square 10 X 10 pixels similar to that in figure 4.4.1.3.1.

Teh and Chin, 1988, show that Legendre moments are very sensitive to noise and not as robust as geometric moments. In addition the computational overhead of determining series of Legendre polynomials makes using this technique unsuitable for PC applications.

Legendre moments can offer an alternative feature for pattern recognition in a similar way to geometric invariants , but only in circumstances where the images are very similar and when

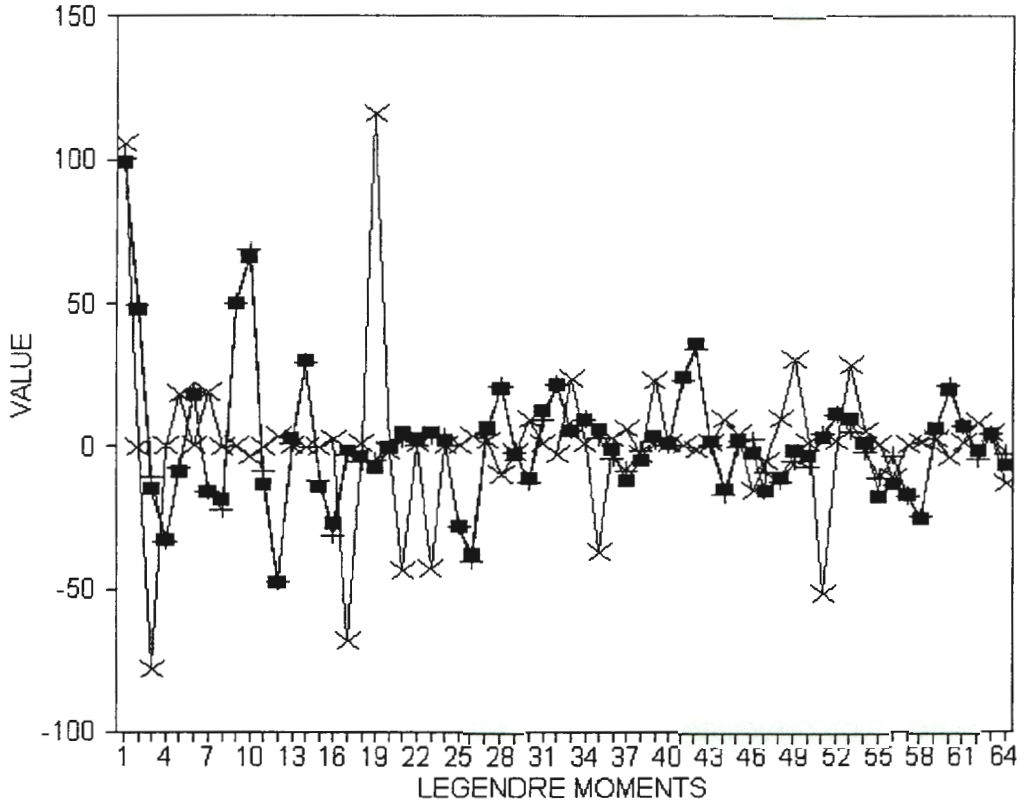


Figure 4.4.2.1.2. *Legendre moments to order 8.*
■ moments for 21 X 21 window.
+ moments of same area as ■ with 31 X 31 window.
X moments of 21 X 21 window around different area.

workstations are available for the computation.

4.4.2.2 ZERNIKE MOMENTS

Zernike moments and invariants were derived by *Teague, 1980*, and applied to image encoding and pattern recognition. These moments use the invariant properties of Zernike polynomials derived by Zernike *[Born and Wolf, 1965]*, inventor of the phase-contrast microscope, and used in the analysis of aberrations in optics.

In practical applications the infinite expansion in (4.4.2.1) is truncated as in (4.4.2.1.5) and can be written in terms of Zernike polynomials as follows *[Teh and Chin, 1988]*

$$f(x,y) = \sum_{n=0}^{N} \sum_{l=-n}^{n} A_{nl} V_{nl}(x,y) \qquad \textbf{(4.4.2.2.1)}$$

where $n - |l| =$ even number and $|l| \leq n$.

$$A_{nl} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^{\infty} [V_{nl}(r,\theta)]^* f(r\cos\theta, r\sin\theta)\, r\, dr\, d\theta \qquad \textbf{(4.4.2.2.2)}$$

$$V_{nl}(x,y) = V_{nl}(r\cos\theta, r\sin\theta) = R_{nl}(r)\, e^{il\theta} \qquad \textbf{(4.4.2.2.3)}$$

$$R_{nl}(r) = \sum_{s=0}^{(n-|l|)/2} (-1)^s \frac{(n-s)!}{s!\left(\frac{n+|l|}{2}-s\right)!\left(\frac{n-|l|}{2}-s\right)!} r^{n-2s} \qquad \textbf{(4.4.2.2.4)}$$

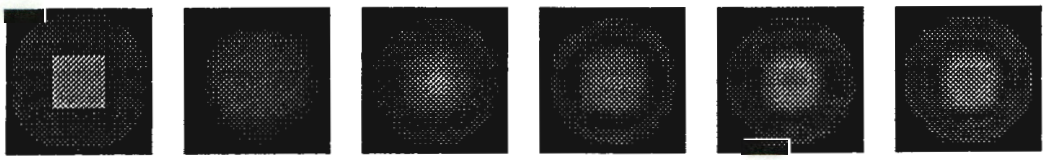
x , y are cartesian coordinates with origin at the centre of the image window chosen. r,θ are the corresponding polar coordinates. Zernike polynomials are complex numbers.

The expansion in (4.4.2.2.1) is carried out with normalized values of x and y so that x and y are in the range [-1,1] over the image window chosen. Correspondingly r is in the range [0,1] and theta in the range [0,2π]. As the expansion is orthonormal over the unit circle only, any points outside of this circle are ignored. The orthonormality condition of the radial function of the Zernike polynomial is given by

$$\int_0^1 R_{nl}(r)\,R_{ml}(r)\,r\,dr = \frac{1}{2(n+1)}\delta_{nm} \qquad \textbf{(4.4.2.2.5)}$$

The integrations in (4.4.2.2.1) are done using Simpson's rule. The terms A_{nl} are known as the Zernike moments and various combinations of these are used to form moment invariants with respect to translation,scale and rotation.

By referring the coordinate system to the centre of the image window translational invariance is introduced as was done by using central moments for geometric moment invariants. The normalization of the coordinates to the unit circle introduces scale invariance. If the coordinate system is rotated by θ then the corresponding Zernike moments are related by $A'_{nl} = A_{nl}e^{-il\theta}$, implying the introduction of a phase factor to the complex moment A_{nl} to obtain the new moment A'_{nl}. Real valued moments A_{nl} remain unaffected by rotation and thus are implicitly rotationally invariant. Elimination of the rotational dependence of complex valued moments can be achieved by suitable combinations of moments and their complex conjugates *[Teague, 1980]*.



N = 4 8 12 16 20

Figure 4.4.2.2.1. *Reconstructed image on the left using equation (4.4.2.2.1) for different values of N.*

Figure 4.4.2.2.1 shows the reconstruction of the image on the extreme left by using the series in (4.4.2.2.1) , truncating after Zernike moment order N. As in the case of the Legendre moments the reconstruction improves with the number of terms included in the series. Zernike moments can therefore be used as descriptors of patterns. In pattern recognition applications the Zernike invariants are used to describe patterns. The first two

invariants are constants and therefore the same for all images , these are $A_{00} = 1/\pi$ and $|A_{11}|^2=0$.Table 4.4.2.2.1 lists the Zernike moment invariants to order 8.*[Teague, 1980]*

Order n	Moment Invariants
2	A_{20} ,, $\|A_{22}\|^2$
3	$A_{33}\|^2$, $\|A_{31}\|^2$, $A_{33}[(A_{31})^*]^3$+c.c., $(A_{31})^2(A_{22})^*$+c.c.
4	$A_{44}\|^2$, $\|A_{42}\|^2$, A_{40}, $(A_{44})^*[A_{42}]^2$+c.c, $A_{42}(A_{22})^*$+c.c.
5	$A_{55}\|^2$, $\|A_{53}\|^2$, $\|A_{51}\|^2$, $(A_{51})^*A_{31}$+c.c, $(A_{53})^*A_{33}$+c.c, $(A_{55})^*(A_{31})^5$+c.c.
6	$A_{66}\|^2$, $\|A_{64}\|^2$, $\|A_{62}\|^2$, A_{60} ,$(A_{66})^*(A_{33})^2$+c.c, $(A_{64})^*A_{44}$+c.c, $(A_{62})^*A_{22}$+c.c.
7	$\|A_{77}\|^2$, $\|A_{75}\|^2$, $\|A_{73}\|^2$, $\|A_{71}\|^2$, $(A_{77})^*(A_{31})^7$+c.c, $(A_{75})^*A_{55}$+c.c, $(A_{73})^*A_{33}$+c.c., $(A_{71})^*A_{31}$+c.c
8	$A_{88}\|^2$, $\|A_{86}\|^2$, $\|A_{84}\|^2$, $\|A_{82}\|^2$, A_{80}, $(A_{88})^*(A_{44})^2$+c.c $(A_{86})^*A_{66}$+c.c, $(A_{84})^*A_{44}$+c.c., $(A_{82})^*A_{22}$+c.c

TABLE 4.4.2.2.1. *Zernike moment invariants up to order 8.* indicates complex conjugate and c.c indicates complex conjugate of previous term.*

In order to show the rotational invariance of the Zernike moment invariants , the moments for the binary images in figure 4.4.2.2.2(a) were determined. The 41 moment invariants to order 8 were calculated and compared. A plot of these is shown in figure 4.4.2.2.2(b). Excepting for two cases the invariants are almost identical. The differences are due to quantization errors as the diagonal lines are more like step functions. The correlation coefficient between the two plotted functions can be used as a matching criteria. The formula for the correlation coefficient is given as

$$\rho = \frac{\sum_{i=1}^{N} f(i)\,g(i)}{\sqrt{\sum_{i=1}^{N} f(i)^2}\sqrt{\sum_{i=1}^{N} g(i)^2}} \qquad \textbf{(4.4.2.2.6)}$$

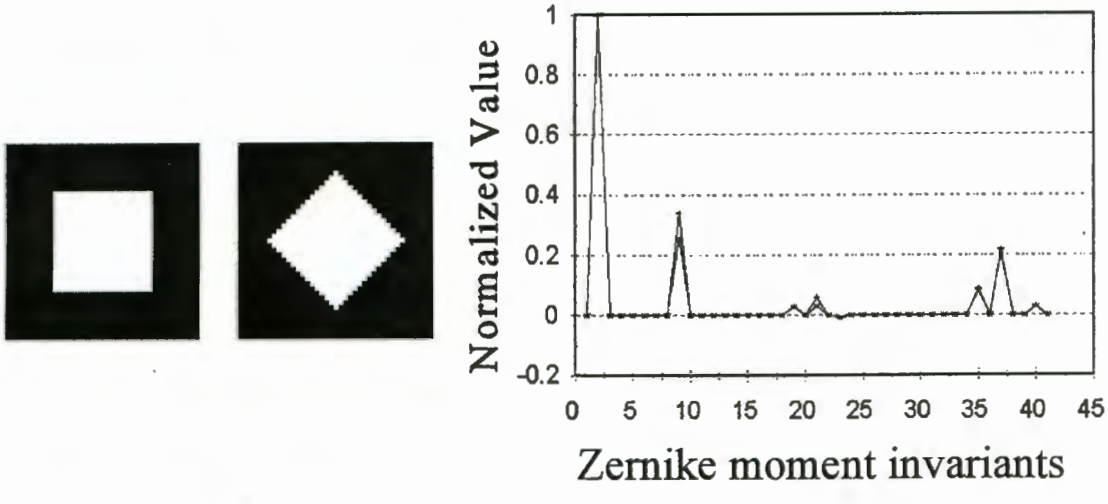
where

ρ is the correlation coefficient.

f(i) and g(i) are the functions being compared.

N is the number of data points.

In the case of figure 4.4.2.2.2 ρ=0.998 indicating almost an identical match. When using the moduli of the moments only to order 8 ,as shown in figure 4.4.2.2.3 , ρ becomes 0.997 which is almost identical.



(a) (b)

Figure 4.4.2.2.2. *Normalized Zernike moment invariants to order 8 for the two images on the left.*

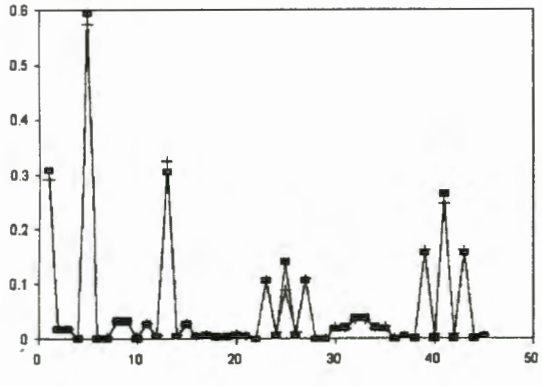


Figure 4.4.2.2.3. *Plot of the moduli of the Zernike moments for the images in figure 4.4.2.2.2(a).*

In order to assess the significance of the window size chosen for identifying patterns,the square region in figure 4.4.2.2.1 with a side length of 10 pixels was scaled up to side length 15 and compared with the smaller square using the same window size. The window size chosen was 31 x 31 so that the larger square would be

completely visible. The result of this comparison was ρ=0.811.

The window size was reduced for the smaller square in proportion to scale to 21 x 21 pixels. The same comparison resulted in ρ=0.998.

The above experiment shows that the window size has to be scaled according to the object size in the image in order to obtain correct matching. Figure 4.4.2.2.4 shows how ρ varies with window size. Although the correlation coefficient reduces it does not go far below 0.7 .

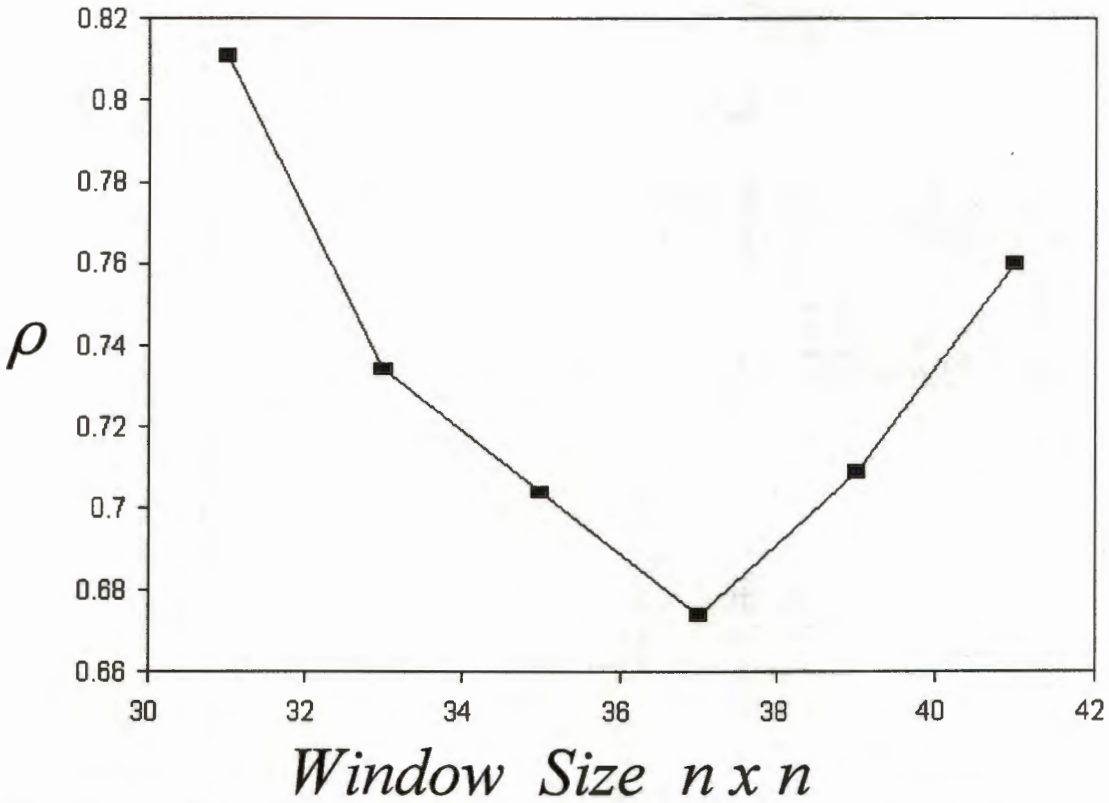
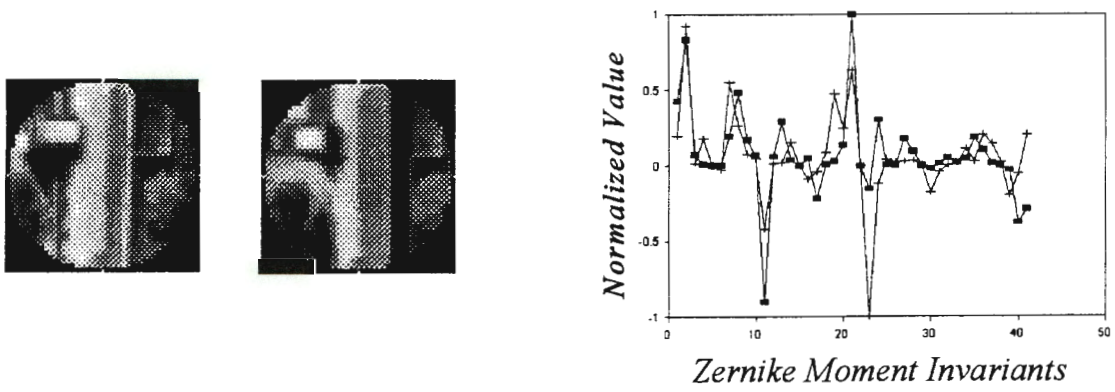


Figure 4.4.2.2.4. *Variation of the correlation coefficient with window size when comparing squares with different pixel sizes (10 x 10 and 15 x 15).*

If only the top corner of the 15 x 15 square is included in the 31 x 31 window and compared with the window surrounding the full square then ρ = 0.335. This indicates that Zernike moment invariants can discriminate between different patterns.

The above indicates that a function consisting of Zernike moment invariants can be used as a pattern descriptor. The correlation coefficient between these descriptors can then be used as a matching criterion.

Figure 4.4.2.2.5 shows the results of a 45 X 45 window taken of the same object from different images. The Zernike invariants are plotted in 4.4.2.2.5(c). They follow a similar pattern but some differ substantially. ρ is only 0.65 in this case and as such is not really a good match. The main reason for this is that the correlation is calculated giving equal weight to each moment invariant. The invariants with large differences will therefore primarily determine the result. A similar argument would apply if the Euclidean distance was used as the matching criteria.



(a) (b) (c)

Figure 4.4.2.2.5. *Plot (c) of the invariants to order 8 of windows (a) and (b) of the same part of a building from different images.*

In order to circumvent this problem a similar approach to that used with the geometric moment invariants can be adopted. A sampling procedure ,or teach-in phase ,to acquire the model invariant vector and its covariance matrix ,with each moment as one component, has to be performed. Diagonalization of the covariance matrix to get the eigenvalues and eigenvectors is then carried out. These eigenvectors and eigenvalues are then used for the whitening transformation on all acquired invariant pattern

vectors. The whitened form of the pattern vectors are then compared by using Euclidean distance.

As at least Zernike moments to order 8 are required for adequate discrimination large covariance matrices need to be diagonalized. The procedure described above is not really suitable for a PC application as considerable computation power is required. The discriminatory power of the Zernike moment invariants should be at least as good as the geometric invariant moments. *Teh and Chin ,1988,* show that Zernike moments have about the same sensitivity to noise as geometric moments and are therefore fairly robust.

The above investigation indicates that Zernike moments can be used as feature descriptors for object location and image matching. Due to the time required to calculate the moments on PCs the method was not used here in an example on image matching. The method can be used effectively on work stations and higher computers.

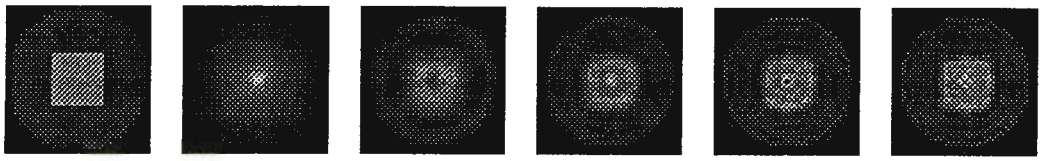
4.4.2.3 PSEUDO-ZERNIKE MOMENTS

As with the Zernike polynomials pseudo-Zernike polynomials are orthonormal in a unit radius circle. The definition of pseudo-Zernike moments is the same as that given by equation (4.4.2.2.1) excepting that the radial polynomial is defined as

$$R_{nl}(r) = \sum_{s=0}^{n-|l|} (-1)^s \frac{(2n+1-s)!}{s!\,(n-|l|-s)!\,(n+|l|-s)!} r^{n-s} \qquad \textbf{(4.4.2.3.1)}$$

where $|l| \leq n$ only.

Moment invariants can be derived in a similar fashion as for Zernike moments. In the case of the Zernike moments there are n+1 invariants at order n. In the case of the pseudo-Zernike moments there are 2n+1 invariants at order n.



N = 4 8 12 16 20

Figure 4.4.2.3.1. *Reconstructed image on the left using equation (4.4.2.2.1) with pseudo-Zernike moments for different values of N.*

Figure 4.4.2.3.1 shows how pseudo-Zernike moments reconstruct the image on the extreme left for different orders. Comparing with figure 4.4.2.2.1 shows that a better reconstruction is achieved at a lower order than in the case of Zernike moments as odd terms are included in the expansion, because n - |l| can be odd and not only even as for Zernike moments.

As in the case of Zernike moments invariants can be used to describe patterns and used in a similar way for pattern matching. *Teh and Chin , 1988*, show that pseudo-Zernike moments are more robust against noise than geometric and Zernike moments. The computational load is twice that of Zernike moments as the odd terms alluded to above are included in the expansion.

4.4.3.OTHER MOMENTS

Two other moments used in pattern recognition are defined. These are not orthogonal and therefore exhibit information redundancy. They are mentioned below for completeness.

4.4.3.1 ROTATIONAL MOMENTS

The rotational moment of order n is defined as

$$D_{nl} = \int_0^{2\pi}\int_0^{\infty} r^n e^{il\theta} f(r,\theta)\, r dr d\theta \qquad \textbf{(4.4.3.1.1)}$$

In the review article by *Prokop and Reeves, 1992*, various applications of these moments are mentioned. The performance of

these moments with noisy images is similar to Zernike moments as shown by *Teh and Chin, 1988*.

4.4.3.2 COMPLEX MOMENTS

Complex moments of order (p+q) are given by

$$C_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x+iy)^p (x-iy)^q f(x,y)\, dxdy \qquad \textbf{(4.4.3.2.1)}$$

and in polar coordinates this becomes

$$C_{pq} = \int_{0}^{2\pi}\int_{0}^{\infty} r^{p-q} e^{i(p-q)\theta} f(r,\theta)\, rdrd\theta \qquad \textbf{(4.4.3.2.2)}$$

Complex moments give moment invariants very easily by using the modulus of the complex moment as invariants. They are also the most robust against noise according to *Teh and Chin, 1988*. Unfortunately they do not form an orthonormal basis and there is more information redundancy, which is explained below, than with geometric moments as shown by *Teh and Chin, 1988*.

4.4.4.GENERAL REMARKS ON MOMENTS

Using complex moments *Abu-Mostafa and Psaltis, 1984* show that moment invariants are not in general good features as they suffer from information loss,suppression and redundancy. Information loss occurs as the higher order moments are left out for which high spatial frequency detail will be lost. Information suppression occurs as the r in equations (4.4.3.1.1) and (4.4.3.2.2) act as a weighting factor, so that detail near the boundary of the circular region is given more weight than near the centre. Complex moments ,geometric moments and rotational moments are not orthogonal and are therefore correlated leading

to information redundancy. For images with little high frequency detail though ,non-orthogonal moment invariants still perform well. Zernike and pseudo-Zernike moment invariants suffer only from information loss and therefore generally perform better. Legendre moments do not have generally simple invariant properties and are very sensitive to noise. Generally it may be concluded that Zernike and Pseudo-Zernike moments have the best image representation properties. Exhaustive experiments on a variety of different images would be required to specifically assess the performance of the different moment based descriptors. Investigations conducted by *Prokop and Reeves, 1992,* also suggest that Zernike and Pseudo-Zernike moments have the best overall performance in terms of noise sensitivity, information redundancy and image representation ability.

In this study for investigating the principle of using moment invariants as a pattern classifier for interest operators, object localization or target location only geometric moment invariants are used as they are computationally faster on a PC computer.

4.5. ANGLE BASED INTEREST OPERATOR

Interest operators based on pattern recognition extract clumps of points surrounding items in the image similar to the selected model pattern. Examples of this were given above when describing the design of a corner interest operator. The size of the clumps is dependent upon the cutoff threshold for the distance metric used to compare the patterns. One disadvantage of this type of interest operator is that it typically gives a relatively sparse distribution of interest points in a general image. The performance is image dependent and is sensitive to radiometric variations. Experiments with the Förstner operator *[Förstner, 1987]* and the Moravec operator *[Moravec, 1980]* show that they give similar results.

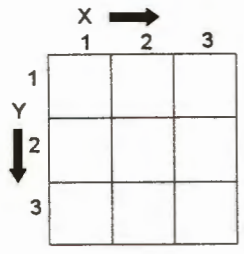


Figure 4.5.1. *3 X 3 Window.*

A general interest operator is described here, which gives plentiful well distributed points. The operator is based on comparing the gradients of neighbouring edgels or edge pixels. Initially the Sobel edge detector is applied and the resulting gradient image is thresholded using Otsu's method although any thresholding technique can be used. The resultant image is used as a search mask. A 3 x 3 window as shown in figure 4.5.1 is placed at the pixel position in the original image corresponding to an edge pixel in the mask. The gradients in the x and y directions are calculated, using the appropriate Sobel masks shown in figure 4.4.1.2.1, for each pixel in the window, from the original grey-scale image .

$\alpha(i,j)$ is the angle that the gradient makes to the X axis given by

$$\alpha(i,j) = \arctan(G_y(i,j)/G_x(i,j)) \qquad \textbf{(4.5.1)}$$

where

$G_x(i,j)$ is the gradient in X direction.

$G_y(i,j)$ is the gradient in Y direction.

i and j are integral indices for the x and y positions in the window in the range [1,3].

The angle determined by equation 4.5.1 is calculated for each pixel in the window and compared with the angle calculated at the central pixel. A counter c, initialized to zero, is incremented according to the result.

$$\begin{aligned} |\alpha(i,j)-\alpha(1,1)| &< \theta \Rightarrow c = c + 1 \\ &\geq \theta \Rightarrow c = c + 0 \end{aligned} \qquad \textbf{(4.5.2)}$$

θ is a preselected threshold. If the condition is true the counter is incremented by one. If any of the points in the window have no gradient or the ratio of the gradients is indeterminate, as in the case of the Y gradient and the X gradient being zero, the counter is incremented by one.

If the counter is more than four then the point is discarded from the mask. After the whole image has been processed the remaining edge points in the mask are taken as the interest points.

The basic principle behind this interest operator is that it selects the edge points with significant changes in gradient direction when compared with their neighbour points. θ determines the number of points retained. If $\theta=0$ then all edge points would be retained. If $\theta=\pi/2$ then few points such as corner points would be retained. A good distribution and number of points is obtained, for the application described in chapter 7, by using $\theta=\pi/18$. Selection of a suitable threshold is image dependent and arrived at heuristically.

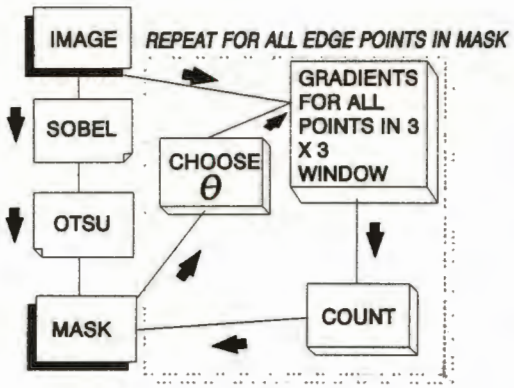


Figure 4.5.2. *Flow diagram of the angle based interest operator.*

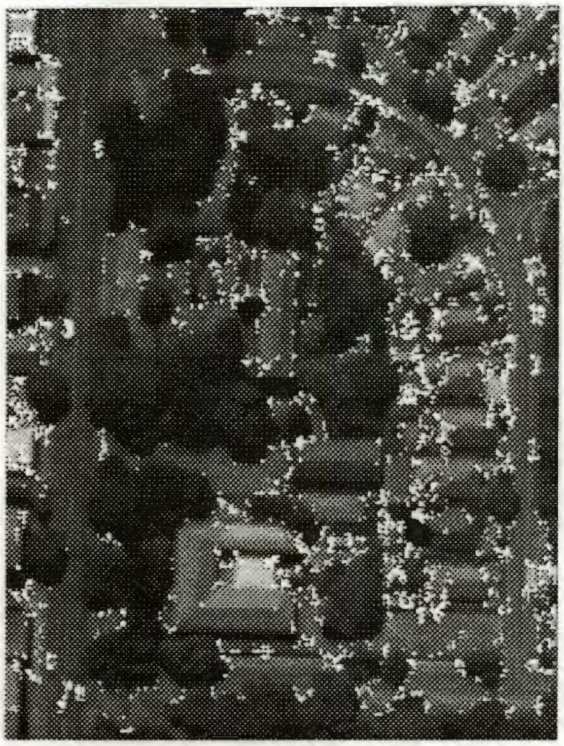


Figure 4.5.3. *Angle based interest operator applied to an aerial image.*

Figure 4.5.2 shows a flow diagram of the angle based interest operator and figure 4.5.3 shows its application to an aerial image. As shown there is a reasonably well distributed number of interest points. In many cases the corners of roofs have been well isolated.

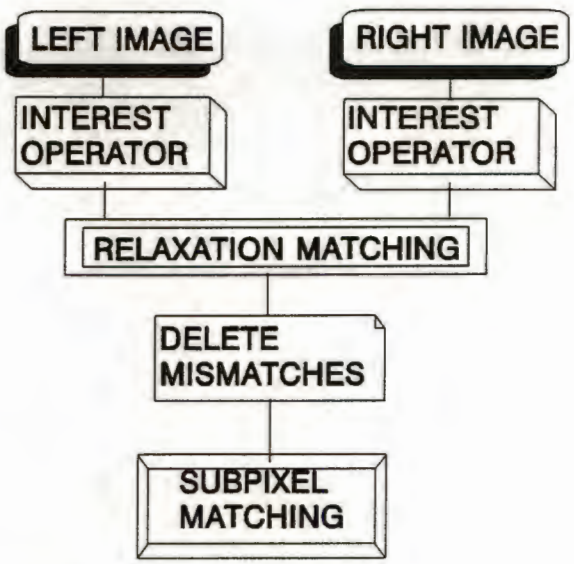


Figure 4.5.4. *Basic flow diagram of a matching strategy using interest operators.*

Applying the angle based interest operator to an aerial stereopair results in many corresponding points being extracted as will be illustrated in chapter 7. Establishing the correspondence between the extracted points is then performed with the continuous relaxation matching algorithm outlined in chapter 6. Once a set of corresponding points is found a correlation matching on the grey-scale values between windows surrounding the matching points can be used to eliminate any obvious mismatches. The pattern vector method using moments can also be used for this purpose. Computer assisted manual editing is then applied to eliminate any remaining mismatches. The remaining matched points are then matched to subpixel accuracy using least-squares or least absolute deviation grey-scale correlation matching which is described in chapter 5. A flow diagram of this general matching strategy is shown in figure 4.5.4.

4.6. FOURIER TRANSFORMS.

The moment based pattern vector approach described in section 4.4 above provides a set of useful quantitative features for pattern recognition and matching. With a similar objective in mind extracting quantitative features from the spatial frequency information in an image was investigated.

The Fourier transform has been used extensively in many applications related to signal processing of which image analysis forms a part. In image processing it is used to transform image data from the spatial domain to the spatial frequency domain. Extracting pattern descriptors from the transformed data seems feasible and is investigated in this section.

4.6.1 ONE DIMENSIONAL FOURIER DESCRIPTORS

The discrete Fourier transform has been successfully applied as a descriptor of object boundaries *[Gonzalez and Wintz, 1987] [Chellappa and Bagdazian, 1984] [Lin and Chellappa, 1987]*.

In most applications the object boundaries are fairly constant and, in analogy with the moment applications, usually with almost noise free backgrounds.

Applying this technique to images of thresholded edges and using the Fourier coefficients as descriptors seems plausible. Location and identification of the object of interest in an image is required in order to ensure the extraction of the correct edge boundary. Generalization of this process for computer automation is not possible at present. An edge following routine is then required to trace around the boundary and determine boundary points to subpixel accuracy. With this data a one - dimensional Fourier descriptor can be determined. Automatic object localization is possible in aerial images using the pattern recognition approach outlined in section 4.4 using quantitative features. The extraction of geometrical features such as edges is complicated by variations in perspective, shadows and feature filled backgrounds. Consequently the one dimensional Fourier descriptor can only be applied in specific applications. For completeness the formulation of the one-dimensional Fourier descriptor is given below:-

A two-dimensional object boundary can be described by a one dimensional discrete Fourier series as follows:-

Assume a closed boundary s(n) with N discrete points. This can be expressed in terms of a Fourier series expansion as follows

$$s(n) = \sum_{u=-NC}^{NC} F(u)\exp(i\frac{2\pi nu}{N}),\ (n=0,\cdots,N-1.) \qquad \textbf{(4.6.1.1)}$$

where

F(u) , u=-NC.....NC are the fourier coefficients and are given by the following formula:-

$$F(u) = \frac{1}{N}\sum_{n=0}^{N-1} s(n)\exp(-i\frac{2\pi un}{N}) \qquad \textbf{(4.6.1.2)}$$

There are 2NC+1 descriptors.

s(n)=x(n)+iy(n) , where x(n) and y(n) are the (x,y) coordinates of the nth point and $i=\sqrt{-1}$.

An edge forming a closed boundary is extracted to sub-pixel accuracy and N points are selected along its length. The set of Fourier coefficients making up the Fourier descriptor is then calculated using equation (4.6.1.2).

4.6.2 TWO DIMENSIONAL FOURIER DESCRIPTORS

A novel alternative attempt at a fourier descriptor is to use a two-dimensional fourier transform of a region containing the targeted object to build up a one-dimensional descriptor. A window of a selected size is positioned on the object to be identified in an image. The two-dimensional fourier transform of this region is calculated. The intensity values of the fourier transform are calculated at each point of a discretized window, with the origin at the centre of the window. The DC term of the fourier transform, i.e. the term at zero frequency is at the centre of the window.By calculating the average intensity over concentric rings centred on the origin, a radial average intensity profile is constructed. This intensity profile is then used as a descriptor of the region and will be referred to as the radial Fourier descriptor.

4.6.2.1 CALCULATION OF TWO-DIMENSIONAL FOURIER TRANSFORMS

Before showing some examples of this method the mathematical definition of the two dimensional fourier transform is given below.

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp[-i2\pi(ux+uy)]\,dxdy \qquad \textbf{(4.6.2.1.1)}$$

where

F(u,v) is the two dimensional Fourier transform and u and v are orthogonal frequency variables.

f(x,y) is a continuous two dimensional signal function described over a cartesian plane.

As digitized images are discrete functions equation (4.6.2.1.1) is given in a discrete form as

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\exp[-i2\pi(\frac{ux}{M}+\frac{vy}{N})] \qquad \textbf{(4.6.2.1.2)}$$

where

F(u,v) and f(x,y) are the discrete versions of the functions in equation (4.6.2.1.1).

f(x,y) has values for x=0,1,.....,M-1 and y=0,1,.....,N-1.

u=0,1,.....,M-1

v=0,1,.....,N-1.

In applying the Fourier transform square windows are usually used so that N=M. A digital image is a discrete sample of points at intervals of 1 pixel in both orthogonal directions of a two dimensional cartesian plane. The interval in the corresponding Fourier transform will be $\Delta u = \frac{1}{N\Delta x}$ where Δx is the interval in each direction of the image plane and Δu is the interval in each direction of the transform plane.

The two dimensional Fourier transform can be calculated by repeated application of the one dimensional transform due to the separable character of the two dimensional function. Equation (4.6.2.1.2) can be re-written as

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} \exp(-i\frac{2\pi ux}{N}) \sum_{y=0}^{N-1} f(x,y)\exp(-i\frac{2\pi vy}{N}) \qquad \textbf{(4.6.2.1.3)}$$

Notice that now M=N and the term 1/MN becomes 1/N only as 1/M is factored into the coefficients. Equation (4.6.2.1.3) becomes

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} F(x,v)\exp(-i\frac{2\pi ux}{N}) \qquad \textbf{(4.6.2.1.4)}$$

where

$$F(x,v) = N[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp(-i\frac{2\pi vy}{N}) \qquad \textbf{(4.6.2.1.5)}$$

The two dimensional Fourier transform is then calculated by calculating the one dimensional Fourier transform of each row in the image and multiplying the result by N as indicated by equation (4.6.2.1.5). The one dimensional Fourier transform along each of the columns of the resultant image F(x,v) is then performed as shown in equation (4.6.2.1.4) to give the final result.

In order to reconstruct the image the inverse Fourier transform has to be determined. The inverse Fourier transform is given in separable form by

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1} \exp(i\frac{2\pi ux}{N}) \sum_{v=0}^{N-1} F(u,v)\exp(i\frac{2\pi vy}{N}) \qquad \textbf{(4.6.2.1.6)}$$

By taking the complex conjugate of equation (4.6.2.1.6) it becomes similar in form to equation (4.6.2.1.3).

$$f^*(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\exp(-i\frac{2\pi ux}{N})\sum_{v=0}^{N-1}F^*(u,v)\exp(-i\frac{2\pi vy}{N}) \qquad \textbf{(4.6.2.1.7)}$$

where * indicate the complex conjugate.

The same method that was used to calculate the forward transform outlined above can be applied in the same way to obtain the inverse transform. The complex conjugate of the Fourier transform has to be calculated first and the complex conjugate of the final output of the method will give the desired result. The Cooley-Tukey algorithm *[Gonzalez and Wintz, 1987]* is used to calculate the one dimensional transforms.

4.6.2.2. EXPERIMENTS WITH A TWO DIMENSIONAL FOURIER DESCRIPTOR

The radial Fourier descriptor mentioned at the start of section 4.6.2.1 was determined for different objects in a number of images. The calculated descriptors were compared to assess their discriminatory ability.

Figure 4.6.2.2.1 (a) and (b) shows a 32 x 32 pixel window of an synthetic object with its Fourier transform. The radial Fourier descriptors for 4.6.2.2.1(a),(c),(d) are plotted in figure 4.6.2.2.2. The plot for figures 4.6.2.2.1(a) and (c) are similar excepting at very low frequencies.The plot for 4.6.2.2.1(d) is different over the whole range considered. This would tend to indicate the similarity that exists between the images themselves as will be evident. Figure 4.6.2.2.3 shows the plot of the descriptors for figure 4.6.2.2.1(e), and its corresponding window in the stereopair partner image shown in figure 4.6.2.2.1(f). The values of the Fourier coefficients are normalized by dividing all the coefficients by the coefficient of the first harmonic.

(a) (b) (c) (d) (e) (f)

Figure 4.6.2.2.1. *(a),(c),(d),(e),(f) are test case windows of different scenes.(b) is the Fourier transform of (a).*

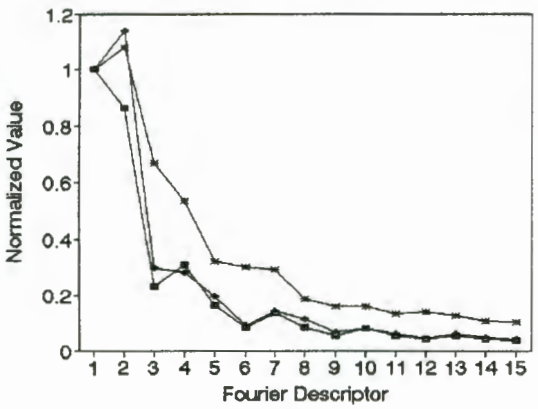


Figure 4.6.2.2.2. *Plots of radial Fourier descriptors .*
■ *Fig. 4.6.2.2.1.(a)*
\+ *Fig. 4.6.2.2.1.(c)*
* *Fig. 4.6.2.2.1.(d)*

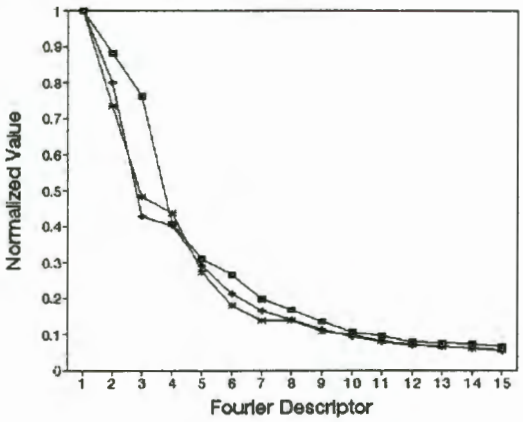


Figure 4.6.2.2.3. *Radial Fourier descriptor plots.*
■ *Fig. 4.6.2.2.1.(e)*
\+ *Stereo partner of 4.6.2.2.1.(e)*
* *Fig. 4.6.2.2.1.(f)*

As is evident from the plots in figure 4.6.2.2.2 the descriptors can discriminate well between the squares and the circle. The plots for the squares are very similar excepting at the very low frequency accounted for by the difference in scale. Unfortunately all three plots in figure 4.6.2.2.3 appear very similar although one scene is vastly different. The noise content of these images render the discriminatory ability of the Fourier descriptors useless. The windows used for figure 4.6.2.2.2 are noise free enabling the Fourier descriptor to discern differences.

A few experimental trials on Sobel edge images of aerial scenes showed that the Fourier coefficients are not sufficiently stable and extremely sensitive to the presence of additional "noisy" edges. This would be expected as the Fourier coefficients describe the spatial frequency of the chosen image window.

Variations in edge lengths, broken edges and additional edges ,as present in most real images, will alter the frequency pattern and destabilize a Fourier descriptor.

In addition to the above reasons the radial Fourier descriptors are not scale invariant and can only be used with images at the same scale. Radial Fourier descriptors can therefore not generally be used to describe patterns in real applications.

The two dimensional Fourier transform can be used as a preprocessor to filter out some noise in images and to analyze the frequency content of images using band-pass filtering. Figure 4.6.2.2.4 shows the frequency content of the synthetic window of figure 4.6.2.2.1(c) for various bandwidths. The edges show up at the high spatial frequencies. Unfortunately the high frequency noise content also shows up in real images and smothers the edge detail.

0<=ν<=4 5<=ν<=8 9<=ν<=12 13<=ν<=16 17<=ν<=20

Figure 4.6.2.2.4. *The synthetic window of figure 4.6.2.2.1(c) filtered at the indicated frequencies using the Fourier transform and its inverse.*

4.6.2.3. IMAGE SEGMENTATION USING FRACTAL DIMENSION

The above investigation indicates that the two dimensional Fourier transform can effectively be used as a descriptor in noise free images. In "noisy" images the Fourier descriptor is ineffective, but can still be used to analyze the fractal nature of an image. This analysis leads to a useful technique for regional image segmentation as, for example to distinguish between urban areas or forests.

Segmentation of images according to texture is widely used in many remote sensing applications *[Haralick and Shapiro, 1993]* and can be applied in an automated photogrammetric system to reduce search space. Texture relates to image properties such as fineness, coarseness, regularity or randomness. Many measures of texture have been developed *[Haralick and Shapiro, 1993]* which have been compared *[Wezka, Dyer and Rosenfeld, 1976]*. The comparison shows that the power spectral techniques using Fourier transforms do not perform as well as other methods, but are computed using familiar concepts. For the latter reason and as an extension to the investigation of using two dimensional Fourier transforms in image analysis and description, the method based on fractal dimension was investigated.

Pentland, 1984, Kube and Pentland, 1988, have shown that different regions in an image, such as forests or urban developments, will have different fractal dimensions. By transforming an image into an image representing its fractal dimension the original image can be segmented according to the values of the fractal dimension. This can be used in an automated photogrammetric system to reduce search space in the localization of any particular object, such as a building in an urban region. Once the urban areas have been isolated using fractal dimension segmentation, the rest of the image can be ignored resulting in a reduced search space.

Mandelbrot, 1983, has shown that natural surfaces and boundaries exhibit a fractal behaviour for which he formulated the concept of fractional (fractal) dimension in order to describe this phenomenon. Essentially he showed that the determination of the length of a natural boundary, such as a coastline, will depend on the length of the measuring ruler. If you used a kilometre ruler you would get a different answer than if you used a metre ruler as the kilometre ruler will ignore finer detail less than a kilometre in length. Also the outline of a one kilometre stretch of coastline would look very similar to the outline of a metre stretch of coastline but at a different scale. To

describe this structural scale independence he modelled the outline using Brownian fractal noise. The behaviour of a one dimensional Brownian fractal curve can be modelled by *[Kube and Pentland, 1988]*

$$<[B_H(t+\delta)-B_H(t)]^2> = A|\delta|^{2H} \qquad \textbf{(4.6.2.3.1)}$$

where

$(t,B_H(t))$ would be the coordinate of any point on the curve.

δ is the length of the measuring ruler.

$H \in [0,1]$ and is related to the Hausdorff-Bescovitch dimension or fractal dimension D by $D = 2 - H$.

<..> indicate the statistical expectation operator or average.

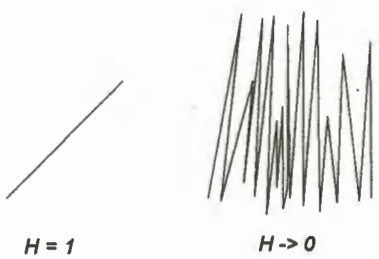


Figure 4.6.2.3.1. *Examples of line structure for different values of H.*

One can interpret (4.6.2.3.1) as the average of many measurements of the distance squared made with a ruler of length δ between many randomly chosen point pairs along the curve. If H=1 then the curve is a straight line. If H is near zero the curve is extremely rugged. Figure 4.6.2.3.1 shows the two cases mentioned. The fractal dimension in these cases will be 1 and almost 2 respectively. The fractal dimension can therefore be used as measure of ruggedness or texture.

This concept can be extended to any dimension. Images can be regarded as two dimensional functions of grey-scale values. The ruggedness or texture of a region is reflected in the grey-scale

values of its image. By treating the grey-scale function as a fractal Brownian function the fractal dimension of any region can be determined. In most applications *[Pentland, 1984] [Kube and Pentland, 1988] [Mallat, 1989]* the two dimensional Fourier transform is used to obtain a power spectrum from which the fractal dimension is determined. The power spectrum of a fractal Brownian function is given by

$$P(f) = kf^{-2H-1} \qquad \textbf{(4.6.2.3.2)}$$

where

f is the spatial frequency.
k is a constant.

The fractal dimension is then given by

$$D = T + 1 - H \qquad \textbf{(4.6.2.3.3)}$$

where T is the topological dimension. The topological dimension is the same as the Euclidean dimension, but the space thus described allows affine transformations.

In order to determine H from equation (4.6.2.3.2) the two dimensional Fourier transform of a selected window is calculated. The radial descriptor described in section 4.6.2.1 is then determined which effectively gives the modulus of the average Fourier coefficients averaged over the perimeter of concentric circles of radius equal to the spatial frequency, as a function of spatial frequency. By taking the logarithms of equation (4.6.2.3.2) one obtains

$$2\log A(f) = (-2H-1)\log f \qquad \textbf{(4.6.2.3.4)}$$

where

A(f) is the Fourier coefficient at frequency f ,which is related to P(f) by $P(f)=A(f)^2$.

Equation (4.6.2.3.4) is that of a line with slope (-2H-1).

Using linear regression will yield the value of H as follows.

$$2H = -1 - \frac{N\sum_{f=1}^{N} 2\log(f)\log A(f) - \sum_{f=1}^{N}\log(f)\sum_{f=1}^{N} 2\log A(f)}{N\sum_{f=1}^{N}\log(f)^2 - \left(\sum_{f=1}^{N}\log(f)\right)^2} \qquad \textbf{(4.6.2.3.5)}$$

where

N=number of spatial frequencies.

The fractal dimension is then given by equation (4.6.2.3.3).

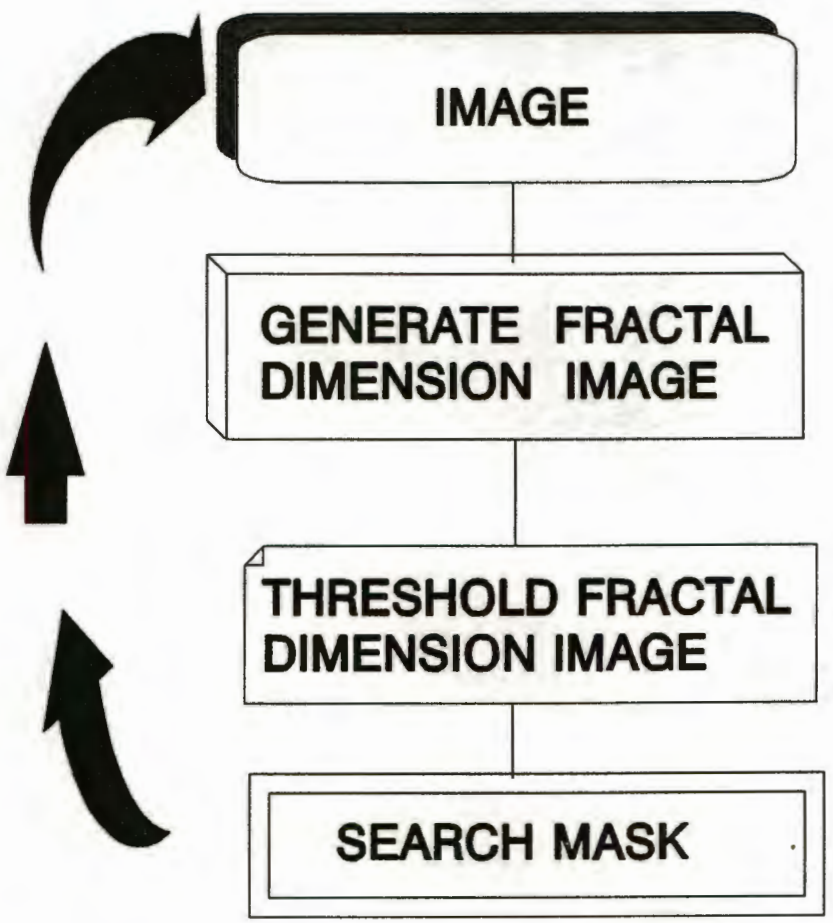


Figure 4.6.2.3.2. *Flow diagram for generation of search mask for image segmentation using fractal dimension.*

Figure 4.6.2.3.2 shows the main steps required in generating a search mask using the fractal dimension technique. The mask is then used to segment the original image as indicated by the arrows.

An example of fractal dimension segmentation is shown in figure 4.6.2.3.3. The fractal dimension of 8 X 8 pixel windows over figure 4.6.2.3.3(a) are displayed as grey-scale values in figure 4.6.2.3.3(b). Window sizes with the side length equal to a power of 2 were chosen as the Cooley-Tukey FFT algorithm works more efficiently in that case. The input data can be modified so that any window size can be chosen by adding zero entries to bring the number of data entries up to the next power of two. As the neighbouring pixel neighbourhoods sometimes have varying fractal dimension figure 4.6.2.3.3(b) is smoothed by averaging over 5 x 5 windows, and the result is shown in 4.6.2.3.3(c). 4.6.2.3.3(a) is then segmented based on the grey-scale values in 4.6.2.3.2(c) by choosing a threshold value. Most of the regions consisting of trees are segmented out of the image as shown in 4.6.2.3.3(d). Unfortunately small segments of urban area are also erased. This problem can be overcome by smoothing with larger windows. The threshold value used was determined heuristically. Figure 4.6.2.3.3(e) shows the result of using the thresholding routine due to Otu described earlier on the image in 4.6.2.3.3(c). In both figures 4.6.2.3.3(d) and (e) border areas can be ignored.

The segmentation method described above can be used to limit the search space in an automated matching routine. Finding a matching building using the geometric moment technique described above would be an example. As non-urban regions are segmented out of the image they can be ignored. In the case where an urban region has been misclassified by the segmentation, additional information such as maximum possible disparity range and the epipolar line can be used to expose such an error. In the latter case a region flagged as non-urban would not be ignored. Alternatively the segmented image can be edited interactively.

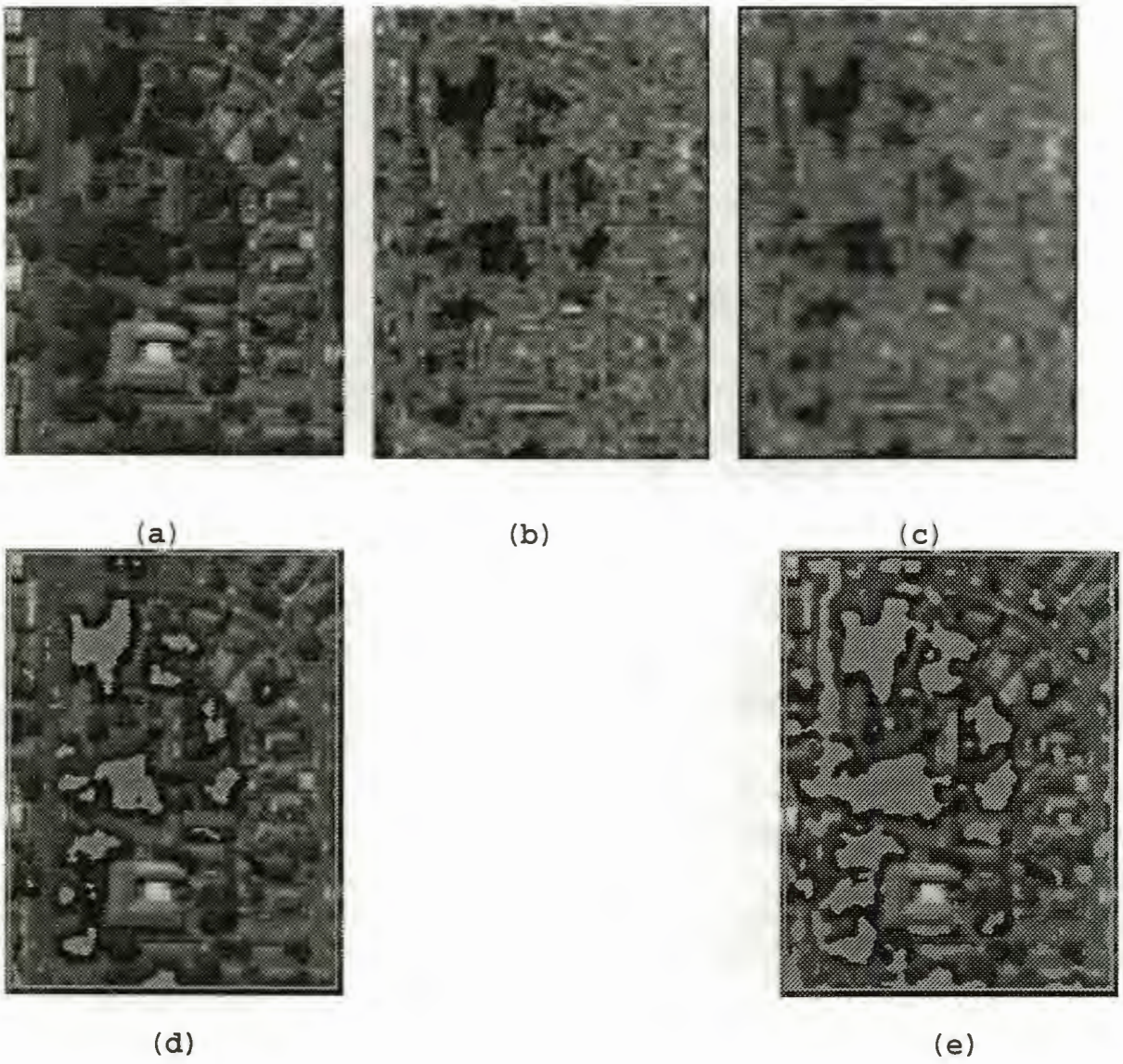


Figure 30. *(a) Original image. (b) Grey-scale fractal dimension image. (c) Averaged grey-scale fractal dimension image. (d) Segmented areas in (a) (manual threshold). (e) Segmented areas in (a) (Otsu threshold).*

4.7. CONCLUSION

The above investigation has endeavoured to show how various mathematical descriptors can be used to uniquely describe image patterns in order to locate matching patterns in similar images. The feature vector description is drawn from an extracted sample of typical windows in order to derive the mean vector and the covariance matrix. The whitened mean vector is then used to

locate corresponding matching patterns by searching similar images.

The main conclusions that emerge are:-

1. Geometric , Zernike and Pseudo-Zernike moment invariants can be used as effective descriptors of image patterns for localization of objects and as interest operators. They exhibit good rotational invariance and scale invariance. Geometric moments have the added advantage of simplicity in calculation but exhibit information redundancy. Zernike and Pseudo-Zernike moments are orthogonal but computationally expensive.

2. Other moments such as Complex, Rotational and Legendre moments are not suitable descriptors. Legendre moments do not have simple invariance properties and are computationally expensive. Complex and rotational moments have information redundancy and suppression.

3. The two-dimensional Fourier transform is a poor descriptor having no simple scale invariance properties and is extremely sensitive to image noise.

4. The two-dimensional Fourier transform can however be effectively used to describe image texture in a relatively simple manner by enabling the computation of the fractal dimension. Images can be segmented on the basis of the fractal dimension to isolate uniform textured regions such as forests, deserts or urban areas. This is especially useful in an automated system to reduce the search space when seeking a building, as all non-urban regions can be ignored.

5. The feature vector formulation for matching patterns can be generalized to include any form of numerical identifier as a component of the vector.

5.SUBPIXEL IMAGE MATCHING FOR OBJECT RECONSTRUCTION

Pattern recognition techniques can be advantageously applied to images for object localization. In order to reconstruct the three dimensional structure of an object from a stereopair of images, precise point to point matching is required to subpixel accuracy.

The main approaches to solving the point matching problem rely on the extraction of points of interest using interest operators *[Förstner, 1987] [Moravec, 1980]* or edge detector operators *[Sarkar and Boyer, 1991] [Canny, 1986] [Sobel, 1970]*. The points extracted in this way are matched directly as pixel points or edgels using techniques such as relaxation methods with additional constraints such as grey-scale correlation, disparity ranges and the epipolar line constraint *[Price, 1985]*. Alternatively the spatial relationships between the points and lines are used to build up descriptors of objects in each image and these descriptors are then matched subject to various problem specific constraints *[Medioni and Nevatia, 1984] [Hellwich and Faig, 1994] [Van der Merwe and Rüther, 1994]*. Once correspondence between matching points has been established the matching precision can be improved to subpixel accuracy using various techniques. If the points are initially extracted to pixel precision then an area-based matching technique can be used to obtain subpixel precision. Alternatively if the points are extracted to subpixel precision then three dimensional reconstruction can commence once the correspondence is established.

When applying the pattern recognition approach in locating objects in a scene, the pattern descriptor used implicitly contains the image structure information, but would require complicated methods to extract the explicit structural information for direct object reconstruction. In order to reconstruct the object from the image data the geometric

structural information is required in an explicit form. Examples of this would be the start and end points of lines or open curves, centres of mass of closed curves, and corner points. The following chapter investigates methods of improving the subpixel precision of area-based matching and attempts to show how pattern recognition techniques can be used to match points to subpixel accuracy. In addition a strategy for extracting the geometric structure of an object such as a building in an image will be discussed.

5.1.AREA-BASED MATCHING

A digital image is essentially a discrete distribution of grey-scale intensity values. The variations in the intensities carries the information that enables the observer to perceive objects in the scene. Consequently similar objects observed from viewpoints that are relatively close when compared to the distance of the object, and in similar lighting should appear similar in the respective images. Therefore the grey-scale distribution representing an object in an image should be similar in the aforementioned images. A measure of this similarity can then be used as a matching criteria and forms the conceptual basis on which area-based matching techniques are based.

5.1.1 GREY-SCALE CORRELATION MATCHING

Direct correlation matching of image windows is the simplest form of area-based matching. A window surrounding an area of interest in one image is selected and is conventionally termed the **template**. The image in which the object is to be located is searched by calculating a distance value between the template and a window of the same size, known as the **patch**, around consecutive pixels over the whole image. The position where an minimum value of the distance function is obtained is taken as the matching region. A distance function that can be used, where the minimum

value would indicate matching is calculated as follows:-

$$\sum_{i=1}^{N} (f_i - g_i)^2 \qquad (5.1.1.1)$$

where

f_i is the grey-scale value at point i in the template.

g_i is the grey-scale value at point i in the patch.

N is the number of points in the window.

Another matching criterion is the maximization of a measure of similarity such as the normalized cross-correlation function given by

$$\frac{\sum_{i=1}^{N} f_i g_i}{\sqrt{\sum_{i=1}^{N} f_i^2}\sqrt{\sum_{i=1}^{N} g_i^2}} \qquad (5.1.1.2)$$

The above method is adequate for pixel matching accuracy. Correlation matching techniques can be extended to give subpixel accuracy by methods which require resampling to increase the resolution of the original image *[Tian and Huhns, 1986]*. In these methods if an accuracy of 0.1 pixel is required then the resolution of the template has to be increased by a factor of 10 in each direction. Bilinear interpolation of the intensity values is adequate for generating the higher resolution templates. The resolution and not the dimensions of the template windows are increased. All these methods are computationally intensive and time consuming.

As area-based matching techniques are dependent on the information provided by the discrete grey-scale intensity distributions in an image, they are affected by factors which influence this distribution. These include geometric distortions due to viewing positions, levels of illumination, shadows, variations in contrast, and occlusions *[Dhond and Aggarwal, 1989]*.

Additionally these methods only solve for the translation (disparity) between matching windows, and not for a full set of affine parameters that model the transformation of the template window into the patch window.

5.1.2 ITERATIVELY REWEIGHTED LEAST-SQUARES MATCHING (IRLS)

The grey-scale correlation method can be extended to model a fully affine transformation between the template and the patch as described by *Gruen, 1985*. A least-squares or L2-norm estimation model is formulated incorporating the projective transformation parameters as the unknowns to be determined. As a subset of the projective transformation, affine parameters are usually determined as an approximation for minor transformations. The transformation required between stereo images that are taken from similar viewpoints can be regarded as a minor projective transformation. The estimation model can be further generalized as a Lp-norm estimation problem where p>0 *[Calitz and Rüther, 1994]*. Least-squares is the same as the L2-norm estimator.

The functions f(x,y) and g(x,y) represent the discrete grey-scale distribution functions of the template and patch respectively. Exact matching occurs when the difference between the grey-scale values at corresponding points is zero. In real images exact correspondence seldom occurs and there is usually a residual grey-scale value remaining. The difference between the template and patch at any point can be expressed as

$$r(x,y) = f(x,y) - g(x,y) \qquad (5.1.2.1)$$

where

r(x,y) is the residual corresponding to the template position (x,y).

Equation (5.1.2.1) is a non-linear function with many local extrema. The location of the local minimum would be taken as the matching position. Only if the complete images are used as the

matching windows can such a global minimum be reliably determined in general. Once a small window is used many possible minima will be detected requiring that the matching method selects the position with the lowest minimum. The least-squares matching method mentioned above uses small windows and needs to commence at a position near the correct match position otherwise it will not converge reliably. A method to determine the correct match position to near pixel accuracy is therefore required before the application of the least-squares matching method. A correlation technique as mentioned above can be used for this purpose, but methods employing a priori information accelerate the location of a matching region ,e.g. the epipolar line constraint based on knowledge of the mutual relative orientation of the images.

A matching criterion in the Lp-norm based on equation (5.1.2.1) requires that

$$\left(\sum_{i=1}^{N} |r_i|^P \right)^{\frac{1}{P}} \rightarrow \min \qquad (5.1.2.2)$$

where

N is the number of points in each window.

The image coordinates of the template and patch are related by an affine transformation as follows *[Gruen, 1985]*:-

$$\begin{aligned} x &= a_{11} + a_{12}x_0 + a_{21}y_0 \\ y &= b_{11} + b_{12}x_0 + b_{21}y_0 \end{aligned} \qquad (5.1.2.3)$$

where

(x,y) are the cartesian coordinates of a point in the template.

(x_0,y_0) are the coordinates of a point in the patch.

a_{11} and b_{11} are the translational parameters of the transformation in the x and y directions respectively.

a_{12} and b_{12} are the x and y scaling parameters respectively.

a_{21} and b_{21} represent x any y axis rotations respectively and are often referred to as the shearing parameters.

Equations (5.1.2.1) and (5.1.2.2) define a nonlinear programming problem. Equation (5.1.2.1) is linearized, according to the Gauss-Newton method, in order to make the dependence on the affine parameters explicit, and assuming only small deformations and shifts so that higher order terms can be ignored.

$$r(x,y) = f(x,y) - \left(g(x_0,y_0) + \frac{\partial g(x_0,y_0)}{\partial x}dx + \frac{\partial g(x_0,y_0)}{\partial y}dy\right) \qquad (5.1.2.4)$$

where

$$dx = da_{11} + da_{12}x_0 + da_{21}y_0$$

$$dy = db_{11} + db_{12}x_0 + db_{21}y_0$$

The left hand side of equation (5.1.2.2) can be rewritten as

$$\begin{aligned}\left(\Sigma|r|^p\right)^{\frac{1}{p}} = \Big(\Sigma\Big|f(x,y) - \big(g(x_0,y_0) \; + \\ g_x da_{11} + g_x x_0 da_{12} + g_x y_0 da_{21} \; + \\ g_y db_{11} + g_y x_0 db_{12} + g_y y_0 db_{21}\big)\Big|^p\Big)^{\frac{1}{p}}\end{aligned} \qquad (5.1.2.5)$$

where

$$g_x = \frac{\partial g(x_0,y_0)}{\partial x}$$

$$g_y = \frac{\partial g(x_0,y_0)}{\partial y}$$

are the gradients of the grey-scale pixel intensity values in the x and y directions respectively.

The problem defined by equations (5.1.2.5) and (5.1.2.2) can be solved by an iterative procedure. Let x be the vector of the differential changes in the affine parameters.

$$x^T = (da_{11}, da_{12}, da_{21}, db_{11}, db_{12}, db_{21}) \qquad (5.1.2.6)$$

For the i^{th} point in the patch let

$$\ell_i = f(x_i, y_i) - g^0(x_i, y_i) \qquad (5.1.2.7)$$

and

$$A_i = (g_{x_i}, g_{x_i}x_i, g_{x_i}y_i, g_{y_i}, g_{y_i}x_i, g_{y_i}y_i) \qquad (5.1.2.8)$$

Equation (5.1.2.5) can then be written in a concise form as

$$(\Sigma|r_i|^p)^{\frac{1}{p}} = (\Sigma|l_i - A_i x|^p)^{\frac{1}{p}} \qquad \textbf{(5.1.2.9)}$$

p=2 gives rise to the least-squares matching method for correlation matching.

Let

$$\sum_{i=1}^{N} |r_i|^2 = \underline{r}^T \underline{r} \qquad (5.1.2.10)$$

where

$\underline{r}$ is the vector of residuals with components $l_i - A_i x$.

N is the number of points in the window. Square windows are usually used with side lengths equal to an odd number of pixels. If M is the length of the side $N=M^2$.

Expanding gives

$$\underline{r}^2 = (\underline{l}-Ax)^T(\underline{l}-Ax)$$
$$= \underline{l}^T\underline{l} - 2x^TA^T\underline{l} + x^TA^TAx \qquad (5.1.2.11)$$

where

$\underline{l}$ is an N component vector of grey-scale differences $f_i - g_i$.

A is an 6 X N matrix known as the design matrix.

x is the 6 component unknown parameter adjustment vector.

Differentiating (5.1.2.9) with respect to x and equating to zero to find the minimum then gives

$$x = (A^TA)^{-1}A^T\underline{l} \qquad (5.1.2.12)$$

Weights can be introduced into equation (5.1.2.12) to allow an observer to have an influence on the parameter adjustments according to their accuracy, with outlier suppression occurring as a byproduct. This method incorporating weights is known as the iteratively reweighted least squares (IRLS) method *[Bloomfield and Steiger, 1983]*. The diagonal N X N weighting matrix P can be incorporated into equation (5.1.2.12) as follows

$$x = (A^TPA)^{-1}A^TP\underline{l} \qquad (5.1.2.13)$$

Various weight functions are used depending on the distribution of residuals of the data. Lp-norm estimators are a subset of a more general class of M-estimators *[Li, 1985]* so that equations (5.1.2.2) and (5.1.2.9) can be reformulated as

$$\Sigma\rho(l_i-A_ix) \Rightarrow \min \qquad (5.1.2.14)$$

where ρ is a function of the residuals and can take on many forms, some of which are listed in table 5.1.2.1.

Both least-squares and least absolute deviation are types of

M-estimators where $\rho=\frac{1}{2}r^2$ and $\rho=|r|$ respectively. Many other M-estimator functions exist and are compared by *Kubik,Merchant and Schenk, 1987* using the IRLS method where the weights are iteratively updated using

$$\frac{\rho(r_i)}{r_i^2+\epsilon} \qquad (5.1.2.15)$$

where
r_i is the residual associated with observation i.

ϵ is an arbitrarily small positive constant to prevent division by zero when the residual is equal to zero.

	$\rho(r)$	
Ordinary Least Squares	$1/2\ r^2$	$\|r\|<\infty$
Least Absolute Residual	$\|r\|$	$\|r\|<\infty$
Huber	$1/2\ r^2$	$\|r\|\le k$
	$k\|r\|-1/2\ r^2$	$\|r\|>k$
Andrews	$A^2[1-\cos(r/A)]$	$\|r\|\le \pi A$
	$2A^2$	$\|r\|>\pi A$
Biweight	$B_2/6\{1-[1-(r/B)^2]^3\}$	$\|r\|\le B$
	$B^2/6$	$\|r\|>B$

Table 5.1.2.1. *Some common M-estimators. k,A and B are heuristically determined constants.*

For the ordinary least squares estimator the weight matrix is maintained as the identity matrix throughout all iterations.

Examples of estimator functions are listed in table 5.1.2.1. As determination of the constants in these estimator formulae are data dependent. Only the least squares and least absolute deviation M-estimators were investigated.

The iterative procedure used to solve for the affine parameters is outlined below:-

1. Initial values are assigned to the affine parameters , $a_{11}=a_{21}=b_{11}=b_{12}=a_{12}=b_{21}=0$. A window size is chosen and a stopping condition is selected.

2. The vector $\underline{l}$ is calculated using the grey-scale values in the template and patch.

3. The matrix A is calculated similarly , with the gradients determined using a Sobel operator.

4. The residuals and the weight matrix terms are calculated at this point. The weights are usually based on the value of the residuals of the previous adjustment. For the first iteration the weight matrix is made equal to the identity matrix.

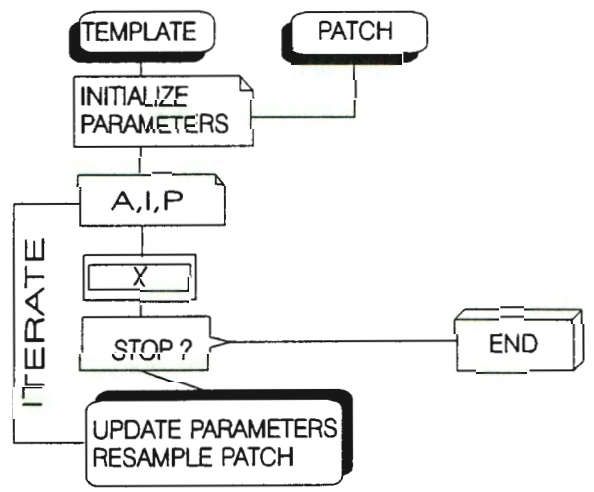


Figure 5.1.2.1. *Basic flow diagram of the Lp-norm matching algorithm.*

5. $A^TP\underline{l}$ and A^TPA are calculated.

6. A^TPA is a positive definite symmetric matrix, and the cholesky method is used to solve for x.

7. The affine parameters are updated using the values found in 6.

8. The new pixel coordinates of the patch are determined. These are non-integral values.

9. The grey-scale values of the newly determined patch are determined using a resampling or interpolation technique. Bilinear interpolation is found to be adequate. Using the newly determined patch the algorithm returns to step 2.

10. Steps 2 to 9 are repeated until the stopping condition is satisfied. This condition is usually based on all the parameter adjustments being smaller than a preselected tolerance value.

In order to assess the precision of the parameters calculated the covariance matrix is determined as

$$K_{xx} = \sigma_0^2 (A^T P A)^{-1} \qquad (5.1.2.16)$$

where

$$\sigma_0^2 = \frac{\underline{r}^T P \underline{r}}{N - c}$$

c=6 is the number of unknown parameters in x.

The covariance matrix is a 6 X 6 matrix and the precision of the ith parameter is measured by the standard deviation

$$\sigma_i = \sigma_0 q_{ii}^{1/2} = K_{ii}^{1/2} \qquad (5.1.2.17)$$

where

q_{ii} is the diagonal element in the ith row of $Q=(A^T P A)^{-1}$.

K_{ii} is the diagonal element in the ith row of K.

To assess the precision of each observation a posteriori and hence identify outlier points the following covariance matrix has to be determined

$$M = \sigma_0^2 A(A^T PA)^{-1} A^T \qquad (5.1.2.18)$$

where

M is an N X N matrix.

The standard deviation associated with equation j is given by
$\sigma_j = m_{jj}^{1/2}$

where m_{jj} is the diagonal element in the jth row of M.

All the Lp-norm methods follow a similar algorithmic form, only that either steps 4,5 and 6 are replaced by different steps or the specific Lp-norm is accounted for by adjustments in the weight matrix *[Kubik, Merchant and Schenk, 1987]*.

5.1.3 LEAST ABSOLUTE DEVIATION (LAD) MATCHING

Least-squares matching is very sensitive to the presence of outliers in the data as would occur with non-corresponding occlusions, hotspots or shadows. Various estimation models have been devised in mathematical statistics to make estimators more robust to the presence of outliers *[Li, 1985]*. The L1-norm has been shown to be more robust than least-squares in multiple regression problems *[Watson, 1980]*.

The L1-norm or least absolute deviation is formulated as a linear programming problem and requires that

$$\sum_{i=1}^{N} |r_i| \rightarrow \min \qquad (5.1.3.1)$$

with the residuals corresponding to the ith observation given by
$r_i = l_i - A_i x$

Let $r_i = u_i - v_i$, i=1,2,....,N .
where $u_i, v_i \geq 0$, i=1,2,.....N, and N is the number of

equations or pixel points in the template and patch. The residuals are expressed in terms of non-negative variables so that

$$r_i > 0 \quad \Rightarrow \quad u_i = r_i \text{ and } v_i = 0$$
$$r_i < 0 \quad \Rightarrow \quad u_i = 0 \text{ and } v_i = |r_i|$$

Similarly let $x_j = c_j - d_j$, $j=1,2,.....n.$
where $c_j, d_j \geq 0$, $j=1,2,.....n$, and n is the number of parameter adjustments to be determined. In this case n=6. The unknown parameters are also expressed in terms of non-negative variables so that

$$x_i > 0 \quad \Rightarrow \quad c_i = x_i \text{ and } d_i = 0$$
$$x_i < 0 \quad \Rightarrow \quad c_i = 0 \text{ and } d_i = |x_i|$$

The problem can now be reformulated in a suitable form for solving using a linear programming simplex algorithm so that *[Watson, 1980]*

$$\sum_{i=1}^{N} (u_i + v_i) \rightarrow \min \qquad (5.1.3.2)$$

subject to the constraints

$$\underline{u} - \underline{v} + A(\underline{c} - \underline{d}) = \underline{l}$$
$$\underline{u}, \underline{v}, \underline{c}, \underline{d} \geq 0$$

where $\underline{u}$, $\underline{v}$ and $\underline{l}$ are N component vectors.

$\underline{c}$ and $\underline{d}$ are n component vectors.

The problem posed by (5.1.3.1) is in the form of a primal linear programming problem. There are a number of different algorithms to solve this linearized optimization problem with a set of overdetermined linear equations as constraints. All are based on the Simplex algorithm originally developed by *Dantzig, 1949*. The methods due to *Barrodale and Young, 1966* (BY) and *Barrodale and Roberts, 1973 and 1974* (BR), are used to solve (5.1.3.1) in this investigation. For LAD grey-scale matching steps 4,5 and 6 of the iterative procedure described in 5.1.2 are replaced by either the BY or BR algorithms. The rest of the iteration scheme remains unchanged.

Wagner, 1959, is attributed as being the first to formulate L1-estimation as a linear programming problem. The essence of the linear programming approach is to search for an optimum feasible solution in an N dimensional space, where N is the number of constraint equations, so as to optimize an objective function such as equation (5.1.3.2). Previously the L1-estimation problem was solved by inherently geometric methods originally formulated by Laplace in the 18th Century *(Bloomfield and Steiger, 1983)*. The Simplex algorithm provides an efficient iterative search strategy for obtaining the optimum solution. The algorithm procedes by structuring the linear programming problem in the form of a tableau, details of this can be found in *Dantzig, 1949*. An initial basis solution is chosen corresponding to a vertex of the hyper-polyhedron defined in N-dimensional space by the intersection of hyper-planes determined by the constraint equations. The objective function is evaluated and the next vertex is chosen by bringing the parameter variable into the basis solution which causes the objective function to decrease by the greatest amount. This procedure is repeated iteratively until the objective function can no longer be decreased without it becoming negative. The basis that exists at this point is chosen as the optimum feasible solution. Details of this algorithm can also be found in *Dantzig, 1949*.

Barrodale and Young, 1966, improved the efficiency of the Simplex algorithm by restructuring the data so that only the essential items are included in the tableaux. *Barrodale and Roberts, 1974,* then improved the algorithm further by developing a strategy whereby a number of vertices in N-dimensional space are bypassed in one iteration. This improved the speed of the algorithm significantly. Details of the improvements mentioned can be found in the respective papers cited. The C codes for the (BY) and (BR) algorithms are given in Appendix A.

Figure 5.1.3.1. shows how the algorithms converge when matching an aerial image with itself. A window size of 9 X 9 pixels was used with a tolerance of 0.00001. The tolerance determines when

the iterations stop. The affine parameter adjustments must all be less than the tolerance before the matching procedure terminates. LSM indicates the ordinary least-squares matching algorithm with weights equal to unity.

The Y values in figure 5.1.3.1 for the LSM graph, are the normalized σ values where normalization is with respect to the value of σ at the first iteration. Similarly the Y values of the BR and BY graphs are the normalized LAD values with respect to the LAD value at the first iteration. σ and LAD are given by

$$\sigma = \sqrt{\frac{\Sigma r_i^2}{N}} \quad , \quad LAD = \frac{\Sigma |r_i|}{N} \qquad (5.1.3.3)$$

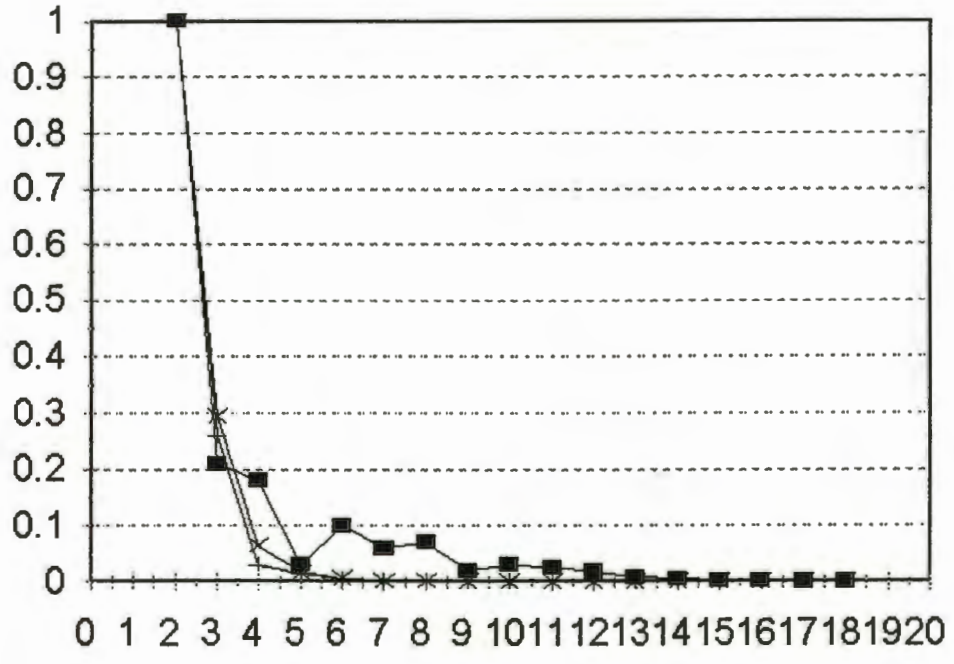


Figure 5.1.3.1. *Convergence comparison of the L-norm algorithms.* ■ *BY , + BR , x LSM.*

All three methods converge to the correct point exactly. The BY algorithm exhibits oscillatory behaviour, but both LSM and BR

converge smoothly.

Each iteration step for each of these methods takes about the same time, but BR and LSM generally converge more rapidly than BY because they require less iterations. Figure 5.1.3.1 shows that the BR algorithm exhibits similar stability to the LSM method, whereas the BY algorithm tends to oscillate.

In order to demonstrate the behaviour of the algorithms in the presence of outliers, a test was performed on a synthetic image with a hotspot as shown in figure 5.1.3.2.

Figure 5.1.3.2. *Template and Patch for outlier test.*

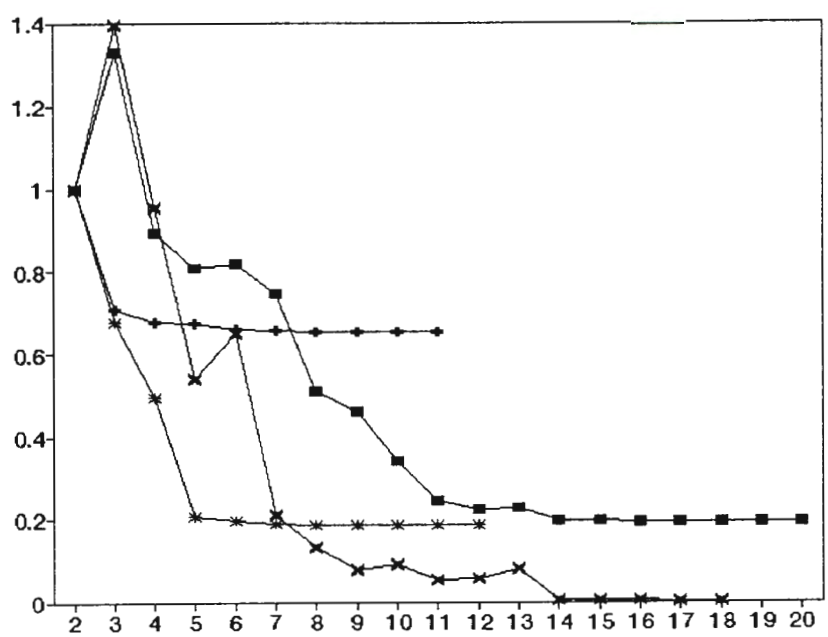


Figure 5.1.3.3. *Convergence of algorithms in the presence of a outlier point.■ IRLS , + LSM , * BR ,X BY.*

Figure 5.1.3.3 shows the convergence behaviour of the various algorithms using a 9 X 9 window including the hotspot. A tolerance of 0.00001 was used for all the algorithms excepting for BY where a tolerance of 0.01 was used. The higher tolerance value was required for BY as it developed cyclic oscillations at the lower tolerance value and did not converge within 50 iterations. The oscillatory behaviour of BY is evident at the lower tolerance value also.

The IRLS algorithm used has iteratively recalculated weights given by

$$P_{ii} = \frac{1}{|r_i| + \epsilon} \qquad (5.1.3.4)$$

where P_{ii} is the diagonal weight matrix element corresponding to observation data point i.

ϵ is an arbitrarily small constant to avoid division by zero.

The weights given by equation (5.1.3.4) are designed to simulate the L1-norm using the IRLS method. Any Lp-norm method with p in the range [1,2] can be simulated in this way using the weight formula given by equation (5.1.3.5).

$$P_{ii} = \frac{1}{|r_i|^{2-p} + \epsilon} \qquad (5.1.3.5)$$

Although the correct point was isolated using IRLS a large number of iterations was required. Using LSM convergence was obtained but the result was 0.4 pixels away from the correct point. The BR algorithm smoothly converged to the correct value in half the number of iterations.

Many researchers *[Schlossmacher, 1973] [Kubik, Merchant and Schenk, 1987]* have investigated using the IRLS algorithm as an alternative to linear programming methods in LAD regression. In 1973 efficient and stable algorithms such as BR did not exist and

the IRLS technique was more stable than the BY algorithm then available. As demonstrated the BR algorithm is stable in matching images with and without outliers. Other recently developed algorithms that perform well in LAD regression applications are the Bartels-Conn-Sinclair and the Bloomfield-Steiger algorithms *[Bloomfield and Steiger, 1983]*.

LAD matching using the BR algorithm can be used as effectively as LSM with the added element of robustness towards outlier points.

The BR and BY algorithms are shown in appendix A with an explanation of the data structure required to use them.

5.2. SUBPIXEL LOCALIZATION USING PATTERN RECOGNITION.

Area-based matching using the Lp-norm methods of section 5.1. results in the matching of a template window in one image with a patch window in another image to subpixel accuracy. An alternative approach to subpixel matching is to extract features from two or more images to subpixel accuracy and then match them with some correspondence algorithm.

In principle the feature vector approach could be used to locate or extract features to subpixel accuracy. The following investigation attempts to show how the feature vector method can be used to achieve this goal. Initially, analogous to the area-based method discussed in section 5.1., an attempt is made to apply the feature vector method to finding a matching point for a selected area in one image, in another image to subpixel accuracy.

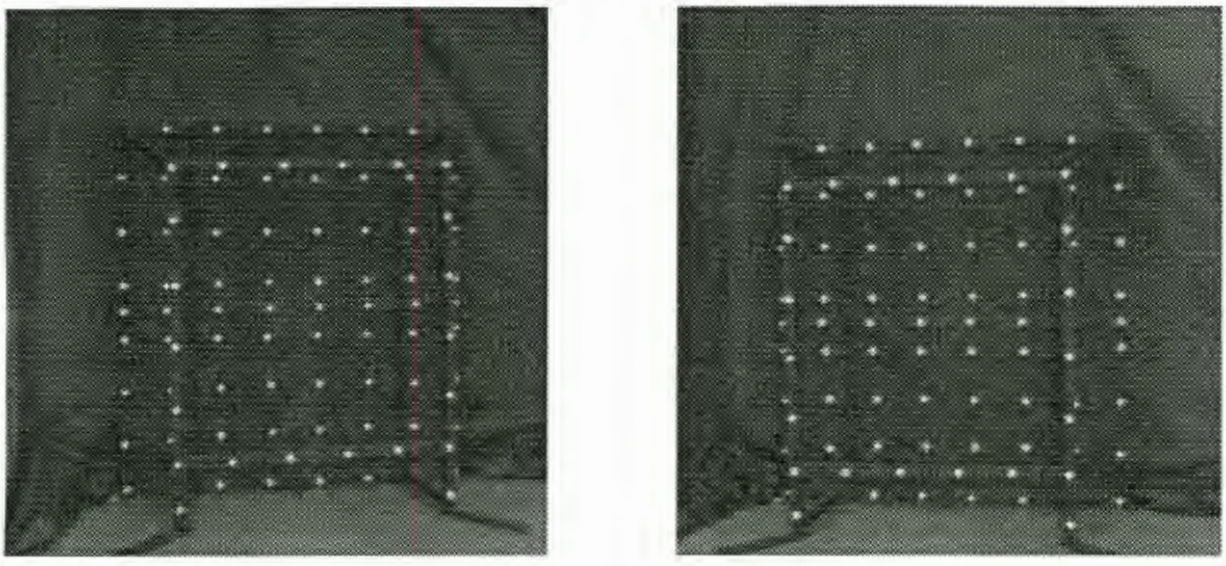
5.2.1 SUBPIXEL FEATURE VECTOR MATCHING

The method of using pattern recognition principles to locate or match objects in a scene outlined in chapter four can be adapted to locate or match objects to subpixel accuracy. The main

application of this technique would be in images of objects close to planar views, such as occurs in vertical aerial photographs, where the effects of perspective do not change the pattern observed significantly. Another application of the moment based techniques discussed in the previous chapter would be the subpixel localization and matching of targets in targeted images, primarily used by close-range photogrammetric applications. This sort of application can be used to illustrate the principle of matching with feature vectors to subpixel accuracy.

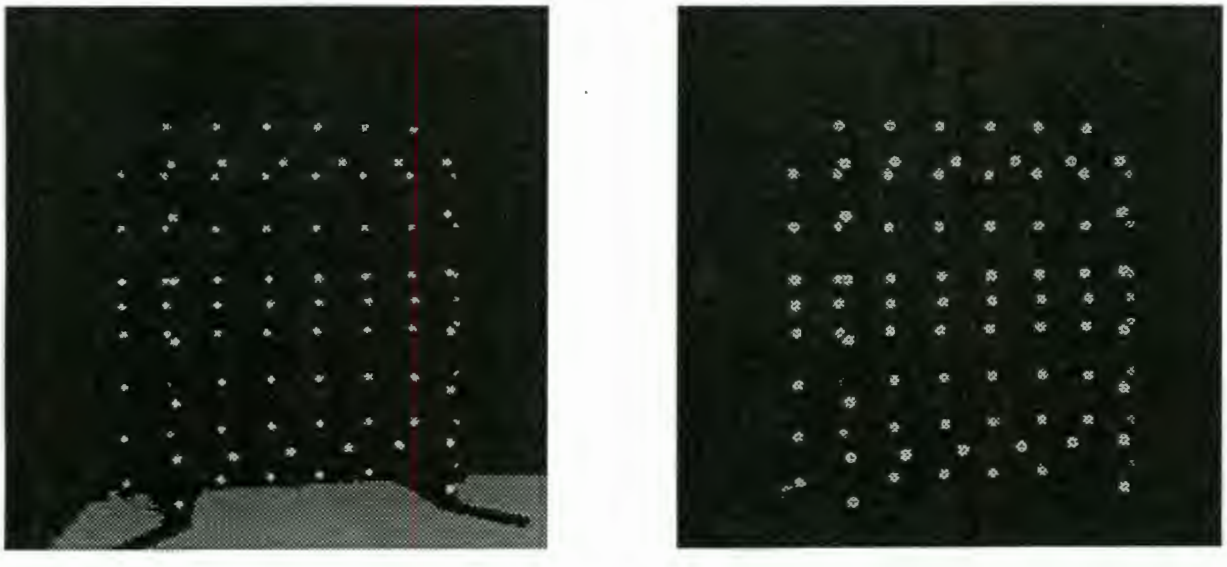
As with least-squares matching a technique for finding the approximate location of the desired object in the scene is necessary. Once a good approximate starting point is obtained the subpixel algorithm can commence. A good approximate location can be obtained using the moment invariant feature vector approach explained previously.

The method of obtaining subpixel location of targets in images will be explained with reference to the targeted images shown in figure 5.2.1.1. The circular targets in figure 5.2.1.1 are to be located to subpixel accuracy. A circle has ∞-fold symmetry and the solid circular patterns produced by the thresholded image will result in the geometric moment invariants being zero excepting for the first moment invariant given by $\mu_{20} + \mu_{02}$. If the targets appeared perfectly circular and there were no quantization effects due to the discrete pixel nature of a digital image, the same would apply to the rings produced by the thresholded gradient image. As the targets will generally appear more like ellipses due to perspective, not all the Hu moments are zero. As an initial experiment to assess the viability of using feature vectors of moment invariants to locate to subpixel accuracy, only the seven Hu moments resulting from the thresholded gradient image were used in the feature vector.



(a) (b)

Figure 5.2.1.1. *Left and right stereopair of a targeted camera calibration frame.*



(a) (b)

Figure 5.2.1.2. *Thresholded images of figure 4(a). (a) Thresholded grey-scale image. (b) Thresholded gradient image.*

The Otsu thresholding method is used to produce the images shown in figure 5.2.1.2. The targets are well brought out due to their

high grey-scale intensity values. In an automated system figure 5.2.1.2(b) can be used as a reference mask to locate the positions of possible matching targets to provide starting points for a subpixel matching technique such as LAD , LSM or the feature vector method.

To illustrate the concept of using moment based feature vectors for subpixel matching the selection of the model pattern is done interactively by pointing with a mouse at the centre of a target circle in the left image of figure 5.2.1.1. The covariance matrix is generated by taking a neighbourhood sample as described previously. The eigenvalues and eigenvectors of the covariance matrix are used to apply the whitening transformation to the vectors after which they can be compared by a Euclidean distance metric. The sample is used to determine the discriminant threshold. The average distance and its standard deviation are determined from the sample and used as shown in equation (4.4.1.3.10) in chapter 4. The model vector is the transformed moment vector of the selected window.

In order to locate the target circle in the right image the mouse cursor was placed near the circle to circumvent the processing required by an automated approximate locating routine. A search procedure based on finding the nearest neighbour region is commenced.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Figure 5.2.1.3. *Numbered pixel points in a 5 X 5 pixel neighbourhood.*

Figure 5.2.1.3 shows numbered pixels in a 5 X 5 pixel neighbourhood. The pattern vector is calculated for the pattern window centred on each numbered pixel and the distance to the model vector determined. The result is a 5 X 5 region of distance values. If the distance value of any of the neighbouring pixels is less than the value at pixel 13, the search proceeds by moving the centre of the 5 X 5 window to the pixel with the

minimum distance value. This is a form of a steepest descent search method. The process continues until no neighbours have distances less than the centre pixel. The subpixel location is then taken as the weighted centre of mass of the 5 X 5 region according to the following formulae:-

$$\bar{x} = \frac{\Sigma w_{xy} x}{\Sigma w_{xy}} \quad , \quad \bar{y} = \frac{\Sigma w_{xy} y}{\Sigma w_{xy}} \qquad (5.2.1.1)$$

$$w_{xy} = \frac{1}{distance_{xy}^{2}}$$

where $distance_{xy}$ is the distance to the model vector at the pixel point position (x,y).

Convergence to the correct central pixel is strongly dependent on the starting position of the search as there may be many local minima. The many methods of finding a starting position used in least squares matching can also be used for this method. The use of a reference mask ,mentioned above ,is probably the fastest method for locating a starting position for the images in figure 5.2.1.2.

In an attempt to avoid the possibility of convergence to an incorrect local minimum ,the pattern distances for a large region centred on a good approximating point can be calculated and used to calculate the centre of mass according to equation (5.2.1.1) without any further steepest descent searches. The contour plot of the distance values for such a region is shown in figure 5.2.1.4. As the circular targets are from real images the contours are not symmetrical due to

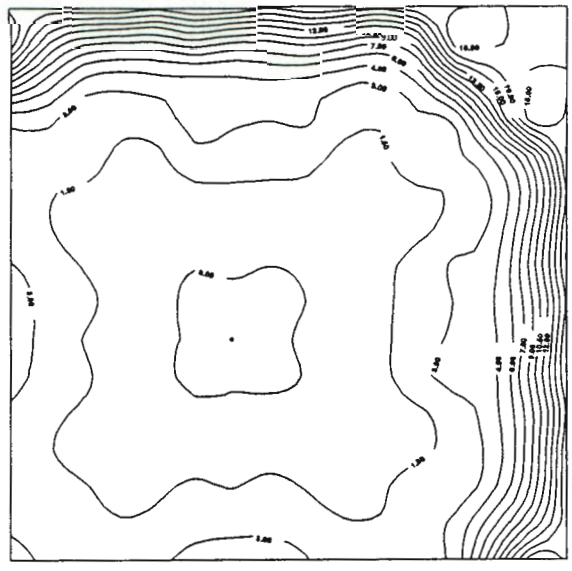


Figure 5.2.1.4. *Contour plot of distance values for a 20 X 20 region including a circular target point.*

quantization effects and intensity variations over the targets. The main advantage of this as opposed to the steepest descent method with a smaller region, is that local image differences can cause the steepest descent method to move to another minimum point. Also, if the initial approximating point for the steepest descent method is on the wrong side of a ridge in distance space then the correct minimum cannot be reached. The main disadvantage of this approach is that neighbouring feature patterns may fall in the larger region and cause the centre of mass calculation to be very inaccurate. Problem specific heuristics can be used to decide on the best approach and the required regional sizes. The target centre is assumed at the minimum distance point of this region. The nature of the contour spacings indicate that a paraboloid fit to the distance value surface may be suitable to determine the global minimum point on this surface for subpixel localization. The general equation of a paraboloid is given by

$$y = ax^2 + by^2 + cxy + dx + ey + 1 \qquad (5.2.1.2)$$

A LAD or Least-squares fitting procedure can be used to determine the coefficients in the equation based on the distance values calculated over the selected region. The position of the minimum point is given by the formulae:-

$$(x, y) = \left(\frac{ce - 2bd}{4ab - c^2} , \frac{cd - 2ae}{4ab - c^2} \right) \qquad (5.2.1.3)$$

In order to obtain an indication of the performance of the subpixel localization methods mentioned above a comparison was done using some of the targets in the image shown in figure 5.2.1.2. The matching point found using the LAD BR algorithm described above was taken as the "correct" matching position. The LAD BR method was used because from the tests with area-based matching techniques in section 5.1. it gave stably converging results that were robust in the presence of outliers. The comparison results , in Table 5.2.1.1, show the image distance in pixel units of the "matching" points of the various pattern

based methods mentioned above from the "correct" position. SD1 and SD2 stand for the centre of mass methods using a steepest descent search with weights 1/distance and 1/distance^2 respectively, where distance is defined as the distance from the model pattern vector in feature vector space. The centre of mass calculation was based on a region of 7 X 7 pixels for SD1 and SD2. REG1 and REG2 represent the regional centre of mass method with weights 1/distance and 1/distance^2 respectively. A region size of 21 X 21 pixels was used. SDPARAB indicates the paraboloid fitting technique with a steepest descent search. The distance values over a region of 7X7 pixels was used for the paraboloid fitting.

Δr

POINT NO	SD1	SD2	SDPARAB	REG1	REG2
1	0.23	0.20	0.98	0.14	0.20
2	2.00	1.05	2.83	0.52	1.46
3	1.77	1.75	1.77	0.80	0.69
4	0.22	0.10	0.51	3.16	3.64
5	1.60	1.40	0.39	1.65	1.79
AVE	1.16	0.90	1.30	1.25	1.56
STD	0.77	0.65	0.91	1.07	1.18

TABLE 5.2.1.1. *Distance from correct position for the different pattern matching subpixel location methods.*

Five different target points were chosen. A more thorough comprehensive test using significantly more points is required to establish true performance. A few points were chosen in order to gain some indicative relative measure of performance and to illustrate the principle of the methods. The results indicate that SD1, SD2 and SDPARAB have the lowest standard deviations and hence the best average performance. REG1 and REG2 perform well excepting for points 4 and 5 because the latter points have similar targets within the region window chosen, and the centre of mass calculation will be distorted as the position of these other targets will be given high weighting. Region size is therefore dependent on the local image contents. Heuristically 7 X 7 pixel regions were found to give more stable results for relatively wide variations in local image content. This is shown by the results of SD2 and SDPARAB. The pattern window size used in the above experiments was 7 X 7 pixels whereas the LAD BR algorithm matching reference points were based on 9 X 9 windows.

The main advantage of the pattern recognition approach for subpixel localization is its relative speed in comparison to the IRLS and LAD methods. Using a PC the SD2 method was about a factor of 4 faster than the LAD BR method for a point matching. Besides the hardware platform used, speed of computation is a function of the efficiency of the written software. The comparison above is based on unoptimized code, but is a good indicator of the relative speeds.

The main disadvantage of the pattern recognition approach as used in the above experiment is that there is no modelling of the affine transformation between the two images. A sampling procedure can be devised to incorporate affine transformation parameter variations as explained in chapter 4, but the computation cost is great.

The pattern recognition approach to subpixel localization is analogous to grey-scale correlation matching discussed in section 5.1 above. The distances are similar to the distance between the

template and patch defined in equation (5.1.2.1). One could therefore do a similar subpixel location based on the distances between template and patch over an image region. The pattern recognition approach would emphasize the structural content of the image as thresholded edges are used whereas the grey-scale function, although implicitly containing the structural information , will be strongly affected by variations in the radiometric quality of the image. This would tend to indicate that the pattern recognition results on this basis would be more robust. Further investigations and comparisons between established centre of mass techniques and the L-norm techniques discussed previously in this chapter are required. The main emphasis here was to illustrate the principle and potential of the method.

5.2.2 PATTERN RECOGNITION BASED DISTANCE IMAGE TRANSFORM

Experimentation with the steepest descent search algorithm in distance space prompted the idea of transforming an image into an image representation of its distance surface. This can act as a visual representation of the areas in the image that are similar to the model feature vector chosen. As such it can be used to segment the image into regions where the object of interest is most likely to be found.

The distance transform can also be used as an intermediary step for incorporating an explicit affine transformation into pattern recognition matching. An image is built up by calculating the distance between the model feature vector and sequentially extracted vectors over the whole image. The distance associated with a window surrounding a particular point is converted into a grey-scale by suitably normalizing the value in the following way

$$gd = \frac{distance_i}{dmax}\, gmax \qquad (5.2.2.1)$$

where

gd is the grey-scale value associated with $distance_i$.

$distance_i$ is the distance associated with point i in the image.

dmax is the maximum distance of the model feature vector from an extracted pattern vector in the image.

gmax is the maximum grey-scale value (gmax=255 in this case).

Figure 5.2.2.1 shows the transformed image of figure 5.2.1.1. The matching pattern positions are brought out indicating the usefulness of this method in extracting specific objects from an image. The other major advantage of this transformation is to convert the piece-wise continuous grey-scale image into a relatively smooth image. The distance measure integrates regional information and therefore effectively smooths the image.

The smooth nature of the transformed image enables IRLS and LAD matching to converge stably at image discontinuities and in some situations where the affine parameters would be indeterminate in the original image *[Baltsavias, 1991]*. An example of this is a perfect corner point where the two scale parameters of the affine transformation would be indeterminate. Figure 5.2.2.2(a) shows a synthetically produced image with square region at a different grey-scale value to the background value. The template is taken as a window centred on the top left corner point. The image corner is then matched to itself by selecting the patch centre near the corner point. Due to the indeterminate nature of the scaling affine parameters in this situation the IRLS method and LAD diverge. By matching the corner point from figure 5.2.2.2(b), the distance transformed image of figure 5.2.2.2(a), both matching methods converge to the correct point.

(a) (b)

Figure 5.2.2.2. *The distance transformed images of the stereopair in figure 5.2.1.1.*

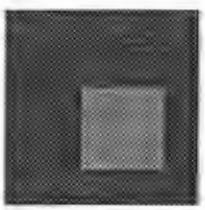


Figure 5.2.2.3 (a) *Synthetic image*

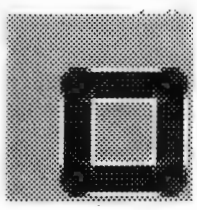


Figure 5.2.2.3 (b). *Distance transform*

Matching of discontinuities or affine indeterminate areas can be done using the distance transform. The template and patch are selected in the original images ,as done with direct area-based methods, and the matching performed on the transformed images. Figure 5.2.2.3 shows a synthetic image with its distance transform.

5.3. MATCHING EXTRACTED FEATURES

Once points are extracted from images to subpixel accuracy they have to be matched. The problem is similar to the consistent labelling problem described in chapter 2. If two images are considered then the correspondence between two lists of point positions to subpixel accuracy has to be established.

As suggested in section 5.2 the lists of point positions can be found using the feature vector approach. The points can then be matched using relaxation matching or dynamic programming matching.

To illustrate the principle of this approach an example where feature points are extracted from two images using a connected

components algorithm and then matched using a dynamic programming method is described.

5.3.1 SUBPIXEL TARGET POINT EXTRACTION WITH CONNECTED COMPONENTS.

An alternative approach to the feature vector method described above using connected components can be used to great advantage for matching high - contrast targeted points in close-range photogrammetry. It also provides a faster means of extracting target centre points to subpixel accuracy.

The image in figure 5.3.1.1(a) is converted to a binary image by thresholding and the targets are labelled using a connected components labelling algorithm. The binary image has pixels which are either 'on' or 'off'. The algorithm searches the image from top to bottom and left to right. The first 'on' pixel to be detected is assigned the label 1. When subsequent 'on' pixels are detected the algorithm checks if they have 'on' neighbours. (A neighbour pixel is any of the immediately adjacent pixels. For a normal square pixel array each pixel will have 8 adjacent neighbours.) If they do then they are assigned the label of their labelled neighbours. If they have no labelled neighbours they are assigned a new label. As only one pass is made over the image from left to right and top to bottom, then for the particular pixel to be labelled, the only possible labelled neighbour pixels will be those above and to the left of it in the image. The final labelled pixels are represented by a grey-scale intensity image with each label having a different grey-scale value as shown in figure 5.3.1.1(b).

For circular target regions, like those in figure 5.3.1.1(a), this one pass connected components algorithm uniquely labels each target correctly. Where there are irregularly shaped connected regions or connected regions with holes in them, different labels can occur within a connected region. At least a two pass algorithm will be required to correctly label these irregular regions. Many other connected component algorithms are

available *[Haralick, 1981] [Rosenfeld and Pfaltz,1966] [Aho, Hopcroft and Ullman,1983] [Horowitz and Sahni,1982] [Lumia,Shapiro and Zuniga,1983] [Ronse and Devijer,1984] [Wildschek, 1989].*

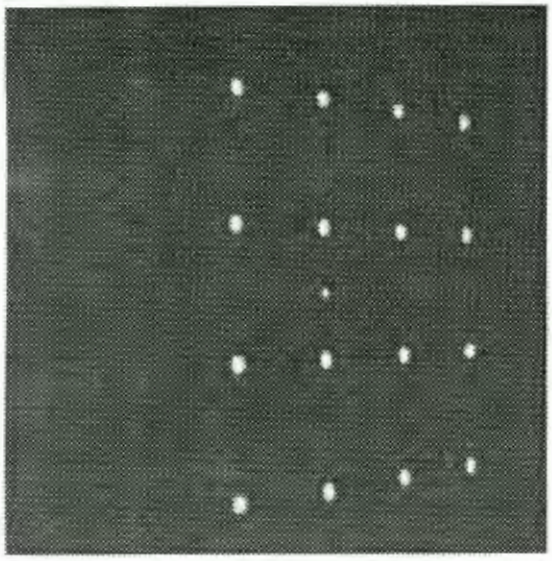


Figure 5.3.1.1(a). *Targeted plane region.*

Figure 5.3.1.1(b). *Labelled connected components of figure 8(a).*

Once all the components are labelled then the centre of mass of each distinct labelled region will give the target location to subpixel accuracy.

$$(\overline{x}^{(L)},\overline{y}^{(L)}) = \left(\frac{\sum_{i}^{n^{(L)}} x_i^{(L)} g_i}{\sum_{i}^{n^{(L)}} g_i}, \frac{\sum_{i}^{n^{(L)}} y_i^{(L)} g_i}{\sum_{i}^{n^{(L)}} g_i}\right) \qquad (5.3.1.1)$$

where

$(\overline{x}^{(L)},\overline{y}^{(L)})$ are the coordinates of the centre of mass of the region labelled L.

$(x_i^{(L)},y_i^{(L)})$ are the coordinates of a point belonging to a region labelled L.

$n^{(L)}$ are the number of points in region L.

g_i is the grey-scale value of the pixel at position $(x_i^{(L)},y_i^{(L)})$

Using this method to locate targets to subpixel accuracy for a stereopair of images similar to figure 5.3.1.1.(a) results in lists of coordinates of the target locations in the left and right images respectively.

5.3.2 MATCHING EXTRACTED POINTS

Having obtained the list of coordinates the next step is to establish correspondence between the matching points. There are various approaches to this problem. The simplest method is to use area-based matching between each point in one list with every point in the second list and selecting the best match as correct. Area-based matching may not be the most efficient or reliable technique due to the similarity of the target point regions.

An alternative method is to use tree search algorithms based on some similarity and consistency criterion. A more efficient adaptation of this method is relaxation matching *[Haralick, 1993]*. Dynamic programming can also be used once a suitable cost function has been defined. Dynamic programming algorithms are often used in the computer recognition of speech *[De Mori and Probst, 1986]* and similar methods can be adapted to image matching.

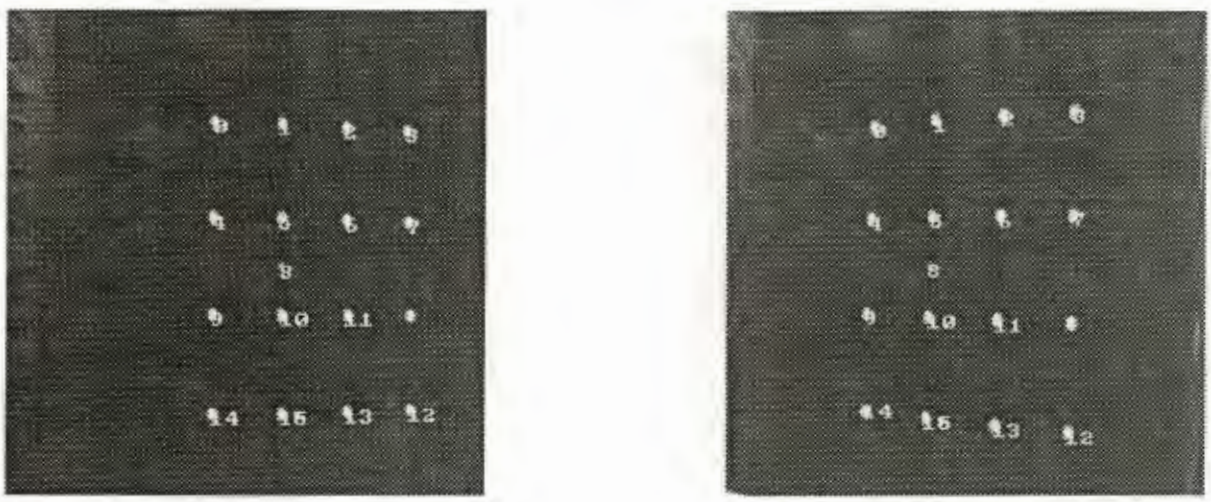


Figure 5.3.2.1. *Result of matching targets using dynamic programming.*

Figure 5.3.2.1 shows the results of matching the located

targets in the left image to those in the right image using a dynamic programming technique.

Let the list of points in the left and right images be in sets A and B respectively. Starting with the first element in A say A1 find all the elements in B that are closer to A1 than a specified disparity range. Let these points be in subset D. Assume A1 is matched to each element of D in turn. For each instance, determine matches for the remaining unmatched points in A. Calculate a cost value using equation (5.3.2.3) for each possible match, and take the match with the minimum cost as the correct match. Accumulate all these minimum cost values for the assignments made. The result is a total cost value for each initial possible match for A1. The point in D that produces the total minimum cost is taken as the correct matching point. The same process is then applied in turn to each successive point in A. Elimination of points that are already matched is carried out to accelerate the algorithm.

The basic equations required for this method are briefly described. {n} is the set of N elements in A. {m} is the set of M elements in B. For every n a subset $\{j\}=D_n$ of B is determined so that the disparity between the points in D_n and n are less than a specified disparity range. Starting with an element n in A determine the total cost associated with each j in its associated subset D_n as

$$C(j) = \sum_{l=n+1}^{N} \sum_{i \in \{\text{unmatched points} \in B\}} c(i,l) \qquad (5.3.2.1)$$

where

C(j) is the cost associated with $j \in D_n$.

c(i,l) is the cost determined by equation (5.3.2.3), where i is an unmatched point in B, and l is an unmatched point in A.

The point $j \in D_n$ with the minimum cost is taken as the matching

point to n.

This process is then applied for each of the remaining unmatched points in A. The total costs of the matching are related at any stage by the following formula:-

$$TC(n+1) = \min_{j \in D_{n+1}} C(j) + TC(n) \qquad (5.3.2.2)$$

where

TC(n) is the total cost after n points have been matched.

The cost function used to evaluate each possible match is given as

$$Cost_{il} = \alpha\, DispD_{il} + \beta\, Correl_{il} \qquad (5.3.2.3)$$

where

$Cost_{il}$ is the matching cost associated with point l in set A and i in set B.

$DispD_{il}$ is the disparity difference between the disparity between n in A and j in D_n and the disparity between point l in A and i in B.

$Correl_{il}$ is the grey-scale difference for windows around points i and l respectively.

α and β are heuristically determined constants.

Using optimization techniques for these types of problems has considerable potential ,as the matching problem can be regarded as an optimization problem.In essence a certain configuration of point assignments needs to be determined so that an optimal value of an evaluation or cost function is achieved. Determining a suitable cost function is the one of the main problems to solve.

5.4. CONCLUSION

As subpixel matching is essential for the surface reconstruction of objects from two or more images, the applicability of the pattern vector approach to subpixel matching was investigated.

Improvements to the frequently used least-squares area-based matching technique were considered. The Barrodale-Roberts LAD algorithm proposed in section 5.1.3 above is found to give better matching accuracy in the presence of outliers in images, with similar convergence stability and computational speed.

A method for using the pattern recognition approach for subpixel matching was proposed and compared with area-based matching techniques. The area-based techniques are more stable and better understood at present, but there appears to be some potential in pursuing the pattern vector approach as computational speed may be improved.

A pattern vector based distance transform was introduced which can facilitate localization of features and may be used for image segmentation. The distance transform may be applied as an image preprocessor to improve area-based matching at affine indeterminate locations. The application of the pattern vector approach for the location of targets was shown as an illustrative example.

A pattern vector approach may be used to extract feature points to subpixel accuracy from one or more images. These are then subsequently matched using some matching algorithm. To illustrate this principle a connected component technique for isolating targets to subpixel accuracy and matching using a dynamic programming technique was investigated. As matching is essentially an optimization problem research in the application of optimization techniques in general may prove advantageous.

6. DETERMINATION OF MATCHING AND RELATIVE ORIENTATION PARAMETERS

6.1. INTRODUCTION

The objective of digital photogrammetry is to automate the process of obtaining three dimensional information from two dimensional views of an object or scene. Most of the techniques developed to date rely on the use of measured control points to provide the exterior orientation of the cameras in the object coordinate system. A minimum of two images of the object taken from different positions is required.

The three dimensional coordinates of the matched points can then be calculated using one of the established photogrammetric algorithms. An interpolation technique in object space is then used to infer the structure of the object.

A general automated procedure would operate in the absence of knowledge of point positions in the object space coordinate system or the object dimensions. The only way this can be achieved is to obtain the object coordinates relative to the positions of the two or more observation cameras. In the method to be discussed only a two camera system is assumed. In order to facilitate the explanation to follow, the cameras will be referred to as the left and the right camera respectively.

The perspective centre of the left camera is taken as the origin of the relative coordinate system. The distance between the two cameras is measured and taken as the x coordinate of the perspective centre of the right camera. This measurement need not be very accurate as the relative orientation algorithm will compensate for this inaccuracy by adjusting the remaining parameters in the solution.

In order to obtain three dimensional information from the images, identifiable points or features need to be unambiguously extracted in each image. Matching or correspondence between the features or points needs to be accurately established. As the object space coordinates of the objects giving rise to the features in the images are unknown, it is impossible to determine the position of the cameras in object space coordinate system. However if one has matching points in each image it is possible to determine the relative orientation of the right camera in the camera world coordinate system of the left camera because of the geometrical relationships that exist between the object and image space.

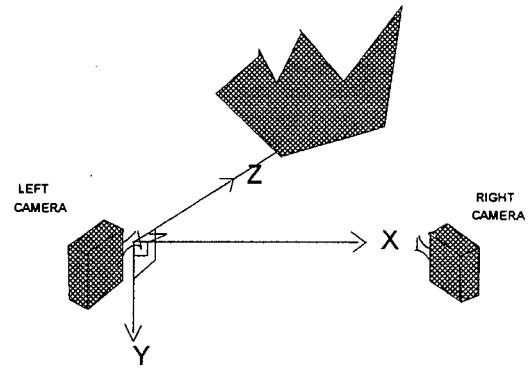


Figure 6.1.1. *Schematic diagram of camera world coordinate system.*

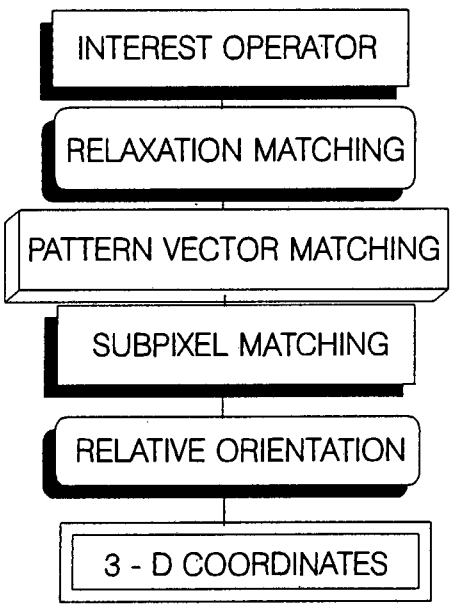


Figure 6.1.2. *Basic strategy for automatically obtaining three dimensional information from an aerial stereopair.*

An general automated system for obtaining three dimensional information from a stereopair of aerial images could be designed with the main components shown in figure 6.1.2. An interest operator is applied over the two images and the location of interesting points determined. A continuous relaxation matching algorithm can then be invoked to establish correspondence between the extracted points in the two images. The provisional matches are then checked with a pattern vector algorithm, to eliminate obvious mismatches. The resulting list of matches are then matched to

subpixel accuracy using a Lp-norm matching technique. Potential matches that do not converge to a subpixel match within a specified number of iterations are discarded as mismatches. The matches in pixel coordinates are then converted to image coordinates. These are then used to determine the relative orientation parameters between the cameras. A novel method of simultaneously checking the consistency of the matches while solving for the relative orientation parameters is suggested. Points that are inconsistent are discarded, and the three dimensional coordinates, in the coordinate system defined by the perspective centre of the left camera, are calculated for the remaining matches.

6.2.THE COPLANARITY CONDITION EQUATION AND RELATIVE ORIENTATION

The geometrical relationship between a stereopair of images enables the derivation of the coplanarity condition equation. This equation is used as the basis for determining the relative orientation between the two images.

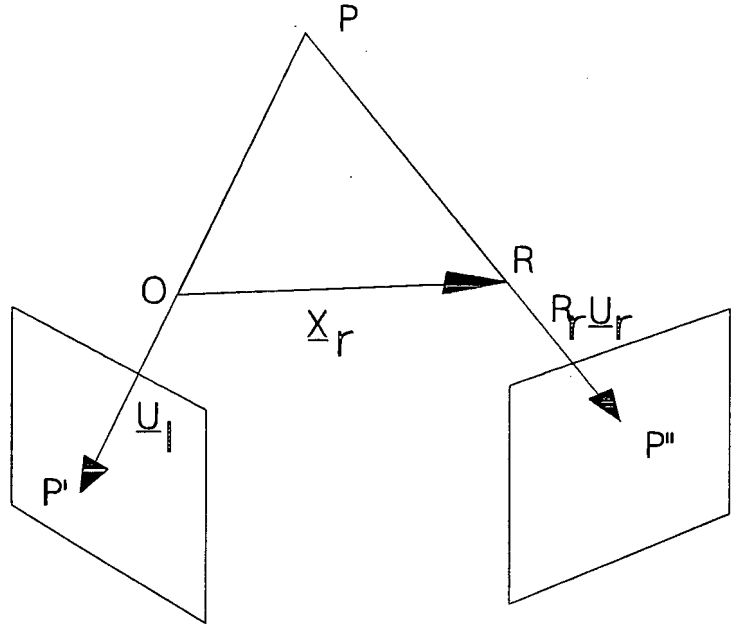


Figure 6.2.1. *Geometry of a general image pair.*

By setting the perspective centre of the left camera O in figure 6.2.1 as the origin of the coordinate system and letting the rotation matrix of the left image be equal to the identity matrix the coplanarity condition equation is given by:-

$$\underline{X_r} \cdot [\, R_r \, \underline{U_r} \times \underline{U_l}] = 0 \qquad (6.2.1)$$

where the terms of the equation are as follows with reference to figure 6.2.1.

$\underline{X_r} = (X_r, Y_r, Z_r)$ is the position vector of the perspective centre of the right camera R.

The perspective centre of the left camera is taken as the origin O.

R_r is the transpose of the rotation matrix with relative orientation angles ω, ϕ, κ to the left image, which is taken as the reference position.

m_{ij} are the matrix elements of the rotation matrix given by

$$R = \begin{bmatrix} m_{11} & m_{21} & m_{31} \\ m_{12} & m_{22} & m_{32} \\ m_{13} & m_{23} & m_{33} \end{bmatrix}$$

where

$m_{11} = \cos\phi\cos\kappa$

$m_{12} = \sin\omega\sin\phi\cos\kappa + \cos\omega\sin\kappa$

$m_{13} = -\cos\omega\sin\phi\cos\kappa + \sin\omega\sin\kappa$

$$m_{21} = -\cos\phi\sin\kappa$$

$$m_{22} = -\sin\omega\sin\phi\sin\kappa + \cos\omega\cos\kappa \qquad (6.2.2)$$

$$m_{23} = \cos\omega\sin\phi\sin\kappa + \sin\omega\cos\kappa$$

$$m_{31} = \sin\phi$$

$$m_{32} = -\sin\omega\cos\phi$$

$$m_{33} = \cos\omega\cos\phi$$

$\underline{U_r} = (u_r, v_r, -f_r)$ are the image point coordinates in the right image .

f_r is the principal distance of the right camera.

$\underline{U_l} = (u_l, v_l, -f_l)$ are the corresponding point image coordinates in the left image.
Image coordinates are measured relative to the perspective centres of their respective cameras.

A brief derivation of the coplanarity condition equation (6.2.1) is given below with reference to figure 6.2.1.

$\underline{u_l}$ is the position vector of image point P' on the left image in the image space system, with O taken as the origin, and the axis direction defined by the vector from O to the principal point in the left image.

$R_r\underline{U_r}$ is the position vector of the image point P'' on the right image relative to point R the perspective centre of the right camera. The coordinates of the image point are rotationally transformed so that the right image is rotated into the image space system coordinates of the left image.

$\underline{X}_r + R_r\underline{U}_r$ is the position vector of P'' relative to O.

Points P' and P'' lie in the plane defined by ORP.

$(\underline{X}_r + R_r\underline{U}_r) \times \underline{U}_l$ defines a vector orthogonal to the plane ORP.

$\underline{X}_r \cdot [(\underline{X}_r + R_r\underline{U}_r) \times \underline{U}_l] = 0$ as $\underline{X}_r$ lies in the plane ORP.

Applying the distributive law for scalar products leads to

$$\underline{X}_r \cdot \underline{X}_r \times \underline{U}_l + \underline{X}_r \cdot R_r\underline{U}_r \times \underline{U}_l = 0$$

and as $\underline{X}_r \cdot \underline{X}_r \times \underline{U}_l = 0 \quad (\underline{X}_r \perp \underline{X}_r \times \underline{U}_l)$

one obtains $\underline{X}_r \cdot R_r\underline{U}_r \times \underline{U}_l = 0$ which is the coplanarity equation given by equation (6.2.1).

Once the image coordinates of the matching points are accurately extracted and matched the coplanarity equation can be invoked to solve for the relative orientation parameters $Y_r, Z_r, \omega, \phi, \kappa$. Usually X_r is assumed known and is taken as the distance between the two cameras. Generally if any one of the six parameters is known or specified then the solution of the other five can be obtained.

6.3.CONTINUOUS RELAXATION MATCHING

Continuous relaxation matching is mentioned in chapter 2 and provides a general framework for a variety of matching applications. This generality makes it an attractive method to consider in a possible automated system as its probabilistic nature incorporates fuzzy logic and degrees of belief. *Price,*

1985 gives the theory of this method with some illustrative applications. The method was used in the experiments conducted in this investigation so that no a priori knowledge need be assumed for the matching of images.

Generally when matching two images, features are extracted from each image with some associated feature value. An example of a feature value would be the grey-scale intensity values of the region surrounding and including the feature. As the first step towards matching the features between images the initial probabilities of the potential matches are determined based on a similarity measure such as the difference between the feature values. These probabilities are then iteratively adjusted according to the contextual compatibility of the potential matches. The compatibility is based on the similarity of the potential matches of neighbouring extracted features to the potential match under consideration. An example would be the disparity between a feature and its potential match in the second image. In this case a large number of neighbouring features would be expected to have a similar disparity with their respective potential matches. If no neighbours have matches with a similar disparity then the amount of neighbour support to the particular match under consideration would be zero. The formula used to update the matching probabilities are designed to increase the probability of a match if neighbour support is significant or alternatively reduce the probability if there is little neighbour support. The basic flow diagram of the iterative relaxation procedure is shown in figure 6.3.1.

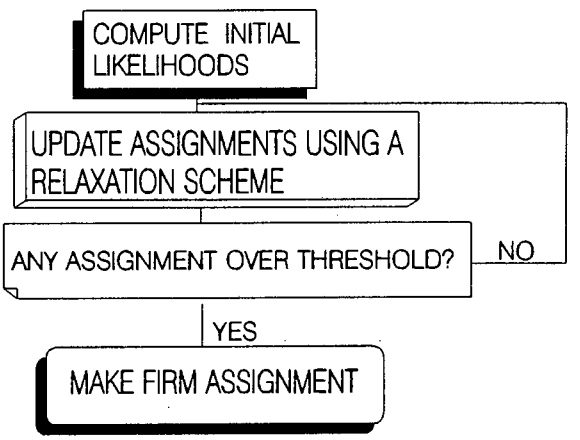


Figure 6.3.1. *Basic flow diagram of the continuous relaxation method.*

6.3.1. DISPARITY RELAXATION MATCHING

In order to illustrate the application of continuous relaxation matching the method due to *Barnard and Thompson, 1980* is discussed in detail, as it is the method used in this investigation to perform initial pixel matching.

Extracting suitable points from each image to use for matching is usually scene dependent and is discussed in detail in chapter 4. In the following it is assumed that there are plentiful strong grey-scale gradients evenly distributed over the scene to provide matchable points. Interest operators can be applied to extract points in both images and are discussed in more detail in chapter 4. For the experiments conducted on this method points were extracted using an angle based interest operator described in chapter 4. For automation to be possible the point extraction has to ensure that corresponding points in the scene are extracted. Corners or edges on objects usually transform to areas in the image with large grey-scale gradients, and usually provide the points which most interest or edge detector operators extract in the images.

Assuming that an evenly distributed set of points is extracted from each image, automatically matching corresponding points without any a priori knowledge of the scene can be done by considering the disparity between the two images.

Disparity is the geometrical shift of corresponding points between the two images. The geometrical shift in the image is caused by viewing the scene from different positions in space. Disparity can be used to match corresponding points.

In order to use disparity to match images the extracted points from each image should have the following properties *(Förstner, 1982)* :-

a) Distinctness - Points must be sufficiently distinct in order to be extracted from both images. Suitably chosen interest operators, edge detectors or pattern vectors based on a priori

knowledge of the scenes to be measured can be used to extract distinct points in each image.

b) Seldomness - The extracted points should be sufficiently rare to reduce the dimension of the matching problem. If the dimension of the problem is not reduced one would be left with the extreme situation of matching every pixel between the images.

c) Similarity - A measure of how closely two points resemble each other needs to be easily established from the nature of the pixel data in the area surrounding the points. This measure is used to establish an initial probability for the two points being in correspondence.

d) Consistency - A measure of how well a particular match conforms to neighbouring matches is required to ensure consistency between matching points in a neighbourhood. A continuous space edge will transform into a continuous image edge. The disparity between two images of the edge should therefore vary continuously. A discontinuity in the disparities between the images may arise at the image boundaries of objects in the scene. The disparity of a point in an image will be similar to the disparities of some points in its neighbourhood. This property can be used to act as support to the likelihood of any particular point in one image matching a point in the other image in a corresponding neighbourhood.

In order to illustrate how disparity can be used to match points in two images a method is briefly outlined for the point marked x_4 in the depiction of two windows of extracted points in figure 6.3.1.1. The size of the window chosen will depend on a priori knowledge of the images by visual inspection and should be greater than the maximum disparity between the two images for any particular pair of matching points. In a truly automated system the whole image could be used. The circle in the right image marks pixel coordinates of point x_4 in the left image. One

can assume that both images are at the same resolution and are at approximately the same scale. The lines in the right window depict all possible disparities between the left and right image for point x_4. A list of all possible disparities for point x_4 is made. The x and y component of each disparity is labelled and stored. Added to the list of potential disparities must be a label indicating that no match is possible. This item is necessary for the situation where for a particular point in the left image no corresponding point was extracted in the right image. Each disparity in the list represents the amount by which the pixel position of an extracted point in the left image has to be shifted to correspond with a potential matching point in the right image.

An initial probability representing the likelihood of a match is assigned to each item in the list of disparities, including an initial probability for the no match item. The probability is based on some measure of similarity between the two potential matching points.

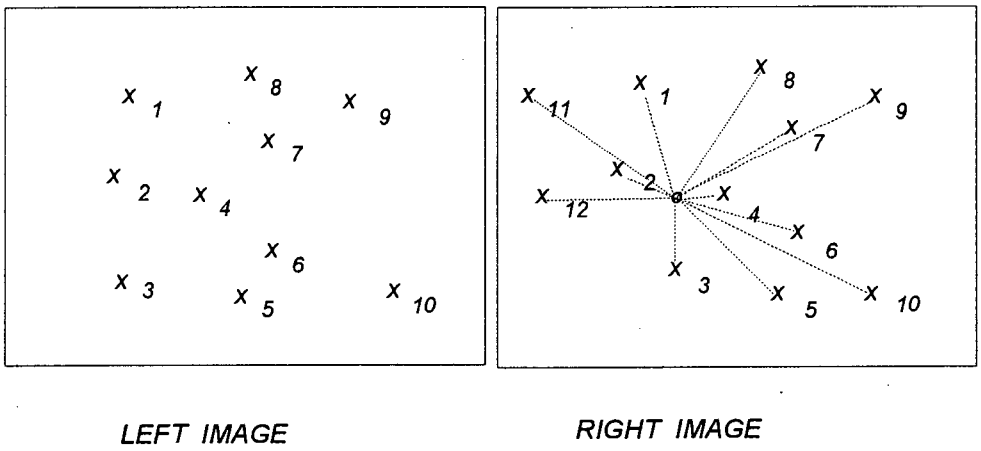


Figure 6.3.1.1. *Extracted interest points in left and right images. Possible matching disparities for point x_4 in left image shown in the right image.*

After initial probabilities are assigned to the potential

disparities of point x_4 , they are iteratively updated by using the consistency criteria mentioned above. The probability attached to a particular disparity is increased if the neighbouring points to x_4 ,e.g. x_5,x_6 have a similar disparity in their disparity lists. The probabilities are iteratively updated and a matching point confirmed when a probability in the list exceeds a preselected probability threshold. The flow of the method is briefly depicted in figure 6.3.1.

In order to derive the formulae for the iterative procedure the general situation is considered where the left image is referred to as image 1 and the right image as image 2. For each point (x_i,y_i) in image 1 construct a set of labels L_i of all possible disparities to points in image 2 within a preselected disparity range. The disparity range is determined by a priori inspection of the images. Research into a possible method to generally computerize this decision needs to be undertaken. The disparity range must be sufficiently large to include the true matching point in the disparity list, but need not include all points in the image. The latter situation would lead to an excessive amount of computation being necessary. Each label in L_i is a vector $l_{ij}=(l_x,l_y)$ such that $lx_{ij}=x_i-x_j'$ and $ly_{ij}=y_i-y_j'$ and (x_j',y_j') is a point in image 2. lx_{ij} and ly_{ij} are integers in the range $[-r,r]$, where r is the preselected disparity range. The element in L_i representing no possible match is indicated by l_i^*.

In order to determine initial probabilities to assign to each disparity in the disparity list associated with a point a measure of similarity is invoked. For each point (x_i,y_i) in image 1 , referred to as a node a_i, and for each disparity l_{ij} in the disparity list of a_i select a small window (typically 5x5) around (x_i,y_i) in image 1 and $(x_{ij}+lx_i , y_{ij}+ly)$ in image 2. Calculate $s_i(l_{ij})$ which is a measure of similarity as follows:-

$$s_i(l_{ij}) = \sum_{m=-d}^{d} \sum_{n=-d}^{d} (f(x_i+m, y_i+n) - g(x_j'+m, y_j'+n))^2 \qquad (6.3.1.1)$$

where d is equal to 2 for a window size of 5X5. f and g are the

grey-scale values of the pixels at the indicated pixel coordinates in image 1 and 2 respectively. The formula shows that the measure of similarity chosen is the difference in the grey-scale values for a window surrounding a particular point. The smaller this value the greater the similarity between the two windows. Hence the probability of matching is a function of the reciprocal of $s_i(l_{ij})$.

For each label l_{ij} a matching probability is determined $p_i(l_{ij})$ such that

$$\sum_j p_i(l_{ij}) + l_i^* = 1 \qquad (6.3.1.2)$$

The summation is over the disparities in the list associated with a_i. Initial estimates of these probabilities are determined by evaluating weights as follows:-

$$w_i(l_{ij}) = \frac{1}{1 + c * s_i(l_{ij})} \quad , \quad l_{ij} \neq l_i^* \qquad (6.3.1.3)$$

where c is a positive constant with a value representing the weighting given to the similarity measure $s_i(l_{ij})$ in the initial probability estimate. c is heuristically chosen according to the range of numerical values associated with $s_i(l_{ij})$ (typically c=10).
An initial estimate for the probability of no match represented by l_i^* is not determined in this way as there is no way of estimating it from the pixel data.

The initial value for the probability that point i has no matching disparity between the two images is chosen as

$$p_i^0(l_i^*) = 1 - \max(w_i(l_{ij})) \qquad (6.3.1.4)$$

This is justifiable as the maximum value of $w_i(l_{ij})$ occurs when

$s_i(l_{ij})$ is a minimum as is most likely to occur at a matching point. Hence $p_i^0(l_i^*) \Rightarrow 0$ as $w_i(l_{ij}) \Rightarrow 1$.

The initial probability that point i has disparity l_{ij} is then given as

$$p_i^0(l_{ij}) = p_i(l_{ij}|i) * (1 - p_i^0(l_i^*)) \quad , \quad l_{ij} \neq l_i^*$$

$$p_i(l_{ij}|i) = \frac{w_i(l_{ij})}{\sum_k w_i(l_{ik})} \qquad (6.3.1.5)$$

where $(1 - p_i^0(l_i^*))$ is the estimate of the probability that point i has a disparity label at all. $p(l_{ij}|i)$ is the probability of the disparity being l_{ij} for a given point i. The summation index k is over all the disparity labels in the set L_i excluding l_i^* as $p(l_{ij}|i)$ assumes that a matchable disparity exists.

The initial probabilities derived will satisfy equation (6.3.1.2) as

$$\sum_j p_i(l_{ij}) = \frac{\sum_j w_i(l_{ij}) * (1 - p_i^0(l_i^*))}{\sum_j w_i(l_{ij})} + p_i^0(l_i^*) = 1$$

After having set up the initial probabilities for all the disparity labels attached to all the points in image 1, the probabilities are iteratively updated by using the consistency property mentioned above. The procedure is described as follows:-

Let l_{jm} be a disparity label from node j a near node a with disparity label l_{in} in image 1. Consistency requires that the disparities of neighbouring points in an image should be very close. Hence

$$\| l_{in} - l_{jm} \| \leq \Theta \qquad (6.3.1.6)$$

where Θ is an appropriate threshold and is usually chosen as 1

or 2 pixels but is dependent on the size of the neighbourhood chosen.

A node a_j is considered near a_i if in image 1 the condition

$$\max(|x_i-x_j|, |y_i-y_j|) \leq R \qquad (6.3.1.7)$$

is satisfied. R is typically chosen as 20 but depends on a priori information provided by viewing the images and estimating a suitable value for R. If neighbouring points are scarce in the vicinity of node a_i R should be increased. On the other hand R may not be so large that points from regions with obviously different disparities are included violating the continuity condition required for neighbourhood support. For an automated system the interest operator chosen should select sufficient points to enable a reasonably small value of R to be used.

At the kth iteration determine intermediate parameter values $q_i^k(l_{in})$, for all l_{in} in the disparity label set L_i associated with node a_i. These values represent the degree of consistency for a particular disparity l_{in} at node a_i with its neighbouring nodes a_j.

$$q_i^k(l_{in}) = \sum_j [\sum_m p_j^k(l_{jm})] \quad , \; l_{in} \neq l_i^* \qquad (6.3.1.8)$$

where the summation index j is over all the disparity labels associated with node a_j which satisfy condition (6.3.1.6). The outer summation is over all the nodes a_j satisfying (6.3.1.7) in the neighbourhood of a_i , but not including a_i.

The q^k_i are used to update the probabilities for the next iteration k+1 by determining an intermediate unormalized value as follows:-

$$\underline{p}_i^{k+1}(l_{in}) = p_i^k(l_{in}) * (A+B*q_i^k(l_{in})) \quad , \; l_{in} \neq l_i^*$$
$$\underline{p}_i^{k+1}(l_i^*) = p_i^k(l_i^*) \qquad (6.3.1.9)$$

Typically A=0.3 and B=3. B determines the rate of convergence. A ensures that the new probability is never zero. A and B are known as damping and gain factors.

$$p_i^{k+1}(l_{in}) = \frac{\underline{p}_i^{k+1}(l_{in})}{\Sigma_r \underline{p}_i^{k+1}(l_{ir})} \qquad (6.3.1.10)$$

Normalization of the intermediate updated probability estimates in order to satisfy equation (6.3.1.2) gives the new probabilities associated with each disparity label for node a_i as

the index r is over the set L_i of disparity labels associated with a_i. The normalization procedure given by equation (6.3.1.10) will determine the next value of $p_i^{k+1}(l_i^*)$.

A potential match is deleted from a disparity label set L_i if its probability falls below a predetermined value , usually 0.01.

If a match exists the probability associated with the matching label will be much greater than that of the remaining labels in the label set. If there are two or more labels which persistently retain probabilities of similar value , then the point is regarded as ambiguous and a match is not determined.

Figure 6.3.1.2. shows a flow diagram of the relaxation matching procedure. As indicated the method is iterated a preselected number of times. Matching points are then extracted from the lists as those with probabilities greater than a suitable

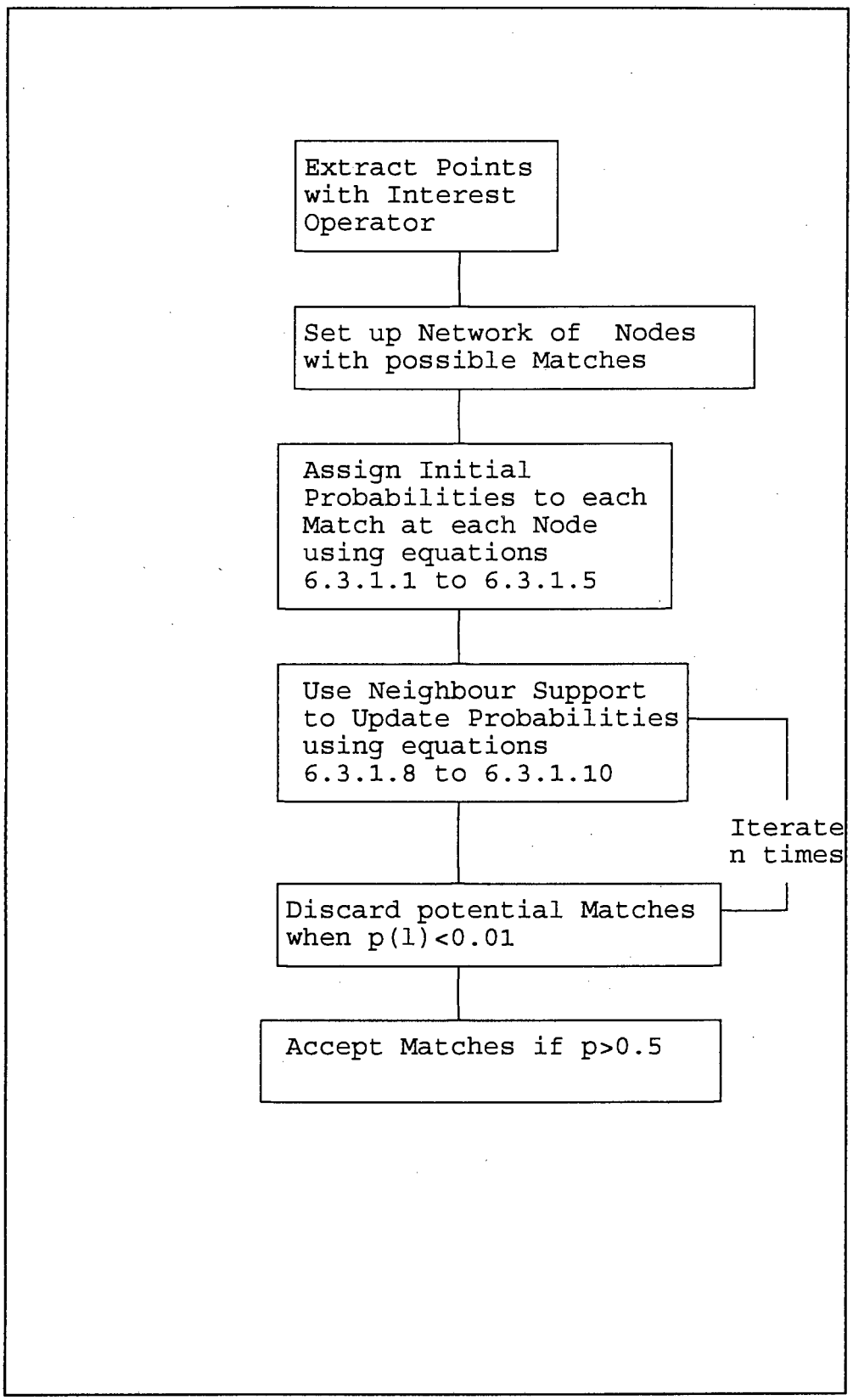


Figure 6.3.1.2. *Structure of the Disparity Matching Algorithm.*

threshold (say 0.5).

Once a set of matching points has been obtained by application of the relaxation algorithm, it will then be possible to perform a standard relative orientation using the coplanarity condition. This would be true under the assumption that the matches were all correct.

6.4.STANDARD RELATIVE ORIENTATION ALGORITHM

One of the standard relative orientation algorithms *[Haralick and Shapiro, 1993]* , given two images with matching points and with the camera base length known, is briefly outlined as follows :-

The coplanarity condition of equation (6.2.1) is reexpressed as:-

$$f(\underline{X}_r, \omega, \phi, \kappa) = \underline{X}_r \cdot [\, R_r \, \underline{U}_r \times \underline{U}_l] = 0 \qquad (6.4.1)$$

where

$\underline{X}_r$ is the position vector of the perspective centre of the right camera. The perspective centre of the left camera is taken as the origin.

R_r is the transpose of the rotation matrix given above with relative orientation angles ω, ϕ, κ to the left image.

$\underline{U}_r$ are the image point coordinates in the right image.

$\underline{U}_l$ are the corresponding point image coordinates in the left image.

If there are N (> 5) matching points equation (6.2.1) leads to

a set of overdetermined non-linear equations in terms of the unknown relative orientation parameters $\omega, \phi, \kappa, Y_r, Z_r$ defined in section 6.2. X_r is taken as equal to the camera base length. By linearizing the non-linear equations using a first order Taylor series expansion a least squares method can be used to solve for the unknown parameters.

Linearizing (6.4.1) gives N equations

$$f_i(\underline{X_r}, \omega, \phi, \kappa) + \nabla f_i \cdot \underline{da} = 0 + v_i \qquad (6.4.2)$$

where

$$\nabla = \left(\frac{\partial}{\partial\omega}, \frac{\partial}{\partial\phi}, \frac{\partial}{\partial\kappa}, \frac{\partial}{\partial Y_r}, \frac{\partial}{\partial Z_r}\right)$$

and

$$\underline{da} = (\delta\omega, \delta\phi, \delta\kappa, \delta Y_r, \delta Z_r)$$

The least-squares method is used to minimize the sum of the residuals squared $\sum_1^N v_i^2$ and iteratively solve for the unknown relative orientation parameters.

Once the relative orientation parameters are found the epipolar lines in each image can be determined. If correspondence is correct matching points will lie on their respective epipolar lines.

6.5.RELATIVE ORIENTATION ALGORITHM INCORPORATING THE EPIPOLAR LINE CONSTRAINT

With most real data there will be a number of mismatches. To try and eliminate more mismatches the epipolar line constraint can be incorporated into the least squares solution for the relative orientation parameters.

For a given point in the left image (x_{l_i}, y_{l_i}) the corresponding point in the right image (x_{r_i}, y_{r_i}) can be determined using the coplanarity condition given by equation (6.2.1). The x_{r_i} value obtained from the disparity matching is taken as a known value , although it is not necessarily correct. The main assumption made is that the majority of points matched by means of the relaxation method are correctly matched. If a point is incorrectly matched then this will show up as a predicted y_r that is far from the y_r value in the observed data set. A multiple regression model can then be set up on the basis that the sum of the squares of the residuals $v_i = y_{r_i}(observed) - y_{r_i}(predicted)$ are minimized.

Assume

$$y_{r_i}(observed) + v_i = y(\omega, \phi, \kappa, Y_r, Z_r) = y_{r_i}(predicted) \qquad (6.5.1)$$

Linearizing by means of a first order Taylor series expansion leads to

$$y_{r_i}(observed) + v_i = y^0_{r_i} + \nabla y^0_{r_i} \cdot \underline{da} \qquad (6.5.2)$$

∇ and $\underline{da}$ are the same as before. $y^0_{r_i}$ is the provisional value of $y_{r_i}(predicted)$.

A least-squares method can be used to solve this overdetermined system of N equations for the five relative orientation parameters that are unknown.

Details of the formulae required are given below using notation adopted from *Wong, 1986*. All image coordinates are taken as relative to the coordinates of the perspective centre in the

image which acts as the origin of the image coordinate system. Assuming that the x coordinate of a matching image point is correct in the right image , the predicted value of the y coordinate in the image will be given by

$$y_{r_i} = \frac{S\xi_i + T\eta_i + Q\zeta_i}{Sm_{21} + Tm_{22} + Qm_{23}} = \frac{N}{D} \qquad (6.5.3)$$

where

$$Q = X_r y_{l_i} - Y_r x_{l_i}$$

$$S = Y_r f_l - Z_r y_{i_i}$$

$$T = Z_r x_{l_i} - X_r f_l$$

f_l is the principal distance of the left camera.

and

$$\xi_i = -m_{11} x_{r_i} - m_{31} f_r$$

$$\eta_i = -m_{12} x_{r_i} - m_{32} f_r$$

$$\zeta_i = -m_{13} x_{r_i} - m_{33} f_r$$

f_r is the principal distance of the right camera.

The m_{ij} are the matrix elements of the rotation matrix , based on the relative orientation angles , given by equation (6.2.2).

In order to evaluate ∇y_r the first derivatives of y_r are given as follows :-

$$\frac{\partial y_{r_i}}{\partial \omega} = \frac{\left(S\frac{\partial \xi_i}{\partial \omega} + T\frac{\partial \eta_i}{\partial \omega} + Q\frac{\partial \zeta_i}{\partial \omega} \right) D - N\left(S\frac{\partial m_{21}}{\partial \omega} + T\frac{\partial m_{22}}{\partial \omega} + Q\frac{\partial m_{23}}{\partial \omega} \right)}{D^2}$$

$$\frac{\partial y_{r_i}}{\partial \phi} = \frac{(S\frac{\partial \xi_i}{\partial \phi} + T\frac{\partial \eta_i}{\partial \phi} + Q\frac{\partial \zeta_i}{\partial \phi})D - N(S\frac{\partial m_{21}}{\partial \phi} + T\frac{\partial m_{22}}{\partial \phi} + Q\frac{\partial m_{23}}{\partial \phi})}{D^2}$$

$$\frac{\partial y_{r_i}}{\partial \kappa} = \frac{(S\frac{\partial \xi_i}{\partial \kappa} + T\frac{\partial \eta_i}{\partial \kappa} + Q\frac{\partial \zeta_i}{\partial \kappa})D - N(S\frac{\partial m_{21}}{\partial \kappa} + T\frac{\partial m_{22}}{\partial \kappa} + Q\frac{\partial m_{23}}{\partial \kappa})}{D^2}$$

$$\frac{\partial y_{r_i}}{\partial Y_r} = \frac{(f_1\xi_i - x_{1_i}\zeta_i)D - N(m_{21}f_1 - m_{23}x_{1_i})}{D^2}$$

$$\frac{\partial y_{r_i}}{\partial Z_r} = \frac{(-y_{1_i}\xi_i + x_{1_i}\eta_i)D - N(-m_{21}y_{1_i} + m_{22}x_{1_i})}{D^2}$$

where

$$\frac{\partial \xi_i}{\partial \omega} = 0$$

$$\frac{\partial \eta_i}{\partial \omega} = -(\cos\omega\sin\phi\cos\kappa - \sin\omega\sin\kappa)\, x_{r_i} + \cos\omega\cos\phi\, f_r$$

$$\frac{\partial \zeta_i}{\partial \omega} = -(\sin\omega\sin\phi\cos\kappa + \cos\omega\sin\kappa)\, x_{r_i} + \sin\omega\cos\phi f_r$$

$$\frac{\partial \xi_i}{\partial \phi} = \sin\phi\cos\kappa\, x_{r_i} - \cos\phi f_r$$

$$\frac{\partial \eta_i}{\partial \phi} = -(\sin\omega\cos\phi\cos\kappa)\, x_{r_i} - \sin\omega\sin\phi\, f_r$$

$$\frac{\partial \zeta_i}{\partial \phi} = (\cos\omega\cos\phi\cos\kappa)\, x_{r_i} + \cos\omega\sin\phi f_r$$

$$\frac{\partial \xi_i}{\partial \kappa} = \cos\phi\sin\kappa \; x_{r_i}$$

$$\frac{\partial \eta_i}{\partial \kappa} = (\sin\omega\sin\phi\sin\kappa - \cos\omega\cos\kappa) \; x_{r_i}$$

$$\frac{\partial \zeta_i}{\partial \kappa} = -(\cos\omega\sin\phi\sin\kappa + \sin\omega\cos\kappa) \; x_{r_i}$$

$$\frac{\partial m_{21}}{\partial \omega} = 0 \qquad \frac{\partial m_{21}}{\partial \phi} = \sin\phi\sin\kappa \qquad \frac{\partial m_{21}}{\partial \kappa} = -\cos\phi\cos$$

$$\frac{\partial m_{22}}{\partial \omega} = -\cos\omega\sin\phi\sin\kappa - \sin\omega\cos\kappa$$

$$\frac{\partial m_{22}}{\partial \phi} = -\sin\omega\cos\phi\sin\kappa$$

$$\frac{\partial m_{22}}{\partial \kappa} = -\sin\omega\sin\phi\cos\kappa - \cos\omega\sin\kappa$$

$$\frac{\partial m_{23}}{\partial \omega} = -\sin\omega\sin\phi\sin\kappa + \cos\omega\cos\kappa$$

$$\frac{\partial m_{23}}{\partial \phi} = \cos\omega\cos\phi\sin\kappa$$

$$\frac{\partial m_{23}}{\partial \kappa} = \cos\omega\sin\phi\cos\kappa - \sin\omega\sin\kappa$$

After the relative orientation parameters are found when the above procedure converges the matching points with large residuals can be excluded from the data based on one of the standard outlier elimination methods. The relative orientation parameters can then be reevaluated with the reduced data set. This procedure can be repeated until there are no "matching points" with large residuals , implying that all retained matches will lie on the epipolar lines determined by the relative orientation parameters found. The procedure is self consistent in the sense that it will converge to relative orientation parameters that will ensure that all the retained matching points lie on the epipolar lines.

The relaxation matching procedure described above is best performed to pixel accuracy. As will be illustrated in an application in chapter 7 this was found to be adequate to obtain approximate relative orientation parameters. Sub-pixel accuracy can be achieved by using the pixel points in the right image as starting points for a least-squared grey-scale correlation matching procedure. This technique is explained in chapter 5. Relative orientation parameters can then be recalculated with these more accurately determined points.

6.6.EXTRACTING 3-DIMENSIONAL INFORMATION

Once corresponding points and relative orientation parameters are determined a rough digital terrain model can be obtained. The object surface coordinates can be determined using a method described by *Haralick and Shapiro, 1993*, based on the collinearity equations.

The collinearity equation for the left image relate the image coordinates to the object coordinates.

$$\underline{X} = \underline{X_L} + \lambda_L R_L \underline{u_L} \qquad (6.6.1)$$

where $\underline{X}$ = (x,y,z) are the coordinates of the object point in a cartesian coordinate system with the perspective centre of the left camera $\underline{X_L}$ = (0,0,0) at the origin. $\underline{R_L}$ is the rotation matrix of the left image which in this case is set equal to the identity matrix. $\underline{u_L}$ = (u_L , v_L , f_L) are the coordinates of the image point in the left image. λ_L is an unknown scale factor. Similarly the collinearity equation for the right image is given as

$$\underline{X} = \underline{X_R} + \lambda_R R_R \underline{u_R} \qquad (6.6.2)$$

where the subscript R indicates the right image.
$\underline{X_R}$ = (x_R , y_R , z_R) are the coordinates of the perspective

centre of the right camera with $\underline{X}_L$ the origin. R_R is the rotation matrix of the right image with the rotation matrix elements defined in terms of the relative orientation angles ω, ϕ, θ as defined in a previous section. λ_R is an unknown scale factor associated with the right image.

Ideally the right hand sides of equations (6.6.1) and (6.6.2) should be equal ,but due to measurement noise this is never the case. One can determine the optimum values of λ_L and λ_R that will minimize the squared residual between the right hand sides of equation (6.6.1) and (6.6.2). The squared residual is given by

$$e^2 = \|\underline{X}_L + \lambda_L R_L \underline{u}_L - \underline{X}_R - \lambda_R R_R \underline{u}_R\|^2 \qquad (6.6.3)$$

By differentiating equation (6.6.3) with respect to λ_L and λ_R and equating to zero one obtains two equations in λ_L and λ_R enabling a solution to these unknowns for each pair of corresponding points. Letting $\underline{p}_L = R_L \underline{u}_L$ and $\underline{p}_R = R_R \underline{u}_R$ then the solution for λ_L becomes

$$\lambda_L = \frac{(\underline{X}_R \cdot \underline{p}_L)\ (\underline{p}_R \cdot \underline{p}_R) + (\underline{X}_R \cdot \underline{p}_R)\ (\underline{p}_R \cdot \underline{p}_L)}{(\underline{p}_R \cdot \underline{p}_R)\ (\underline{p}_L \cdot \underline{p}_L) - (\underline{p}_L \cdot \underline{p}_R)^2} \qquad (6.6.4)$$

Substituting this value into equation (6.6.1) gives the object coordinates $\underline{X}$. The object coordinates for each object point corresponding to each pair of matching image points can therefore be determined. A rough digital terrain model is consequently obtained. Various forms of interpolation techniques can be applied to obtain a denser digital terrain model. Interpolation can also be performed on the disparity map prior to the object point determination.

6.7. CONCLUSION

A combination of the methods outlined in this chapter can be used as part of a fully automated system for three dimensional surface reconstruction. The main steps are shown in figure 6.7.1. After the points are matched to subpixel accuracy the relative orientation algorithm which incorporates the epipolar line constraint can be applied iteratively to eliminate outliers in the data.

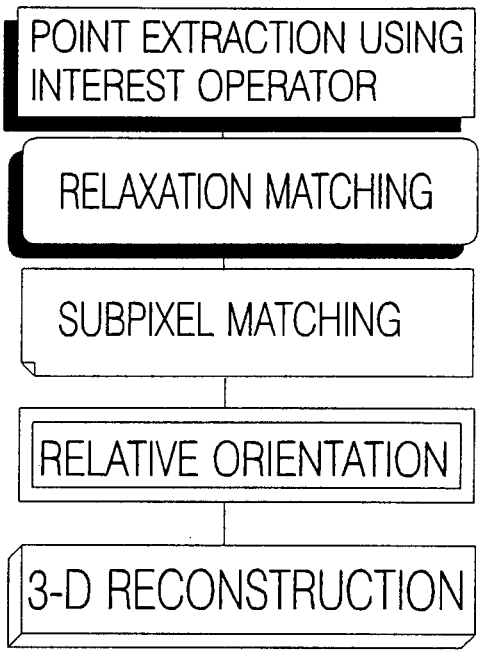


Figure 135 *Figure 6.7.1. Main steps in an automated surface reconstruction system.*

As the above methods require considerable computational power, they are not suitable for practical PC computer applications. Although with the advent of faster computer hardware this may soon change.

7.APPLICATION OF SOME METHODS TO AERIAL IMAGES

7.1. LOCATION AND MEASUREMENT OF A BUILDING

Some of the techniques developed in the previous chapters can be illustrated by an application to the location and measurement of a particular building in an aerial scene. The building could be in a knowledge base so that a user could hypothetically enter a request to the computer to locate and extract the specific building for a number of purposes.

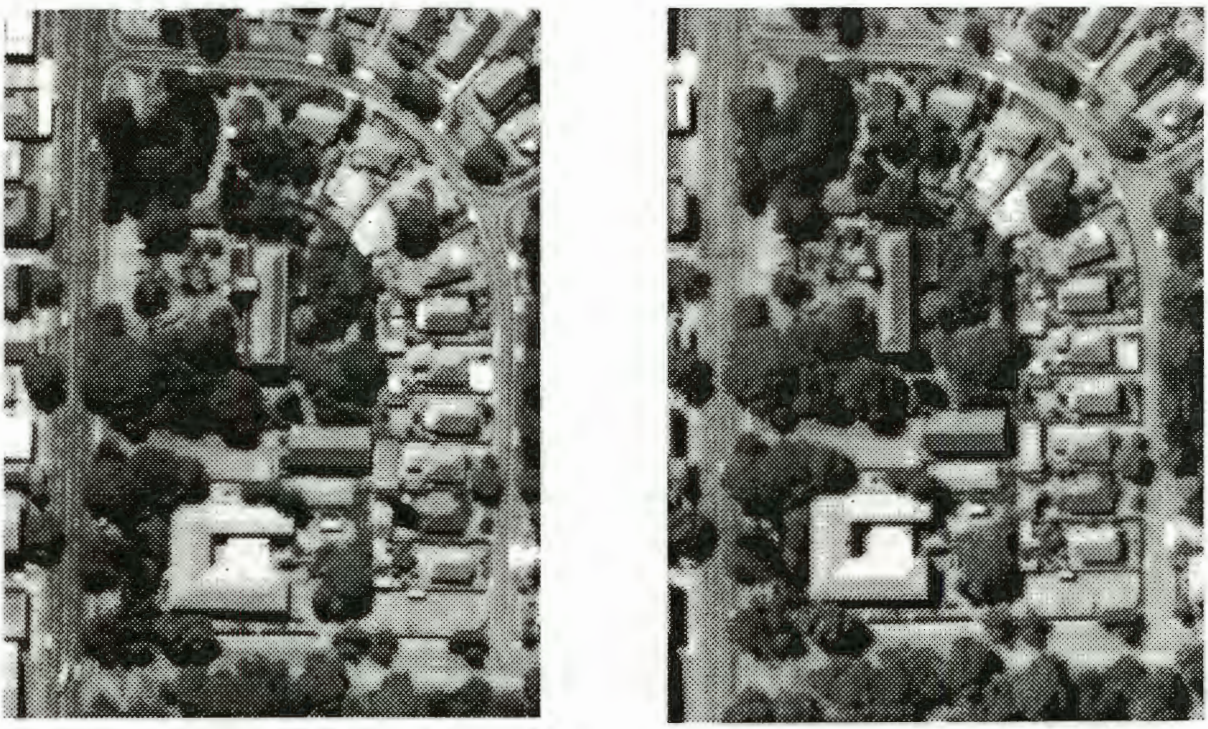
The test case was selected from aerial photographs taken of a section of the suburban area in Durban, South Africa. Only a portion of the photograph was digitized using a digital scanner, due to the limitations of disk space and the experimental nature of the investigation.

The two digitized images are shown in figure 7.1.1. Figure 7.1.1(a) was taken from the 'left' photograph and 7.1.1(b) from the 'right' photograph of a stereo pair.

(a) (b)

Figure 7.1.1. *Images extracted from a stereopair of a Durban suburban area.*

The left image is slightly darker than the right image. In order to improve the similarity between the photometric quality of the images both images were transformed by histogram equalization described in Appendix B. The resulting images are shown in figure 7.1.2.



(a) (b)

Figure 7.1.2. *Figure 1 after grey-scale equalization.*

The various techniques adopted for measuring a building in figure 7.1.2. are discussed under subheadings according to the logical sequence required in their application.

7.1.1. INITIAL DATA REQUIRED

In order to obtain a 3-dimensional reconstruction of a building the following metric data is required.

A) Distance between the airstations. This is necessary as the relative orientation algorithms used require one known parameter.

B) Pixel and photograph coordinates of calibration points for affine transformation parameters for converting between the different coordinates systems. The calibration points are ground points clearly marked by white paint markers in the shape of circles or crosses. The three dimensional coordinates of these points in object space have been determined by surveying techniques and the values of these coordinates are available. These points are also used for the determination of the relative orientation parameters of the cameras. These are identifiable in the images and their image coordinates can be determined manually and with least squares matching for subpixel accuracy.

Digitization of aerial photographs results in distortions introduced by digital scanners. Optical aberrations are introduced by the camera. Further distortions are included when developing photographs. Modelling of these distortion parameters should be included in the relative orientation algorithms if greater precision is required. These abberations where not considered as the main emphasis of this example was to illustrate the methods discussed in the thesis.

The affine transformation parameters between the image and the photograph is necessary because the size of the fully digitized aerial photographs is very large and impractical to operate on when using a PC (personal computer). It is therefore necessary to split the whole photograph into subsections and work on these subsections. In order to relate the subsections to the whole photograph via the affine transformation parameters at least 6 calibration points must be present in each subsection.

The affine transformation would not be necessary if one uses multiresolution image pyramids. The whole aerial image can be reduced to a resolution of $R/2^n$ (R=original resolution , n=integer)to reduce the size of the image and the initial matching can be performed on the reduced resolution images. The corresponding points found by this method can then be used to obtain the relative orientation between the two cameras.

Unfortunately the limitations of Pcs makes this procedure time consuming. If there are no calibration points then this will be the most practical means of obtaining the relative orientation. In a fully automated system this would be essential.

In this example the calibration points were available and were used to reduce the computation time.

7.1.2. SUBSECTION SELECTION

As a result of the requirements of 7.1.1 subsections of similar regions containing at least 6 calibration points need to be extracted from the corresponding whole images. In order to relate the subsections to each other and eventually to reconstruct the complete region covered by the stereo photographs, a certain amount of overlap between subsections should exist to avoid information at subsection borders being omitted. To illustrate the methods in the thesis a subsection of the same region was selected from the stereo images as shown in figure 7.1.1.

7.1.3. APPLICATION OF INTEREST OPERATOR

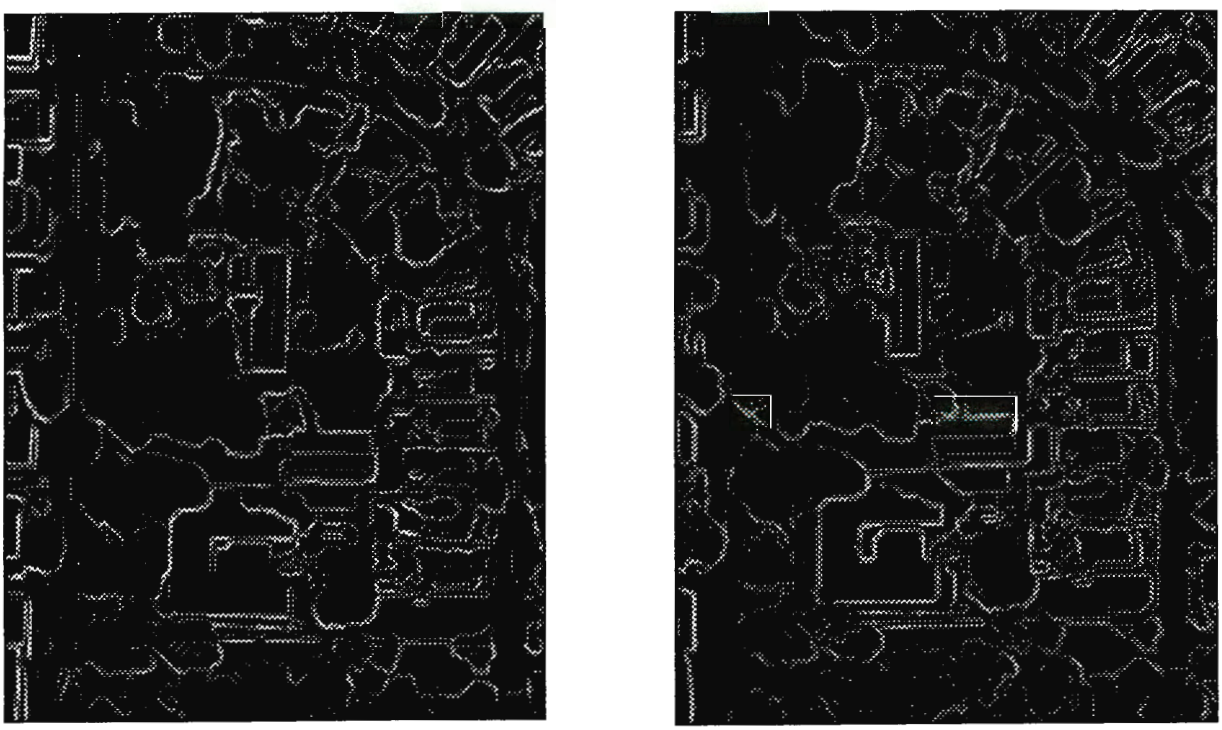
In order to extract matchable points an interest operator must be applied over each subsection. The angle based interest operator described in chapter 4 was chosen as it gives a good distribution and density of points over the image.

First the sobel edge detector is applied as shown in figure 7.1.3 and then the interest operator selects the points shown in figure 7.1.4.

7.1.4. RELAXATION MATCHING

The relaxation method due to Barnard and Thompson described in a previous chapter is applied to the selected points to identify corresponding points. The angle-based interest operator detected

5854 points in this particular example. A disparity range of 30 was used to set up the initial labels. This resulted in up to 200 possible disparity labels per initial image point, only the possible disparity labels with high initial probability were retained. Five iterations were carried out, with A=1.0 and B=10.0 in equation (6.3.1.9) to accelerate convergence. Labels with probabilities below 0.01 are discarded. For each point the label with the maximum probability was selected as the disparity between the corresponding points in each image. The image disparities for each point are shown in figure 7.1.5(a). Points were selected with probabilities above 0.5 and these are shown in figure 7.1.5(b). The disparities are fairly similar over the whole image as would be expected from a relatively distant view. Figure 7.1.5(a) clearly includes a few mismatches.



(a) (b)

Figure 7.1.3. *Images after Sobel edge detection.*

(a) (b)

Figure 7.1.4. *Points selected by angle interest operator applied to Figure 7.1.3.*

(a) (b)

Threshold p=0.0 Threshold p=0.5

Figure 7.1.5. *Disparities of points selected with different probability thresholds. (a) has 5854 points. (b) has 353 points.*

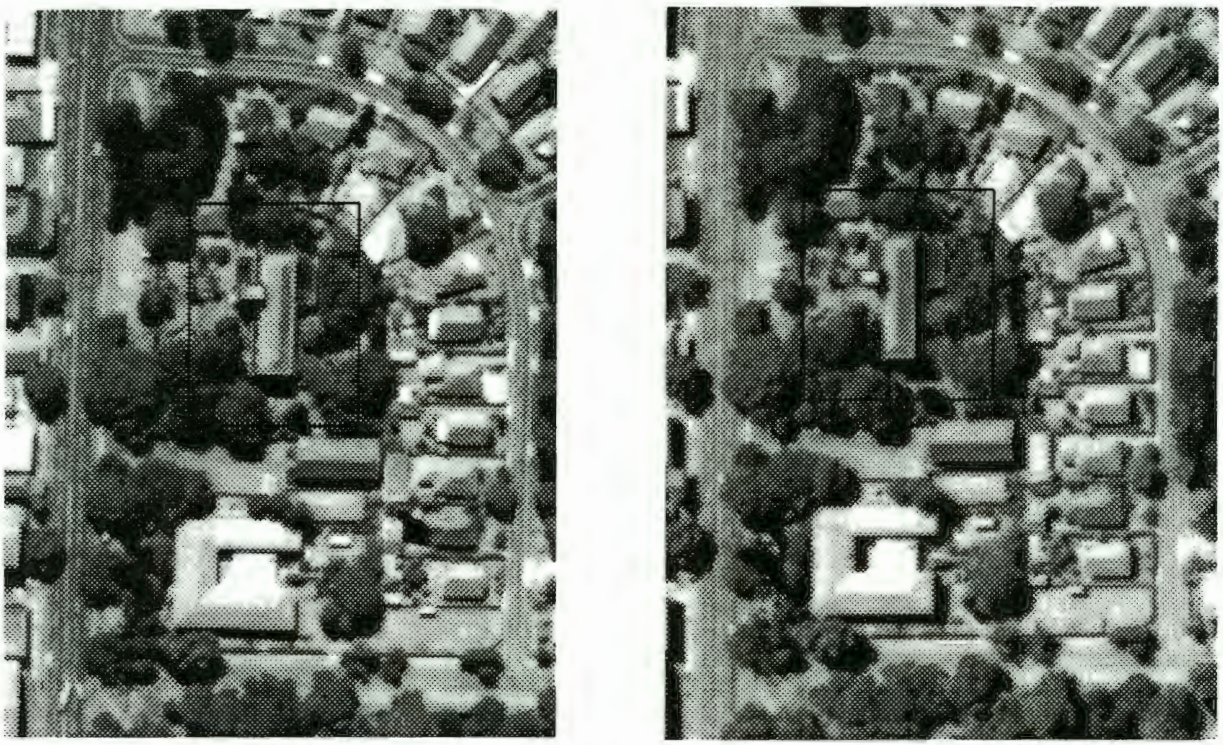


Figure 7.1.6. *Building to be measured in each image.*

7.1.5.COMPUTER-ASSISTED MANUAL EDITING OF PIXEL MATCHED POINTS

In order to remove incorrect matching points for the calculation of the relative orientation parameters computer screen visual editing of the matched points can be performed. If points are insufficient or badly distributed return to step 7.1.3 and change the angle threshold in equation (4.5.2) as required.

7.1.6. CALCULATION OF RELATIVE ORIENTATION PARAMETERS

Notional relative orientation parameters are found based on the pixel data to enable epipolar line searching. RELAT1 or RELEPI are used.

The standard relative orientation (RELAT) algorithm and the epipolar constrained algorithm (RELEPI) were used with the

corresponding pixel points in each image and assuming the centres of each image as the respective principal points. Arbitrary values of 1 were chosen for the distance between the two cameras and for the principal distances of both images. A notional set of relative orientation parameters is thereby obtained in order to use the epipolar line constraint to guide further matching.

	STANDARD ALGORITHM	EPIPOLAR CONSTRAINED ALGORITHM
	n=1324	
ω	-0.000027 +/- 0.000003	0.000008 +/- 0.000003
ϕ	-0.000020 +/- 0.000004	-0.000070 +/- 0.000005
κ	0.029952 +/- 0.000245	0.025860 +/- 0.000260
Δy	0.749552 +/- 0.002938	1.331166 +/- 0.000448
Δz	0.000050 +/- 0.000029	0.000065 +/- 0.000005
	n=345	
ω	-0.000026 +/- 0.000009	-0.000034 +/- 0.000028
ϕ	-0.000018 +/- 0.000025	0.000039 +/- 0.000038
κ	0.029797 +/- 0.000317	0.025531 +/- 0.001406
Δy	0.774678 +/- 0.004431	1.331032 +/- 0.004193
Δz	-0.000015 +/- 0.000141	-0.000085 +/- 0.000064

Table 7.1.1. *Relative orientation parameters results using n matched pixel data points.*

Table 7.1.1 shows the relative orientation parameters calculated using two different data sets and using the two different relative orientation algorithms. The angles are in radians and the distances are in the same units as the baseline and principal distance. The precision is higher for both algorithms when more points are used as would be expected. RELEPI was found to converge more rapidly compared to RELAT usually within 20 iterations. The convergence rate is dependant on the parameter tolerance input by the operator. The parameter tolerance indicates the relative change in parameter values between iterations. A parameter tolerance of 0.1% gives

adequately precise values with convergence, whereas lower values can lead to oscillations in the iterative solution without converging.

The actual parameter values produced by the two algorithms differ consistently in actual values , but they are of the same order. The difference in values can be attributed to one of the underlying assumptions in RELEPI. The x pixel coordinates are assumed correctly matched between images and the difference in the y coordinates minimized. If matching is to pixel precision, then this will be unlikely to be generally true. Mismatches will therefore also have a more significant effect on the relative orientation parameters determined than when using RELAT. Following this argument RELEPI would give good results with data where points are matched to subpixel accuracy and where there are no mismatches. More research needs to be conducted to establish the relative robustness of RELEPI compared to RELAT and other relative orientation algorithms.

(a) (b)

Figure 7.1.7. *Epipolar lines on the left and right images.The brighter lines are from the RELAT parameters. The fainter from RELEPI.*

7.1.7. OBJECT SELECTION

An object and its surroundings, such as a building, is selected, interactively from the left image or from a knowledge base, for measurement. In this example the building shown in the left image, in figure 7.1.6, is to be located in the right image.

7.1.8. EPIPOLAR LINE SEARCH

By using the relative orientation parameters found in 7.1.6 a pattern recognition based search for the corresponding object in the right image along a narrow band following the epipolar line can be performed.

Figure 7.1.7 shows the epipolar lines for selected points in the left image. The epipolar lines are calculated using equation (6.5.3). The brighter lines are produced by using the relative orientation parameters from RELAT. The fainter lines are produced using the RELEPI parameters. Each intersection of the epipolar lines in the left image indicates the position of one of the points selected. As shown the lines produced by RELAT are closer to the corresponding points in the right image than those of RELEPI. In the case of the point lying on the building that is to be measured (the building selected is shown in figure 7.1.6) both lines are about 4 or 5 pixels away from the corresponding point. This is indicative of the fact that the pixel data used in calculating the relative orientation parameters are mostly at ground, tree and house roof level. Not sufficient weight is given to the significantly fewer points on high buildings. The RELAT lines pass almost exactly over the corresponding points in the right image for points at house roof or ground level.

The epipolar lines produced above are adequate for doing epipolar line guided matching, where the search space is taken as a band about ten pixels wide centred on the epipolar line. For a better set of relative orientation parameters using points matched to subpixel accuracy a band of 2 to 3 pixels would be

acceptable. This will enable a selected region in the left image to be located in the right image. A disparity threshold can be derived from the average disparity over the whole image obtained from the relaxation matching. This threshold can be used to eliminate any obviously wrong points that may be chosen by a pattern matching technique.

A grey-scale correlation search can be used along an epipolar line band and the window most closely correlated to the selected left image window chosen is taken as the correct location in the right image. Alternatively the pattern recognition approach described in chapter 4 can be used to locate the building by searching along the epipolar line band with the nearest neighbour point taken as the matching location.

A comparison of these two methods was made using different window sizes in order to locate the building shown in figure 7.1.6 in the right image. Table 7.1.2 shows the results of this comparison.

Window Size (N X N)	Grey-Scale Correlation Method	Feature Vector Method
9 X 9	7.2	4.2
17 X 17	7.8	2.0
33 X 33	14.8	4.5
65 X 65	6.1	9.0

Table 7.1.2. *Distances in pixels of located window from correct position for different window sizes.*

Table 7.1.2 shows the distances in pixels of the centres of the located windows from their correct positions. In all cases the results are not very good due to the nature of the region in the image chosen. The influence of the difference in distributions of shadows between the two images is the main contributing factor to the poor accuracy. Shadows usually give very strong edges. Excepting for the large window size the feature vector method gives significantly better results than the

grey-scale correlation technique. Global regional information appears to assist the grey-scale correlation technique in obtaining better localization for large window sizes. In this sense the two methods complement each other as the grey-scale technique is significantly faster than the feature vector method especially for large window sizes. For this particular example, the feature vector method can be used for windows up to about 40 x 40 pixels and the correlation technique can be used for larger window sizes. The exact size of the windows and the method to be used will be image dependent.

7.1.9. WINDOW EXTRACTION

A window including the object and its surrounding are extracted from the located positions in each image.

Figure 7.1.8 shows the selected window on the left and the located window on the right. A 17 x 17 feature vector search window was used to give the result shown, as it appears to give the best results of the window sizes chosen. The centre of the right window is only about 2 pixels from the correct position. The size of the extracted windows around the located building is determined heuristically in this particular example. A method for automatically establishing this size could be devised in principle from information in a knowledge base combined with information derived from a bundle adjustment camera calibration procedure.

Figure 7.1.8. *The selected window on the left and the located window on the right.*

7.1.10.APPLICATION OF INTEREST OPERATOR TO WINDOWS

As with step 7.1.3 an angle-based interest operator is used to extract matchable points.

7.1.11.RELAXATION MATCHING OF WINDOWS

The same technique applied in step 7.1.4 is used. The relaxation matching procedure is performed between these two windows using the points selected by the angle interest operator. Of the 917 points selected by the interest operator 143 were matched using relaxation disparity matching.

7.1.12.COMPUTER-ASSISTED MANUAL EDITING OF PIXEL MATCHED POINTS

To eliminate badly matched points the same method as that used in step 7.1.5 is used. If the remaining points are insufficient or badly distributed return to step 7.1.10 and change the angle threshold in equation (4.5.2) as required.

114 of the 143 points found in 7.1.11 were good matches.

7.1.13.WEIGHTED LEAST-SQUARES MATCHING

In order to obtain sub-pixel locations of the matching points a weighted least squares method is used with the pixel match point position as an initial approximation. The weights used approximate the behaviour of the L1-norm as described in chapter 5.

7.1.14.AFFINE TRANSFORMATION PARAMETERS

The 6 calibration points in the image subsections are used to determine affine transformation parameters between the images and the photograph coordinates using least-squares estimation. This can be calculated at any time after step 7.1.2. These points are found interactively.

The affine transformation is used to convert the pixel coordinates to photograph coordinates. The affine transformation parameters are obtained by using a least squares estimation procedure based on the coordinates of six calibration points in the images depicted in figure 7.1.1(a) and (b). Two different affine transformations parameter sets are determined corresponding to each image of the stereopair. The affine transformation parameters are shown in table 7.1.4.

	LEFT IMAGE		RIGHT IMAGE	
	Forward	Backward	Forward	Backward
a0	19.2003	-217.2845	-56.5643	656.2266
a1	0.0867	11.5257	0.0870	11.4949
a2	-0.0010	0.1422	-0.0016	0.2132
b0	27.9095	326.0990	-28.1886	329.2244
b1	-0.0007	0.0946	0.0002	-0.0194
b2	0.0851	11.7498	0.0853	11.7187

Table 7.1.4. *Affine transformation parameters. Forward is from pixel to photograph and backward the reverse.*

The forward affine transformation converts the point positions from the pixel coordinates in the images of figure 7.1.1 to the corresponding photographic coordinates in millimetres. The backward transformation performs the inverse process.

7.1.15.COORDINATE TRANSFORMATION

The matched points are transformed to their corresponding photograph coordinates using the parameters determined in 7.1.14.

7.1.16.RELATIVE ORIENTATION PARAMETERS

In order to obtain the three dimensional structure of the scene

in figure 7.1.8 the relative orientation of the cameras has to be known. The photographic coordinates of identifiable calibration points in the photographs were determined using a stereoplotter. Using these coordinates and knowing the airbase of the stereopair RELAT and RELEPI can be used to obtain the relative orientation parameters of the original cameras. Using an airbase of 667 metres and 8 well distributed calibration points the results of the relative orientation are as shown in table 7.1.3.

	STANDARD ALGORITHM	EPIPOLAR CONSTRAINED ALGORITHM
	$\Delta x=667$	$\Delta x=667$
ω	0.009688 +/- 0.000008	0.009688 +/- 0.000008
ϕ	-0.005578 +/- 0.000013	-0.005578 +/- 0.000013
κ	-0.023872 +/- 0.000005	-0.023872 +/- 0.000005
Δy	-3.648951 +/- 0.013417	-3.648885 +/- 0.013401
Δz	-5.148068 +/- 0.006529	-5.148107 +/- 0.006540

Table 7.1.3. *Relative orientation parameters of the cameras.*

7.1.17. DETERMINATION OF 3-D COORDINATES

Having obtained the photographic coordinates of the matching points and using the relative orientation parameters given in table 7.1.3 the three dimensional coordinates in the object space of the matching points are calculated using the method described in chapter 6. The perspective centre of the left camera is taken as the origin of the three dimensional coordinate system.

7.1.18.DIGITAL ELEVATION MODEL

With the three-dimensional points as data an existing contouring package was used to interpolate points, plot contours and give a perspective view of the digital terrain model. In this

application the SACLANT GRAPHICS PACKAGE operating on a VAX computer was used. These plots are shown in figure 7.1.9.

The building is +/- 12 metres above local ground level. Ground level is at +/- 1322 metres below the left camera. (The perspective centre of the left camera is taken as the origin of the object space coordinate system.) The flying height of the aeroplane is about 1340 metres above sea level. The values obtained compared favourably with those produced by a stereoplotter. The digital elevation model clearly shows the building rising from the ground level and also the presence of a tall tree at the south-west corner.

To improve precision modelling of the camera aberrations and the distortions introduced by the digital scanner in the relative orientation algorithms and the affine transformation parameter estimation algorithms is necessary. Matching precision is limited by the noise content of the images and a perfect reconstruction of the scene is not possible.

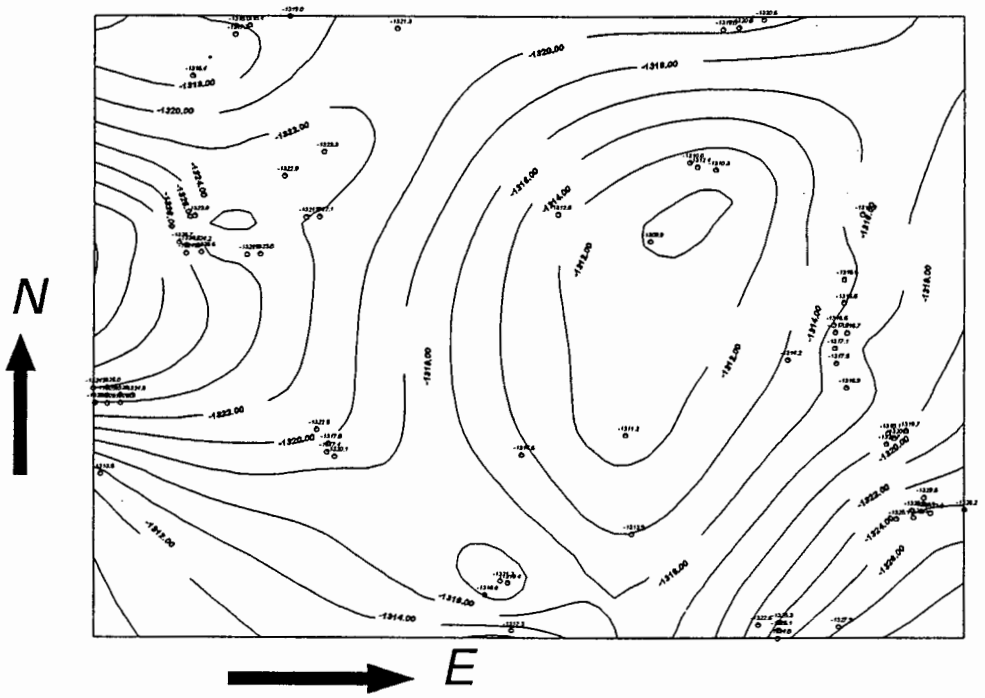


Figure 7.1.9.(a) *Contour plot of the window shown in figure 7.1.8.*

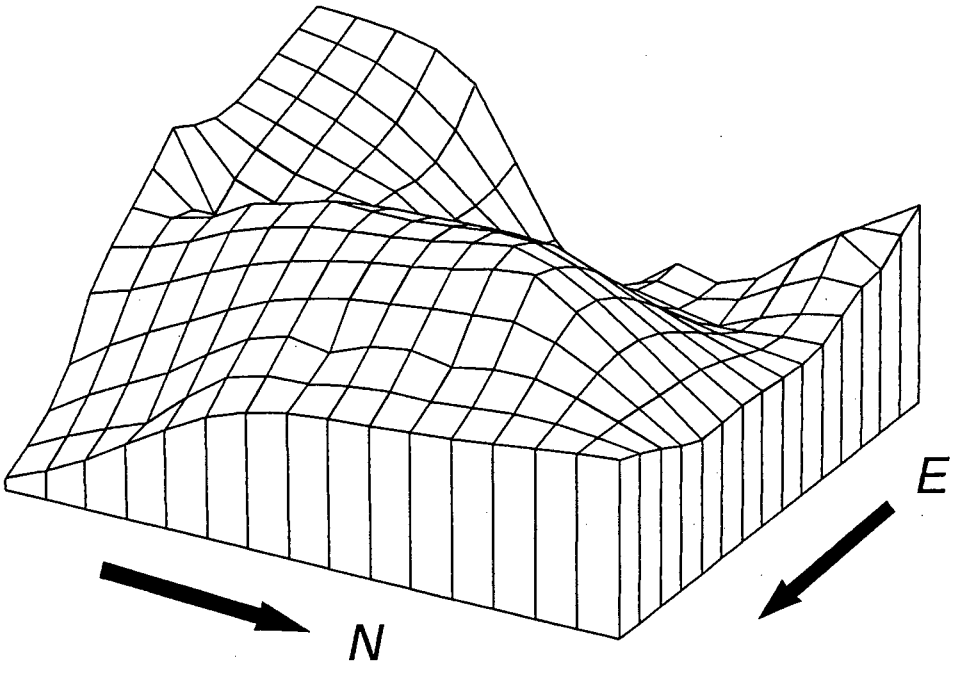


Figure 7.1.9.(b) *Digital elevation model of the window shown in figure 7.1.8.*

Figure 7.1.10 shows the flow diagram of the method used for the example described in this chapter. The digital elevation model of figure 7.1.9 is the result given by the final step 18.

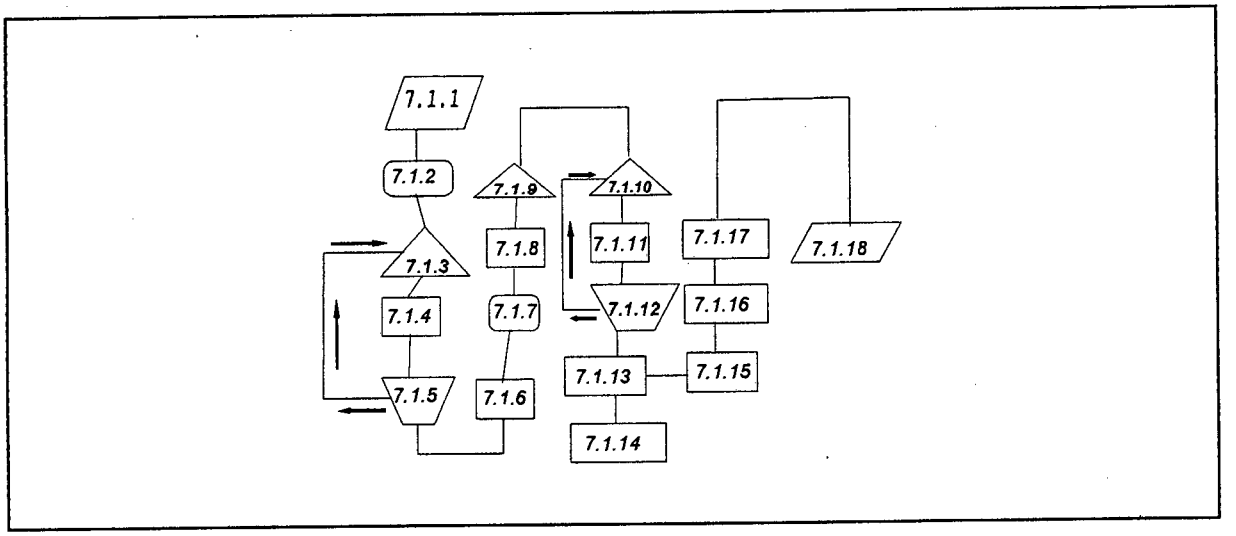


Figure 7.1.10. *General flow diagram of the steps described in the text. Feedback loops are indicated.*

7.1.19 EXTENSIONS TO STRATEGY

The above example illustrates the use of the methods in chapter 6 combined with the pattern recognition localization method and the angle-based interest operator techniques in chapter 4. The iteratively reweighted least squares method described in chapter 5 is used to obtain subpixel accuracy. With sufficiently powerful computers the strategy adopted can be fully automated if combined with an image pyramid approach so that the complete digitized photographs can be used as opposed to interactively selected sections. Other methods discussed in this thesis could be included as additional steps in the strategy shown in figure 7.2.1. The image segmentation technique based on fractal dimension mentioned in chapter 4 could be used to reduce the search space further. The distance transform technique of image segmentation outlined in chapter 5 could also be used to dramatically decrease the search space. Both techniques would require powerful computers for a practical system. The LAD methods for subpixel localization described in chapter 5 could be used in place of the iteratively reweighted least squares method. The pattern recognition based subpixel localization method described in chapter 5 could be used to eliminate obvious mismatches. More research needs to be conducted to establish the range of applicability of the pattern recognition subpixel localization method. The area-based L-norm methods can then be used to establish subpixel matching accuracy. The L-norm techniques proposed in chapter 5 can be used in place of the least squares techniques wherever the latter is applied. An example would be in the relative orientation algorithm. The robustness of the L1-norm methods towards outliers in the data can result in more precise results. Further research needs to be conducted in this respect.

The various combinations of techniques discussed in this thesis that can be used to solve a particular photogrammetric problem, would be dependent on the heuristics of the particular problem, and the computational tools and imaging technology

available.

7.2. LOCALIZATION AND MEASUREMENT OF AN AREA FROM NORMALIZED STEREO IMAGES.

A stereopair of images, normalized according to the method described under epipolar line matching in chapter 2, was obtained from the University of Stuttgart and used to test some of the methods described in this thesis. The images are shown in figure 7.2.1.

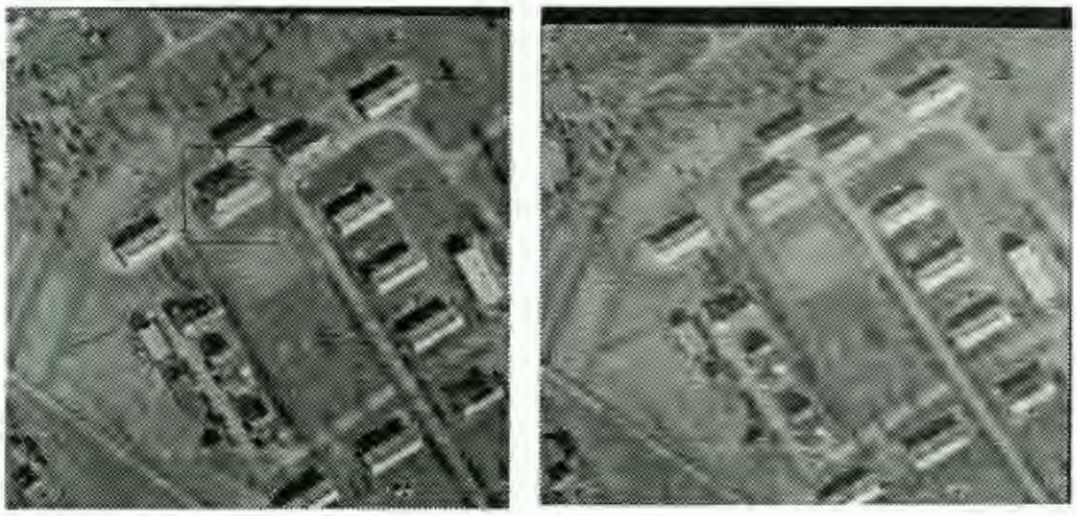


Figure 7.2.1. *Normalized aerial stereopair. The building to be located is contained in the square.*

7.2.1. LOCALIZATION OF BUILDING USING MOMENTS

Some of the moment based pattern recognition methods described in chapter 4 were compared by their ability to locate the building, indicated in figure 7.2.1, in the image shown on the right. As the images are normalized the search space is reduced to one dimension. In addition as the point disparity between any two matching points will not exceed a certain maximum disparity , the search space can be limited to this maximum value on either side of a particular point. The maximum disparity can be assessed by interactively inspecting the images on the computer.

The Hu moment method used in the example in section 7.1. was compared with the Zernike and Pseudo-Zernike moment methods. In the case of the latter two moment methods the complex moments to the 8th order were used as pattern descriptors. The norm of the difference between extracted descriptors was used as a matching criterion. The minimum value obtained in the search was chosen as the matching point.

Window Size	Hu Moments	Zernike	Pseudo-Zernike
9 X 9	3	8	2
17 X 17	2	1	1
23 X 23	2	0	0

Table 7.2.1. *The pixel error of the located point for the methods indicated.*

Table 7.2.1 shows the results of the localization experiment in terms of the pixel error from the `correct` position. The `correct` position was determined by interactive computer inspection of the images and should be accurate to about one pixel. The Zernike and Pseudo-Zernike methods appear to give good localization results as the window size is increased.

(a) (b)

Figure 7.2.1.1 *(a) is a 96 X 96 pixel window centred on area of interest shown in the left image of figure 7.2.1. (b) is the corresponding window located in the right image of figure 7.2.1.*

Figure 7.2.1.1 shows the 96 X 96 pixel windows surrounding the area of interest. The right window was located by searching with a 23 X 23 pixel window pseudo-Zernike moment method.

7.2.2. POINT MATCHING WITH PSEUDO-ZERNIKE MOMENTS.

Interest points were found in figure 7.2.1.1 (a) using the angle interest operator. The points are shown in figure 7.2.2.1. As the stereoimages are normalized the matching points in figure 7.2.1.1.(b) can be found by a one dimensional search in the x direction. The pseudo-Zernike moments corresponding to a 24 X 24 pixel window surrounding an interest point in the left image, were used to locate the matching point in the right image. The same criteria used in locating the corresponding windows described in section 7.2.1 was used. In order to get subpixel accuracy a parabola was fitted to the three norm values centred on the minimum value corresponding to the located pixel point. The subpixel value corresponding to the minimum of the fitted parabola is taken as the matching point.

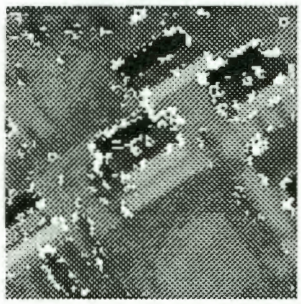


Figure 7.2.2.1. *Interest Points found in figure 7.2.1.1.*

After the matching points are found, the three dimensional object space coordinates can be determined using a formula provided by the University of Stuttgart. The parameters in the formula were specifically determined for the normalized stereopair shown in figure 7.2.1.

Figure 7.2.2.2 shows the contour plot of the heights determined from the matching points. The contour plot is inverted in the vertical direction with respect to the image shown in figure 7.2.1.1.(a). Generally the location of some buildings are evident and so is their distribution over the area concerned. The z coordinate values are quite reasonable at around 100 metres as the mean height over the whole region shown in figure 7.2.1. is given as 100 metres. The extreme variation in heights shown in

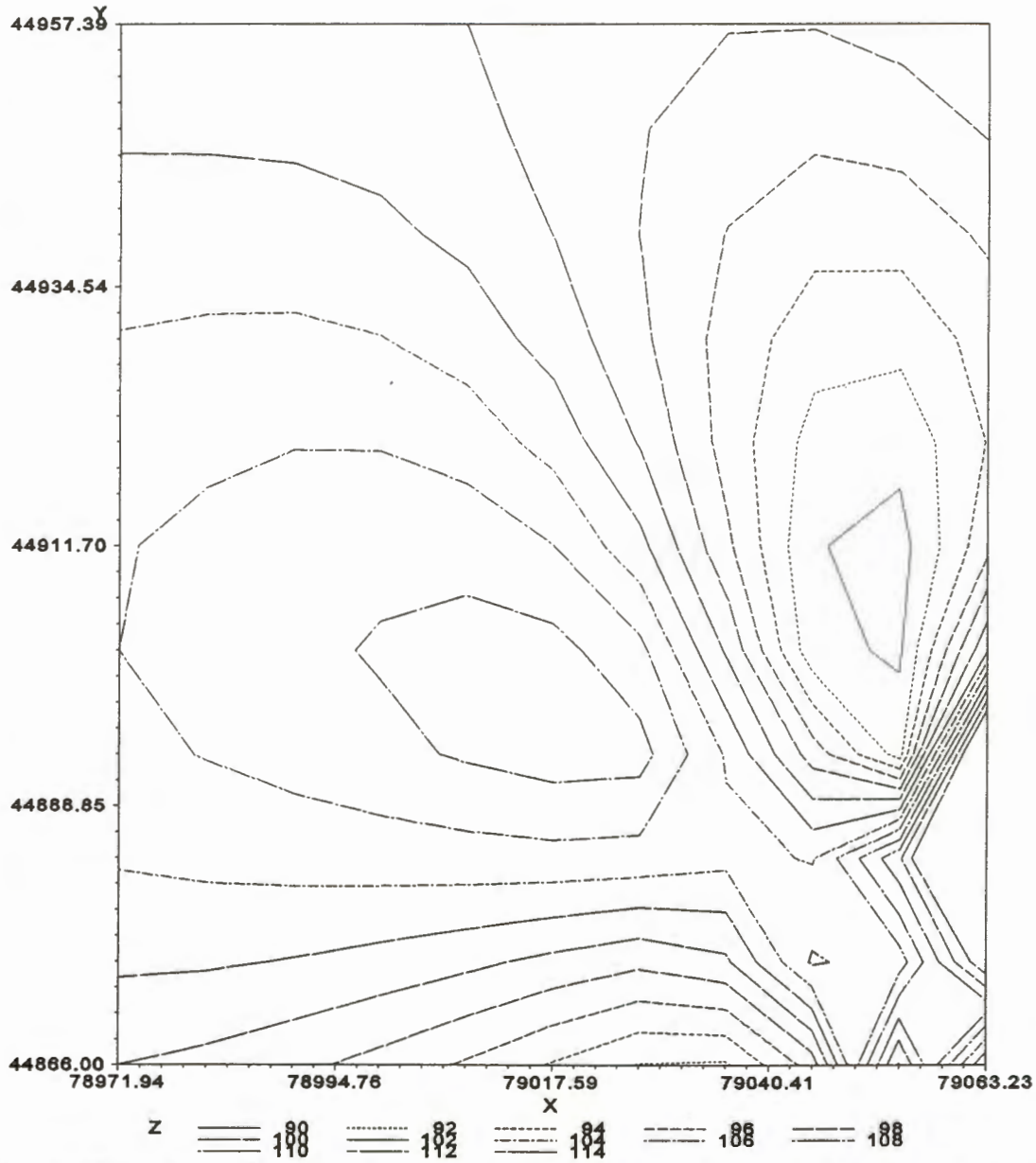


Figure 7.2.2.2 *Contour plot of the area shown in figure 7.2.1.1 (a). (values are in metres)*

some areas are evidence of mismatches. Interactively editing the matching points manually can help to eliminate some of the mismatches. Alternatively a matching strategy combining the above method for locating potential matches with a feature based method based on matching extractable geometrical shapes can be used. Extracting suitable geometric features requires considerable image processing and is a major area of current research.

7.3. CONCLUSION

The examples in this chapter indicate the manner in which the various techniques described in the thesis can be combined to extract three dimensional information from stereo images. With sufficiently powerful computers the strategies adopted could be fully automated.

8.CONCLUSIONS AND RECOMMENDATIONS

The main objective of this thesis was to investigate the application of some image processing, pattern recognition and computer vision techniques towards the development of an automated photogrammetric system. The system that is ultimately envisaged is an "intelligent measurement robot" that has measurement and image understanding abilities.

8.1 CONCLUSIONS

1. The ill-posed nature of the general inverse problem of object reconstruction from images requires that problem constraints need to be introduced to obtain a solution. For the foreseeable future this places limits on the range of applicability of any particular automated system.

2. Excepting in well prepared environments, digital image matching is not completely automated. By using pattern recognition techniques, supervised and unsupervised learning can be introduced into a photogrammetric system enhancing its interpretive abilities. Matching objects can then be located and classified in images of the same scene. When used in conjunction with other matching techniques and with a knowledge base an automated system for object identification and measurement is possible.

3. Two automated industrial quality control applications illustrated the operation of simple image understanding systems. Image processing techniques were used to extract features from the images.

In the one application a knowledge base in the form of a specific barcode string convention enabled the extracted strings to be decoded. In the other application a knowledge base of various features extracted from label view classes was used to enable the machine identification of any particular label.

4. Moment invariants were discussed as feature vector components when using a statistical pattern recognition approach for object classification and localization. Geometric, Zernike and Pseudo-Zernike moments were found to be effective descriptors of objects in an image.

5. Provided the images being matched are at approximately the same scale and radiometrically similar, feature vectors with moment invariant components can be used to locate matching objects between images. They can also be used to locate specific patterns such as corners and therefore act as interest operators. Image processing techniques are invoked to preprocess the images prior to matching.

6. An interest operator based on the angles between neighbouring edge pixels was developed and found to provide evenly distributed points when applied to aerial scenes of suburban areas.

7. A descriptor based on the two-dimensional Fourier transform was found to be very sensitive to image noise and not effective as a general pattern descriptor. The two-dimensional Fourier transform can however be used effectively for determining image texture based on fractal dimension and used for image segmentation. With predetermined parameters in a knowledge base this method of image segmentation can be incorporated into a system for identifying regions in images, such as urban areas or forests.

8. An automated measurement system requires accurate point matching between images. Various methods of obtaining accurate point matching were investigated and compared. The L1-norm method exhibits greater robustness to outlier points and can improve the accuracy of subpixel area-based point matching.

9. A method for subpixel feature extraction using a pattern recognition approach was investigated and shown to be effective

in extracting simple patterns such as those used on targeted objects.

The idea of a feature space distance transform emerged from this investigation. The transform can be used for obtaining the possible locations of particular objects in an image, and therefore as a means of segmenting the image.

10. A technique for matching feature points extracted to subpixel accuracy between images based on dynamic programming was investigated. As matching is a constrained optimization problem application of optimization techniques is appropriate. The function to be optimized is found to be problem specific.

11. A non-contact measurement system without a-priori information about the objects to be measured, requires a robust means of determining the relative orientation between the imaging devices used to capture the images. A relative orientation algorithm incorporating the epipolar line constraint was discussed, and can be used for obtaining relative orientation parameters from points matched to subpixel accuracy.

12. A basic system was designed that could be automated for generating digital elevation models from aerial scenes. The system was illustrated by determining the digital elevation model of a building from an aerial scene. Such a system would require faster computer hardware than is currently available on a PC to be practical in a production environment.

13. The continuous relaxation matching technique can be invoked in an automated system as no a priori information about a scene is required to apply this method. This method requires a suitable means of extracting points to subpixel accuracy, such as the pattern recognition techniques described in this thesis, to generate lists of points to be matched between images.

14. The pattern recognition and matching techniques described in this thesis can be integrated into an image understanding system

that has both measuring and interpretation abilities. Currently the main constraint to the development of such a system is the considerable computational power required. Once low cost gigaflop computers are available such a system can be used in a near-real-time production environment.

8.2 RECOMMENDATIONS

A large amount of research and development is being undertaken at present in knowledge-based computer vision systems. Some of the developed systems have a very limited domain of application. ACRONYM *(Brooks,1983)*, MOSAIC *(Herman and Kanade, 1986)* and SPAM *(McKeown, Wilson and McDermott, 1985)* are among the currently developed systems.(Details of these references can be found in *Haralick and Shapiro, 1993.*)

The methods discussed in this thesis could be combined into a similar system with a more general domain of application. The statistical pattern recognition approach is suitable for building up a knowledge-base by supervised learning. Descriptors of various objects in an image can be built up from training samples of images of the object. These descriptors are then used to locate an object in different images of the same scene. Points are then extracted with interest operators and matched using a combination of relaxation and area-based techniques. A surface reconstruction is then performed and compared with specific known dimensional data of the object in the knowledge-base. By iteratively refining this procedure, and depending on the amount of detailed information in the knowledge-base, a detailed three dimensional description of a scene can be built up.

Further investigations are required concerning the detailed operation of the methods described in this thesis. As the surface reconstruction problem is essentially an optimization problem, more research should be undertaken in applying established and newly developed optimization techniques in this area.

REFERENCES

The following abbreviations have been used:

PAMI. IEEE Transactions on Pattern Recognition and Machine Intelligence.

SIAM J. Numer. Anal. SIAM Journal of Numerical Analysis.

CACM. Communications of the Association for Computing Machinery.

IAPRS. International Archives of Photogrammetry and Remote Sensing.

SMC. IEEE Transactions on Systems, Man and Cybernetics.

PERS. Photogrammetric Engineering and Remote Sensing.

Phil. Trans. R. Soc. Lond. Philosophical Transactions of the Royal Society of London.

CVGIP. Computer Vision, Graphics and Image Processing.

Proc. R. Soc. Lond. Proceedings of the Royal Society of London.

SPIE Society of Photo-Optical Instrumentation Engineers.

Abu-Mostafa Y. and Psaltis D.,1984. Recognitive Aspects of Moment Invariants. Vol.PAMI-6, No.6., pp 698-706.

Ackermann F.,1984. Digital Image Correlation:Performance and Potential Applications in Photogrammetry, Photogrammetric Record, 11(64), pp. 429-439.

Aho A.,Hopcroft V.J.E.,and Ullman J.D.,1983. Data Structures and Algorithms., Addison-Wesley, Reading, MA.

Baltsavias E.P.,1991. Multiphoto geometrically constrained matching, Mitteiliungen nr. 49, Institut fur Geodasie and Photogrammetrie, Zurich.

Barnard S.T.,Thompson W.B.,1980. Disparity analysis of Images. Vol PAMI-2, No.4., pp 333-340.

Barrodale I. and Young A.,1966. Algorithms for best L_1 and L_∞ linear approximations on a discrete set. Numerische Mathematik, 8, pp.295-306.

Barrodale I. and Roberts F.D.K.,1973. An improved algorithm for discrete l_1 linear approximation. SIAM J. Numer. Anal., Vol.10, No. 5, pp.839-848.

Barrodale I. and Roberts F.D.K., 1974. Algorithm 478. Solution of an overdetermined system of equations in the L_1 norm, CACM, 14, pp. 319-320.

Bernard M.,1984. Automatic stereophotogrammetry: A method based on feature detection and dynamic programming, Photogrammetria, Vol. 39, No. 4, pp. 141-146.

Beyer H.A.,1992(a). Advances in Characterization and Calibration of Digital Image Systems. IAPRS, Vol. 29 , pp V:552.

Beyer H.A.,1992(b). Geometric and Radiometric Analysis of a CCD-Camera based Photogrammetric Close-Range System.ETH-Zurich, Mitteilungen Nr. 51.

Bhanu B. and Faugeras O.D., 1984. Shape Matching of Two-Dimensional Objects, PAMI-6, No. 2., pp. 137-155.

Bloomfield P and Steiger W.L.,1983. Least Absolute Deviations-Theory,Applications and Algorithms. Birkhäuser.

Born M. and Wolf E.,1965. Principles of Optics. Pergamon Press.

Brown D.C.,1971. Close-Range Camera Calibration, Photogrammetric Engineering, pp. 855-866.

Brown D.C.,1976. The bundle adjustment - progress and prospects. IAPRS, Vol.21, B3.

Burkinshaw N.,1992. Going Behind Barcodes, Electronics World + Wireless World, pp. 164-167.

Calitz M.F. and Rüther H, 1994. L-Norm Methods in Area-based Matching, IAPRS Vol. 30 V, pp. 22-27.

Canny J.,1986. A Computational Approach to Edge Detection. Vol.PAMI-8, No.6, pp 679-698.

Chellappa R. and Bagdazian R., 1984. Fourier Coding of Image Boundaries. Vol. PAMI-6, No.1., pp 102-105.

Dantzig G.B., 1949. Programming of interdependent activities , Econometrica, 17, pp. 200-211.

De Mori R and Probst D, 1986. Computer Recognition of Speech, in Handbook of Pattern Recognition and Image Processing, eds T.Y. Young and K-S Fu, pp.499-525.

Dhond U.R. and Aggarwal J.K., 1989. Structure from Stereo-A Review, SMC-19, No.6., pp.1489-1510.

Dudani S.A., Breeding K.J., McGhee R.B.,1977. Aircraft Identification by Moment Invariants. IEEE Transactions on Computers, pp 39-45.

Ebner H. and Heipke C., 1988. Integration of Image Matching and Object Reconstruction, IAPRS Vol. 27, B11, III, pp. 534-545.

El-Hakim S.F.,1986. Real-Time Image Metrology with CCD cameras.PERS , Vol.52, No. 11, pp. 1757-1766.

Faugeras O.D. and Price K.E., 1981. Semantic Description of Aerial Images using Stochastic Labelling, PAMI-3, No.6, pp. 633-642.

Förstner W., 1982. On the Geometric Precision of Digital Correlation, IAPRS, Vol. XXIII, Comm. III, pp. 176-189.

Förstner W. ,Gülch E., 1987. A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular features.Proc. of Intercommission Conference on Fast Processing of Photogrammetric Data., pp 281-305.

FORTRAN Scientific Subroutine Library, Peerless Engineering Service,1984. John Wiley & Sons, Inc.

Fukunaga K.,1990. Introduction to Statistical Pattern Recognition. (second edition). Academic Press.

Gigus Z. and Malik J.,1988. Computing the Aspect Graph for Line Drawings of Polyhedral Objects, Proceedings of IEEE Conference on Computers Vision and Pattern Recognition, pp. 654-661.

Gigus Z.,Canny J.,and Seidel R.,1988. Efficiently Computing and Representing the aspect Graphs of Polyhedral Objects, Proceedings of Second International Conference on Computer Vision, pp. 30-39.

Gonzalez R.C. and Wintz P.,1987. Digital Image Processing. Addison-Wesley Publishing Company.

Gottschalk P.G.,Mudge T.,1988. Efficient Encoding of Local Shape: Features for 3-d Recognition. SPIE , Vol. 1002, Intelligent Robots and Computer Vision, pp. 46-57.

Greenfeld J.S.,1991. An operator based matching system, PERS, Vol. 57, No. 8,pp. 1049-1055.

Grimson W.E.L.,1981. A Computer Implementation of a Theory of Human Stereo Vision, Phil. Trans. R. Soc. Lond., B 292, pp. 217-253.

Grimson W.E.L.,1982. A Computational Theory of Visual Surface Interpolation, Phil. Trans. R. Soc. Lond., B 298, pp. 395-427.

Grimson W.E.L. and Huttenlocher D.P.,1990. On the Sensitivity of Geometric Hashing, pp.334-338.

Gruen A. 1994. Digital Close-Range Photogrammetry - Progress Through Automation. IAPRS, Vol. 30 ,pp. V:122-135.

Gruen A. and Baltsavias E.P.,1985. Automatic 3-D Measurement of Human Faces with CCD-cameras. Proc. of SPIE, Vol. 1030, pp. 106-116.

Gruen A.,1985. Adaptive Least Squares Correlation - A Powerful Image Matching Technique, South African Journal of Photogrammetry , Remote Sensing and Cartography, 14(3), pp.175-187.

Gustafson P.C.,1988. The application of real-time and near real-time photogrammetry in industry: A test of accuracy. IAPRS, Vol. 27/B10, pp. V:198-205.

Helava U.V.,1978. Digital correlation in photogrammetric instruments, Photogrammetria, Vol. 34, pp. 19-41.

Helava U.V.,1988. Object-Space Least-Squares Correlation, PERS, Vol.54(6)., pp. 711-714.

Haralick R.M. and Shapiro L.G.,1993. Computer and Robot Vision , Vol.2., Addison-Wesley Publishing Company.

Haralick R.M. and Shapiro L.G.,1993. Computer and Robot Vision , Vol.1., Addison-Wesley Publishing Company.

Haralick R.M.,1981. Some Neighbourhood Operators, Real-Time Parallel Computing Image Analysis, M. Onoe,K. Preston,Jr.,and A. Rosenfeld(eds.), Plenum, New York.

Haralick R.M. and Shapiro L.G., 1979. The Consistent Labelling Problem I, PAMI-1, pp. 173-184.

Haralick R.M. and Shapiro L.G., 1980. The Consistent Labelling Problem II, PAMI-2, pp. 193-203.

Haralick R.M. and Kartus J, 1978. Arrangements, Homomorphisms and Discrete Relaxation, SMC-8, pp. 600-612.

Hecht E. and Zajac A.,1974. Optics. Addison-Wesley Publishing Company.

Hellwich O. and Faig W., 1994. Graph-Based Feature Matching Using Descriptive and Relational Parameters, PERS, Vol. 60., No.4, pp. 443-450.

Horn B.K.P.,1975. Obtaining Shape from Shading Information. in The Psychology of Computer Vision (Winston, P.H. Ed.) , pp. 115-155.

Horn B.K.P. and Brooks M.J.,1986. The Variational Approach to Shape from Shading. CVGIP , Vol. 33, pp. 174-208.

Horowitz E. and Sahni, 1982. Fundamentals of Data Structures., Computer Science Press, Rockville, MD.

Hu M.K.,1962. Pattern Recognition by Moment Invariants. Proc. IRE, vol. 49, pp. 1428.

Ikeuchi K.,1987. Generating an Interpretation Tree from a CAD Model for 3D-Object Recognition in Bin-Picking Tasks, International Journal of Computer Vision, Vol.1, pp. 145-165.

Ikeuchi K. and Hong K.S.,1989. Determining Linear Shape Change: Toward Automatic Generation of Object Recognition Programs, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 450-457.

Koenderink J.J. and Van Doorn A.J., 1979. The Internal Representation of Solid Shape with Respect to Vision, Biological Cybernetics, Vol.32, pp. 211-216.

Kube P. and Pentland A.P.,1988. On the Imaging of Fractal Surfaces. Vol. PAMI-10, No.5., pp 704-707.

Kubik K., Merchant D., Schenk T., 1987. Robust estimation in photogrammetry, PERS, Vol. 53, No. 2,pp. 167-169.

Lamdan Y., Schwartz J.T. and Wolfson H.J.,1988. Object Recognition by Affine Invariant Matching, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition., pp 335-344.

Lemmens M.,1988. A survey on Stereo Matching Techniques, IAPRS, Vol. 27, Part B 8, pp. v11-v23.

Li G.,1985. Robust regression. In: Exploring data, tables, trends and shapes. Eds. Hoaglin D.C., Mosteller F., and Tukey J.W.,John Wiley & sons, New York, pp. 281-344.

Li J.C. and Schenk T., 1990. Aerial image matching using Ψ-S representation, Tech. Notes in Photogrammetry, Dept. of Geodetic Science and Surveying, Ohio State University, No. 9.

Lin C.C. and Chellappa R.,1987. Classification of Partial 2-D Shapes Using Fourier Descriptors. PAMI-9, No.5., pp686-690.

Loser R., Luhmann Th.,1992. The programmable optical 3D measuring system POM - applications and performance. IAPRS , Vol.29/B5, pp. V:533-540.

Lumia R.,Shapiro L.G., and Zuniga O.,1983. A New Connected Components Algorithm for Virtual Memory Computers, CVGIP, Vol 22., pp. 287-300.

Mallat S.G.,1989. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. Vol. PAMI-11, No. 7., pp. 675-693.

Mandelbrot B.B.,1983. The Fractal Geometry of Nature , W.H.Freeman.

Marr D and Poggio T.,1979. A computational theory of Human Stereo Vision, Proc. R. Soc. Lond., B 204,pp. 301-328.

McKeown D.M., 1988. Building Knowledge-based Systems for Detecting Man-made Structures from Remotely Sensed Imagery, Phil. Trans. R. Soc. Lond. A 324, pp. 423-435.

Medioni G. and Nevatia R., 1984. Matching Images using Linear Features, PAMI-6, No.6., pp. 675-685.

Moravec H.P.,1977. Towards automatic visual object avoidance. Proceedings of 5th International Joint Conference on Artificial Intelligence, pp. 584.

Moravec H.P.,1980. Object avoidance and navigation in the real world by a seeing robot rover, Ph. D. Desertation, Stanford Univ., Stanford.

Muller J.P.A.L.,1988. Key Issues in Image Understanding in Remote Sensing, Phil. Trans. R. Soc. Lond. A 324, pp. 381-395.

Nandhakumar N. and Aggarwal K.,1985. The Artificial Intelligence Approach to Pattern Recognition - A Perspective and an Overview. Pattern Recognition, Vol 18., No 6., pp. 383-389.

Nasrabadi N.M., Li W. and Choo C.Y.,1990. Object Recognition by Hopfield Neural Network, IEEE 3rd International Conference on Computer Vision, pp. 325-328.

Nayar S.K.,Ikeuchi K and Kanade T,1991. PAMI , Vol 13, No. 7,pp. 611-634.

Ohta Y. And Kanade T., 1985. Stereo by intra and inter scanline search using dynamic programming, PAMI-7, pp. 139-154.

Otsu N.,1979.A Threshold Selection Method for Grey-Level Histograms, Vol. SMC-9, pp.62-69.

Pattern Analysis and Machine Intelligence (PAMI) Vol. 10, No.1, January 1988.(Special Issue devoted to applications of machine vision).

Pattern Analysis and Machine Intelligence (PAMI) Vol. 14, No.2, February 1992.(Special Issue devoted to 3-D object modelling).

Pavlidis T., 1968. Computer Recognition of Figures through Decomposition, Information and Control, Vol. 14.,pp. 526-537.

Pavlidis T.,Swartz J. and Wang Y.P.,1990. Fundamentals of barcode information theory, Computer, April, pp. 74-86.

Pavlidis T.,Swartz J. and Wang Y.P.,1992. Information encoding with two-dimensional bar codes, Computer, June ,pp. 18-28.

Pilgrim L.,1992. Two Dimensional Image Matching and Difference Detection, Australian Journal of Geodesy, Photogrammetry and Surveying, No.56, pp. 1-36.

Plantinga W.H., and Dyer C.R.,1986. An Algorithm for Construction of the Aspect Graph, Proceedings of IEEE Symposium on Foundations of Computer Science, pp. 123-131.

Prokop R.J., and Reeves A.P.,1992. A survey of Moment-Based Techniques for Unoccluded Object Representation and Recognition,CVGIP, Vol 54, No. 5., pp. 438-460.

Pentland A.P.,1984. Fractal-Based Description of Natural Scenes. Vol. PAMI-6, No.6., pp. 661-674.

Price K.E., 1985. Relaxation Matching Techniques-A Comparison, Vol. PAMI-7, No.5., pp. 617-623.

Ronse C., and Devijer P.A.,1984. Connected Components in Binary Images:The Detection Problem, Research Studies, Letchworth, Herts, England.

Rosenfeld A., 1974. Compact Figures in Digital Pictures, Vol. SMC-4, pp. 221-223.

Rosenfeld A., Hummel R.A., and Zucker S.W., 1976. Scene Labeling by Relaxation Operations, Vol. SMC-6, pp. 420-433.

Rosenfeld A. and Pfaltz J.L.,1966. Sequential Operations in Digital Processing. Journal of the Association of Computing Machinery, Vol. 13, pp. 471-494.

Sanz J.L.C. and Dinstein I.,1987. Projection-based Geometrical Feature Extraction for Computer Vision:Algorithms in Pipeline Architectures", Vol PAMI-9, pp. 160-168.

Sarkar S. and Boyer K.L. , 1991. On Optimal Infinite Impulse Response Edge Detection Filters, PAMI-13, No. 11., pp 1154-1171.

Schenk, T.,1990. Computation of Epipolar Geometry, Report No. 5., Department of Geodetic Science and Surveying, Ohio State University.

Schlossmacher, E.J.,1973. An iterative technique for absolute deviation curve fitting, Journal of the American Statistical Association, 68, pp. 857-865.

Sobel, I.E.,1970. Camera Models and Machine Perception, Ph. D. Thesis, Stanford, CA.

Stockman G., Kopstein S. and Benett S., 1982. Matching Images to Models for Registration and Object Detection via Clustering, pp. 229-241.

Streilen A.,1992. Digital photogrammetric techniques in architectural design. IAPRS, Vol.29/B5, pp. V:825-831.

Szeliski R., 1991. Fast Shape from Shading. CVGIP:Image Understanding , Vol. 53, No.2, pp. 129-153.

Teague M.R.,1980. Image Analysis via the general theory of moments. Journal of the Optical Society of America, Vol.70, No.8, pp. 920-930.

Teh C.-H. and Chin R.T.,1986. On Digital Approximations of Moment Invariants, CVGIP, vol.33, pp. 318-326.

Teh C.-H. and Chin R.T.,1988. On Image Analysis by the Methods of Moments.Vol. PAMI-10, No.4., pp. 496-513.

Terzopoulos D.,1988. The Computation of Visible-Surface Representations, PAMI-10, No.4, pp. 417-438.

Tian Q. and Huhns M.N., 1986. Algorithms for Subpixel Registration, CVGIP, Vol.35., pp. 220-233.

Tou J.T. and Gonzalez R., 1981. Pattern recognition principles, Addison-Wesley, Reading, Massachusetts.

Van der Merwe N. and Rüther H., 1994. Image Matching through a Combination of Feature - and Area Based Matching, IAPRS Vol. 30, pp. V:407-413.

Van der Vlugt G. and Rüther H.,1992. A Real-time photogrammetric system for patient positioning in proton therapy. IAPRS , Vol.29/B5, pp. V:880-884.

Van der Vlugt G. and Rüther H.,1994. The Development of an Automated Surface Measurement System, IAPRS Vol. 30, pp. V:414-429.

Wagner H,M.,1959. Linear Programming Techniques for Regression Analysis, J. Amer. Statist. Assoc., 54, pp. 206-212.

Waltz D., 1975. Understanding Line Drawings of Scenes with Shadows, in The Psychology of Computer Vision, ed. P.Winston, McGraw-Hill.

Watson G.A., 1980. Approximation Theory and Numerical Methods. Wiley.

Wezka J.S.,Dyer C.R. and Rosenfeld A.,1976. A Comparative Study of Texture Measures for Terrain Classification. Vol. SMC-6, pp. 269-285.

Wildschek, R, 1989. Surface Capture using Near-Real-Time Photogrammetry for a Computer Controlled Milling System, M.Sc. Thesis, University of Cape Town.

Wolfe W.J.,Weber-Sklair C.,Mathis D. and Magee M.,1988. Locating known objects in 3-D from a single perspective view. SPIE, Vol. 1002, Intelligent Robots and Computer Vision, pp. 550-566.

Wong Kam W. and Ho Wei-Hsin.,1986. Close range mapping with a solid state camera. PERS, Vol. 52, No. 1, pp. 67-74.

Wrobel B.,1987. Facets Stereo Vision (FAST)- A new Approach to Computer Stereo Vision and to Digital Photogrammetry, Proceedings of Conference on Fast Processing of Photogrammetric Data, Interlaken, pp. 231-258.

Zheng Y-J,1993. Digital Photogrammetric Inversion: Theory and Application to Surface Reconstruction, PERS, Vol. 59, No. 4, pp. 489-498.

Zhou X. and Dorrer E., 1994. An Automatic Image Matching Algorithm Based on Wavelet Decomposition, IAPRS, Vol. 30, Part 3/2, pp. 951-960.

APPENDIX A

C PROGRAMS FOR THE L1-NORM ALGORITHMS

A.1. BARRODALE-ROBERTS ALGORITHM FOR L1 NORM

Variables and data

**a is a double pointer variable to a two dimensional array a[m2][n2]

where

m2 = m+2 and m is the number of equations.

n2 = n+2 and n is the number of unknowns.

n=6 when the algorithm is used to determine the affine parameters in the matching method described in section 5.1.3.

The matrix of the coefficients of the equations are stored in a[i][j]

where

i ∈ [1,m] and j ∈ [1,n]

The column vector representing the right hand side of the equations is stored in b[i]

where

i ∈ [1,m]

x[j] where j ∈ [1,n] contains the L1-norm solution on exiting the routine.

e[i] where i ∈ [1,m] contains the residuals on exiting the routine.

a[m+1][n+1] contains the minimum sum of the absolute value of the residuals.

a[m+1][n+2] contains the rank of the matrix.

a[m+2][n+1] contains an exit code of 1 for successful completion or 2 for premature completion.

a[m+2][n+2] contains the number of iterations required.

TOLER should be set to $10^{-2d/3}$ where d represents the number of

decimal digits of accuracy required.

The Barrodale-Roberts Simplex Subroutine.

```
int L1BR(int m,int n,int m2,int n2,double **a,double
b[300],double toler,double x[9],double e[300])
{double sum,min,max,d,pivot;
 int out,stage,test,s[300];
 int m1,n1,i,j,in,k,l;
 int kount,kr,kl;
 double big=1.0e+75;
/* INITIALIZATION */
m1=m+1;
n1=n+1;
for(j=1;j<=n;j++)
{a[m2][j]=j;
 x[j]=0;
 }
for(i=1;i<=m;i++)
{a[i][n2]=n+i;
 a[i][n1]=b[i];
 if(b[i]>=0)goto lab1;
 for(j=1;j<=n2;j++)
 a[i][j]=-a[i][j];
 lab1: e[i]=0;
 }
/*COMPUTE THE MARGINAL COSTS */
for(j=0;j<=n1;j++)
{ sum=0;
  for(i=1;i<=m;i++)sum=sum+a[i][j];
  a[m1][j]=sum;
  }
/* STAGE I */
/* DETERMINE THE VECTOR TO ENTER THE BASIS */
stage=1;
kount=0;
kr=1;
kl=1;
lab2: max=-1.0;
for(j=kr;j<=n;j++)
{if(fabs(a[m2][j]>n))goto lab3;
d=fabs(a[m1][j]);
if(d<=max)goto lab3;
max=d;
in=j;
lab3:
}
if(a[m1][in]>=0)goto lab4;
for(i=1;i<=m2;i++)a[i][in]=-a[i][in];
/*DETERMINE THE VECTOR TO LEAVE THE BASIS */
lab4: k=0;
 for(i=kl;i<=m;i++)
 { d=a[i][in];
  if(d<=toler)goto lab5;
```

```
   k=k+1;
   b[k]=a[i][n1]/d;
   s[k]=i;
   test=1;
lab5:
  }
  lab6:if(k>0)goto lab7;
  test=0;
  goto lab9;
  lab7:min=big;
  for(i=1;i<=k;i++)
  {if(b[i]>=min)goto lab8;
  j=i;
  min=b[i];
  out=s[i];
  lab8:
  }
  b[j]=b[k];
  s[j]=s[k];
  k=k-1;
  /*CHECK FOR LINEAR DEPENDENCE IN STAGE I */
  lab9:if(test || !stage)goto lab10;
  for(i=1;i<=m2;i++)
  {d=a[i][kr];
  a[i][kr]=a[i][in];
  a[i][in]=d;
  }
  kr=kr+1;
  goto lab17;
  lab10:if(test)goto lab11;
  a[m2][n1]=2;
  goto lab23;
  lab11:pivot=a[out][in];
  if((a[m1][in]-2.0*pivot)<=toler)goto lab12;
  for(j=kr;j<=n1;j++)
  {d=a[out][j];
  a[m1][j]=a[m1][j]-2.0*d;
  a[out][j]=-d;
  }
  a[out][n2]=-a[out][n2];
  goto lab6;
  /*PIVOT ON a[out][in] */
lab12:for(j=kr;j<=n1;j++)
{if(j==in)goto lab13;
a[out][j]=a[out][j]/pivot;
lab13:
}
for(i=1;i<=m1;i++)
{if(i==out)goto lab15;
d=a[i][in];
for(j=kr;j<=n1;j++)
{if(j==in)goto lab14;
a[i][j]=a[i][j]-d*a[out][j];
lab14:
}
lab15:
```

```
}
for(i=1;i<=m1;i++)
{if(i==out)goto lab16;
a[i][in]=-a[i][in]/pivot;
lab16:
}
a[out][in]=1.0/pivot;
d=a[out][n2];
a[out][n2]=a[m2][in];
a[m2][in]=d;
kount=kount+1;
if(!stage)goto lab18;
/*INTERCHANGE ROWS IN STAGE I */
kl=kl+1;
for(j=kr;j<=n2;j++)
{d=a[out][j];
a[out][j]=a[kount][j];
a[kount][j]=d;
}
lab17:if((kount+kr)!=n1)goto lab2;
/* STAGE II */
stage=0;
/*DETERMINE THE VECTOR TO ENTER THE BASIS */
lab18: max=-big;
 for(j=kr;j<=n;j++)
 {d=a[m1][j];
 if(d>=0)goto lab19;
 if(d>-2.0)goto lab20;
 d=-d-2;
 lab19:if(d<=max)goto lab20;
 max=d;
 in=j;
 lab20:
 }
 if(max<=toler)goto lab21;
 if(a[m1][in]>0)goto lab4;
 for(i=1;i<=m2;i++)a[i][in]=-a[i][in];
 a[m1][in]=a[m1][in]-2;
 goto lab4;
 /*PREPARE OUTPUT*/
lab21:l=kl-1;
for(i=1;i<=l;i++)
{if(a[i][n1]>=0)goto lab22;
 for(j=kr;j<=n2;j++)a[i][j]=-a[i][j];
lab22:
 }
a[m2][n1]=0;
if(kr!=1)goto lab23;
for(j=1;j<=n;j++)
{d=fabs(a[m1][j]);
if((d<=toler)||((2.0-d)<=toler))goto lab23;
}
a[m2][n1]=1;
lab23:for(i=1;i<=m;i++)
{k=a[i][n2];
d=a[i][n1];
```

```
if(k>0)goto lab24;
k=-k;
d=-d;
lab24:if(i>=k1)goto lab25;
x[k]=d;
goto lab26;
lab25:k=k-n;
e[k]=d;
lab26:
}
a[m2][n2]=kount;
a[m1][n2]=n1-kr;
sum=0;
for(i=k1;i<=m;i++)sum=sum+a[i][n1];
a[m1][n1]=sum;
return 0;
}
```

A.2. BARRODALE-YOUNG ALGORITHM FOR L1 NORM

Variables and data

**q is a double pointer variable to a two dimensional array q[N2][n3]

where

N2 = N+2 and N is the number of equations.

n2 = n+3 and n is the number of unknowns.

n=6 when the algorithm is used to determine the affine parameters in the matching method described in section 5.1.3.

The matrix of the coefficients of the equations are stored in q[i][j]

where

$i \in [1,N]$ and $j \in [1,n]$

The column vector representing the right hand side of the equations is stored in q[i][0]

where

$i \in [1,N]$

After convergence the solution vector l[j] where $j \in [1,n]$ is extracted from the matrix column q[j][0] using the routine getl() given below.

q[0][0] contains the minimum sum of the absolute value of the residuals.

q[N+1][n+2] contains an exit code of 1 for successful completion or 2 for premature completion.

q[0][n+2] contains the number of iterations required.

The Barrodale-Young Simplex Subroutine.

```
int L1BY(double **q,int n,int N)
{int i,j,t,in,out,it;
double a,b,d,p,z;
int B;
B=0;
for(j=1;j<=(n+1);j++){q[0][j]=0;
                      q[(N+1)][j]=N+j;
                      }
q[0][0]=q[(N+1)][0]=0;
for(i=1;i<=N;i++){q[i][n+2]=i;
                  a=0;
                  if(q[i][0]<0)B=1;
                  for(j=0;j<=n;j++){
                       if(B==1)q[i][j]=-q[i][j];
                       a=a-q[i][j];
                       q[0][j]=q[0][j]+q[i][j];
                       }
                   if(B==1)q[i][n+2]=-q[i][n+2];
                   q[i][n+1]=a+q[i][0];
                   q[0][n+1]=q[0][n+1]+q[i][n+1];
                   B=0;}
     it=-1;
step1:
       a=1.0e-8;t=0;it=it+1;q[0][n+2]=it;
       for(j=1;j<=(n+1);j++){z=q[0][j];
          if(q[(N+1)][j]>N)goto L1;
          if((-z-2)>a){in=j;
                       t=2;
                       a=-z-2;
                       goto L2;
                       }
L1:     if(z>a){in=j;
          t=1;
          a=z;
          }

L2:     }

       if(t==0){q[(N+1)][n+2]=1;
          goto OUT;
          }

step2: b=1.0e+10;
       for(i=1;i<=N;i++){d=q[i][in];
                         if(t==2)d=-d;
                         if(d<1.0e-8)goto L3;
                         d=q[i][0]/d;
                         if(d<b){b=d;
                                 out=i;
                                 }
L3:    }
       if(b>=1.0e10){q[(N+1)][n+2]=2;
```

```
                goto OUT;
                }
        if(t==2){q[(N+1)][in]=-q[(N+1)][in];
                q[0][in]=-a;
                }
step3:  p=q[out][in];
        for(i=0;i<=N;i++){if(i==out)goto X;
                d=q[i][in]/p;
                for(j=0;j<=(n+1);j++)
                if(j==in)q[i][j]=-d;
                        else
                        q[i][j]=q[i][j]-d*q[out][j];
X:                              }
        p=fabs(p);
    for(j=0;j<=(n+1);j++)if(j==in)q[out][j]=1.0/p;
                        else
                        q[out][j]=q[out][j]/p;
                i=q[out][n+2];
                q[out][n+2]=q[(N+1)][in];
                q[(N+1)][in]=i;
                goto step1;
OUT:                            return 0;
}
```

Subroutine to extract solution from matrix q[][].

```
int getl(double l[],double **q,int nn,int N)
{int i,j,ok,ex;
 double compare;
 for(i=(N+1);i<=(N+nn+1);i++)l[i-(N+nn+1)]=0;
 for(i=(N+1);i<=(N+nn+1);i++)
 {ok=0;
  compare=i;
   for(j=1;j<=N;j++)
    if(q[j][nn+2]==compare)
     {l[i-(N+1)]=q[j][0];
      ok=1;}
   if(ok==0)ex=i;
   }

if(ex!=(N+nn+1))for(i=(N+1);i<=(N+nn);i++)l[i-(N+1)]=l[i-(N+1)
]-l[nn];
  return 0;
  }
```

APPENDIX B

HISTOGRAM EQUALIZATION

Histogram equalization can be used to compensate for radiometric differences between images of the same scene. The resultant equalized images will have similar grey-scale distributions.

Consider an N X M pixel image with 256 grey-scale values ranging from 0 to 255. In order to obtain the equalized image the grey-scale values in the image are altered as follows:-

$$c_N = \frac{255\ h(c_o)}{h(255)} \qquad \text{(B.1)}$$

where

$h(c_o) = \sum_{c=0}^{c_o} g(c)$ is the cumulative grey-scale distribution function of the original image.

g(c) is the number of pixels with grey-scale value c in the original image.

c_o is the grey-scale value in the original image.

c_N is the grey-scale value in the equalized image.

ORTHOGONAL MOMENTS

C.1 ZERNIKE MOMENTS

The controling subroutines for capturing the pixel data from the image, determining the Zernike moments and then recreating the image are as follows:-

```
get_patch(f,dim,xp,yp);
determine_coeff(f,l,dim,num,x,y);
recreate_patch(f,l,dim,num,x,y);
```

where

num is the highest order of Zernike moment that is to be used.

**f* is a pointer to a one dimensional array containing the two dimensional pixel data reading from left to right and top to bottom over the image patch.

dim is the length of side of an image patch and should be an odd number.

**x* and **y* are pointers to one dimensional arrays containing the respective x and y pixel coordinate of the pixels in the patch in the same sequence as the data in **f*.

xp and *yp* are the x,y coordinates of the patch selected in the full image.

**l* is a one dimensional complex array containing the calculated Zernike moments.

The detailed subroutines plus supporting routines are given below.

```
double atanc(double y,double x)
/* DETERMINES ARCTANGENT IN ALL FOUR QUADRANTS */
{double pi=3.14159;
 double a,b,ret;
 ret=0;
 if(y>0){ if(x>0)ret=atan(y/x);
           if(x<0)ret=pi+atan(y/x);
          if(x==0)ret=pi/2.0;}
        else
        { if(x>0)ret=2.0*pi+atan(y/x);
          if(x<0)ret=pi+atan(y/x);
          if(x==0.0)ret=3.0*pi/2.0;
         }
   return ret;
 }
```

```
complex Z(double n,double l,double x,double y)
/*CALCULATES ZERNIKE POLYNOMIAL ORDER nl
V_nl EQUATION (4.4.2.2.3)*/
{double m,r,R;
 double f,f1,f2,f3,f4,theta;
 complex s;
 int mm,i,j;
 double l1,l2,l3,l4,t;
 r=sqrt(x*x+y*y);
 theta=atanc(y,x);

 m=(n-fabs(l))/2.0;
 mm=m+0.01;
 s=complex(1.0,0.0);
 if(n==0)goto end;
 s=complex(0,0);
 for(i=0;i<=mm;i++){
   f=1;f1=1;f2=1;f3=1;f4=1;l1=1;l2=1;l3=1;
   for(j=1;j<=i;j++)f=f*(-1.0);
   for(j=1;j<=(n-i);j++)f1=f1*j;
   for(j=1;j<=i;j++)f2=f2*j;
   for(j=1;j<=(mm+fabs(l)-i);j++)f3=f3*j;
   for(j=1;j<=(mm-i);j++)f4=f4*j;
   for(j=1;j<=n-2.0*i;j++)l1=l1*r;
   t=l1*f*f1/(f2*f3*f4);
   s=s+complex(t*cos(l*theta),t*sin(l*theta));
}
end:
return s;
   }

int get_patch(double f[],int dim,int xp,int yp )
/* GETS PIXEL DATA FOR PATCH WITH SIDES dim CENTERED ON xp,yp*/
{int i,j,ii,jj;
 double sx,sy,r;
 for(j=yp-dim/2;j<=yp+dim/2;j++)
 {jj=j-yp+dim/2;
 for(i=xp-dim/2;i<=xp+dim/2;i++)
 {ii=i-xp+dim/2;
  sx=ii-dim/2;
  sy=jj-dim/2;
  r=sqrt(sx*sx+sy*sy);
  f[ii+jj*dim]=getpixel(i,j);
  if(r>dim/2)f[ii+jj*dim]=0;}
  }
return 0;
}


complex double_simp(double f[],double x[],double y[],int n,int
l,int dim)
/*PERFORMS DOUBLE INTEGRATION TO DETERMINE ZERNIKE MOMENTS
A_nl EQUATION (4.4.2.2.2)*/
{double wx,wy,r,r1,theta,x1,y1;
```

```
 double pi=3.14159;
 complex s,t;
 int i,j,inc;
 s=0;
 inc=10;
 for(j=0;j<=360;j=j+inc)
 for(i=0;i<=dim/2;i++)
{ r=2.0*i/(double)dim;
  if(r<=1.0){
  if((j==0)||(j==360))wy=1;
       else
     {if(((j/2)*2)==j)wy=2;
        else
        wy=4;
        }
  if((i==0)||(i==dim/2))wx=1;
       else
     {if(((i/2)*2)==j)wx=2;
        else
        wx=4;
      }
  x1=r*cos((double)j/180*pi);
  y1=r*sin((double)j/180*pi);


t=r*conj(Z(out,n,l,x1,y1))*wx*wy*(int)getpixel((int)(100+x1*di
m/2),(int)(200+y1*dim/2))/255.0;
  }
  else
  t=complex(0.0,0.0);
  s=s+t;
  }
  s=s*(n+1)/9.0*4.0*pi/360.0*inc/(double)(dim-1)/pi;
  return s;
}


int     determine_coeff(double  f[],complex  l[],int  dim,int
num,double x[],double y[])
/*CONTROLS DETERMINATION OF MOMENTS*/
{int i,j,k;
 for (j=0;j<num;j++)
  for (i=-j;i<=j;i++)
{k=j-abs(i);
 if((k/2)*2==k)l[i+2*j*num]=double_simp(out,f,x,y,j,i,dim);
 else
 l[i+2*j*num]=0;
 }
return 0;
 }


double recreate_pixel(complex l[],double x,double y,int num)
/*RECREATES PIXEL AT x,y USING EQUATION (4.4.2.2.1)*/
{complex s;
 double nn;
```

```
 int i,j,k;
 s=complex(0.0,0.0);
 for(j=0;j<num;j++)
{ for(i=-j;i<=j;i++)
 {k=j-abs(i);
 if((k/2)*2==k)s=s+Z(out,j,i,x,y)*l[i+2*j*num];}}
 nn=abs(s);
 return nn;
 }


int  recreate_patch(double f[],complex l[],int dim,int num,double
x[],double y[])
/*CONTROLS RECREATION OF PATCH*/
{int i,j,k;
 double s,r;
 for (j=0;j<dim;j++)
  for (i=0;i<dim;i++)
{r=x[i+j*dim];
s=y[i+j*dim];
r=sqrt(r*r+s*s);
if(r<=1.0)f[i+j*dim]=recreate_pixel(l,x[i+j*dim],y[i+j*dim],nu
m)*255.0;
}
return 0;
}
```

C.2 PSEUDO-ZERNIKE MOMENTS

The control structure for the pseudo-Zernike moments is the same as for Zernike moments. The subroutine Z() which calculates the polynomials is modified as follows:-

```
complex Z(double n,double l,double x,double y)
/*CALCULATES PSEUDO-ZERNIKE POLYNOMIAL OF ORDER nl */
{double m,r,R;
 double f,f1,f2,f3,f4,theta;
 complex s;
 int mm,i,j;
 double l1,l2,l3,l4,t;
 r=sqrt(x*x+y*y);
 theta=atanc(y,x);
 m=(n-fabs(l));
 mm=m;
 s=complex(1.0,0.0);
 if(n==0)goto end;
 s=complex(0,0);
 for(i=0;i<=mm;i++){
   f=1;f1=1;f2=1;f3=1;f4=1;l1=1;l2=1;l3=1;
   for(j=1;j<=i;j++)f=f*(-1.0);
   for(j=1;j<=(2*n+1-i);j++)f1=f1*j;
   for(j=1;j<=i;j++)f2=f2*j;
   for(j=1;j<=(mm-i);j++)f3=f3*j;
```

```
    for(j=1;j<=(n+fabs(l)+1-i);j++)f4=f4*j;
    for(j=1;j<=n-i;j++)l1=l1*r;
    t=l1*f*f1/(f2*f3*f4);
    s=s+complex(t*cos(l*theta),t*sin(l*theta));
}
end:
return s;
    }
```

C.3. LEGENDRE MOMENTS

The control structure is once again the same as for the Zernike moments, but details of some of the supporting subroutines are different and given below.

```
double P(double n,double x)
/*CALCULATES LEGENDRE POLYNOMIAL OF ORDER n AT COORDINATE x
EQUATION (4.4.2.1.3)*/
{double m;
 double s,f,f1,f2,f3;
 int mm,i,j;
 double l1,l2,l3,l4,t;
 m=n/2.0;
 mm=m;
 s=1;
 if(n==0)goto end;
 s=0;
 for(i=0;i<=mm;i++){
   f=1;f1=1;f2=1;f3=1;l1=1;l2=1;l3=1;
   for(j=1;j<=(2*n-2*i);j++)f=f*j;
   for(j=1;j<=i;j++)f1=f1*j;
   for(j=1;j<=(n-i);j++)f2=f2*j;
   for(j=1;j<=(n-2*i);j++)f3=f3*j;
   for(j=1;j<=(n-2*i);j++)l1=l1*x;
   for(j=1;j<=n;j++)l2=l2*2.0;
   for(j=1;j<=i;j++)l3=-1.0*l3;
   t=l3*f/(f1*f2*f3*l2)*l1;
   s=s+t;
}
end:   return s;
    }
```

```
double double_simp(double f[],double x[],double y[],int p,int
q,int dim)
/*PERFORMS DOUBLE INTEGRATION TO DETERMINE LEGENDRE MOMENTS
λpq EQUATION (4.4.2.1.2)*/
{double s,t,wx,wy;
 int i,j;
 s=0;
 for(j=0;j<dim;j++)
 for(i=0;i<dim;i++)
{
  if((j==0)||(j==(dim-1)))wy=1;
```

```
        else
      {if(((j/2)*2)==j)wy=2;
         else
         wy=4;
         }
  if((i==0)||(i==(dim-1)))wx=1;
        else
      {if(((i/2)*2)==j)wx=2;
         else
         wx=4;
       }
  t=P(p,x[i+j*dim])*P(q,y[i+j*dim])*wx*wy*f[i+j*dim];
  s=s+t;
  }

s=s*(2.0*p+1)*(2.0*q+1)/4.0/9.0*4.0/(double)(dim-1)/(double)(d
im-1);
  return s;
}

int  determine_coeff(double f[],double l[],int dim,int num,double
x[],double y[])
/*CONTROLS THE CALCULATION OF THE MOMENTS TO ORDER num*/
{int i,j;
 for (j=0;j<num;j++)
  for (i=0;i<num;i++)
 l[i+j*num]=double_simp(f,x,y,i,j,dim);
return 0;
 }

double recreate_pixel(double l[],double x,double y,int num)
/*RECREATES PIXEL ACCORDING TO EQUATION (4.4.2.1.1)*/
{double s;
 int i,j;
 s=0;
 for(j=0;j<num;j++)
{ for(i=0;i<=j;i++)
 s=s+P(j-i,x)*P(i,y)*l[j-i+i*num];}
 return s;
 }


int  recreate_patch(double f[],double l[],int dim,int num,double
x[],double y[])
/*CONTROLS RECREATION OF IMAGE PATCH*/
{int i,j;
 double s;
 for (j=0;j<dim;j++)
  for (i=0;i<dim;i++)
f[i+j*dim]=recreate_pixel(l,x[i+j*dim],y[i+j*dim],num);
return 0;
}
```

APPENDIX D

EIGENVECTORS AND EIGENVALUES

The following subroutine mateig() calculates eigenvalues and eigenvectors.

lra and n are equal to the dimension of the square matrix.

a[] contains the elements in the matrix .

s[] contains the eigenvectors at completion.

The diagonal elements of a[] are equal to the eigenvalues at completion.

This subroutine is used to diagonalize the covariance matrix given by equation (4.4.1.3.2) in chapter 4. The input matrix to the subroutine is in a[].

a[] and s[] are treated as one dimensional arrays.

At completion the eigenvectors can be printed out using

```
for(j=0;j<n;j++){
for(i=0;i<n;i++)
fprintf(data," %le ",s[j*n+i]);
fprintf(data,"\n");
}
```

```
int mateig(int lra,int n,double a[],double s[])
/*SUBROUTINE FOR DETERMINING EIGENVALUES AND EIGENVECTORS OF
MATRIX a[] WITH DIMENSION n */
{double b,anorm,constf,fnorm,e=1.0e-6,u,v,w,snn,csn,snn2;
 double c,stc,app,aqq,csn2;
 int m,in,i,j,ia,is,iq,ja,ka,la,ip,iia,iib,ii,ij,ik,il;
 constf=n;
 in=0;
 for(j=0;j<n;j++)
 for(i=0;i<n;i++)
 {is=lra*j+i;
  if((i-j)==0)s[is]=1.0;
   else
   s[is]=0;
 }
 anorm=0;
 for(j=0;j<n;j++)
 for(i=0;i<n;i++)
 if((j-i)!=0){ia=lra*j+i;
            anorm=anorm+a[ia]*a[ia];
            }
```

```
  anorm=sqrt(anorm);
  fnorm=anorm*e;
iter: anorm=anorm/constf;
iter1:  for(iq=1;iq<n;iq++)
  {m=iq-1;
   for(ip=0;ip<=m;ip++)
   {ia=lra*ip+iq;
    ja=lra*iq+ip;
    ka=lra*ip+ip;
    la=lra*iq+iq;
    if(fabs(a[ja]-anorm)>=0.0)
    {in=0;
     u=-a[ja];
     v=(a[ka]-a[la])/2.0;
     w=u/sqrt(u*u+v*v);
     if(v<0.0)w=-w;
     snn=w/sqrt(2*(1.0+sqrt(1.0-w*w)));
     snn2=snn*snn;
     csn=sqrt(1.0-snn2);
     for(i=0;i<n;i++){
      iia=lra*ip+i;
      iib=lra*iq+i;
      if(((i-ip)!=0)&&((i-iq)!=0)){
      b=a[iia]*csn-a[iib]*snn;
      a[iib]=a[iia]*snn+a[iib]*csn;
      a[iia]=b;
      }
      c=s[iia]*csn-s[iib]*snn;
      s[iib]=s[iia]*snn+s[iib]*csn;
      s[iia]=c;
     }
     csn2=csn*csn;
     stc=snn*csn;
     app=a[ka]*csn2+a[la]*snn2-2.0*a[ja]*stc;
     aqq=a[ka]*snn2+a[la]*csn2+2.0*a[ja]*stc;
     a[ja]=(a[ka]-a[la])*stc+a[ja]*(csn2-snn2);
     a[ia]=a[ja];
     a[ka]=app;
     a[la]=aqq;
     for(i=0;i<n;i++){
     ii=lra*i+ip;
     ij=lra*ip+i;
     ik=lra*i+iq;
     il=lra*iq+i;
     a[ii]=a[ij];
     a[ik]=a[il];
     }
     }
     }
     }
     if((in-1)==0){in=0;
     goto iter1;
     }
     if((anorm-fnorm)>0.0)goto iter;
     return 0;
     }
```

APPENDIX E

FOURIER TRANSFORM ALGORITHMS

The following routines calculate the two-dimensional Fourier transform and its inverse for an image patch. The method used is that described in section 4.6.2.1. The one-dimensional Fast Fourier Transform algorithm due to Cooley-Tukey is used.

```
int invfft2d(int wid,int hght)
 { /* DETERMINES THE INVERSE FOURIER TRANSFORM OF A FORWARD
TRANSFORM FOR AN IMAGE PATCH OF WIDTH wid AND HEIGHT hght. THE
FORWARD TRANSFORM IS IN FILE fftx.dat. THE OUTPUT IS DISPLAYED
ON THE COMPUTER SCREEN. TEMPORARY DISK FILES ARE CREATED DURING
THE OPERATION OF THIS ROUTINE.*/
   FILE *out,*in;
   int jj,ii,ll,sgn,nn,w,w1,nn1,hand;
   long offset;
   filefft="fftx.dat";
    nn=log(wid)/log(2)+1;
    w=exp(nn*log(2))+1;
    nn1=log(hght)/log(2)+1;
    w1=exp(nn1*log(2))+1;

  out=fopen(filefft,"rb+");
  hand=fileno(out);
       for(ii=0;ii<w1;ii++){
    for(jj=0;jj<w;jj++){fread(&g[jj+1],16,1,out);
                g[jj+1]=conj(g[jj+1]);}
             fft(g,nn);
   rewind(out);
   offset=(long)ii*w*16;
   fseek(out,offset,0);
   for(jj=1;jj<=w;jj++)
   write(hand,&g[jj],16);
   }
      for(ii=0;ii<w;ii++){
    for(jj=0;jj<w1;jj++)
    {offset=(long)(jj*w+ii)*16;
      rewind(out);
      fseek(out,offset,0);
      fread(&g[jj+1],16,1,out);
      g[jj+1]=g[jj+1]*w;
   }
   fft(g,nn1);
    for(jj=0;jj<w1;jj++)
    {offset=(long)(jj*w+ii)*16;
      rewind(out);
      fseek(out,offset,0);
      write(hand,&g[jj+1],16);
```

```
   }
  }
  rewind(out);
      for(ii=0;ii<w1;ii++){
    for(jj=0;jj<w;jj++)
   {fread(&g[jj],16,1,out);
    ll= sqrt(norm(g[jj]));
      putpixel(512+jj,352+ii,ll);
   }
   }
 fclose(out);
 getch();
return 0;
}


 int fft2d(int xx,int yy,int wid,int hght)
/*CALCULATES THE FORWARD FOURIER TRANSFORM FOR AN IMAGE PATCH OF
WIDTH wid AND HEIGHT hght WITH CENTRE AT PIXEL COORDINATE xx,yy.
THE INPUT IS READ FROM THE COMPUTER DISPLAY MEMORY, AND THE
OUTPUT IS THE TRANSFORM WHICH IS STORED ON A DISK FILE fftx.dat
INTERMEDIATE TEMPORARY DISK FILES ARE CREATED DURING THE
OPERATION OF THIS ROUTINE. CALCULATION OF THE FRACTAL DIMENSION
IS ALSO DONE. */
 { FILE *out,*out2,*in;
   int jj,ii,ll,sgn,nn,w,w1,nn1,hand,cnt[100],ix,iy;
                                d     o     u     b     l     e
radf[100],r,sx,sy,sxx,syy,sxy,slope,fd,dd,x,y,theta,gv,gv1,dx,dy;
   double pi=3.14159;
   long offset;
   char buff[20];
   filefft="fftx.dat";
   filefft1="fftx1.dat";
   out=fopen(filefft,"wb+");
    nn=log(wid)/log(2)+1;
    w=pow(2.0,nn);
    nn1=log(hght)/log(2)+1;
    w1=pow(2.0,nn1);
       out2=fopen("fftwin.img","wb+");
       for(ii=0;ii<w1;ii++){

for(jj=0;jj<w;jj++){if((abs(jj-w/2)>wid/2)||(abs(ii-w1/2)>hght
/2))ll=0;
                              else
                              ll=getpixel(xx+jj-w/2,yy+ii-w1/2);
                              fputc(ll,out2);
                    if(((jj+ii)/2*2)==(jj+ii))sgn=1;
                           else
                           sgn=-1;
                    g[jj+1]=sgn*complex(ll,0);}

   for(jj=1;jj<=w;jj++)
   fwrite(&g[jj],16,1,out);

   }
```

```
    fclose(out);
    fclose(out2);
   out=fopen(filefft,"rb+");
   hand=fileno(out);
        for(ii=0;ii<w1;ii++){
     for(jj=0;jj<w;jj++)fread(&g[jj+1],16,1,out);
              fft(g,nn);
    rewind(out);
    offset=(long)ii*w*16;
    fseek(out,offset,0);
    for(jj=1;jj<=w;jj++)
    write(hand,&g[jj],16);
    }
       for(ii=0;ii<w;ii++){
     for(jj=0;jj<w1;jj++)
     {offset=(long)(jj*w+ii)*16;
       rewind(out);
       fseek(out,offset,0);
       fread(&g[jj+1],16,1,out);
       g[jj+1]=g[jj+1]*w;
    }
    fft(g,nn1);
     for(jj=0;jj<w1;jj++)
     {offset=(long)(jj*w+ii)*16;
       rewind(out);
       fseek(out,offset,0);
       write(hand,&g[jj+1],16);
    }
   }
   rewind(out);
       out2=fopen("fftfft.img","wb+");
       for(ii=0;ii<w1;ii++){
     for(jj=0;jj<w;jj++)
    {fread(&g[jj],16,1,out);
     ll= sqrt(norm(g[jj]));
      fputc(ll,out2);
       putpixel(512+jj,352+ii,ll);
    }
    }
  fclose(out);
  fclose(out2);
/*CALCULATION OF FRACTAL DIMENSION*/
  out=fopen("fftcirc.dat","wt+");
       for(ii=0;ii<100;ii++){radf[ii]=0;cnt[ii]=0;}
   for(ii=0;ii<w1/2;ii++){
     for(jj=0;jj<360;jj++)
    { theta=pi*jj/180.0;
      x=(double)ii*cos(theta);
      y=(double)ii*sin(theta);
      ix=x+w1/2;
      iy=y+w1/2;
      dx=x+w1/2-ix;
      dy=y+w1/2-iy;

gv=dx*getpixel(ix+1+512,iy+352)+(1.0-dx)*getpixel(ix+512,iy+352);
```

```
gv1=dx*getpixel(ix+1+512,iy+1+352)+(1.0-dx)*getpixel(ix+512,iy
+1+352);
      gv=dy*gv1+(1.0-dy)*gv;
      radf[ii]=radf[ii]+gv;
      cnt[ii]=cnt[ii]+1;
    }
    }
    r=0;
 for(ii=0;ii<100;ii++)if(cnt[ii]!=0)radf[ii]=radf[ii]/cnt[ii];
 for(ii=0;ii<100;ii++)fprintf(out,"%d %lf \n",ii,radf[ii]);
 sx=sy=sxx=syy=sxy=0;nn=0;
 for(ii=1;ii<100;ii++){if(radf[ii]!=0){nn=nn+1;
                        r=2.0*log(radf[ii]);dd=log((double)ii);
                        sx=sx+dd;sy=sy+r;sxx=sxx+(double)dd*dd;
                        syy=syy+r*r;sxy=sxy+r*dd;}}
slope=((double)nn*sxy-sx*sy)/((double)nn*sxx-sx*sx);
fd=3.0+(slope+1)/2.0;
fprintf(out,"fd=%lf \n",fd);/*OUTPUT FRACTAL DIMENSION*/
 fclose(out);
 gcvt(fd,10,buff);
 setcolor(255);
 outtextxy(400,50,buff);
 invfft2d(wid,hght);
 getch();
 setcolor(0);
 outtextxy(400,50,buff);
 return 0;
}

int fft(complex f[513],int ln)
/*CALCULATION OF ONE-DIMENSIONAL FAST FOURIER TRANSFORM USING THE
COOLEY-TUKEY METHOD*/
{
 complex u,w,t;
 int n,nv2,nm1,k,l,le,le1,ip;
 int i,j;
  float pi=3.141593;
 n=1;
 for(i=1;i<=ln;i++)n=n*2;
 nv2=n/2;
 nm1=n-1;
 j=1;
 for(i=1;i<=nm1;i++)
    { if (i>=j) goto lab1;
      t=f[j];
      f[j]=f[i];
      f[i]=t;
 lab1:k=nv2;
 lab2:if (k>=j) goto lab3;
       j=j-k;
       k=k/2;
       goto lab2;
 lab3:j=j+k;
       }
 for(l=1;l<=ln;l++)
    {le=1;
```

```
 for(i=1;i<=l;i++)le=le*2;
 le1=le/2;
 u=complex(1,0);
 w=complex(cos(pi/le1),-sin(pi/le1));
 for(j=1;j<=le1;j++){
 for(i=j;i<=n;i=i+le){
   ip=i+le1;
   t=f[ip]*u;
   f[ip]=f[i]-t;
   f[i]=f[i]+t;}
 u=u*w;}}
  for(i=1;i<=n;i++)f[i]=f[i]/float(n);
return 0;
}
```